CECD/ME Symposium on Computation-Enabled Materials Discovery
Summary Page

A symposium was held on May 20, 2015 in the Center for Engineering Concepts Development, Department of Mechanical Engineering at the University of Maryland to take a retrospective look at some of the most prominent science and engineering software and codes (SESC) used today to discover and design materials.

Significant lessons can be learned, both scientifically and programmatically, from the experiences of the people and programs that have produced some of the present-day *de facto* infrastructure.

Invited speakers were requested to speak on their current research interests, the experiences of the SESC capability they had a hand in developing and, in particular, the technical and programmatic challenges they overcame. Questions they were requested to answer included: a) Identify the major periods in the life of the SESC? b) How did stakeholders change as developments matured? c) How did the teams and workforce change? d) Are some challenges unique to your sector (government, academia, industry)? e) What are the greatest virtues and worst threats? f) What is the intellectual property and which models for protection work?

The primary findings include:

- The SESC life cycle is nonlinear. Related efforts can therefore be difficult to classify clearly as fundamental or applied research using existing taxonomy.
- Performers, or the developers of SESCs, stretch across varied sectors. No clear sector exclusively performs basic or applied (or beyond) research.
- Stakeholders are diverse and can be well-defined as a function of the development maturity of the SESC. Due to the nonlinear life cycle, stakeholders cannot be statically defined.
- Sectors (government, academia, business) have strengths and weaknesses that are under-defined. Some strengths are under-utilized.
- The transition from "algorithmic ideas" to *de facto* infrastructure is not evident in less than 10 years.
- Open source licensing is a means to protect IP. But serves large, complex SESC efforts best and may make potential privatization difficult in the future.

Speakers were selected for their acknowledged contributions to recent SESCs that may be considered a part of today's infrastructure for materials discovery and innovation. The invited speakers (and their presented efforts) included: Douglass Post (DoD CREATE), Robert J. Harrison (NWCHEM), Gerhard Klimeck (NEMO & nanoHUB), Michael Mehl (NRL-TB), Steve Plimpton (LAMMPS), A. (Tom) Arsenlis (ParaDIS), Stefano Curtarolo (AFLOW), Rose McCallen (ALE3D), and Tom McGrath (DYSMAS). Additional speakers were also invited that included distinguished faculty and scientists from the University of Maryland and government laboratories located near the University of Maryland who attested to the value of these and other SESC capabilities.

Symposium Organizing Committee: Peter W. Chung, Davinder. K. Anand, Balakumar. Balachandran, Ania Picard, Dylan Hazlewood, Mukes Kapilashrami, Millard S. Firebaugh (RADM, Ret), and James M. Short.

CECD/ME Symposium on Computation-Enabled Materials Discovery

Final Report

November 2015

Prepared by:

Peter W. Chung

Center for Engineering Concepts Development

Department of Mechanical Engineering

University of Maryland

Sponsored by:

Participants

Davinder K.	Anand	CECD/Mechanical Engineering, UMD
Tom	Arsenlis	Lawrence Livermore National Lab
Wade	Babcock	NSWC IHEODTD
Bala	Balachandran	Mechanical Engineering, UMD
Robert	Briber	Materials Science and Engineering, UMD
Aris	Christou	Materials Science and Engineering, UMD
Peter	Chung	Mechanical Engineering, UMD
John	Clayton	Army Research Lab/OAEE, UMD
Stefano	Curtarolo	Duke University
Abhijit	Dasgupta	Mechanical Engineering, UMD
Ruth	Doherty	NSWC IHEODTD
Dev	Ettisserry	Electrical and Computer Engineering, UMD
Millard	Firebaugh	CECD/Mechanical Engineering, UMD
Mark	Fuge	Mechanical Engineering, UMD
Neil	Goldsman	Electrical and Computer Engineering, UMD
Robert	Harrison	Stony Brook University/ Bookhaven
Dylan	Hazelwood	CECD/Mechanical Engineering, UMD
Mukes	Kapilashrami	CECD/Mechanical Engineering, UMD
Gerhard	Klimeck	Purdue University
Teng	Li	Mechanical Engineering, UMD
Krista	Limmer	Army Research Lab
Kenny	Lipkowitz	Office of Naval Research
Sultana	Mahmooda	NASA
Rose	McCallen	Lawrence Livermore National Lab/UC Davis
Patrick	McCluskey	CALCE, Mechanical Engineering, UMD
Thomas	McGrath	NSWC IHEODTD
Michael	Mehl	Naval Research Lab
Yifei	Mo	Materials Science and Engineering, UMD
Lynn	Munday	Army Research Lab
Joe	Myers	Army Research Office
Mahesh	Neupane	Army Research Lab
Michael	Peng	Mechanical Engineering, UMD
Ania	Picard	CECD/Mechanical Engineering, UMD
Steve	Plimpton	Sandia National Lab
Carolyn	Plitt	Mechanical Engineering, UMD
Douglass E.	Post	Army-HPCMO
Amir	Riaz	Mechanical Engineering, UMD
Berend	Rinderspacher	Army Research Lab
James	Short	CECD/Mechanical Engineering, UMD
Timothy	Sirk	Army Research Lab
Ichiro	Takeuchi	Materials Science and Engineering
Frank	VanGessel	Mechanical Engineering, UMD
Uzi	Vishkin	UMIACS, UMD
Eric	Wachsman	Materials Science and Engineering
James	Warren	NIST
Rose	Weisburgh	Mechanical Engineering, UMD

Preface

This document presents a summary report of the Symposium on Computation-Enabled Materials Discovery hosted by the Center for Engineering Concepts Development (CECD) in the Department of Mechanical Engineering at the University of Maryland in College Park Maryland on May 20, 2015. The report contains an overview of presentations from invited speakers along with a distillation of facts that identifies a retrospective set of lessons-learned in the inception, deployment, and adoption of computational scientific infrastructure. The Appendix contains the transcripts of the speakers' presentations.

Advances in computing capabilities have today enabled unprecedented physical realism and the least presumptive simulations and models that have ever been seen. Such simulations and models are fundamentally enabled by an infrastructure made up of science and engineering software and codes (SESC) that, in many cases, have taken decades to reach its current state. This SESC infrastructure is an integral part of nearly all research and development communities, and their value to innovation today is unmistakable. Entire communities of researchers rely critically on these capabilities to study new materials and materials technologies. Virtual testing, screening, concept integration and evaluation, and design can first occur within a parametric model that can be adjusted, augmented, embellished, or discarded long before the first tooling. These are important capabilities to have in high precision industries related to nanotechnology, biotechnology, medicine, electronics, power and energy, and national defense, among others. They minimize costs, optimize concept evaluation processes, and overall accelerate the ability of scientists and engineers to bring innovative technologies to market. With time, analogous tools have also permeated other industries ranging from consumer products to financial services. In light of these facts, the SESC infrastructure serves as an ever-improving set of baseline capabilities. They also help to educate future scientists and engineers whether via training and professional development or in the augmentation of classical pedagogies in higher education.

Out of a recognition of the value and need for SESC infrastructure and recent national and community dialogues on Cyberinfrastructure and the like, the CECD organized a symposium to examine the experiences of individuals who were directly involved in creating the codes and software tools today used as infrastructure capabilities in the study of materials. As the physical accuracy of materials modeling techniques have improved over the last half century, the communities of interest are necessarily disparate and diverse. The purpose was to assemble different groups to share in their experiences, many that span multiple decades, in the development of SESC capabilities that today are seen by some as de facto standards. Remarkable consistency was seen in these experiences despite the varied participating disciplines and organizations. This suggests that it is possible to harness these ideas to develop educational and research programs that are more directed and deliberate in the creation of new and enabling SESC infrastructure.

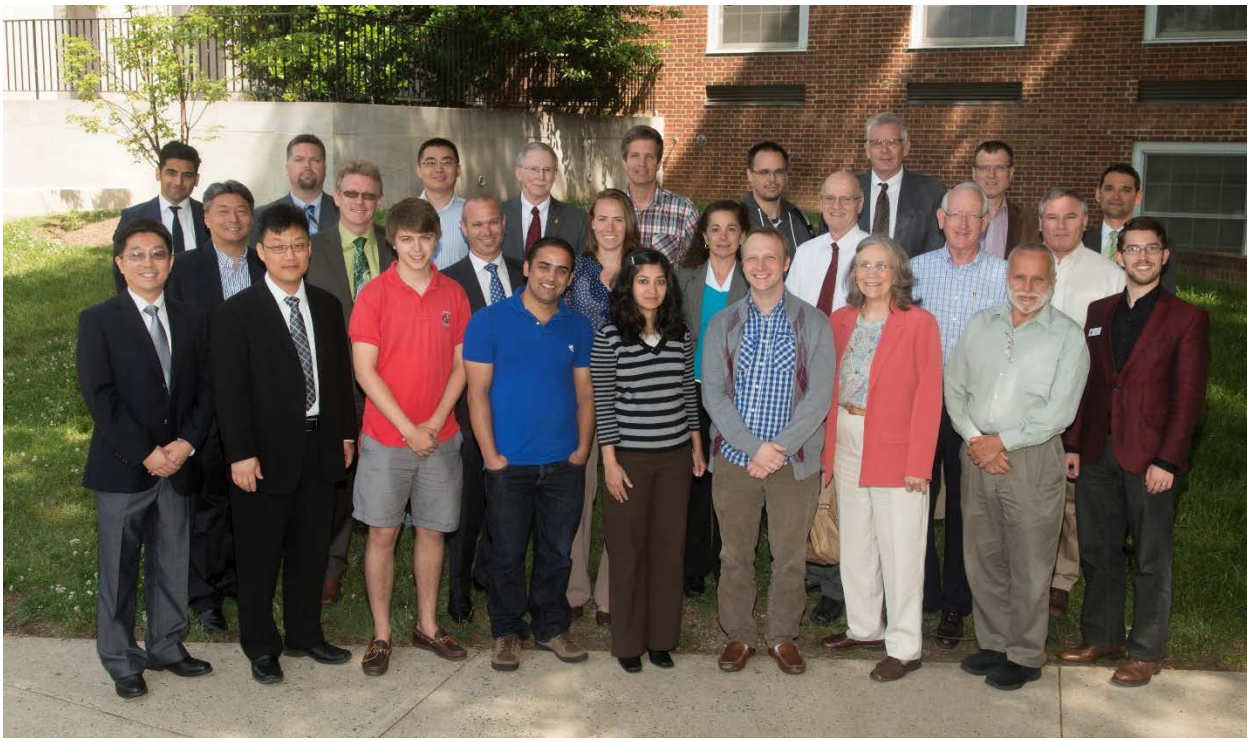
The symposium was made possible by the advice and encouragement of Dr. Joseph D. Myers (Army Research Office) and Dr. Massimo Ruzzene (National Science Foundation). Their support is gratefully acknowledged. Within CECD, the organizing committee was instrumental in the planning and execution of the event. The committee consisted of Emeritus Prof. Davinder K. Anand, Prof. & Department Chair Balakumar Balachandran, Ms. Ania Picard, Mr. Dylan Hazlewood, Dr. Mukes Kapilashrami, Prof. Millard Firebaugh (RADM, Ret), and Dr. James M. Short.

The symposium also greatly benefited from the participation of individuals with direct first-hand knowledge of the development of SESC infrastructure. The event would not have been possible without the involvement of this accomplished and august group. Our sincere gratitude is extended to Dr. Douglass Post, Prof. Robert J. Harrison, Prof. Gerhard Klimeck, Dr. Michael Mehl, Dr. Steve Plimpton, Dr. A. (Tom) Arsenlis, Prof. Stefano Curtarolo, Dr. Rose McCallen, and Dr. Tom McGrath. Invited speakers also included distinguished faculty and scientists from the University of Maryland and government laboratories located near the University of Maryland whose substantial contributions were instrumental: Prof. Ichiro Takeuchi, Prof. Neil Goldsman (through Mr. Dev Ettiserry), Prof. Teng Li, Dr. John D. Clayton, Dr. Lynn Munday, Prof. Amir Riaz, and Prof. Yifei Mo.

As with all such efforts, numerous students ensured the smooth running of the Symposium and diligently transcribed the talks in the following months. They include Rachel Flanagan, David Nguyen, Jie Peng, Carolyn Plitt, Frank VanGessel, and Rose Weisburgh.

Finally, this report describes observations made by the symposium organizing committee and is not endorsed by any of the organizations or government agencies represented by the symposium participants or attendees. The views in the summary statements that follow only reflect those of the symposium organizing committee.

Peter W. Chung
CECD Energetics Group Lead
Associate Professor, Department of Mechanical Engineering
University of Maryland, College Park MD



Rear (left to right): Mukes Kapilashrami, Dylan Hazlewood, Yifei Mo, Douglass Post, Steve Plimpton, Berend Rinderspacher, Michael Mehl, Robert Harrison, Tom Arsenlis

Middle: Ichiro Takeuchi, Gerhard Klimeck, James Warren, Krista Limmer, Rose McCallen, James Short, Millard Firebaugh, Joseph Myers

Front: Teng Li, Peter Chung, Mahesh Neupane, Sultana Mahmooda, Timothy Sirk, Ruth Doherty, Kenny Lipkowitz, Mark Fuge

Executive Summary

With national and international attention turning towards fundamental scientific capacity and ability of the technology base to innovate more rapidly, a palpable desire exists for scientific capabilities to be improved in number, sophistication, and accessibility. Today modeling and simulation are recognized as valued members in the cooperative team that includes theory and experiment, *as enabled* by science and engineering software and codes (SESC). As such, SESCs can become – or in some cases, already are – the infrastructure that enables scientific or engineering enterprise. They are an instantiation of theory and empirical knowledge that, owing to reliability, accuracy, or ease of use, becomes a primary pillar in a technical effort. They enable solutions to problems to confirm results of the past, and/or proceed in to regions of the parameter space where past theoretical or experimental solutions are not as easily extended. When in use across multiple groups, they become a capability that underpins a community.

SCIENTIFIC AND ENGINEERING SOFTWARE AND CODES (SESC)

But where do SESCs come from and how are they developed? This symposium was organized, in part, to share perspectives that help convey the meaning of both this question and its many answers. If a future of discovering materials relies on computing, will new SESCs be required to realize that future or are the present-day SESCs sufficient? If the latter, who will produce SESCs and what are the means through which they will do so? The symposium brought together experts who have developed SESCs that today serve important roles in enabling the discovery of materials or expediting their transitions into technologies. By reviewing the histories and sharing the stories from this list of exemplars, we began to identify echoing themes that suggest that the future of SESCs is not as clear as it may have once been. This report describes the summary outcomes and the detailed presentations.

This symposium was held under an existing backdrop of numerous national dialogues including Cyberinfrastructure, Materials Genome, Integrated Computational Materials Engineering, Big Data, and STEM education, to name a few. The visions and strategies are extensively articulated in, among others, the Materials Genome Initiative (MGI)[1], the National Research Council Report on Integrated Computational Materials Engineering[2], and the National Science Foundation's vision for Cyberinfrastructure[3]. These have grown, in part, from the greater need for integrative modeling and simulation. National studies, such as the National Science Foundation's Simulation-Based Engineering Science Blue Ribbon Panel Report[4] and the WTEC Panel Report on International Assessment of Research and Development in Simulation-Based Engineering and

[1] Executive Office of the President, Office of Science and Technology Policy, "Materials Genome Initiative for Global Competitiveness," June 2011.

[2] National Research Council. National Academy of Engineering Report of Committee on Integrated Materials Engineering (ICME). National Academies Press, Washington, D.C., 2008.

[3] National Science Foundation Cyberinfrastructure Council, "Cyberinfrastructure Vision for 21st Century Discovery," March 2007.

[4] Oden, J.T., T. Belytschko, J. Fish, T.J.R. Hughes, C. Johnson, D. Keyes, A. Laub, L. Petzold, D. Srolovitz, and S. Yip. Simulation-based engineering science: Revolutionizing engineering science through simulation. Arlington, VA, National Science Foundation, 2006.

Science[5] have characterized the broad landscape of simulation-based engineering and science while identifying numerous needs and opportunities. These and other studies acknowledge the important roles of computing, mathematical, physical, and engineering sciences that feed into and culminate in a codified computing capability that are instrumental in creating SESCs.

Thus the topics covered in this Symposium are not disconnected. In fact, SESCs are the capabilities through which models and simulations can be constructed and studied on computers. In spite of this vital role, communities still struggle to agree on their true cost and value. It is not difficult to see why. The histories of most SESCs are circuitous and span decades. Their efforts are labor- and time-intensive with a fraught mix of scientific, engineering, and management challenges that are very rarely evident to an end-user. Almost all modern SESCs have experienced different research classifications, vacillating among basic, applied, and developmental research at different periods in their histories. This migratory behavior occurs for many reasons but most predominantly for the sake of maintaining or acquiring support in an environment where resources are growing scarcer for all. The research classifications, once accepted, generally are not easily forgotten. So ironically, developers must work to overcome the inertia created by their own programmatic decisions. Contrary to the conventional succession from fundamental, to applied, to development, and beyond, SESCs frequently undergo periods of reinvention where a developmental effort one year becomes the core platform for scientific inquiry the next. Eventually, with time, patience and vision, the most successful SESCs become widely used and developed, and entire communities rely on them.

IN SPITE OF THIS VITAL ROLE, COMMUNITIES STILL STRUGGLE TO AGREE ON THE TRUE COST AND VALUE OF SESCS.

Such paths are not uncommon across scientific and engineering communities but SESCs face one unique challenge that looms large in the future. Hardware capabilities used in high performance computing systems, such as processors, memory, storage, networking, and certain programming languages, are continuing to benefit in part from the tangential interests that drive a global trillion-dollar computing and electronics economy. New hardware architectures are forcing more transistors onto a smaller area on a semiconductor die resulting in greater speed or efficiency. These architectures soon will render older programming paradigms obsolete. As multicore chips on multisocket motherboards also continue to develop and memory is made to work more collaboratively with cache, an SESC's performance will be limited if it was written using any one (or no) particular parallelization technique. Hardware improvements will be perpetual as hardware developers continue to release on an annual schedule and as processing and fabrication technologies make continuous improvements into the foreseeable future.

[5] Glotzer, S.C., S.T. Kim, P.T. Cummings, A. Deshmukh, M. Head-Gordon, G. Karniadakis, L. Petzold, C. Sagui, and M. Shinozuka, "International Assessment of Research and Development in Simulation-Based Engineering and Science," World Technology Evaluation Center, Inc., April 2009.

What this means for SESCs is stagnation at best. The greater efficiency and scalability of the hardware will have little effect. SESC capabilities will be anchored to a significantly slower rate of improvement than the actual specifications of emerging hardware suggest. Thus, grave concerns continue to grow that SESCs will be limited in addressing emerging needs or will grow increasingly dated as scientific advances rewrite the theories upon which they rely.

SESC CAPABILITIES WILL BE ANCHORED TO A SIGNIFICANTLY SLOWER RATE OF IMPROVEMENT THAN THE ACTUAL SPECIFICATIONS OF EMERGING HARDWARE SUGGEST.

For researchers and developers of SESCs, finding support is the primary challenge. While a complete rewrite of SESCs to improve compatibility with new hardware is certainly a technically feasible option, traditional funding sources rarely permit the reinvention of an existing capability unless as a secondary, and often incidental or deemphasized, objective. Private industry support is also certainly possible, but a business plan is most sensible in a for-profit model and where a market for the product is already established and robust. Current examples of successful SESCs categorically have origins in small research codes with a handful of potential users. It was only after a significant germination period that the number of users began to climb. Furthermore, the greater use of public license agreements and the crowd-sourcing of developments may make late-stage privatization no longer legally tenable. Late-stage SESCs require curation and up-keep, where there are significant costs that may not be easily handled in an open-source model or without clear licensor rights. This is especially true in light of the relatively small sizes of some SESC user communities.

And should funding be obtained, the second challenge facing developers are the issues surrounding personnel and career development. Government, university and private sector incentive structures often lack a clear valuation system for SESC developments. Compounding this is that new computers demand increasingly specialized programming skills. A workforce doubly-trained as experts in a traditional scientific discipline as well as the critically-important computing skills will only grow increasingly more difficult to find or educate.

The evidence from the symposium indicates that overcoming these challenges may be possible but the past shows this is systemically difficult. Educational curricula changes, creation of new faculty & researcher incentives, national laboratory staff participation, and a holistic examination of the market for SESCs are some of the potential directions to pursue.

It should also be noted that at present, programs of all sizes and scales ranging from student fellowships to multi-institutional centers, have the mission for improving scientific capacity. Many arguably deal with SESCs. Yet little appears to be openly shared about their arduous journeys. Here, the past must serve as prologue. The approaches for overcoming common hurdles are mostly propagated informally through word-of-mouth anecdotes, leaving few beneficiaries of these valuable lessons-learned.

HERE, THE PAST MUST SERVE AS PROLOGUE.

In short, we took a careful look at the past to examine and understand what history tells us about creating SESC infrastructure. To understand true cost and value, these stories must be told. In

a measured approach, the Center for Engineering Concepts Development at the University of Maryland assembled recognized experts in the conception, development and deployment of SESC infrastructure. The Symposium was organized as a forum where experiences could be shared from varied disciplines. The meeting was attended by mechanical engineers, electrical engineers, computer scientists, materials scientists, physicists, and chemists. Despite the variations in disciplines, the stories had familiar refrains and best practices could be heard echoing throughout the day.

This report summarizes the key observations made during the single-day event, and transcripts of the presentations follow in the Appendix. The brevity of the meeting means the day is necessarily remiss in capturing the categorical experiences across countries, sectors, and institutions. However, common themes were observed and summary points are grouped according to a) Life Cycle of Science and Engineering Software and Codes, b) Performers, c) Stakeholders, d) Sector-unique challenges, e) Periodic virtues and threats to SESC efforts, and f) Intellectual Property.

Table of Contents

Integrated materials discovery engine

Appendix: Rose McCallen

Evolution of ALE3D, challenges, enabled science, return on investment, and a vision for a path forward

Appendix: Thomas McGrath

An Overview of DYSMAS Coupled Code Development and Application

Appendix: Amir Riaz

New Advances in Direct Numerical Simulation of Multiphase Flow

Summary of Observations:

1. ***Life cycle of Science and Engineering Software and Codes (SESC)***
    a. Early SESC development histories resemble basic research ventures. The histories are circuitous and often marked by near-death experiences for the project. Early metrics are based on peer-reviewed publications. Principal Investigator(s) is(are) highly involved in attracting funds and sustaining the efforts.
    b. Late development is driven partly by scientific interest and partly by engineering need even after the SESC is openly shared and can be considered infrastructure. It is rare that the drivers are exclusively scientific or mission-need.
    c. Among the SESCs considered in this study, none appears to have been in development less than ten years.
    d. The performers and stakeholders differ between the early- and late-stages of SESC development. In some cases, the roles reverse with time – performers become the stakeholders and the stakeholders become the performers. Once developmental versions of SESCs reach a point of being infrastructure to a community, the community is more equipped to contribute back to the scientific and engineering interests of the organization that originally performed or supported its development.
    e. SESCs life cycles are highly nonlinear. Basic, applied, and developmental research are continuously intermingled if not cyclic. A single SESC may have several aspects in concurrent developing each having a unique Technical Readiness Level (TRL). The cyclic nature of the development history seems to be found in the history of every SESC. It is often presumed each step in the cycle is mutually exclusive of the others, making it difficult to classify SESC development clearly as any particular one (fundamental, applied, or development) and, as a result, is often classified incorrectly.

2. ***Performers***
    a. Early development occurs principally, but not solely, in universities.
    b. Some algorithms, if not available in the scientific literature, are developed and studied by government researchers. These efforts appear to be tied to a clearly defined broader mission seeking to develop a particular code capability.
    c. Strongest transitions appear to involve graduating students moving from the university sector into a government laboratory, taking their early coding experiences with them.
    d. Less explicit transitions appear to occur via dissemination through the scientific literature. Infrastructure creation through this approach appears to be less deterministic.

e. Crowd-sourced code development is primarily made up of students or personnel at government labs. Private industry involvement occurs but apparently to a lesser degree.
f. Private sector business emphasizes improvements in interoperability and code connections for user-based requirements. New code developments appear to be focused on increased capability and other improvements for the user experience rather than the evaluation and testing of algorithms and calculation methodologies.
g. Sustained development requires funding sources and workforce incentivized by potential solutions to problems of a scientific or mission-driven nature.
h. Traditional performer roles generally appear to emphasize
    i. Universities: First instantiation, workforce education, computability
    ii. Government: Transition, scale-up, validation
    iii. Private Industry: Usability

3. ***Principal Stakeholders***

a. Academic faculty and students involved in early development have stakes in developments in the form of funded research, scientific journal publications and education during early periods. These incentives seem to conflict with the push to share early source code.
b. There is evidence to suggest that sustained research and development of SESCs involve early stakeholders of SESCs becoming its users in later stages as the SESC becomes useful for further investigation of other scientific/engineering problems.
c. As the value of the SESC becomes more apparent, larger investments are made by government laboratories for internal efforts. This seems to be accompanied by a clearer community-wide vision that prompts university efforts and, in turn, results in more focused adoption of methods (and related coding schemes) in internal government programs.
d. The government's stakes are in the capability to perform design and testing in larger programs and systems that may be of either scientific or engineering significance. Clarity in its vision and mission aids in strengthening community involvement.
e. When a market of sufficient size exists, regardless of internal government adoption, private businesses may be formed, typically in connection with the original academic research groups. Software and support services may serve to generate revenue.
f. Universities, government laboratories, and industry have under-defined roles as stakeholders of SESCs. Questions still persist regarding responsible parties for SESC infrastructure support, maintenance, and workforce training.

4. ***Sector-unique challenges***

a. Universities, government laboratories, and industry appear to have naturally recurring roles in the development of SESCs due to
    i. Workforce differences
    ii. Mission differences
b. The academic workforce, apart from university faculty, is composed principally by students and post-doctoral associates. The longevity is often limited by graduation rates or other career prospects. Large numbers of the workforce are foreign nationals, which may limit their ability to transition into internal government programs. Projects spend the majority of its resources on training of the workforce. Turnover is relatively rapid. Due to the career development incentives of university faculty, research must culminate in publications.
c. Government efforts are composed principally of and/or led by career scientists and engineers. Incentives include the establishment, development, or sustainment of novel internal programs competitively awarded through mostly government-only solicitations. Competitions consider in part publications and scientific reputation of the principals, but longer efforts consider transitions of deliverables into larger programs of record. Turnover of the workforce is limited to conditional or non-permanent employees.
d. Private sector efforts are driven exclusively by revenue. A clearly defined market must exist for SESCs to have internal corporate support or must narrowly focus on support for a product. The workforce turnover rate is higher than in the government but lower than in academia. The inherent risk associated with SESCs is mitigated by closing source code and acquiring market share through mergers and acquisition. SESC-focused subsidiaries appear to be formed to provide internal value to other internal businesses.

5. ***Periodic virtues and threats to SESC efforts***

a. The cyclic lifespan of SESCs
    i. There is either a limited taxonomy or limited awareness of taxonomy by both performers and stakeholders alike. Basic, applied, and developmental research are frequently mischaracterized.
    ii. Both early and late SESC development efforts appear to be sustained in part through fundamental questions that frame opportunities and needs.
    iii. Quality control protocols often justify continued support for development in the later stages of programs where SESCs are shared with users but, by definition, makes it difficult to classify efforts as fundamental research.
b. There does not appear to be any evidence that SESCs have been developed exclusively or predominantly through private support or funding.

c. Successful SESCs appear to experience sustained development, with often trickling support, over 10 or more years. The value of SESCs to physical experimentation has been demonstrated with sufficient time.

6. Intellectual Property

a. Measurability of intellectual property in SESCs is continuing to improve through approaches that recognize early career faculty contributions to open-source developments. The degree of adoption is still limited to select universities and government institutions.
b. After an initial period of small scale development, crowd-source development and public license models work well to simultaneously increase the user-base and the number of developers. However, this also makes it more difficult to convert to a different model (e.g., license-for-fee or service-for-fee) when maintenance costs become substantial or crowd-sourcing is no longer viable.
c. The use of publically-licensed SESCs for government-only purposes is not necessarily subject to the terms of the license as long as the intellectual property is not introduced into a commercial or non-governmental setting.
d. The portability of computer code makes the challenges of protecting this form of intellectual property distinct from other forms of intellectual property.

Appendix A: Presentation Transcripts

The following presentation transcripts are accompanied by the presentation materials as permitted by the respective security offices.

Transferring Basic Research to Engineering Design with Physics-based Software

Dr. Douglass Post
Department of Defense High Performance Computing Modernization Program (HPCMP)
Carnegie Mellon University Software Engineering Institute
Pittsburgh, PA

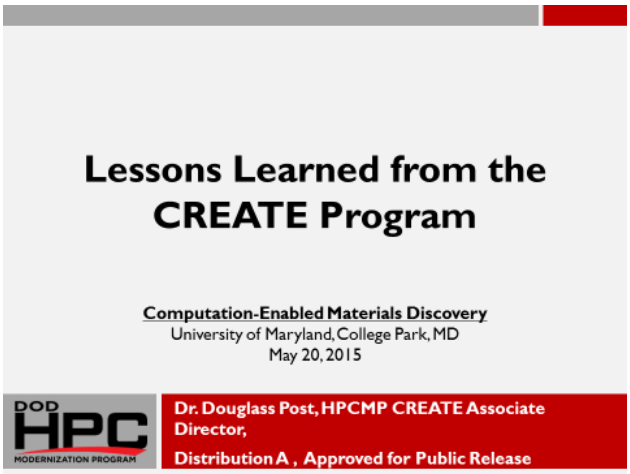
Abstract

Producing practical applications of the results of basic research has always been a major challenge. The continued exponential growth in computing power since the invention of computers is beginning to offer a new process for accomplishing this. The first step is to capture the existing knowledge of a field together with the new knowledge from research in that field and incorporate it into physics-based application software for high performance computers. These "tools" can then be used by "engineers" to design real systems that include the benefits of the advances in the state of knowledge in the field (or fields) of interest. I will discuss how we have been accomplishing this in the DoD HPCMP Computational Research and Engineering Acquisition Tools and Environments (CREATE) Program. The CREATE program involves the development and deployment of 13 computational tools that can accurately predict the performance of military air and ground vehicles, ships and radio frequency antennas. These tools are beginning to provide decision data early in the product design and development process, allowing the identification of design flaws and performance shortfalls, and reducing the costly (time and money) rework required to fix them. Goodyear tire adopted this paradigm and was able to cut their time to market by a factor of four, and increase the number of new products from 10 per year to 40 per year. The CREATE tools are being developed by government-led distributed, non-collocated teams embedded in the customer organizations where the technical expertise resides. I will discuss how this code development and deployment process might be transferred to the construction of physics-based high performance computing tools for the design of materials. I will discuss the many similarities as well as many differences between product development of large weapon systems and the potential product development of materials and offer suggestions for how the latter might be achieved.

Biography

Dr. Post is the Associate Director of the US Department of Defense (DoD) High Performance Computing Modernization Program (HPCMP) Computational Research and Engineering Acquisition Tools and Environments (CREATE) Program and the former Chief Scientist of the HCPMP (2005-2014). He is an IPA from the Carnegie Mellon University Software Engineering Institute. The CREATE Program has been developing and deploying 13 physics-based high performance computing engineering software applications that are being used by DoD engineers to accurately predict the performance of military air and ground vehicles, ships and radio frequency antennas. These tools are being used to identify design flaws and performance shortfalls in these systems early in the design process, well before metal has been cut. This will reduce the costly rework required to fix them, and reduce the risks, time and cost of many major DoD acquisition programs.

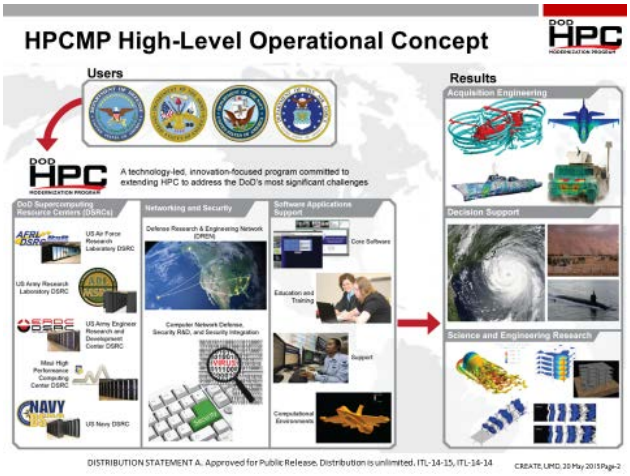
Doug received a Ph.D. in Physics from Stanford University in 1975 where he held National Science, Woodrow Wilson and Hertz Foundation Fellowships. He is Associate Editor-in-Chief of the AIP/IEEE publication "Computing in Science and Engineering". He led the A and X Division Nuclear Weapons Simulation programs at the Lawrence Livermore National Laboratory and the Los Alamos National Laboratory nuclear weapons code development programs (1998-2005). Doug led the International Thermonuclear Experimental Reactor (ITER) Physics Team (1988-1990) for which he received the American Nuclear Society (ANS) Outstanding Technical Achievement Award for fusion science and engineering in 1992. From 1993-1998, Doug led the ITER In-Vessel Physics Team (1993-1998). He established and led the tokamak modeling group at the Princeton University Plasma Physics Laboratory (1975-1993). He is a Fellow of the American Nuclear Society, the American Physical Society, and the Institute of Electronic and Electrical Engineers (IEEE). He received the American Society of Naval Engineers 2011 Gold Medal Award in February, 2012, for establishing and leading the CREATE Program. He has written over 250 publications with over 7,000 citations.



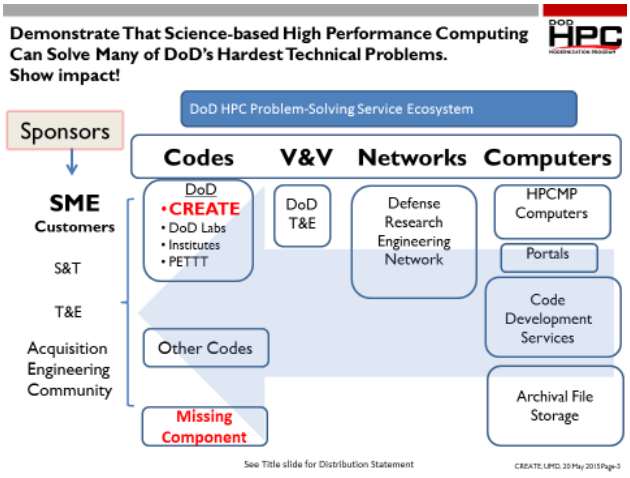
I was invited to this symposium on materials and materials development to discuss our experience with the DoD CREATE program on how to turn research codes into practical engineering tools. We are building codes to design and predict the performance of ships, airplanes, antennas, and ground vehicles, including generating the geometry and meshes. That's not strictly a materials issue (your problem), but it does go back to my roots, so I brought one of my first papers which I wrote with some people at Livermore entitled "Steady State Radiative Cooling Rates for Low-Density High Temperature Plasmas", which illustrates the multi-scale challenges we all face. We were able to compute, in 1977, the soft X-ray and UV emission of tungsten (Z=74) and uranium (Z=92) and other elements from the multi-charged ions in high temperature plasmas. We were only able to do it by confronting the multiscale issues straight on. For tungsten in a one kilovolt plasma, the conditions in the controlled fusion experiments at Princeton in the late 1970s, there were millions and millions of emission lines. Computing the strength of each of those lines and summing up the emissions was a task beyond the capability of the existing computers. The CDC7600s we had a four Megahertz central processor with just four Megabytes of memory. Today we use computers with hundreds of thousands of 2 Gigahertz processors, each with 2 GigaBytes of memory, to address such problems. Solving a large complicated atomic physics problem with CDC 7600 would seem to be completely hopeless. So we concentrated on the essential quantity of interest (QOI), the total energy emission rate in coronal equilibrium. To find that, we made a lot of simplifying approximations keeping a focus on the QOI. We used an average ion model instead of calculating each multi-charged ion. We didn't calculate each one of the millions of possible transitions. We used the oscillator sum strength rule (https://en.wikipedia.org/wiki/Sum_rule_in_quantum_mechanics) , a kind of conservation law that stated that for a given set of transitions within a level, the sum of the oscillator strengths was one. This allowed us to get a sufficiently accurate answer by calculating only the major transitions. With these and other simplifications, we were able to put together a code that could calculate the emission of tungsten in one kilovolt plasma in 15 to 20 minutes on a CDC7600, which was a practical solution (Post, D.E., et al., *Steady-State Radiative Cooling Rates For Low-Density High Temperature Plasmas.* Atomic Data and Nuclear Data Tables, 1977. **20**(November): p. 397-439.) Even today, on today's computers, at the forefront of what we can do, it takes 15 to 20 minutes to calculate all the lines. Our method yielded a reasonably accurate calculation of the total emission. The most accurate calculation today comes within a factor of two of what we got 40 years ago for the total emission. Of course, we didn't do the hard problem, which is crucial for calculating the opacity due to the need to include all the millions of lines. The practical implications of this were very large. Tungsten was used as a refractory material in Tokamaks, the mainline approach for controlled fusion. In the late 1970s, it was proving impossible to increase the plasma temperature above 1 keV, even with very intense heating methods. Our calculations indicated that even a small amount of tungsten ablated from the tungsten limiters would suffice to radiate all the heating energy to the vacuum vessel walls. The calculations also indicated that carbon limiters would have very low radiation losses since at 1 keV, carbon would be fully ionized. The tungsten limiters were

replaced with graphite limiters, and the central temperature in the tokamak experiments at Princeton jumped up to 7 keV. This result led to the construction of the next generation of tokamak experiments.

Thus I have roots in this community.



I am part of the DoD High Performance Computing Modernization Program. We have about 20 supercomputers in 5 major sites around the US. We have high speed networks to connect the customers to the computers. And we have software. What I've been doing for the last 10 years at the HPCMP is to organize and lead a major software development program (the Computational Research and Engineering Acquisition Tools and Environments, CREATE) to develop and deploy nine multi-physics high performance computing software applications to design and analyze military aircraft, ships and Radio Frequency Antennas for the Department of Defense acquisition engineering organizations.



Computational Engineering requires an ecosystem. You must have computers. You can't compute without computers. You need networks to connect the users to the computers because the computers are most likely at a few central places, in our case, DoD facilities or military bases. The users are spread around many universities, military facilities and defense industries. Generally, they're located at centralized sites so a lot of users can get access to them. Application software, of course, is worthless without verification and validation. You must have data that represents nature to check the codes and guide your code development. Then you must have application software. Computers can be used to calculate almost anything. They can be used to analyze data for the large hadron collider, or data from a telescope, or to compute materials properties, or to design armor for tanks. A network is also somewhat generic. But software is usually specific to each application, and that's really the long pole in the tent. Networks and computers are almost a commodity, although now they are getting complicated. You write software one code at a time. To make all of this work, you need to have people run the codes to make this whole system work. These subject matter experts can come from various communities: science and technology, test and evaluation, engineering. But they're all subject matter experts, because none of these codes are black boxes. I do interviews of fellowship applicants and follow-ups for the Hertz Foundation. Since I have a background in computational chemistry and computational atomic physics, I typically get some of those students, and I'm usually appalled. These days, the really bright kids

in high school seem to be publishing papers, and certainly in college they're doing that. They sit down in front of us, and start talking, and they say:

"Oh, I've been doing this molecular dynamics thing, look at my three papers."

I say:

"Walk me through what you have done. What is molecular dynamics? What are you trying to do?"

And there's a pause.

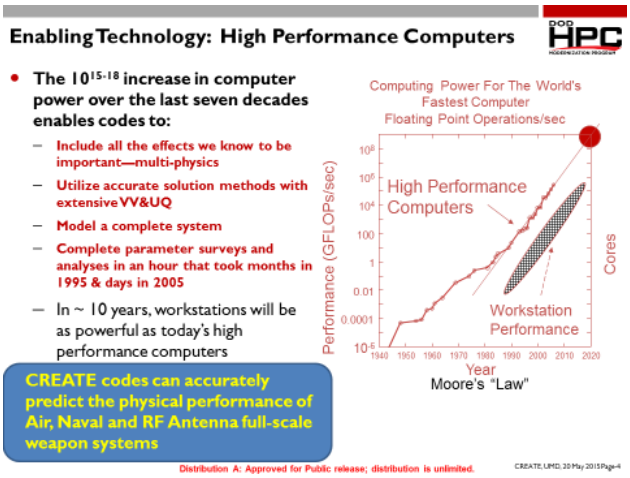
"Well, you know, I have this way of getting a potential model."

I then ask for more details. They have some idea of these particles bouncing around. Okay. F equals ma. That's good old Newtonian physics.

And then they go on. And I say:

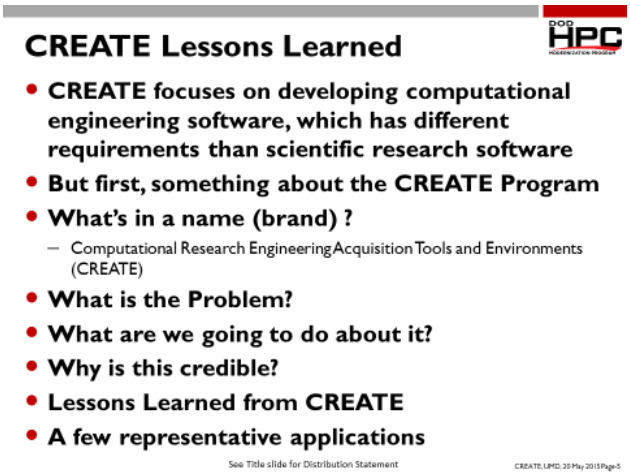
"Where does the quantum mechanics come in?"

They don't know. You really can't calculate atomic and molecular structure and bonding without bringing quantum mechanics. Again there is a big pause. The kids are really good at the craft, but many don't really understand what's going on in the code. Thus, I think we really have a lot to do in that area.



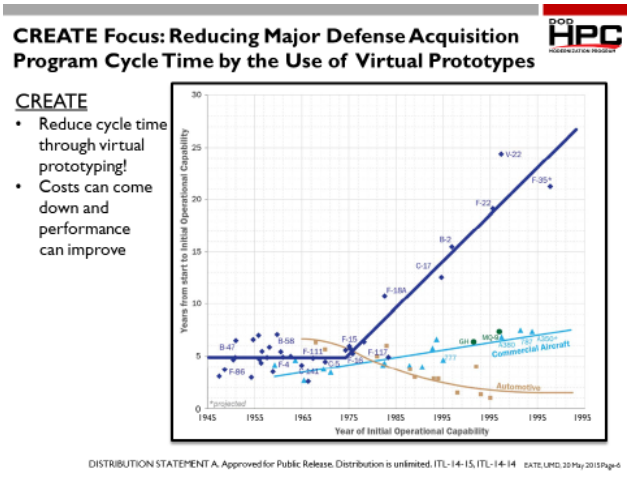
What's the enabling technology that is the basis for what we are trying to do? It's the growth in high performance computing. We're approaching the time where maybe we'll have an exa-scale machine in 2020. We already have peta-scale machines now. China has a 60 petaflop machine. In the United States, the Department of Energy has just let a large contract to produce 300 petaflops across two labs. This is a huge increase, given that we started with one flop right around World War II. We are at a point where we can include all the effects we know to be important. We can use accurate methods and have adequate resolution for the problems. We can model complete systems: a whole airplane, or a whole ship. We can complete parameter studies in hours, not years. Workstations are moving up to where they'll be incredibly advanced. My iPhone has more computing power than the computers we used to design the present nuclear stockpile. Now, it's possible to make accurate predictions of complex behavior of complex systems. We can actually compute how an airplane flies, including the propulsion, the airflow, and the modification of the structures due to those loads, and be able to change the control surfaces to control the flight of the airplane.

Right now, the development of such codes takes a multi-disciplinary team of 10 to 15 professionals about 10 years. We need to learn to do it faster and more cheaply, and the lessons-learned I will describe can contribute to that.



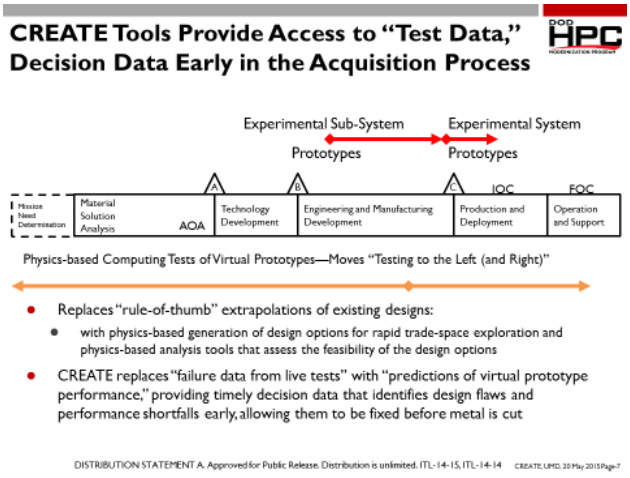
To make it work, you have to follow a code development process. We've been focused on that for the last 10 years. But first, something about the CREATE Program. What we're doing in CREATE, which stands for Computational Research Engineering Acquisition Tools and Environments, is transitioning scientific research into engineering practice. What do we know about being able to compute how an airplane flies? There are many good Computational Fluid Dynamics codes. There are many structural mechanics and structural dynamics codes. Allusions in other talks were made to NASTRAN, one of them. There are pretty good models for propulsion systems, and we know how to move control surfaces and things like that. But that's it. We don't have all these capabilities integrated into one code. Our goal in CREATE is to put them all together in an integrated multi-physics code, and produce a tool that aeronautical engineers and design engineers can use to predict and analyze the performance of real airplanes, in our case military airplanes.

The second issue is that you need to establish a brand. If you're going to do something like this, you'll really need to concentrate on the name, because you're going to have to sell your program continuously since it needs support. You want to establish a brand, get a good name, and have it resonate with people. Over my career I've had people put names together which were pretty awful. There was person working for me who wrote a code to calculate X-ray emission from accretion from white dwarf stars onto a black hole. He called the code CRETIN, not a good choice for a brand name. So I'm going to discuss what the problem is and what we're going to do about it, and why it is credible. This is what you really have to do. And then I'll discuss the lessons learned from doing this.

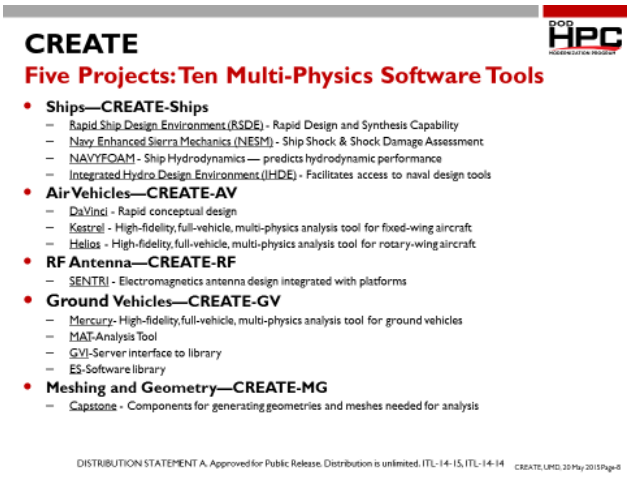


To focus on the problem, what we're trying to do with CREATE is to fix the acquisition process, i.e. reduce acquisition costs, schedule and risks and improve system performance. This slide shows the time it takes to develop and deploy new military aircraft. On the X-axis is the year industry delivered the first operating airplane of that type, and on the Y-axis is the years it took to do that. It took about five years to develop and deliver a new airplane up to about 1975, and then the time began to increase. Now we're looking at 20 to 25 years for the F-22, the Osprey Tilt Rotor (V-22), and the F-35. This is not sustainable! It reminds everyone of Augustine's 16th Law, "In the year 2054, the entire defense budget will purchase just one aircraft. This aircraft will have to be shared by the Air Force and Navy 3 ½ days each week, except for leap year, when it will be made available to the Marines for the extra day." (N. Augustine, Augustine's Laws, AIAA Press (1977), p. 107.)

Eventually, the next generation of bomber or fighter could take 30 or 35 years. The Soviets built six generations of surface to air missiles in the time that it took the US to build the F-22 . We have really got to pull this time back down. Commercial aircraft are still taking around for six or seven years. Automobiles are in a similar position. This is a real problem for the DoD. We've got to get the cycle time down.

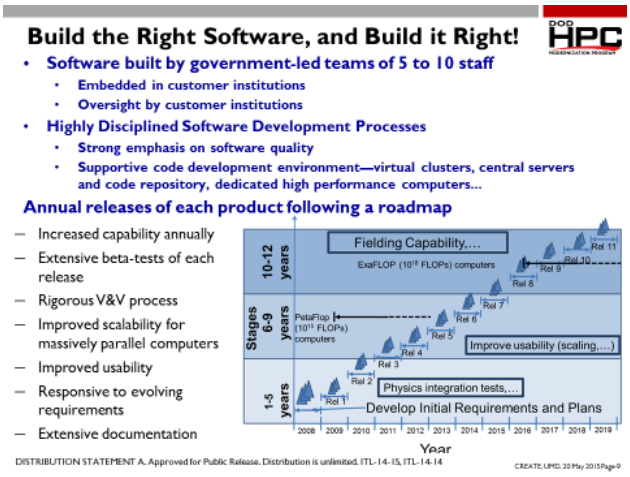


The way we're trying to do it is by using virtual prototypes. We're going to use our tools to generate virtual prototypes and then accurately predict the performance of those virtual prototypes. This will allow us to obtain virtual test data for the prototype, i.e. provide decision data early in the process. The next slide shows the DoD acquisition process. Essentially, the requirements are defined, candidate designs are developed and assessed, and then the technology is assessed and matured. Then detailed engineering is done followed by manufacture and production. You really don't get data until you've actually got physical prototypes for live tests, and in the acquisition process you only get full-scale prototypes around what's called Milestone C. But that's too late! If you discover problems with your airplane, and we do with every single airplane, then you've got a major problem. The plane is not going to work very well, and it's too late to make major engineering changes. You really want to get live testing early but you can't do that because it takes a long time to the point where you can build a physical prototype, and even longer before you have access to a full-scale experimental prototype. Virtual prototypes can be constructed quickly and then analyzed by high fidelity codes to predict their performance, a "virtual test". You first build a virtual prototype of the airplane. You then use software to test it which you can do that at the beginning of the process, at any point of the development process. You don't have to wait for tests. That means you can find and fix the defects of the design and the shortfalls in performance, and fix them before you cut metal. Of course the codes need to be extensively verified and validated to be credible.

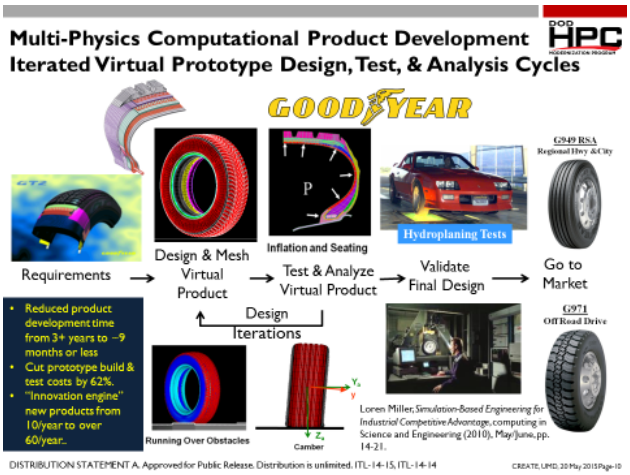


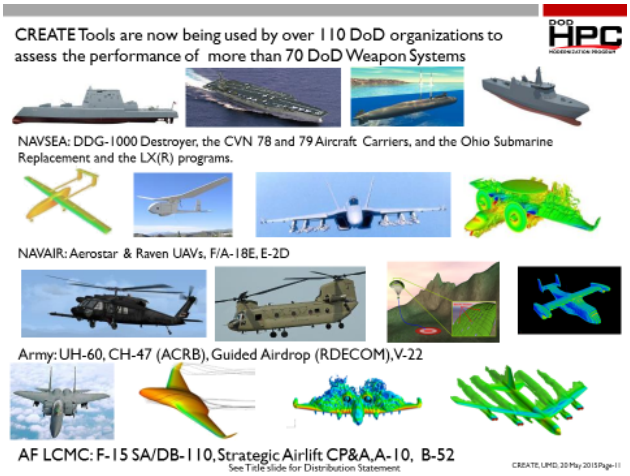
CREATE consists of five projects and thirteen Multi-Physics software tools. The CREATE-Ships tools are used to generate and assess naval ships, using the CREATE rapid design capability and detailed design analysis. Similarly, CREATE-AV tools are used to design and analyze military aircraft, using rapid design tools to generate conceptual designs, prototypes, and high-fidelity tools for detailed analysis of fixed- and rotary-wing air craft. CREATE-RF tool are used for the design of electromagnetic including Radio Frequency antennas. We recently added CREATE-GV

Ground Vehicles project. And you need a mesh to do any analysis, so we established the CREATE Meshing and Geometry project to make the meshes the analysis tools.



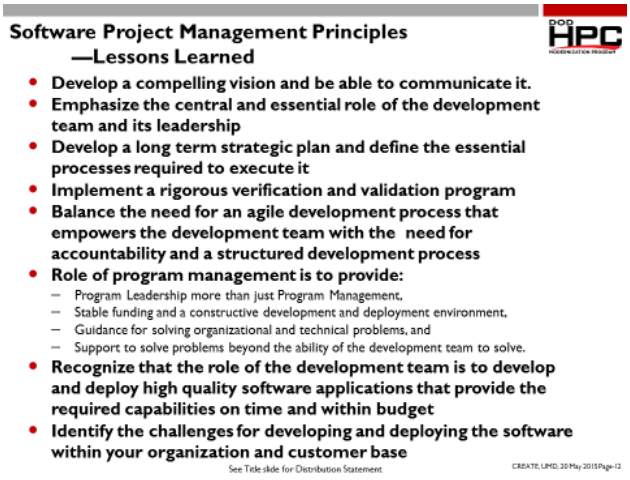
We not only build the right software, we build it right. It's built by government-led teams of five to 15 staff. The staff are embedded in the DoD customer organizations, which is very important. If you lose track of customers, you won't have any. The customers must be highly involved in this. We have a highly disciplined software development process, with a lot of emphasis of software quality and supportive code development. You also must have an infrastructure that supports code development. We also have a good release process that results in a release every year. Since it takes about ten years to put these kinds of codes together, it's important to deliver code capability early, in the first few years, and then continue delivering new capability every year. Keeping your support for that length of time is a challenge in Defense Department.



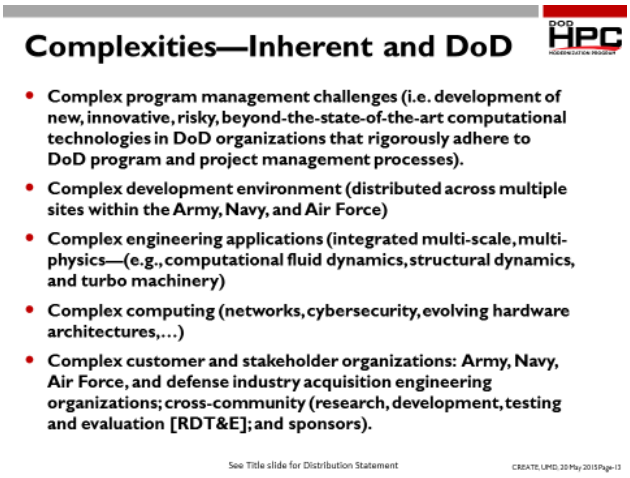


Why is this approach credible? This paradigm has been used in nuclear weapons programs since the 1950s (Francis, S., *Warhead Politics, Livermore and the Competitive System of Nuclear Weapon Design*, 1995, Lawrence Livermore National Laboratory: Livermore, CA. p. 247.) The nuclear weapons designer only got a handful of tests, maybe three of four, to do their whole design. That means that a lot of exploratory design and testing was just not going to happen. So they adopted (actually invented) the virtual prototyping paradigm. The virtual prototyping paradigm involves generating a numerical model of the product geometry, building a mesh from that geometry, then using a multi-physics high fidelity code to predict the performance of the system. The goal of computational engineering is to get a competitive advantage, which means it usually becomes a trade secret so you don't really hear about it much. Livermore and Los Alamos were not interested in in telling the rest of the world how to design nuclear weapons. In fact, the details were (and still are) highly classified for very good reasons. Similarly, Goodyear Tires put together a tire design tool with help from the Sandia National Laboratories(Miller,

L.K., *Simulation-Based Engineering for Industrial Competitive Advantage.* Computing in Science and Engineering, 2010. **12**(3): p. 14-21.). They aren't interested in telling the rest of the world how to build such a code. Nor does Pratt-Whitney tell you how they use computers to design jet engines, and so forth. Goodyear used virtual prototyping to reduce their product development time from more than three years to nine months. Dropping the time to market by a factor of four gave them a large competitive advantage. They also were able to cut their prototype testing costs and increase the innovation rate from the normal 10 products per year to 60 products per year. Slide 11 shows some examples of successful CREATE applications.

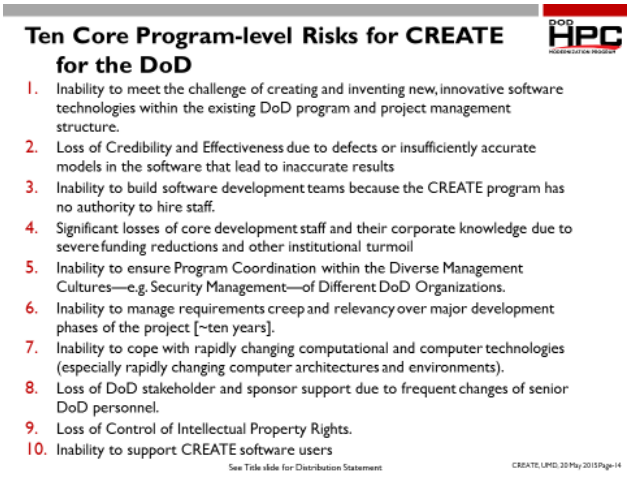


Virtual prototyping is a credible concept. So what is the process for developing the software applications required to implement the process? There's a set of software project management principles that you can use displayed on the slide 12. You must have a compelling vision and be able to communicate it. You must be able to explain to people what you're doing and why: you must be able to explain it to your team, and to the sponsoring agency. You need to emphasize the central role of the team and its leadership. That's the key. If you want one predictor of whether a project will be successful or not, it's whether you picked the right team leaders, and whether they can pull the right team together. Verification and validation is crucial. It's essential for success. You've got to balance the need for a structured development process together with accountability; the team has to know they're on the hook to deliver something. But you need to cut them enough slack and give them the flexibility to develop and deploy the codes. The role of program management is to provide leadership and support, stable funding, guidance to avoid problems, and help for problems the team can't solve. Likewise, the team has to understand their job is to develop and deploy software and provide the capabilities that are needed on time and on budget. Finally, you must tailor your management and development processes to your organization and your customers. You and the development teams have to be very agile.



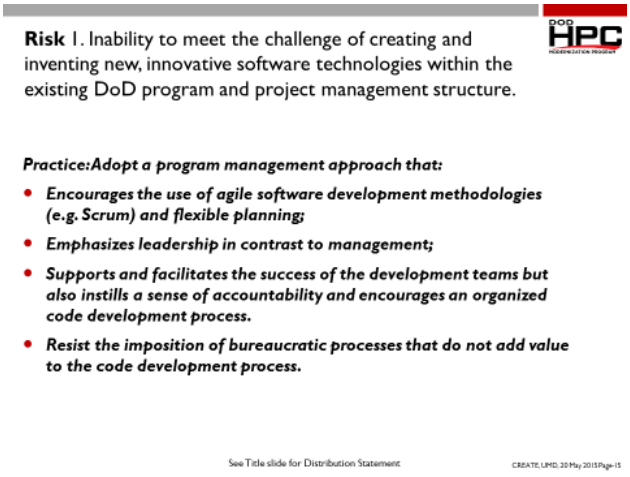
Inventing new, complex technical software in the Department of Defense or any large, established organization is really challenging. However, we've succeeded using our approach. There are a lot of complexities in our world starting with complex program management challenges. We're trying to invent codes in a system that's not designed to invent things. It's a struggle for large, mature organizations to be innovative. It is a very complex development environment. You have to invent codes and solution methods to run computers that don't yet exist. These types of code are complex technical and engineering applications. The technical challenges are pretty

major, and the computing is pretty complex. And ultimately, you have to put this back into the defense industry and the services. Thus, it's a very complex environment we swim in. Your organization won't have exactly the same challenges, but unless it's very small it will have a lot of them.

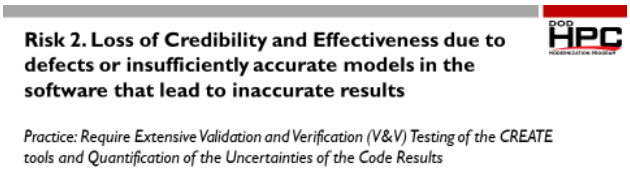


Recognizing the complexity of what we were trying to accomplish, we identified the ten major risks that we faced in developing and deploying complex technical software in the Defense Department. The first is the risks due to the challenge of developing this kind of software in a very large, mature organization. Software that produces inaccurate results (because your codes didn't have enough validation or verification) will cause your program to lose credibility. Next, the CREATE Program needed to recruit and retain software development teams. The CREATE Program was allocated no government civilian job slots, and had no authority to hire people. It thus had to rely on recruiting staff assigned to the CREATE program from existing DoD organizations and support contractors for those organizations. Stable funding and minimal institutional turmoil is essential. If the development teams perceive that the funding base is unstable, they will seek employment with another organization that does have stable funding. The employment market for highly skilled engineers with good computing skills is highly competitive, with much higher salaries and less bureaucracy than the DoD. The diverse cultures among the Army, Navy and Air Force introduce challenges for coordinating the code development among the multi-institutional distributed teams. Computer security restrictions lead to additional challenges. For good reasons, every military base severely restricts the transfer of computer data across the site boundary. Yet, development of complex software by distributed, multi-disciplinary teams requires rapid and reliable data transfer among the three or four participating organizations. And of course, the standard rule of thumb is that requirements change about one percent a month. That means, during the development phase of your program, about ten years, the requirements will have changed substantially and the software development requirements and deliverables will have to track them. Additionally, the computing environment is changing radically right now. Next generation computers will have very different computer architectures and the CREATE codes will need to run on them as well as on current computers. Then there's a lot of turnover at the top in the federal government and in the DoD. Political appointees hold positions for two to four years. Senior service leaders, generals, admirals, and colonels will be reassigned in two to four years. The program management and the software developers must continually justify their program the upper level management. Intellectual property is a major issue for software. If you are going to distribute software, you need to "own" it, i.e. have the legal right to control the distribution. I had never thought that as a physicist I would have had to worry all that much about intellectual property, but it's a really crucial issue. You've got to own and have cognizance and ownership of every bit of code. And of course, once you've got a successful product out there, you've got to support the user community, otherwise it's useless.

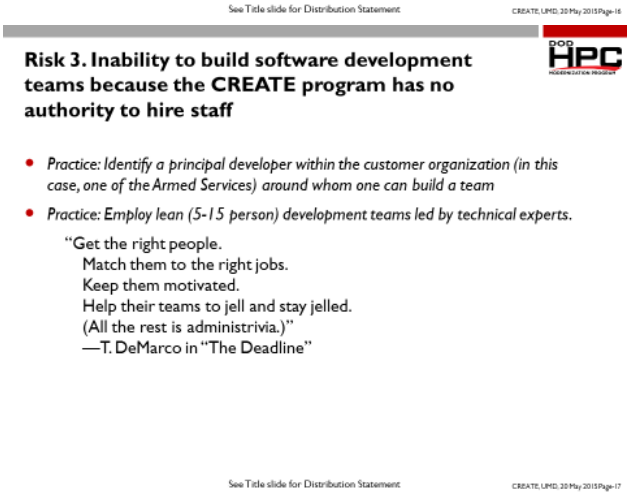
For each of these risks, we developed a series of practices to reduce and avoid the risks and mitigate them if they are realized.



Our approach for creating new, innovative, technical software in a large, mature organization such as the Defense Department, with many goals that have higher priorities than our software development projects, is to strike the right balance between the need for agility and flexibility so that the teams can successfully develop the complex software with the need for accountability and a disciplined code development process to ensure software quality and delivery of software more or less on time within budget. We encourage the use of agile development processes, emphasize leadership in contrast to management, and resist the imposition of non-essential bureaucratic processes.

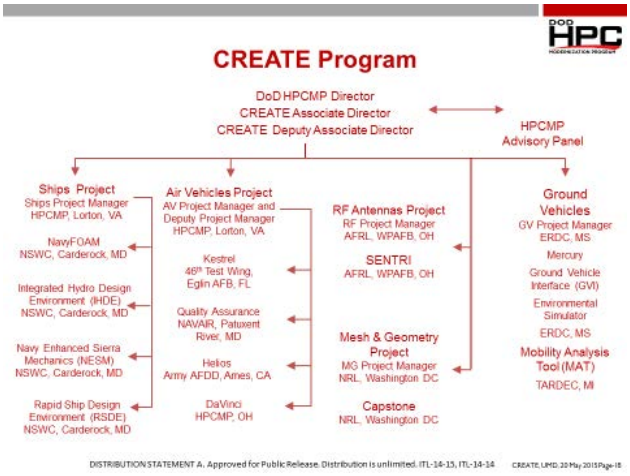


For ensuring credibility and effectiveness, the solution is extensive validation and verification.

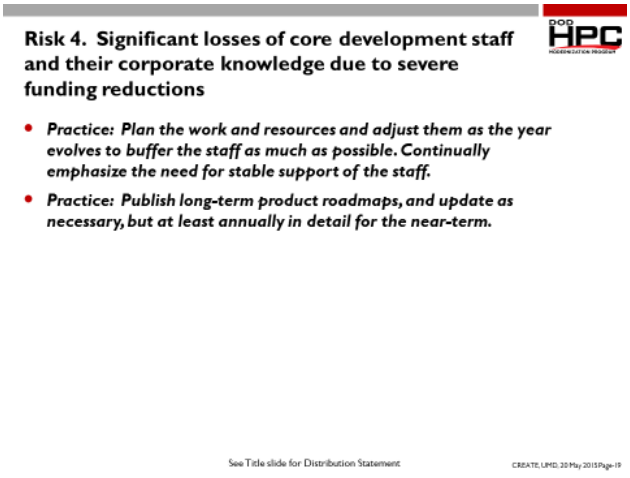


Without the authority to directly hire the development teams, we made lemonade out of our lemons. We first identified the DoD acquisition engineering customer organizations that had the expertise to develop the software and who had responsibility for executing the design and analysis of the weapon systems targeted by the CREATE tools. Within each customer organization, we identified a principal developer, who possessed: extensive domain knowledge in at least one of the major areas and could understand the salient points of the rest of them, good computational and program management skills, the ability to represent the project to senior management and to the team, and could recruit and build a team and inspire the members of the team. Typically, you don't find all of that in one person, but you should come close as close as you can. Successful

team managers really get a career boost when they are successful. The overall competence of the team leader is the biggest predictor for a successful code. If you haven't got the right person to lead this, it's dead. Only with the right team leader do you get a good team that can work well together. There's a great book in the form of a novel entitled The Deadline by Tom DeMarco (Dorset House, 1997) about code development and software development that I highly recommend. Most project management books are a sure cure for insomnia, but this one you can actually pick up and read right through. DeMarco states that the path to success is "to get the right people, match them to the right jobs, keep them motivated, and help the team jell and stay jelled. All the rest is administrivia." My experience is that he has correctly identified the most important element of successful software projects.

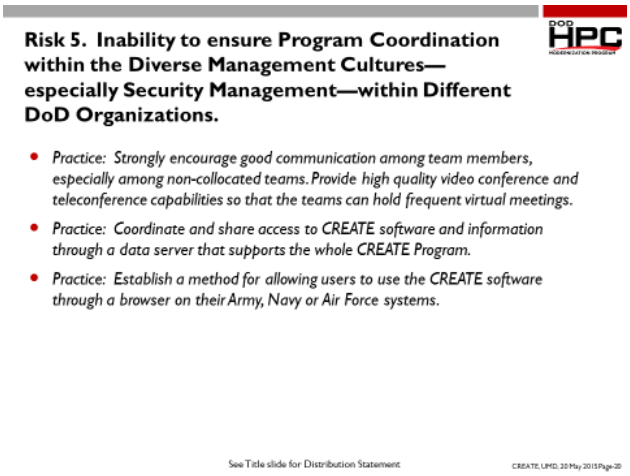


The CREATE program is a distributed, multi-institutional multi-organizational program. As noted, each of the codes is located in this customer organization, and the leader is someone who is a senior person in that area, and has responsibility for leading that area in that particular part of the Defense Department. As seen in slide 18, there are senior program manager for ships and for air vehicles. The Ships Project is led by groups at the Naval Surface Warfare Center at Carderock, because that's the organization the Navy relies on for Ship design. The Air Vehicles Project has elements in the all the services, because each of the services has an aviation program. The antenna program is led by the Air Force Research Lab Sensors Directorate, which is the informal coordinator for electromagnetics in the Defense Department. There is a meshing project at NRL, because a mesh is the starting point for a detail analysis of a weapon system. The Ground Vehicles Project is located at the major DoD organizations for ground vehicle design, the Army Corps of Engineers Engineer Research and Development Center, and the U.S. Army *Tank Automotive Research Development and Engineering Center*. It was started around 2013 and hasn't yet had a release.

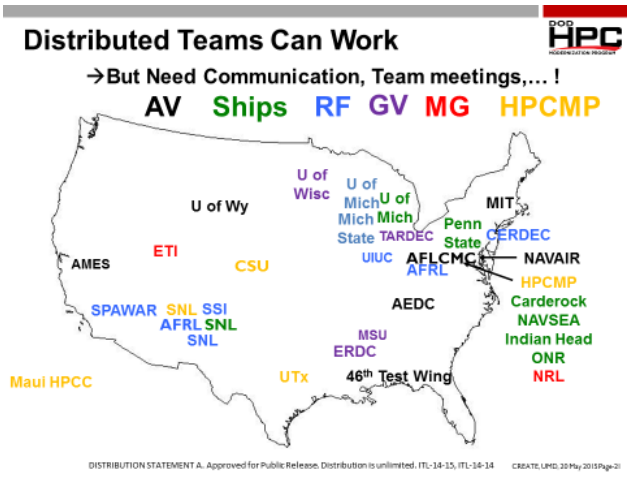


The CREATE development teams are more than 90% of the assets of the CREATE Program. Recruiting and retaining the team members is essential for success. Our approach is to protect the team from institutional turmoil, funding uncertainties, interruptions, etc. as much as possible. Developing code requires uninterrupted, very concentrated work. People have to sit in front of a terminal and work, and concentrate to get into it. It is intellectually demanding, and can't be done if the developers are constantly being pulled away from their work. That's difficult because there's always some sort of crisis. You also absolutely cannot plan out a ten year detailed code development

program. In my experience it can't be done. So you want a roadmap of where you're going to go, things you want to deliver, an ultimate goal. You need pretty detailed plans for the next six months. Less detailed plans for a year, even less detail for two years out. For longer term goals, a general vision of the desired capabilities is generally adequate. We developed 12-year roadmaps, and updated them every year as new requirements emerged. Now we are using a rolling 5 year roadmap.



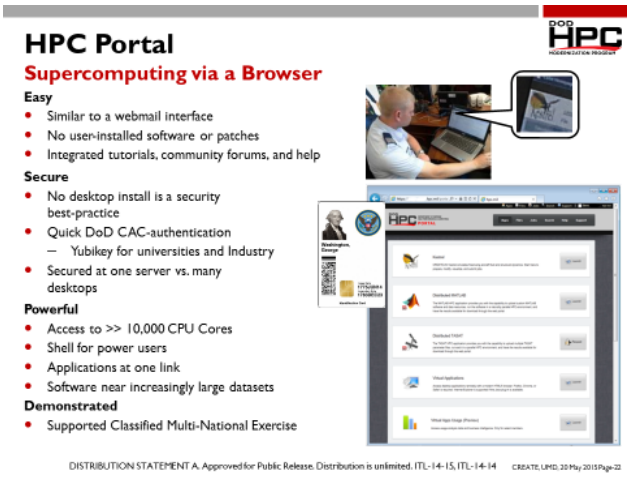
It's essential to cope with computer security requirements within the different DoD organizations. The DOE and other federal agencies have a similar set of issues as do corporations. We are working with Sikorsky Helicopter. Sikorsky has no interest in letting any of their IP, or any other of their information, offsite. We install the codes on their computers, and they run our codes on their computers, and we don't really know what they do with them because it's their intellectual property. However we work with them, and make sure the codes are validated.



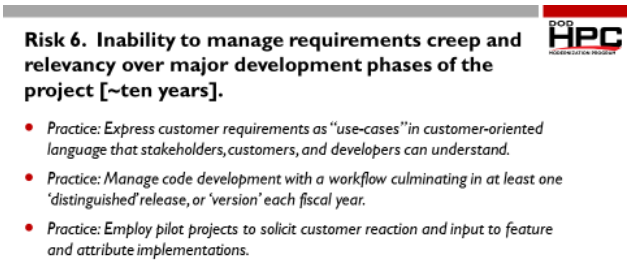
The map of the US on slide 21 shows the location of the 29 organizations participating in CREATE. The CREATE Program Office needs to coordinate all of them. Three years ago, the Defense Department decided that for a year or more due to the GSA Las Vegas conference scandal complicated by sequestration and furloughs, etc. that no one in the DoD civilian sector could travel. But "management by walk-around" is the most effective way of keeping things going, and if you can't travel to talk to people, except over the telephone, it's pretty tough to keep track of things. We utilized really good video conferencing capabilities, something I would suggest for everyone. You have to have the security people sign off on it though, and that's always an issue. In addition you need a central data server which supports the whole program, where you can create a document repository, and configuration management systems where you can check in your code. Blogs and forums are useful to allow developers and user to document their exchange on views and experienced. Issue trackers help keep the work coordinated. A central server accessible to the whole development community has allowed us to keep things together, and allows groups to work together and stay coordinated. The users have access to the document repository, the test plans and data, and the forums and blogs. This helps them support themselves.

The final issue is perhaps unique to the DoD. Most of the engineers in the defense department have a Windows PC (usually a laptop), Microsoft Office, and a browser. They are also not expert supercomputer users. How much engineering can they do with Excel? Not very much. All the

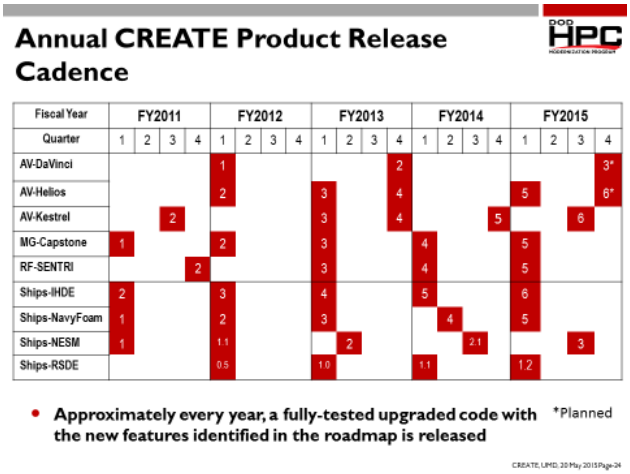
engineers told us that all the software we're developing is really great, but security restrictions will not allow them to access a remote supercomputer.



To solve this problem, we put together a software interface we call a portal installed on the HPCMP supercomputers. It allows the users to employ a browser on their local Windows computer (or other system) to securely access remote supercomputers over an encrypted network connection with two-factor authentication. Now users can employ their browser to log onto the supercomputer, set their problem up, run it on the supercomputer, store the results, then analyze the results and ship the pixels back to their laptop. This also means that they don't have to ship the large datasets that were generated from the run over the network to their local computer, a big advantage. This works very well, and it helps with industry too, because company networks for defense contractors like Boeing, Raytheon and Lockheed, have their own network security protocols and restrictions so that access for them is often problematic without the CREATE portal.

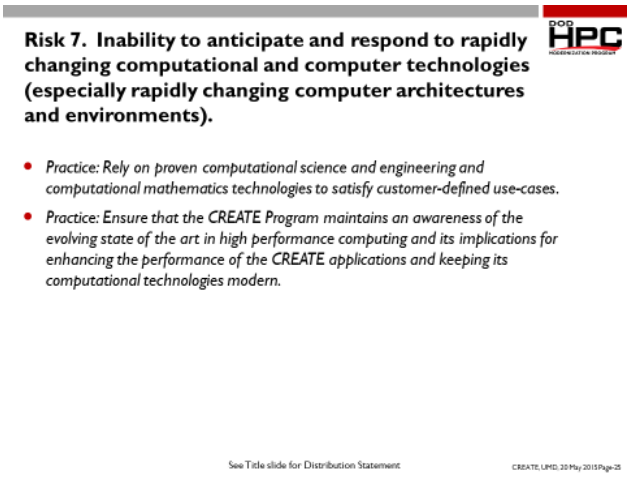


Managing requirements is a major challenge. The Defense Department has a very elaborate system for tracking requirements. We decided on a streamlined process based on the standard software development process of Use Cases: define exactly what the code needs to do, and test to ensure that it fulfil the use case. Use cases allow you to explain the requirements in a way that can be understood to the developers, the users and senior management and measure and document achievement of the requirements.

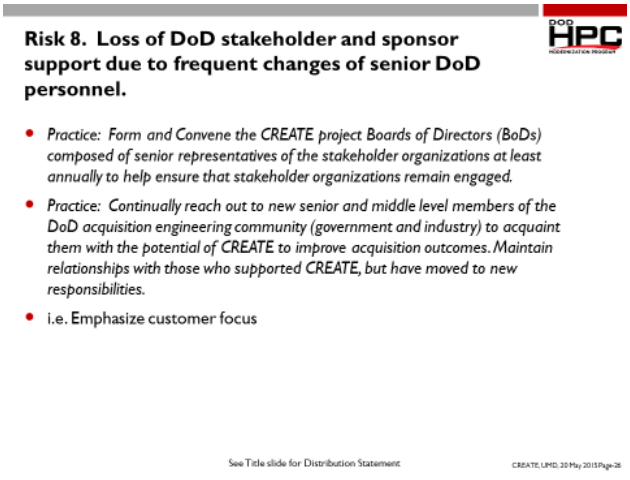


Annual releases are one of the most important development and management practices. This is FY2015, and as seen in the chart on slide 24, the CREATE codes have established a good cadence. The development of the CREATE software started in 2008, and by 2011 we were releasing out the first versions of the software. And every year, in almost all cases, there has been an annual release of each code with increased capabilities. The release went out to the customers, they tried it and used it, they were able to get access to these capabilities, and they reported their experiences (good and bad) back to us. The code development groups reached closure each yare. They didn't hang around for

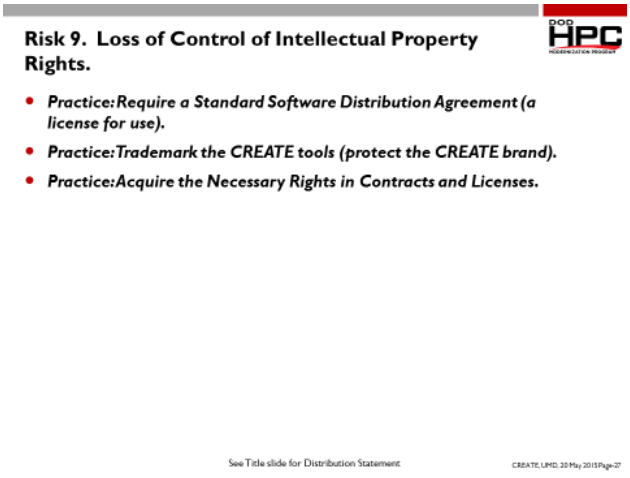
ten years trying to get things going. They had to get something out in each year and move on. The customers got a crack at it to tell them what was wrong.



Another important principle is: "Don't do research unless absolutely necessary." The development of computational engineering tools is about capturing mature scientific knowledge and making it accessible to the engineering community. I've done research for much of my career, but you can't do fundamental research if you're putting together engineering tools. You need to take sound, mature algorithms and methods and implement them into practice. However you will need to do some research because you can't anticipate everything that you will need for the code to solve the specified problems. You also need to keep track of what's going on in computer architectures because you must be able to keep your codes working on present and next generation computers.



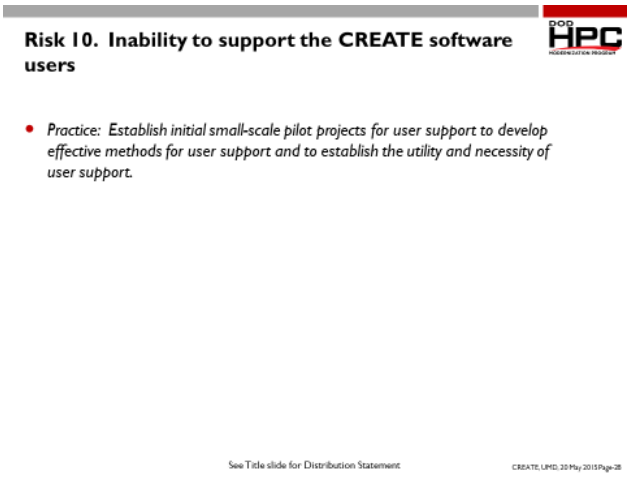
The turnover rate of the senior people in the Defense Department is noticeable. That's the way the world is, so you have to adjust. You have to make it your job to educate the new management as they rotate in every few years. In the DoD aviation engineering acquisition community, we've gone through three major changes in leadership at Wright-Patterson. You educate one senior leader, and two years later, you got to convince their successor. In our office, we're on our third director. Coping with the revolving door for senior leaders is a challenge that must be met.



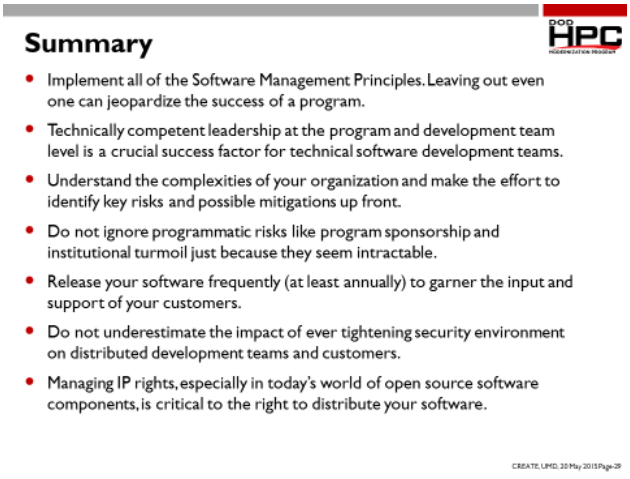
We've established Boards of Directors for each project to ensure senior oversight, accountability and stability and establish a close connection to the relevant Services Engineering Acquisition Communities. Each of the Boards is composed of senior people from the service acquisition engineering communities. This provides stability, customer outreach, and advocacy for each CREATE project.

With regard to intellectual property and standard distribution agreements, trademark each code. This provides some protection

against misuse by others. Additionally, make sure you have a legal subject matter expert for intellectual property issues on staff to review contracts, distribution agreements and licenses, and make sure you have the legal ability to distribute your codes. I live in fear of this because we've had some near misses with IP issues. A bright young engineer, a code developer, one code team thought: "Why should we spend two months developing some code to handle a specific task on my own, when I can just grab it from web and pull it in and use it and save a lot of time and effort?" The reason that you can't do it is because if somebody else wrote the code, they have the rights to it. Then you've got a time bomb in your code which eventually can cause major problems. You need to ensure that you get all the necessary legal rights to distribute the code.

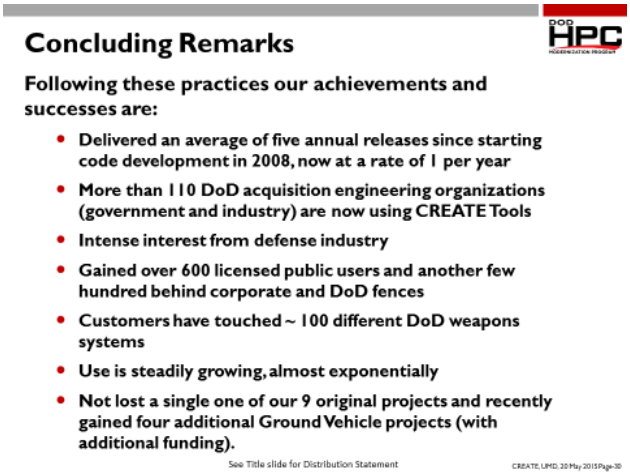


If you built the software, but can't support the use of the software, then no one's going to use it because they will not be able to use it with confidence or even get it to run. So you've got to figure out how to provide support. We've established support groups for each code, and emphasized self-support by establishing good documentation for users, live and on-line tutorials, user blogs and forums that users can query and search to find out if anyone else has encountered their problem, and what that person did to fix it. We've done small scale pilots to learn the best ways to provide support. We are now supporting a user group of close to 800 users without taxing the development groups to an intolerable degree. Since user support scales with the number of users, we have asked the customers (the Army, Navy and Air Force acquisition engineering organizations) to pay for user support, and we're actually getting the services to contribute staff for user support.



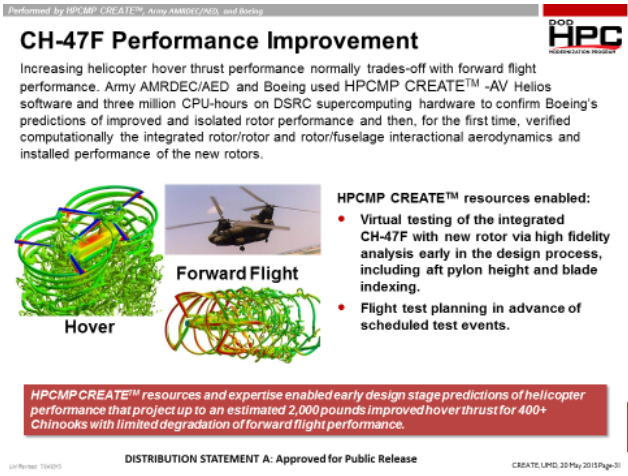
In summary, I recommend following all of the practices I've described. By following these practices, we've had more than five annual releases since we started and are rolling along at one release per code each year. That level of achievement, in seven to eight years, is pretty amazing to me. Over 110 DoD organizations, both in government and in industry are using the codes. We've have intense interest from industry. In fact, some major Aerospace industries are assessing the use of our software for use in their commercial as well as their military design and analysis processes. For the major US Aerospace firm to be interested in adopting our tools for use in their design processes, is a strong validation of our vision and our ability to execute. We now have over 600 licensed users, and another few hundred are out there, hiding behind corporate firewalls and DoD classified programs. Customers have touched about 100 different DoD weapons systems. Use is growing steadily, almost exponentially. In fact, it's growing faster than I'm comfortable with given our ability to provide support. The biggest metric we've got is that we started out nine original projects

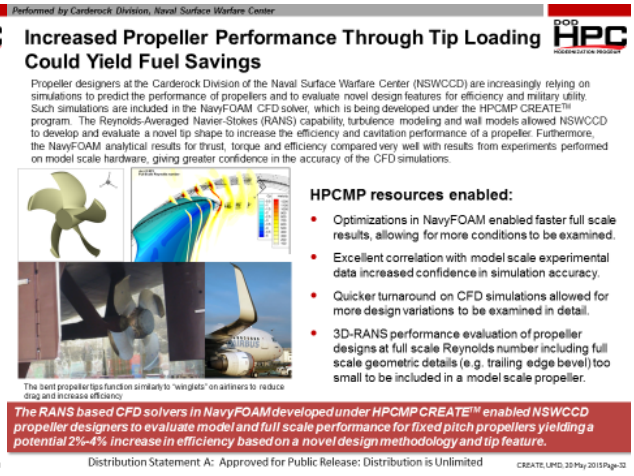
and today we have nine projects still going on. The standard success rate for this kind of code development when I look across the history of computational engineering codes is between 30 or 40 percent. Over 50% of similar development projects development efforts have failed in the past. We tried to learn from the mistakes and successes of the past and it's worked well. We haven't lost anybody.



I have included two success stories just to illustrate the kinds of calculations the CREATE codes can performance. The CREATE high fidelity rotary wind code, Helios, was used to assess a proposal by Boeing to use an improved set of rotor blades to increase the lift during hover for the CH-47F by 2000 pounds (about 10%) with no degradation of flight performance. NavyFOAM, the high-fidelity ship hydrodynamic code, was used to assess the performance advantages of putting winglets on propellers, for both submarines and ships. The preliminary estimate of the NavyFOAM analysis is that it would decrease fuel use by 2 to 4%, and reduce the vibration level.

Thank you.





Seeking a sustainable model for scientific simulation beyond the exa-scale

Dr. Robert J. Harrison
Director – Institute for Advanced Computational Science, Stony Brook University
Stony Brook, NY
Director – Center for Computational Science, Brookhaven National Laboratory
Upton, NY

Abstract

As we progress toward and beyond exa-scale computation, two disruptive changes are derailing our progress toward realizing the full potential of HPC in urgent imperatives such as designing sustainable energy technologies, understanding and controlling our environment, managing our society, and growing our national economies. First, while the high performance made possible by massive distributed-memory systems, multi-core processors with specialized vector instruction sets, and GPU architectures is a huge boon to the HPC community, achieving portable performance across different systems is virtually impossible today, and tomorrow brings new complexities and architecture shifts.

Second, our ambitions for scientific computing are leading us to ask increasingly large and complex questions that already cause the corresponding complexity of our software to exceed the capabilities of current programming systems. For instance, the huge equations of many-body physics and chemistry have transcended human ability to translate directly into software, and much modern science and engineering is at the interface of disciplines forcing the composition of multiphysics applications with diverse numerical representations, solvers, data structures, and software suites. Previously successful strategies for maintaining productivity and performance, such as frameworks and expert-written libraries, have been undermined by the disruptive pace of change in architecture and programming models, which will continue and even accelerate for the foreseeable future.

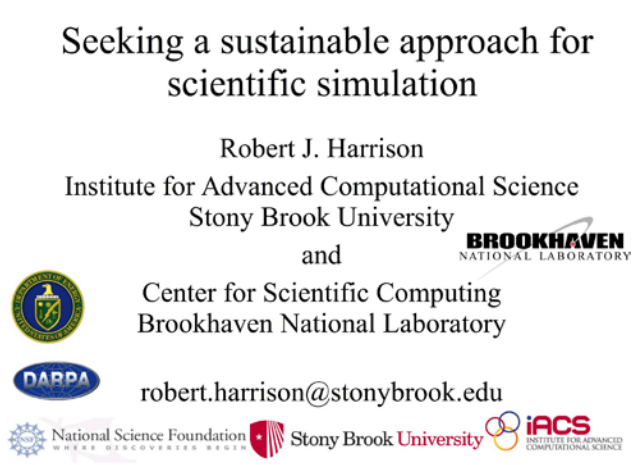
As a result, many are now questioning whether our current approaches to developing software for science and engineering are sustainable. Can we deliver to the world the full benefits expected from high-performance simulation at the peta and exa-scales? Or is innovative science being stifled by the increasing complexities of all aspects of our problem space (rapidly changing hardware, software, multidisciplinary physics, etc.)?

Focusing on applications in chemistry and materials science, and motivated by co-design of exa-scale hardware and software, I will discuss many of these issues including how chemistry has already been forced to adopt solutions that differ quite sharply to those in the mainstream, and how these solutions position us well for the technology transitions now underway.

Finally, producing a software infrastructure for computationally intensive science and engineering that overcomes these technical challenges advances, and yet is sustainable long-term, will require unprecedented cooperation between the scientists within a science domain as well as across disciplines/activities including computer architects, applied mathematicians, computer scientists, resource providers, and educators.

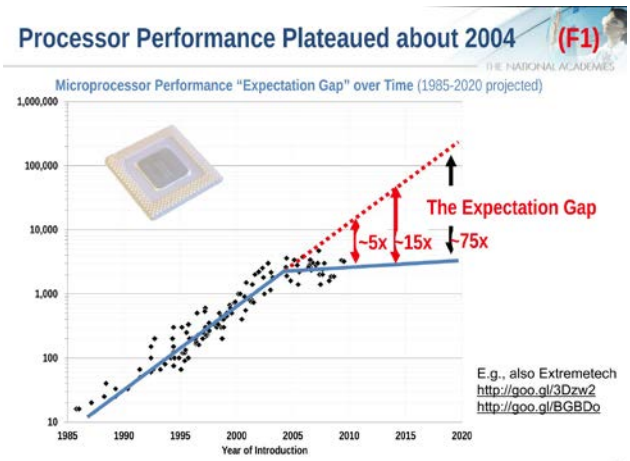
Biography

Since October 2012, Robert J. Harrison is a professor in the Applied Mathematics & Statistics Department at Stony Brook University where he also directs the new Institute for Advanced Computational Science. He is jointly appointed with Brookhaven National Laboratory where he directs the Center for Computational Science. Previously, he was director of the Joint Institute for Computational Sciences (JICS) at the University of Tennessee, Knoxville (UTK) and Oak Ridge National Laboratory (ORNL), with an appointment in the Department of Chemistry at UTK. JICS is home to the National Institute for Computational Sciences (NICS), one of the National Science Foundation supercomputer centers. He has many publications in peer---reviewed journals in the areas of theoretical and computational chemistry, and high---performance computing. His undergraduate (1981) and post---graduate (1984) degrees were obtained at Cambridge University, England. Subsequently, he worked as a postdoctoral research fellow at the Quantum Theory Project, University of Florida, and the Daresbury Laboratory, England, before joining the staff of the theoretical chemistry group at Argonne National Laboratory in 1988. In 1992 he moved to the Environmental Molecular Sciences Laboratory of Pacific Northwest National Laboratory, conducting research in theoretical chemistry and leading the development of NWChem, a computational chemistry code for massively parallel computers. In August 2002 he started a joint faculty appointment with UTK/ORNL and became director of JICS in 2011. In addition to his Department of Energy Scientific Discovery through Advanced Computing (SciDAC) research into efficient and accurate calculations on large systems, he has been pursuing applications in molecular electronics and chemistry at the nanoscale, and with support from the National Science Foundation he has been working toward making general numerical computation on high--- performance computers much more accessible and scientifically productive. In 1999 the NWChem team received an R&D Magazine R&D100 award. In 2002 he received the IEEE Computer Society Sidney Fernbach Award, and in 2011 another R&D Magazine R&D100 award for the development of MADNESS.

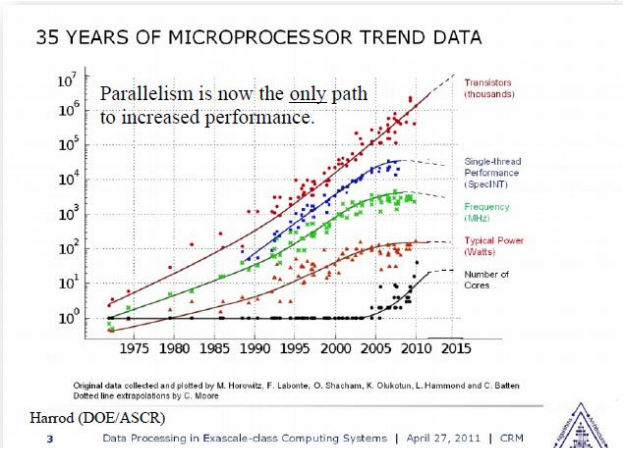


Thank you very much for the welcome, and thank you Peter for the invitation and also organizing this event, which is focused on a very important and very current topic. Brookhaven has a long and very broad history of leadership in scientific computing with a strong tie to the defense agencies. I come primarily from the Department of Energy side of things with an interest in fundamental physics and chemistry. Now, at the Institute for Advanced Computational Science (IACS), we have a broad prospective on computing. In IACS, we have faculty from sociology and atmospheric sciences. Of course, many people cross doing materials and chemistry. We are actively recruiting in the space of computer science and applied math where we are trying to build up a stronger foundation in the techniques that enable computation to happen. This is precisely directed towards the main concerns that I'll be talking about today, which is how do we keep the scientific computing enterprise alive in the face of all of this change and everything that is happening. My talk is going to focus on what technology is doing and how do we have to respond to that, because the future is not as the past was. Most people in the audience, especially those who compute regularly, will be aware of this.

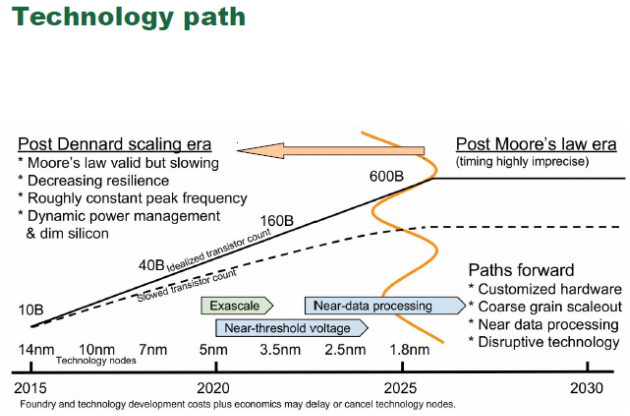
This is a very nice report, and even though it's now about four years old it's still very fresh and current. If you haven't seen it, it's well worth looking at.



One of the main drivers is on this slide from the underlying presentation from that report, which outlines how processor speed and frequency has not really changed much over nearly a decade, basically because of fundamental physics limiting power characteristics of processors. We simply cannot continue to operate processors faster and faster. As people make chips bigger, what we're getting are more cores, not getting faster processors.



The technical term, if you're familiar with Moore's Law, is called Dennard Scaling, which basically says if you make the features on a chip smaller, you could operate the chip faster. We're not able to do that anymore. Looking forward in time, we're not getting faster processors, what we're doing is getting more processors, and different types of processors.



Moore's Law is, in the end, not a law, but a self-fulfilling prophecy that basically says every so many months or years, the number of transistors on a chip will double by making the features on the chip smaller. But of course, you don't get to do that forever. Ultimately, you'll reach some fundamental level of physics. If you look back, this has continued over an enormous amount of time. It's simply ingrained into everyone's understanding about computer technology. For our entire professional lives, Moore's law has really held. Exponential growth is unfamiliar to us for most other contexts. Most peoples' experience with exponential growth is probably when they get infected with a cold or flu virus. That thing grows over a few days and then suddenly overwhelms you. Moore's Law has been continuing in that vein for a few decades. Basically, the party is over. Sometime after the year 2020, it's going to stop. What does that mean going forward?

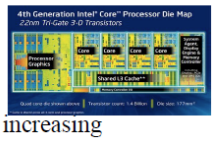
The parallel future

- Science students commonly not prepared for this future
  - Taught sequential not parallel programming
  - Little awareness of performance or architecture
- Many computer programs written now will live for at least 10 years into the future
  - But designed to run on computers from 10 years past
- Individual research groups, small institutions, disciplines new to computing, most companies, do not have the awareness, skills, resources, to navigate this transition

We're clearly not preparing our students for this future, since we ourselves are not prepared for it. But the computer programs that Doug is talking about, and something that I have been working on and everyone in the room has been working on, are going to live a long time. One of the main codes that I've worked on for much of my career started in 1992: NWChem, a large quantum chemistry materials code. It's now over 20 years old. It certainly wasn't designed to run on computers of the current era, because we simply couldn't imagine them back then. We thought we were being bold in designing these programs to run on ten thousand processors. But we didn't envision nodes with thousands of cores inside a single node. We didn't imagine GPUs. We certainly didn't imagine what's going to happen in the post-Moore's Law era. The skills, and even the awareness simply don't exist in most groups: certainly not in most university groups, and probably not in most software engineering groups.

Looking forward – no escape

- It's not just the core count – it's total concurrency
  - SIMD width, pipeline depth, multi-instruction issue, ...
- What is the lifetime of your code?
  - 2012 – Intel KNC 64 cores/socket
  - 2016 – 128+ cores / socket
  - 2020 – 512+ cores / socket
  - Vector length and # pipelines also increasing



- In 2020 1+M cores will be a campus resource
  - I.e., exascale technology is relevant to you

7

There's no escape from this. The path towards exo-scale computing is an important topic. Doug mentioned peta-scale computing, with computers that can perform at 10^{15} operations per second. But these computers are in hand now. We're progressing through computers that are capable of tens of petaflops. The Department of Energy is in the process of fielding three computers capable of roughly somewhere between 150 to 250 petaflops individually at a collective price tag of close to half a billion dollars. Those are going to be hitting the ground sometime in 2018 or 2019. That's going to get us on the path to exo-scale computing, which is 10^{18} operations per second. This is just a path going forward. People speak of exo-scale computing as if it's a point in time, but it's not: it's a pathway to the future. People think most supercomputing technologies are somehow different from what people would be computing individually. Most of us aren't going to get to use these big supercomputers, so most of us will be computing on personal computers. What do you think is going to be in these personal computers? It's the same thing that's going to be on the supercomputers, because it's all commodity technology. There's no custom technology in these big computers anymore. A lot of the same concerns that we have in mobile devices are the same concerns that we have in the big supercomputers, because it's all about power consumption, battery life, and high performance. These are contradictory things almost. The codes live a very long time, which also means the underlying technologies will be changing. Creating an exo-scale computer would translate into peta-scale computing for everyone else. If an exo-scale computer fits into a hundred racks inside a large computer room, then a peta-scale computer would be a thousandth of that size, which means that a typical university could easily afford several of these things, and they

would fit in a small box. That's coming, and it's coming very soon, but it's going to be with technology that reflects that.

Do new science with

O(1) programmers
O(100,000) nodes
O(100,000,000) cores
O(1,000,000,000) threads & growing

- Increasing intrinsic complexity of science
- Complexity kills … sequential or parallel
  - Expressing concurrency at extreme scale
  - Managing the memory hierarchy
- Semantic gap (Colella)
  - Why are equations O(100) lines but program is O(1M)
  - What's in the semantic gap – and how to shrink it?

8

Somehow, with the skills and the few talented individuals that we have, we have to corral the hordes out there. Gamers might recognize this shot on the slide of a popular Microsoft game, with a bunch of zombies to kill in the background. That's basically the massive threat, the complexity, the parallelism that we have to overcome. That's the challenge of any software enterprise: overcoming complexity. In the end it doesn't really matter where the complexity comes from. But our problem in high-performance computing is that first phrase: high-performance. We want high-performance. High-performance has become a level one correctness issue. And that translates into all of these horrible details of the underlying architecture coming up to haunt us in ways that they don't haunt other parts of the software industry that aren't so concerned with performance. Another aspect is a phrase that I attribute to Colella: why is it that our equations are relatively compact, and yet our computer programs are so big? NWChem, depending on how you measure it, is somewhere between three to five million lines of codes. Yet we can write down the equations at a very high level for quantum mechanics on a few pages. What's up with this? What's this semantic gap between how we think about and reason about science and how it's implemented?

Wish list

- Eliminate gulf between theoretical innovation in small groups and realization on high-end computers
- Eliminate the semantic gap so that efficient parallel code is no harder than doing the math
- Enable performance-portable "code" that can be automatically migrated to future architectures
- Reduce cost at all points in the life cycle

- Much of this is pipe dream – but what can we aspire to?

9

For large codes, once you got something that's a few million lines long, it's not going to change really fast; it's kind of frozen. How do we move that forward? If something that big is also complicated, how do we structure these things for the software and also the communal enterprise that's producing these things to maintain productivity? In the chemistry and materials world, innovation is happening in small university groups and small research labs. We can't require every one of those to have the expertise to manage all of these issues. How do we connect this innovation pipeline with the ability to realize ideas as rapidly as possible in big production codes capable of using large parallel computers? Of course, we also have to worry about cost. Government funding, or even funding in industry, is certainly not growing in this space. We have to do more with the same or do more with less very often. How much of this can we actually do?

Scientific vs. WWW or mobile software

- Why are we not experiencing similar exponential growth in functionality?
  - Level of investment; no. of developers?
  - Lack of software interoperability and standards?
  - Competition not cooperation between groups?
  - Shifting scientific objectives?
  - Are our problems intrinsically harder?
  - Failure to embrace/develop higher levels of composition?
  - Different hardware complexity?

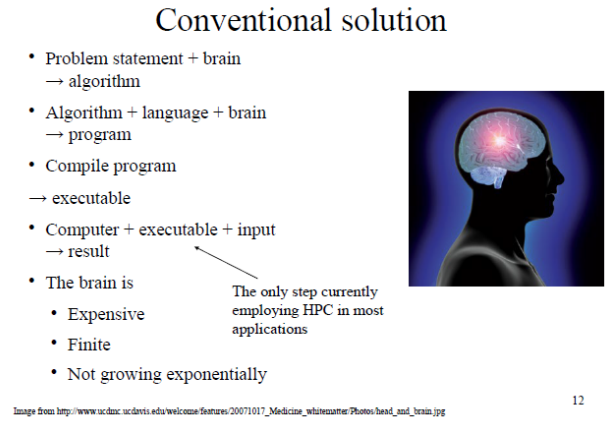
Part of this is learning from practices elsewhere; we have to change what we're doing. Most people who have written a parallel program will, especially if you're a certain age, realize that things have not really improved in a few decades. This here on the bottom right of the slide is some Fortran code written using OpenMP, which is a standard for shared memory programming: using a computer like a single PC or a single server, but using all of the processors in that single shared memory environment. You can see here it is very low level; the instructions have loops and indices and all these directives here. That's almost state-of-the-art parallel programing in some sense, at least in terms of mainstream standard programming. If you're doing a production project, you have no choice but to code to existing standards. This here at the top right of the slide is a snapshot of an iPhone screen. You can be sure the people writing iPhone apps, in which we've seen an explosion of productivity and tools, aren't writing like this. What's different about these faces? Of course, there's a difference in budget. There's a different scale it's after. One other major difference is in the science community, there's really not a very large market for this software, so often the person writing the software is the same person using it. In this world, we have ten developers for an app, but there's probably tens of millions of users. There's a different scale here, so we have to factor all of that into issue. But I don't necessarily buy that scientific software is more complex. I think there are things we can learn from what they're doing.

How do we write code for a machine that does not yet exist?

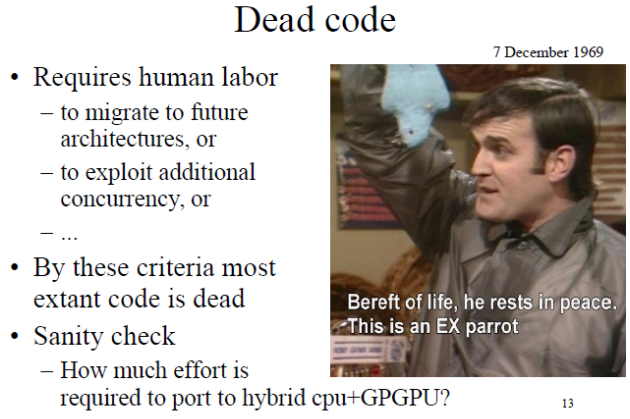
- Nothing too exotic, e.g., the mix of SIMD and scalar units, registers, massive multi-threading, software/hardware managed cache, fast/slow & local/remote memory that we expect in 2018+
- Answer 1: presently cannot
  - but it's imperative that we learn how and deploy the necessary tools
- Answer 2: don't even try!
  - where possible generate code from high level specs
  - provides tremendous agility and freedom to explore diverse architectures

11

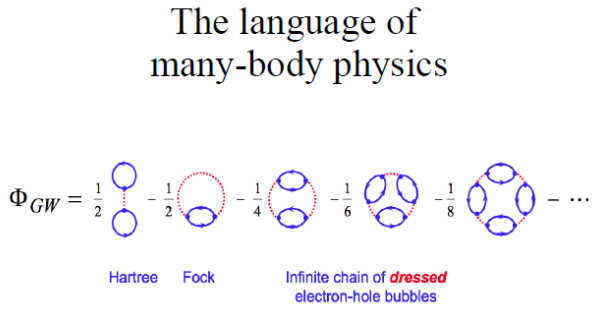
As I have already mentioned we have this shifting technical landscape. We are having to think about a code that's going to last for ten or twenty years without knowing what that code is going to be running on. We can't really imagine that hardware. So how do we write software for a machine if we don't know what it will look like? The answer is that we don't really know how to do that right now. But there are some concrete ideas, and they involve changing what we software developers do on a daily basis.



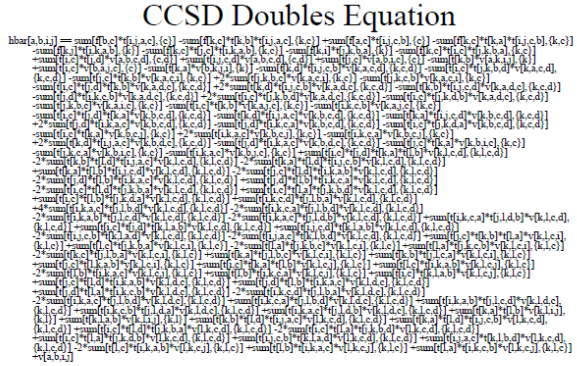
To gain some perspective on this, let us discuss how we write software right now. Humans are doing it. We have to translate our problem into an algorithm, and then translate that into a program. We then get the computer to help us translate our high-level specification program into executable machine code, and then we finally get to run the program on the computer. That is the only step in which we are using high-performance computing. How do we use the computer more effectively to do all of the other steps? Our own brain is finite; it's not growing exponentially, which is good if you think about it. You'd come to a very short end.



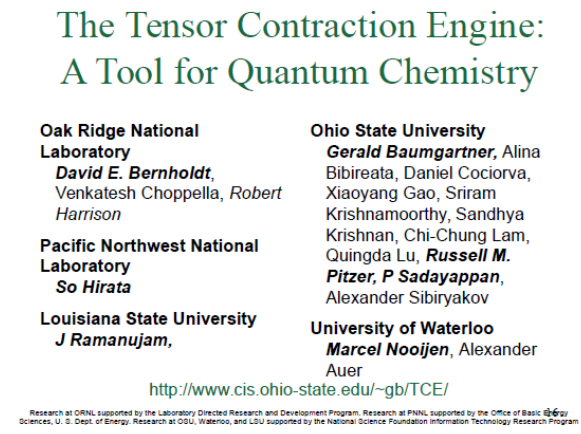
Thinking about our software then, traditional software has a finite life. My message here is that if we think about traditional software, Fortran, C, etcetera, then how do we move that software forward? There's no automated process usually to do that, although there are a few tools out there that help. To move that software onto another platform, onto another architecture, often involves a complete rework. By that sense, most existing code is dead. It's the way it is, and that's the way it's going to be forever. The sanity check on that are these GPGPUs, General Purpose Graphical Processing Units. People who compute realize these are very high-performance computers if you have code that is a good algorithmic match for them, but it often requires a complete rework. There's a very successful project called SPIRAL that generates high-performance transforms like FFT and so on, and what they do is they start on a high-level specification of all of the mathematics. Once they have captured all of that, as well as a detailed model of the machine architecture they want to implement it on, they conduct a brute force search, or maybe a more nuanced search that eventually delivers the high-performance code that often realizes 99 percent of peak speed on that architecture. They have a very nice saying: once you start writing code, you immediately stamp an expiration date on that code. It's because you have all these design decisions that ossify into it: how much memory you can use, how much parallelism, what are you going to parallelize, on what type of hardware, etcetera. Yet the mathematical specification lives forever. That is something that is really timeless.



14



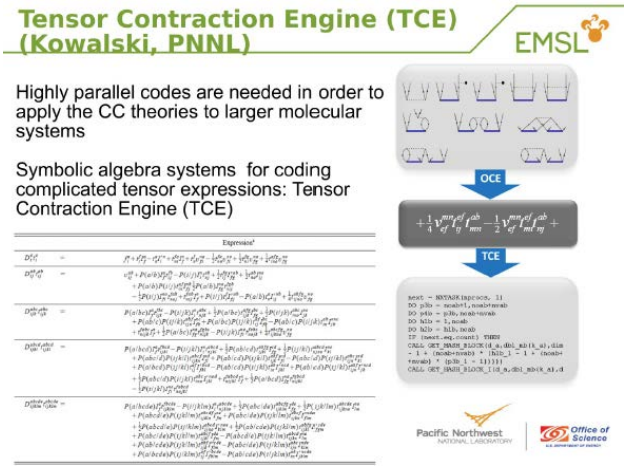
15



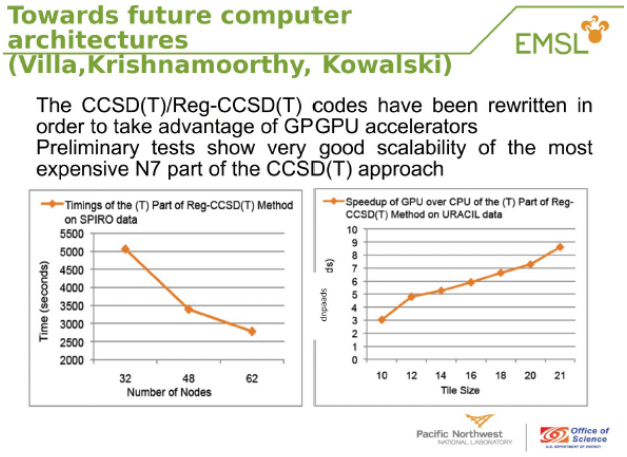
Thinking about my own fields of chemistry and physics, this here on the slide is a language. These are Feynman-like diagrams. It's not a complete specification of this particular problem, which is a many-body method for modelling materials, because there are details of the basis set and the problem we want to reply to. But nevertheless, this suffices to communicate a very large fraction of the underlying software. We want to write code that looks like this.

This chemistry equation down here at the bottom of the slide is equivalent to a diagram like the one depicted earlier. You could create diagrams that would translate into this statement. This is the residual for the couple clusters singles and doubles model for electronic structure. We basically want to solve this residual equal to zero. This here on the screen is where it begins. What you can immediately see is that this equation actually translates to something with a lot of structure when we're contracting multi-dimensional arrays against each other.

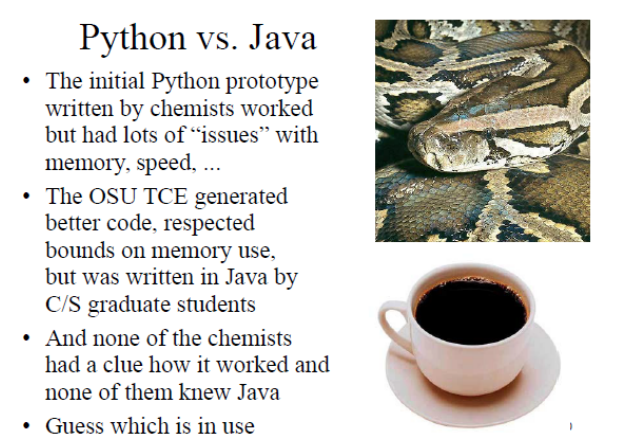
That structures gives rise to the possibility of doing other things. Instead of having humans write code, they can write high-level specifications. There was a very nice pair of projects, one a DoE-funded funded SCIDAC project and the other an NSF-funded activity. Sadayappan from OSU was the lead on the NSF side. I was the PI on the DoE side, but actually David Bernholdt and in particular So Hirata were lead actors on that side.



It was Kowalski at PNNL that turned this into a production capability. We can basically take the diagrams and, using the rules of quantum physics, translate them into pages and pages of these things shown in the middle of the flowchart on the slide. These things get turned into literally millions of lines of code that do massively parallel solutions of these equations. This has now transcended human ability. It used to take, if we think about the couple cluster equations, a graduate student's entire career to get to the point where they could write a high-performance parallel couple cluster code. Now, we've gotten to the point where we can, in a single morning, generate new equations that would have taken an army of graduate students working for several careers to realize, at which point you're asking: would the human-generated code be correct? Probably not. Would it be as fast as machine-generated code? Possibly, but again probably not. What happens on future architectures? The key point here is that the Python code generator that Karol Kowalski (PNNL) has is about ten thousand lines of Python instead of a few million lines of Fortran. You rewrite the code generator.



The only piece of NWChem that is working well on multicore GPGPUs is the Tensor Contraction Engine generator code, because they reworked the generator instead of rewriting the entire code face. This is possible because everything is so well-structured in that domain. But there are things there for us to learn.



Another aspect of learning is the sociological one. The Tensor Contraction Engine project at OSU, led by Sadayappan, actually generated a really powerful tool. But unfortunately, it wasn't done in a sustainable way. The chemists and physicists didn't understand how it worked, and it was using a language that none of them knew. Going forward, it was the less capable Python code that the physicists and chemists understood inside out. That also points to a funding-level challenge. On the DoE side, there's more continuity for funding, or at least historically there was, for computer scientists and applied mathematicians to

collaborate with application teams. On the NSF side, back when this was happening, there was more of a model. We'd fund your project for five years, and that's it. We'd move on to something else. That model is changing, and I'll come back to that at the end.

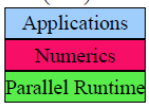
Another project I've been involved in is this MADNESS activity. MADNESS stand for Multiresolution ADaptive Numerical Environment Scientific Simulation.

It is a very interesting collaboration between applied mathematicians, computer scientists, and chemists working on several topics.

Here are pictures of their faces.

What is MADNESS?

- A general purpose numerical environment for reliable and fast scientific simulation
  - Chemistry, nuclear physics, atomic physics, material science, nanoscience, climate, fusion, ...
- Want robust and fast algorithms that scale correctly with system size and are easy to write
- Semantic gap
  - Why are equations O(100) lines but codes O(1M) ?
- Facile path from laptop to exaflop



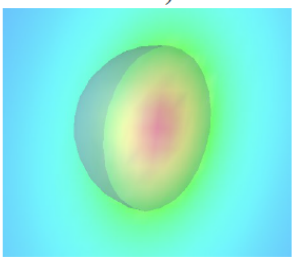
http://code.google.com/p/m-a-d-n-e-s-s
http://harrison2.chem.utk.edu/~rjh/madness/

MADNESS started off with looking at computations in chemistry and asking what the main problems were. The problems were accuracy and speed. Basically, you can't get both; yet we want both. We also want our software to look more like our physics equations; we want this high-level composition, and we want to run on the big computers. How do we make that happen?

E.g., with guaranteed precision of 1e-6 form a numerical representation of a Gaussian in the cube $[-20,20]^3$, solve Poisson's equation, and plot the resulting potential
(all running in parallel with threads+MPI)

```
Let
  Ω = [-20,20]^3
  ϵ = 1e-6
  g = x -> exp(-(x0^2+x1^2+x2^2)) * π^-1.5
In
  f = 𝓕 g
  u = ∇^-2(-4*π*f)
  print "norm of f", ⟨f⟩, "energy", ⟨f|u⟩ * 0.5
  plot u
End
```

output: norm of f 1.00000000e+00 energy 3.98920526e-01



There are only two lines doing real work. First the Gaussian (g) is projected into the adaptive basis to the default precision. Second, the Green's function is applied. The exact results are norm=1.0 and energy=0.3989422804.

```
        print "iter", i, "norm", ||ϕ||, "eval", λ
    end
End
```

We want to write equations that look like what's depicted on this slide.

This is actual code in our highest level environment. It's not a production compiler. It really exists to instruct students and new users about how to think when they're computing inside MADNESS. This code compiles, generates in C++ the commanding code, and then it runs in parallel using MPI and threads. This is solving some equations in electronic structure.

The math behind the MADNESS

- Multiresolution

$$V_0 \subset V_1 \subset \cdots \subset V_n$$
$$V_n = V_0 + (V_1 - V_0) + \cdots + (V_n - V_{n-1})$$

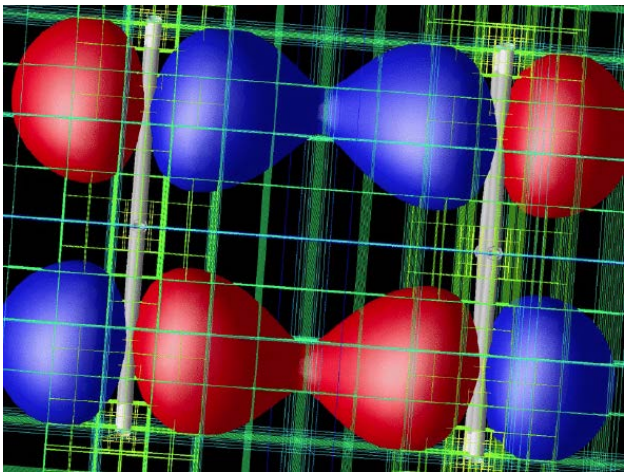
- Low-separation rank

$$f(x_1,\ldots,x_n) = \sum_{l=1}^{M} \sigma_l \prod_{i=1}^{d} f_i^{(l)}(x_i) + O(\epsilon)$$
$$\left\| f_i^{(l)} \right\|_2 = 1 \qquad \sigma_l > 0$$

- Low-operator rank

$$A = \sum_{\mu=1}^{r} u_\mu \sigma_\mu v_\mu^T + O(\epsilon)$$
$$\sigma_\mu > 0 \qquad v_\mu^T v_\lambda = u_\mu^T u_\lambda = \delta_{\mu\nu}$$

What we did here is, in order to realize these objectives of more speed, higher-level composition, and control of accuracy, we tossed out a whole bunch of things. We adopted a different numerical representation and focused not just on having accuracy but also having this representation be well-matched to the underlying trends in computation, the underlying basis.



Here on the slide is a molecular orbital from some molecule, I think benzene dimer. It looks like a traditional adaptive mesh. Our basis functions in each of these mesh boxes is a discontinuous spectral element basis. Inside each box we have a bunch of fairly high-order Legendre polynomials. We typically start computing at order six, but we'll go up to ten or twenty or even higher depending on the problem. That gives us dense lumps of computation that are very well suited for modern processors.

Frameworks and libraries

- No longer sufficient to address the pace and nature of technology change
  - Written by humans: not sustainable, not nimble
  - Rigid partition between application and library
  - Constrains composition and expression of all available concurrency
  - Premature binding of application to H/W model
- Still valuable tools in our toolbox
  - Reuse, exchange between disciplines, aggregation of capability, community action

Part of what's going wrong is that this pace of change in the software and the underlying technology is breaking previously successful techniques we had for dealing with complexity. Frameworks and libraries are architecting solutions dealing with complexity, and that complexity can be inherent in the application or it can be inherent in the technology or the computer. But it's also a great way to interface different domains. If there's some library with a well-defined interface and we're familiar with, say the linear algebra world, with linpack and lapack, I don't need to know how the eigensolver works. I only need to know what interface to call and have some guarantee that good things are going to happen. This has been a very successful approach to handling software complexity. However, now that computer architecture in the programming world is changing so much, this model is still an issue.

Disruptive change demands disruptive solutions

High-Level Semantic Environments (HLSEs)

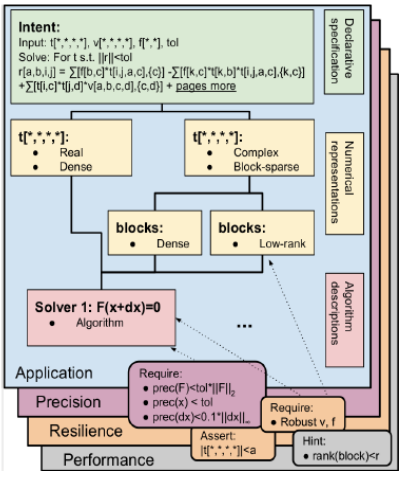
SBU: Harrison, Chowdhury, Liu, van de Rijt, Sheppard

OSU: Sadyappan, Rountev, Parthasarathy, Fosler-Lussier

Utah: Hall, Gopalakrishnan, Balasubramonian, Might

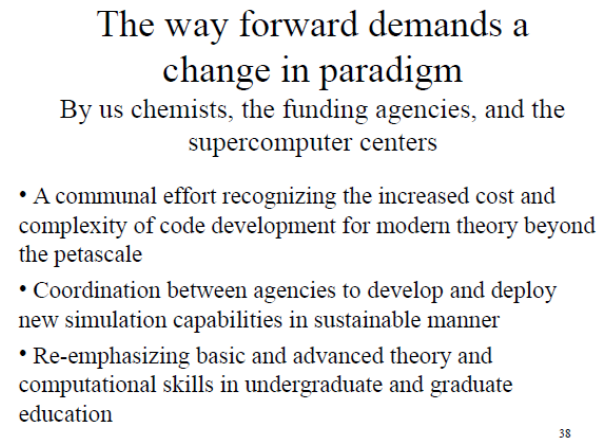
Delaware: Gao, Siegel

Colorado: Strout



The domain-specific languages that I was speaking about earlier in the context of the Tensor Contraction Engine are a part of that solution, but it's still not a complete solution. Looking forward, we're going to have to be synthesizing in other things. Other aspects of computation are important in realizing efficiency, not just in terms of performance but in terms of power and other attributes of computation. We want to be able to specify what precision computation should happen. For instance, when data is stored into memory, you don't have to store double precision numbers all the time because often most people aren't interested in those trailing bits. They need those trailing bits when they are computing sums and

reductions and the like. We need to specify computation more carefully, and we also have to have statements about resilience. Maybe applications will be able to tolerate certain types of errors that will be arising on the architectures and so on.

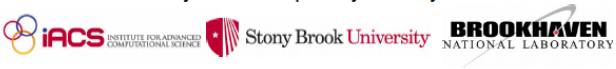


There's a lot of complexity here. What you will see in trends in this space is a reemergence of an ongoing narrative that really started in the DoE world. There's a very successful program called Scientific Discovery through Advanced Computing, the SCIDAC Program at the DoE. It really started a very clear consensus about co-funding scientists, computer scientists and applied mathematicians to work together in long-lived teams and also promoted an understanding of the fact that scientific software is long-lived and has to be managed and supported in the same way that we support facilities.

NSF, in its sustainable software initiative, has really taken this to heart and is in the process of reengineering communities as well. I'm involved in this activity, which was funded as a conceptualization of a project for a sustainable software institute in chemistry and materials science. We're actually in the process of finalizing a proposal to implement this institute. I invite you to reach out to any of the PIs on the slide here that you know, or me, if you're interested in getting more information. The basic story is we want to have a grassroots, community-led effort that implements, with a very long vision, a sustainable approach to our software that brings all the skills together in a sustainable way, so that not everyone has to be an expert in everything, and the community has the resources to move ahead. The institute will operate with a core group of staff with a diverse set of skills, located at Virginia Tech. A sizable fraction of the resources will be going out into software fellowships for students and post-docs to work on projects embedded in the community in their own groups. Of course, there will obviously be strong ties coming back into the central activity hub.

Summary

- We need radical changes in how we compose scientific S/W
  - Complexity at limits of cost and human ability
  - Need extensible tools/languages with support for code transformation not just translation
- Students need to be prepared for computing and data in 2020+ not as it was in 2000 and before
  - Pervasive, massive parallelism
  - Data-centric, bandwidth-limited computation and analysis
  - An intrinsically multidisciplinary activity



I'm going to wrap up there. Thank you very much.

Accelerating Materials Discovery using First Principles Computation Techniques

Dr. Yifei Mo
Department of Materials Science and Engineering, University of Maryland
College Park, MD

Abstract

The design and discovery of new materials have been pursued through a trial-and-error manner based on human intuition and serendipity. This traditional materials design process is time consuming and labor intensive, which have significantly delayed the research and development for novel materials. Computational techniques based on first principles are capable of predicting materials properties accurately with little experimental input, and have the potential to accelerate the materials design. In this presentation, I will share our success stories of using first principles computation techniques in the design and discovery of new solid electrolyte materials in Li-ion batteries and solid oxide fuel cells. I will illustrate the development of first principles computation methods in predicting the phase stability and chemical stability, which are crucial for the accelerated design of new materials. Our computation results and predictions are in good agreements with multiple experimental studies.

Biography

Dr. Yifei Mo is an Assistant Professor of Materials Science and Engineering at the University of Maryland (UMD). Dr. Mo has been conducting research on the computational design of novel materials using first principles calculations and on nanoscale mechanics using large-scale atomistic modeling. Mo has published research articles in peer-reviewed journals including Nature, Nature Materials, Energy and Environmental Science, Nano Letter, Chemistry of Materials, etc.

Accelerating Materials Discovery using First Principles Computation Techniques : A Case Study of Ionic Conductor Materials

Yifei Mo
Assistant Professor

Department of Materials Science and Engineering
University of Maryland, College Park

Thank you Prof. Li. I would like to thank Peter and other organizers for holding this nice symposium and having me here to introduce my research.

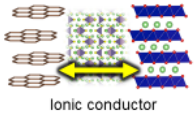
Since I am from material science, my talk will focus on how we can use first principle computation and infrastructure based on the first principle computation data to help us accelerate discovery of new materials.

What material scientists are really interested in is a new compound of composition with certain crystal structure that serves functionalities with amazing properties. Here I will share a case study with you, a successful case of ionic conductor material where we predict a material from first principles computation to validate experiments in the lab then then goes to commercialization.

Fast Ion-conducting Materials

Fast ion-conducting materials are critical for many technological applications, esp. in electrochemical devices.

- Batteries
- Fuel cells
- Electrochemical sensors and membranes



Ionic conductor

Technical challenges

- Improving rate capabilities for batteries
- Lowering the operational temperature of solid oxide fuel cells.

Materials design challenges: design and discover new materials with

- Improved the ionic conductivity,
- while maintaining other attribute (stability, etc.) of the materials

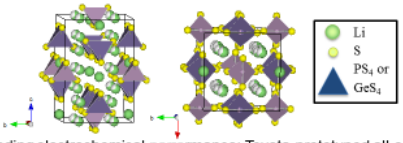
2

Here is a brief introduction. Why do we care about fast ion-conducting materials? Because it is critical for a lot of electro-chemical or solid-state electrochemical devices, such as batteries and fuel cells. Here I show a diagram of the Lithium ion battery. In this battery, you need to have very fast transport of Lithium ion. And this is also true for fuel cells and electrochemical devices such as sensors and membranes. So to enable these devices, we need to develop a new material with higher ionic conductivity so that you can improve rate capabilities of batteries and also lower the operational temperatures of solid oxide fuel cells.

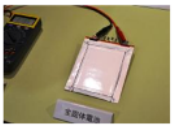
Case study: $Li_{10}GeP_2S_{12}$ (LGPS) - a new superionic conductor

$Li_{10}GeP_2S_{12}$ (LGPS): a new material discovered by Kamaya et al. ***Nature Materials***, 10, 682-686 (2011)

- Highest Li^+ conductivity ever reported for Li solid electrolyte: σ = 12 mS/cm at room temperature. Comparable to liquid electrolyte!



- Outstanding electrochemical performance: Toyota prototyped all-solid-state battery with LGPS solid electrolyte.



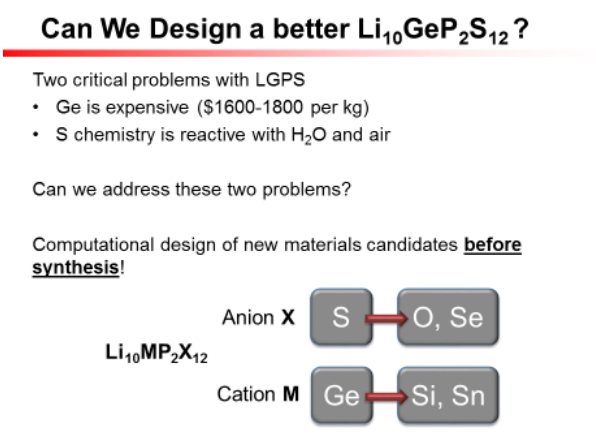
- Ultimate safety
- High energy-density

3

So the materials design challenges we are facing are designing and discovering new materials with improved ionic conductivity and meanwhile maintaining other good attributes of the material such as phase and electrochemical stability.

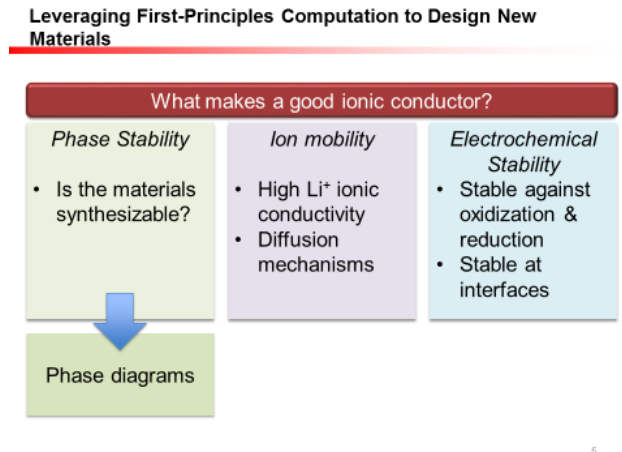
The case I will present today is the LGPS based material discovered by the Tokyo Institute of Technology and Toyota. They discovered that this compound has a really interesting and distinct crystal structure. What is nice about it is that it has the highest Lithium ion conductivity ever reported in solids. It is about 12 mS/m at room temperature which is comparable to the commercial liquid electrolyte that we have in Lithium battery.

Due to its outstanding electrochemical performance, Toyota took this material and prototyped an all-solid-state Lithium battery. One point worth mentioning is the safety incidents of the Lithium battery which is mostly because the liquid electrolyte is flammable so that it can catch fire. The idea here is to replace the current Li-ion battery technology with a solid-state-based battery technology as shown in the figure, in order to achieve the ultimate safety and also higher energy density.



So we are interested in this material, coming along with some critical problems. First of all, Germanium is very expensive, making it a non-starter for large-scale applications. And also, Sulfide is difficult to handle. It is moisture sensitive, and also reacts with air.

What we are trying to do here, is to design this material using first principle calculations so that we can address the two critical problems of this amazing new material before synthesis. We proposed to replace Sulfur with Oxide, which is easy to handle, and Selenium, for scientific interest. And also another thing is whether you can replace the very expensive element Germanium with some cheap and abundant elements in the same row in the periodic table of elements such as Silicon and Tin.

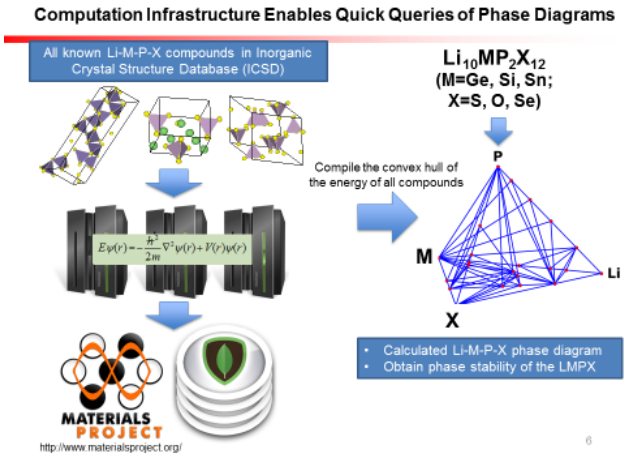


We want to leverage our first principle computation to evaluate these potential material candidates. So how should we do that?

We know that to become a good solid electrolyte, you need all these following good materials characteristics. First of all is phase stability. It is very important, because you want a material, especially a new material to be stable so that you can make sure you can synthesize it in the lab. Second of all is the ionic mobility such as Lithium ionic conductivity because it is the critical material property you are looking at in this particular. And also you want electrochemical stability because you want this material to be stable against oxidization and reduction that will happen due to battery recycling. And also because it will work with other materials, you also want it to be stable at interfaces.

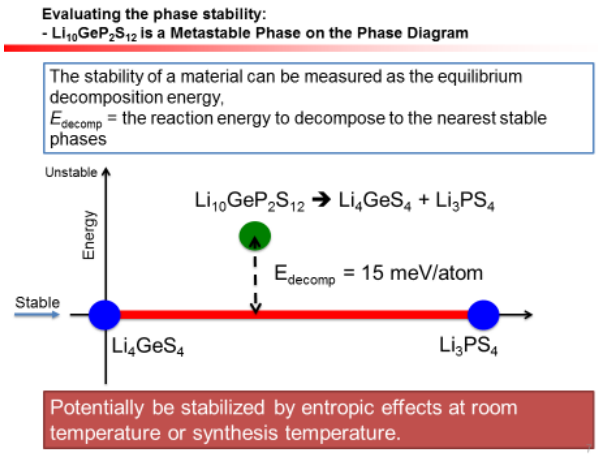
So the idea to design the material is to evaluate these key material properties in first principle calculations. If we have accurate predictions of these properties in first principle calculations, we can essentially predict and evaluate new materials on computers.

I will talk about each of these properties respectively. I will start with phase stability. I will show you that using first principle calculation and the infrastructure based on it, it is actually incredibly simple and straightforward to evaluate the phase stability of any material you come up with.



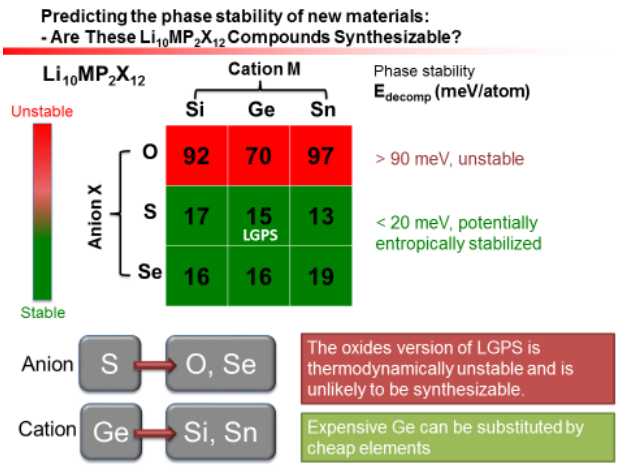
So here what we do is using the infrastructure of first principle calculations. Now with the development of programs such as *Materials Project* which holds inorganic first principle energetic data, it is fairly easy. We can simply generate the relevant phase diagram using the first principle database. What we need to do with the new compound material, $Li_{10}MP_2X_{12}$, is that we can just do one single first principle calculation for each compound. Then put it on the phase diagram to evaluate its phase stability. It is incredibly simple to do by accessing the database using a web browser.

And for us, we use a codebase to do that.



So in this way, you can evaluate phase stability of a new material such as $Li_{10}GeP_2S_{12}$. It comes up with the energy as a whole that describes the stability of materials, which is comparing the energy of the $Li_{10}GeP_2S_{12}$ which you obtain from your first principle calculation to the relevant phases in its composition space, i.e. Li_4GeS_4 and Li_3PS_4. And then you can have the decomposition energy which evaluates how relatively stable the $Li_{10}GeP_2S_{12}$ is against its decomposition phases with which you can have a benchmark of whether the $Li_{10}GeP_2S_{12}$ has good stability

and is experimentally synthesizable.



So the only caveat here is in first principle calculation, you are evaluating energy instead of free energy at 0K. You have to consider that effect when you deal with real materials.

Here is the data we get for all these materials with different cation and ion substitutions.

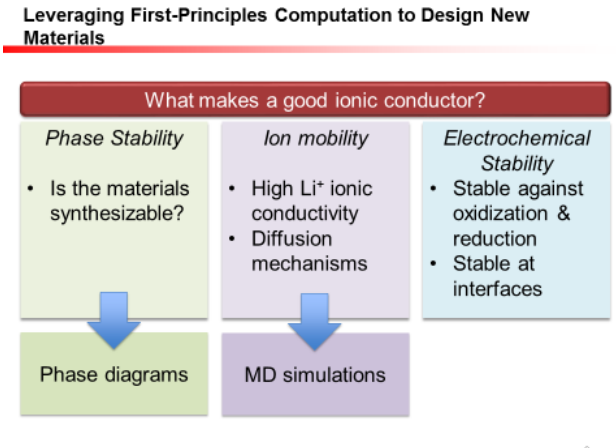
The original $Li_{10}GeP_2S_{12}$ is in the middle with decomposition energy of 15 meV/atom. As you can see here, when you change cation Ge into Si or Sn, the decomposition energy does not go much higher, which means that for $Li_{10}SiP_2S_{12}$ and $Li_{10}SnP_2S_{12}$, they have similar stability as

$Li_{10}GeP_2S_{12}$, as is demonstrated by Toyota. And they are likely to be synthesizable.

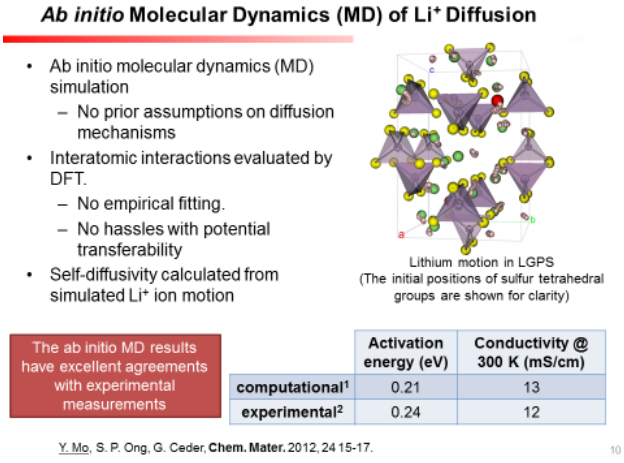
If you change the ion from S to O, you can see that the decomposition energy increases dramatically which means that the new compound obtained by substituting S in $Li_{10}GeP_2S_{12}$ by O would be likely unstable and difficult to synthesize and use.

The good news is that we indeed find that, we can substitute Ge into Si or Sn. The question is after substitution, whether the new material would have good performance to be used as a good solid electrolyte material.

So what we will do next is to evaluate the ionic mobility, specifically the Lithium ionic mobility in this material.



In the last part, I will show you how we design new materials using our computational methods developed.

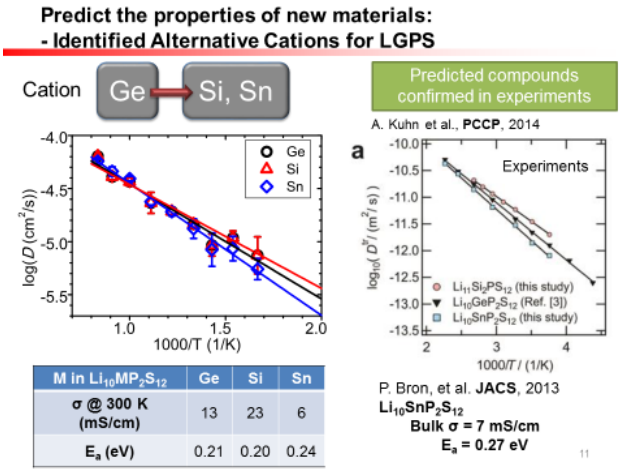


We use first principle, Ab initio Molecular Dynamics simulation to do that.

Here is a movie from the simulation. The good thing about MD is that typically when people do diffusion first principle, we do NEB type of calculation where we assume there is a path for specific mobile carrier to evaluate the diffusion. In that case you assume the diffusion mechanism. But in Ab initio molecular dynamics simulation, you make no assumption. You just let the Lithium atom diffuse and observe what happens, and extract the relevant diffusion properties such as diffusivity and ionic conductivity. And you can also capture the correlated motion.

Moreover, this is a first principle calculation where you have no hassle worrying about a good empirical potential. The good transferability of quantum mechanics makes you confident to work with new materials.

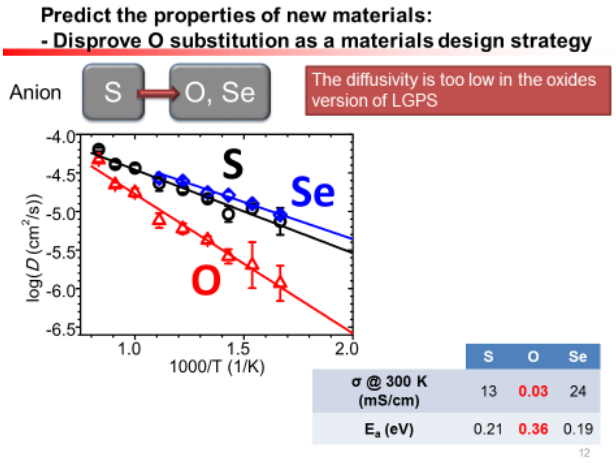
The activation energy and conductivity at 300K given by Ab initio MD simulation are 0.21 eV and 13 mS/cm, while the experimental results are 0.24 eV and 12 mS/cm. They agree very well.



Then we can leverage the technique to evaluate properties of the material we have, which is obtained by substituting cation Ge into Si and Sn.

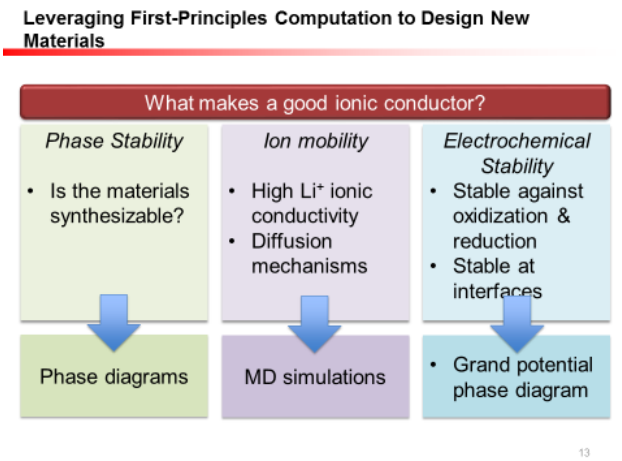
The figure here is log of diffusivity VS reciprocal of temperature for different compounds. You can see that when you change Ge into Si or Sn, the diffusional property is largely similar. The activation energy and conductivity at 300K of $Li_{10}GeP_2S_{12}$, $Li_{10}SiP_2S_{12}$ and $Li_{10}SnP_2S_{12}$ are close.

This indeed shows that the newly substituted materials are great materials. Multiple groups have confirmed the results given by our computational simulations as is shown in the figure on the right. This is the one from PCCP which shows similar plot as ours. There is another paper on JACS showing that the activation energy and conductivity at 300K of $Li_{10}SnP_2S_{12}$ is 7 mS/cm and 0.27 eV, which is close to our results which are 6 mS/cm and 0.24 eV. These results verify our prediction in first principles calculations. Now, $Li_{10}SnP_2S_{12}$ has already been commercialized by a company in New Jersey.



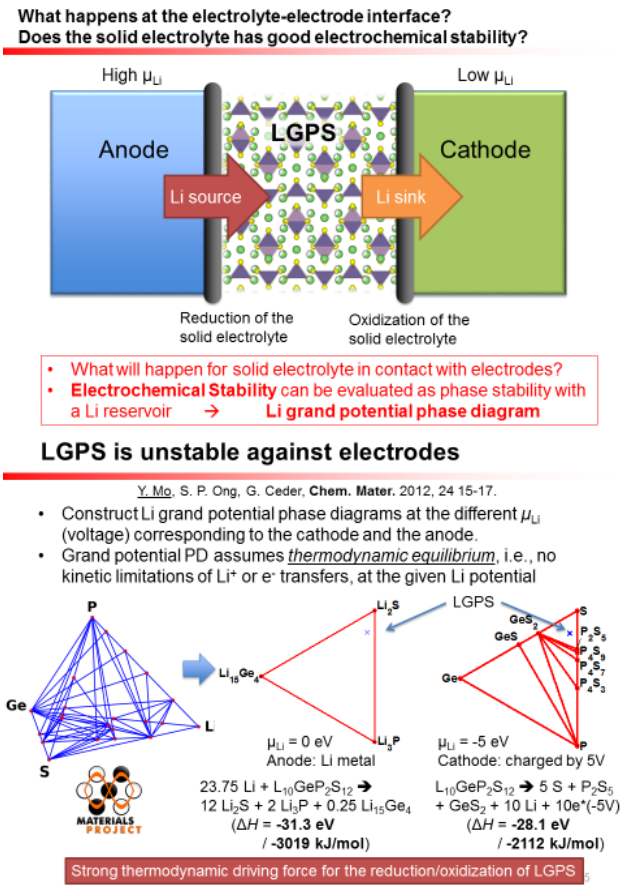
Furthermore, we also did the ion substitution with S changing to O and Se.

We originally showed that when you do the oxygen substitution, the phase is going to be highly unstable. But still we can get the property from first principles showing the plot of log of diffusivity VS reciprocal of temperature for $Li_{10}GeP_2S_{12}$, $Li_{10}GeP_2O_{12}$ and $Li_{10}GeP_2Se_{12}$. We see that when you do the ionic substitution, the diffusional property drops significantly which means that the ionic substitution is not only unstable, but also has poor performance. It is not a viable strategy for improving the LGPS material.



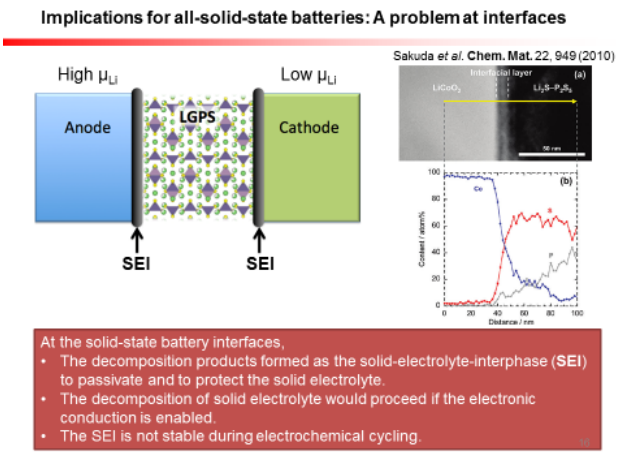
Considering the time, I will go over the last part fairly quickly.

As I mentioned, we need the solid electrolyte to have very good electrochemical stability. And because your material will never work alone in the device, often interfaces can be a big problem and you will need the interface compatibility between different materials.

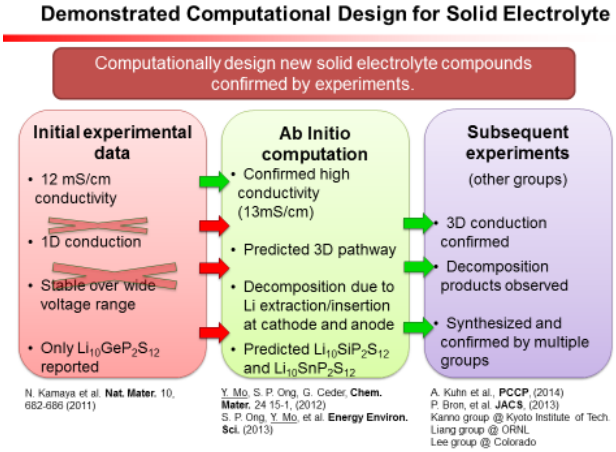


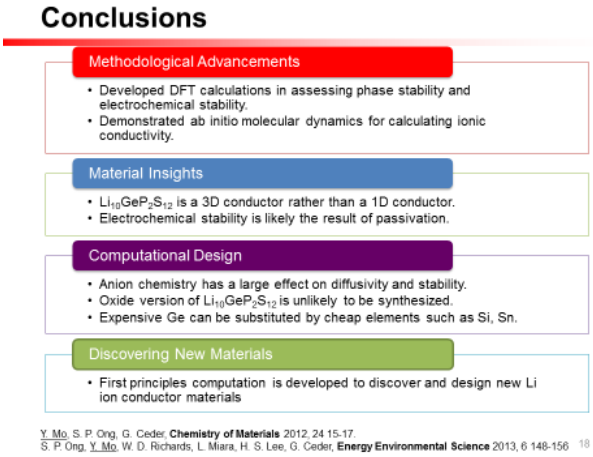
We can also evaluate these properties by first principle calculations thanks to the scientific infrastructure we have. In a real battery, you will put LGPS between anode and cathode which you will have reduction and oxidization.

To evaluate the highly reducing and oxidized environment, we can use the grand potential phase diagram which is shown by the figures on the right to evaluate material stability under different conditions. Still based on the scientific infrastructure, we have the phase equilibrium at different chemical potential of Li which gives us the stability of the material in the different environment and at interfaces.



What happens for the LGPS solid electrolyte when you put them in an actual battery.

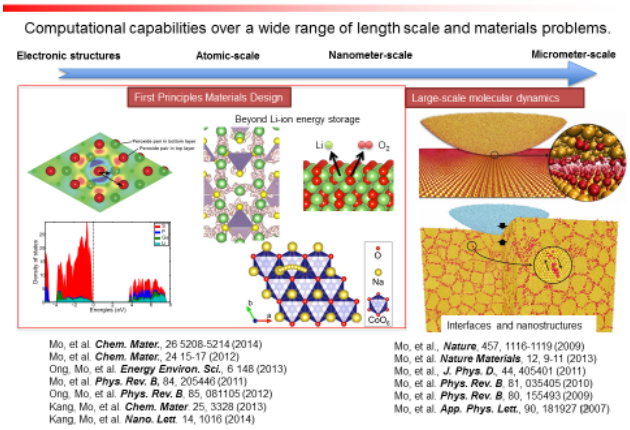




To conclude, basically we have shown the first principle calculations and with new type of development for method, we verify the calculation for phase stability, electrochemical stability and diffusion. We provide some valuable insights into the new LGPS material. We further leverage first principle calculation and show the computational design that we can change expensive Ge into cheap element such as Si and Sn while the material still works well. We also show that some other material design strategies such as using O substitution is unlikely to work. This helps us to discover new materials.

Here I only show one example, but as you can imagine we can leverage this into studying a lot of other new materials because first principles calculation can work with any combination of elements and chemistry that you want.

Research Background



In the end is a little bit advertisement for myself. We are a new group here, we have the capability of dealing with a large range of expertise and length scale. Results shown in the figure on the left is first principles materials design. We are able to calculate the electronic conductivity at the atomistic scale and at surfaces. We also do things at large-scale molecular dynamics in which we extend the length scale to hundreds of nanometers where we use reactive forces to investigate problems at surfaces, interfaces and complex nanostructures and microstructures.

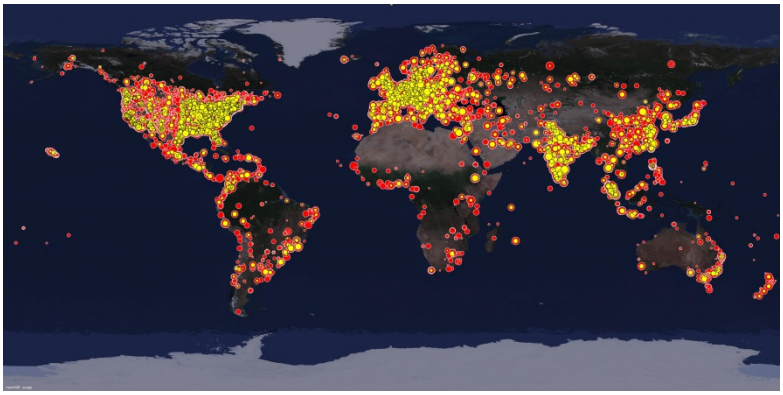
I would like to thank you and I am happy to take questions!

Development of the NEMO tool suite: from basic physics to real industrial devices and to global impact on nanoHUB.org

Dr. Gerhard Klimeck
Birck Nanotechnology Center, Purdue University
West Lafayette, IN

Abstract

NEMO5 is the fifth edition of the NanoElectronics MOdeling Tool set. It incorporates the core concepts and insights gained from 20 years of development of NEMO-1D, NEMO-3D, NEMO-3D-Peta and OMEN. NEMO5 licensing agreements for academic and commercial use are available. NEMO5 is free, with restrictions, for academic use.



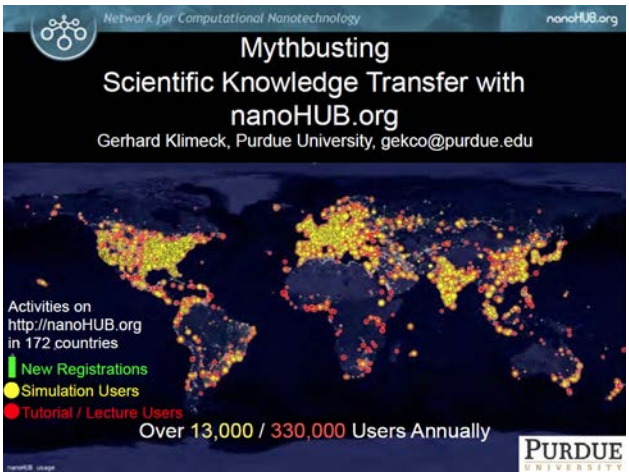
The core capabilities of NEMO5 lie in the atomic-resolution calculation of nanostructure properties: strain relaxation, phonon modes, and electronic structure using the tight-binding model, self-consistent Schrödinger-Poisson calculations, and quantum transport. This presentation overviews various aspects of NEMO5 capabilities, interactions with academia and industry, and its deployment on nanoHUB.

More than 330,000 users in 172 countries annually participate in nanoHUB.org, a science and engineering gateway providing the capability to perform online simulation through a web browser without the installation of any software. nanoHUB is an online meeting place for simulation, research, collaboration, teaching, learning and publishing. Over 12,000 users run simulation software from their browser in nanoHUB's science computing cloud. Cumulatively over 20,000 students in over 1,000 classes utilized nanoHUB simulations in classrooms and over 2,200 authors referenced nanoHUB in over 1,100 scientific publications. The platform has spawned nanoHUB-U and, in turn, Purdue HUB-U, interfaces for online courses that are broadly accessible around the world.

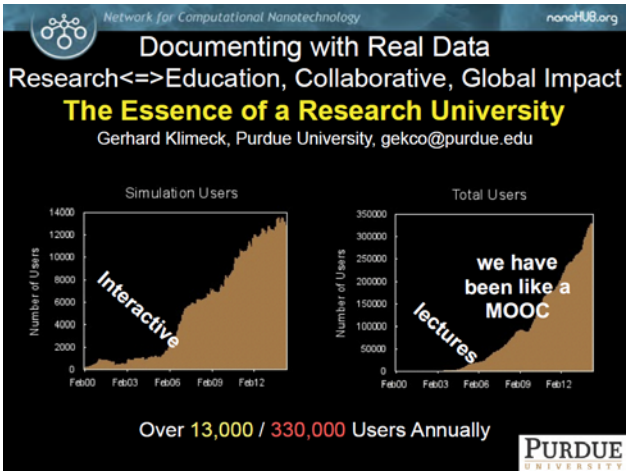
In collaboration with: Michael Povolotskyi, Tillmann Kubis, James Fonseca, Bozidar Novakovic, Jun Huang Yu He, Yaohua Tan, Mehdi Slamani Jelodar, Daniel Mejia, Zhengping Jiang, Hesameddin Ilatikhameneh, Prasad Sarangapani, Tarek Ameen, Junzhe Geng, Yuling Hsueh, Daniel Lemus, Saima Sharmin, Ahmed Reza, Pengyu Long, Harshad Sahasrabudhe, James Charles, Sicong Chen, Ganesh Hgde, Saumitra Mehrotra, Santiago Perez, David Bermeo, Krishna Madhavan, Lynn Zentner, Michael Zentner, Michael Mclennan

Biography

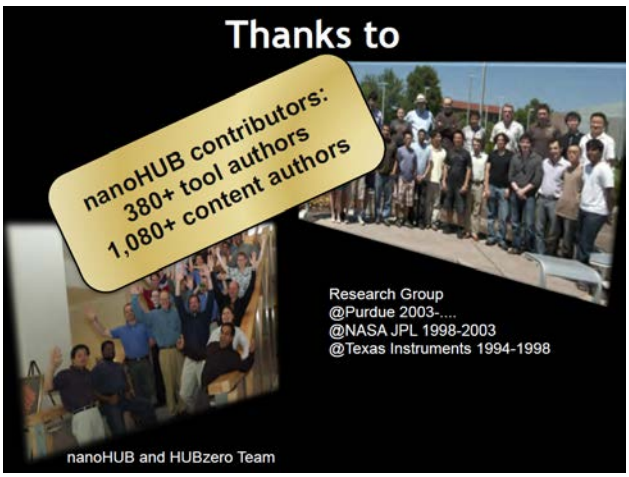
Gerhard Klimeck is the Reilly Director of the Center for Predictive Materials and Devices (c-PRIMED) and the Network for Computational Nanotechnology (NCN) and a Professor of Electrical and Computer Engineering at Purdue University. He was previously with NASA/JPL/Caltech and Texas Instruments leading the Nanoelectronic Modeling Tool development (NEMO). At Purdue he leads the development of nanoHUB.org which serves over 320,000 annually. His research is documented in over 400 peer-reviewed journal and proceedings articles and over 210 invited and 400 contributed conference presentations. He is a fellow of the IEEE, American Physical Society, and the Institute of Physics.



I was going to talk about the end of Moore's Law and the devices that we're modelling, and the tools that we build to analyze those devices, but in light of the talks this morning and their general scope I completely changed what I was going to talk about. Today I'm going to entice you with something that might jolt your willingness to share codes and the meaning of how you can share codes. I think that is more important than me talking about one particular topic. So I'm going to talk about nanoHUB. What you see here on the slide is an animation of usage from February 2013, two years ago. You see people coming to the site, logging in and signing up. The bigger the symbol at a particular location, the more people there are. These turn into simulation users that simulate, on nanoHUB, through a web browser. We have about 13,000 people annually that run simulations without installing any software. The red dots we have roughly 20 times more of compared to the simulation users. Those are people that look at seminars, tutorials, and lectures. This project is non-infrastructure; it runs 24/7, with half a day of downtime throughout the year. It's really up and running; it's part of the NSF infrastructure. I want to convince you that you can do scientific knowledge trends in a facility like that, from one group of people to a completely different group of people. Then I'm going to talk about mythbusting. I'm going to tell you all about the things we were told we couldn't be doing, and how people were stymied by these efforts. Here on the slide is a static image of a year's worth of users. We have users all over the globe that light up the night sky. Clearly, nanotechnology is very interesting to people.



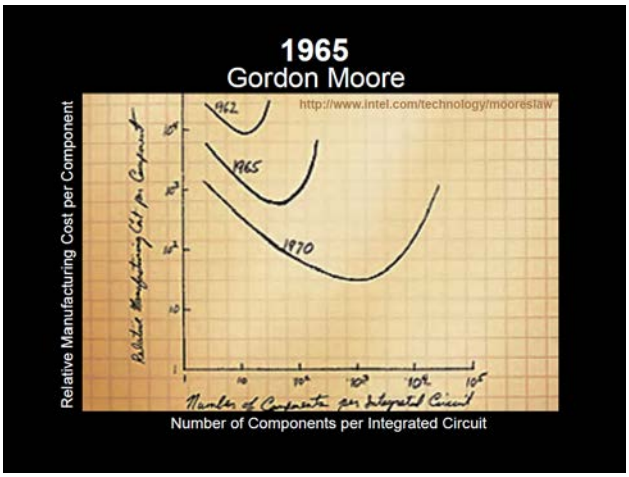
We've seen large growth; we started with 500 users when Mark Lundstrom founded this whole thing. We changed the system dramatically by making tools interactive. We moved away from web forums, similar to what your bank has today, towards fully interactive engineering tools that are actually appealing to users. Then we saw a little bit of growth. We also deployed lecturers because we knew we had to reinvent the system. There's a lot of talk about MOOCs now. If you define Massively Online Open Courseware at around 100,000 users, we've been doing that since 2009. To me, this is the essence of a research university. We connect research and education, we collaborate, we have global impact, and we're going to document all of this with real data to you.



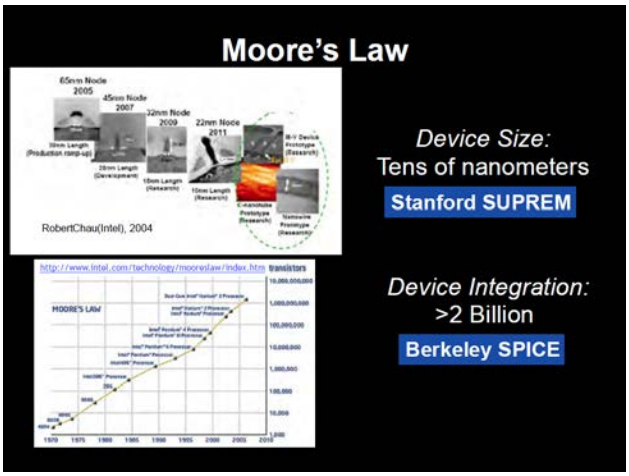
Obviously I don't do this work by myself. This is one picture of my research group. It has been a 20 year career of building this NEMO tool that I was going to talk about. I have a hub team that runs it all; those are the professionals that really operate this website. Then there are over a thousand content authors that actually contribute to the site, without whom we wouldn't have information to serve. There are really three different entities that I need to give thanks to.

It's a website, so that's one way to look at it. But it's much more than a website. Here is a very brief overview. These images on the slide are visualizations you can do on nanoHUB. Anybody can do it without installing software. There's a whole set of lectures and tutorials. There are over 4000 content items you can dig into and search through. These lectures here are Adobe type lectures. The key element is really simulation with the launch of a button. This launches a Unix tool that is VPNed into your browser, allowing you to do all these visualizations. These simulations can run fast, and they can also run relatively slow. Here on the slide is a full transistor design tool, and this will be dispatched in several clusters or into a grid computing structure. This is a biopore, a tool from Illinois. There's also a whole MOOC environment.

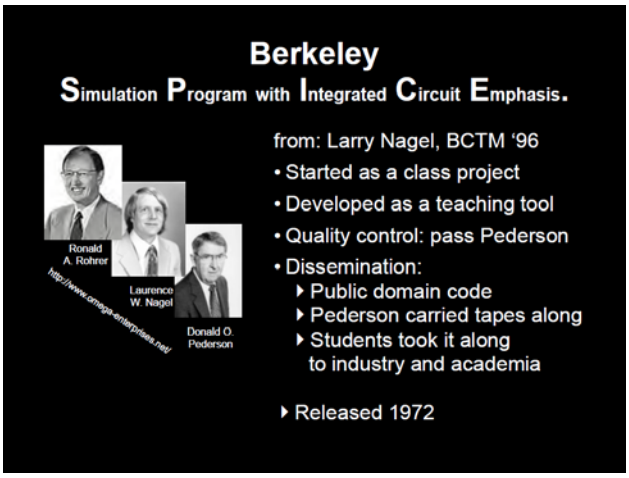
And yes, it's free. We're not asking for a credit card or anything. It's not free to me because I have to sell this thing all of the time, but it's free to the users.



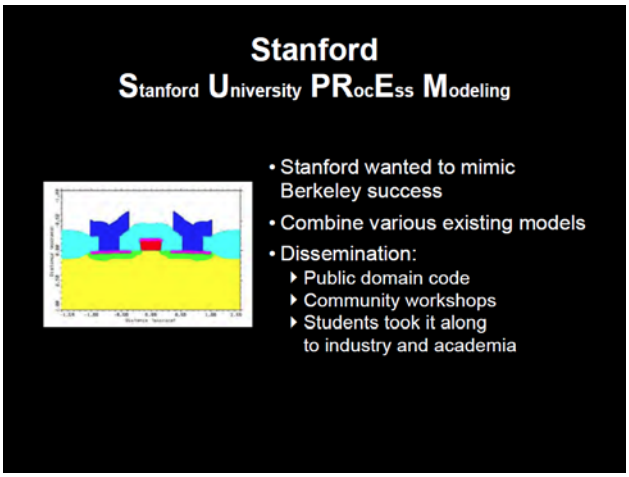
Before I take you into the future of what I see computing to be, I'm going to take you into the past, to the year 1965. It's an incredibly important year: I was born. But more importantly, a person sketched something into his notebook, and on the slide is the original sketch. It's the number of components per integrated circuit versus the relative manufacturing cost. Moore's Law was an economic law, not necessarily a technology law.



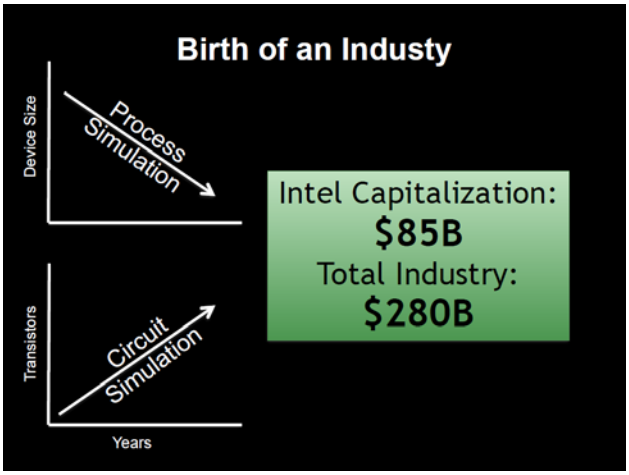
You've seen Moore's Law mostly like this, as seen here on this slide, where you plot the number of transistors versus development, and it's growing exponentially. We're at over two billion transistors. That's almost a third of the world's population working together without having a war. That's how big that number is. And that's an incredibly strong engineering feat to do. This version of Moore's Law talks about device integration. You've also seen Moore's Law maybe in this form; things are getting smaller, and that's my area of research expertise. Today, we're at tens of nanometers. We're material scientists, and we get that. You might have also heard that device integration and circuit simulation is being done with a tool called SPICE. You might have heard of processing modelling being done with the ancestor of the tool, SUPREM. But I bet you don't really know where these tools come from.



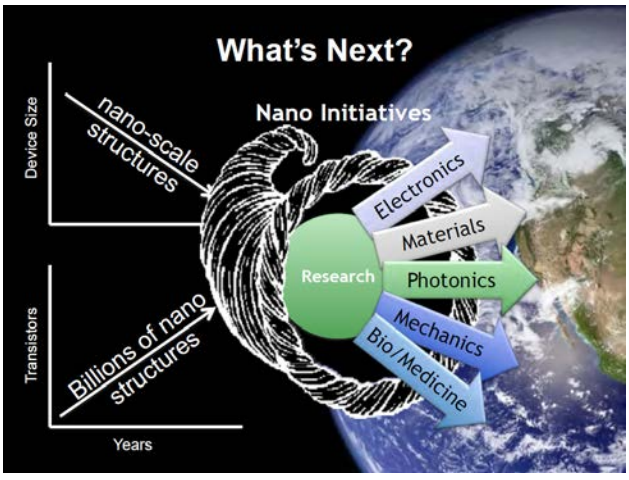
SPICE (Simulation Program Integrated Circuit Emphasis) started as a class project between a master's student and a professor. It was one of the first open-source tools. The creators carried it around on tapes. There was no assumption of perfectness. There was an assumption that there would be bugs in there, and the community would be involved in fixing them, and the creators would be involved in with them. Then, students took it along to industry and academia, and that's how that tool flourished.



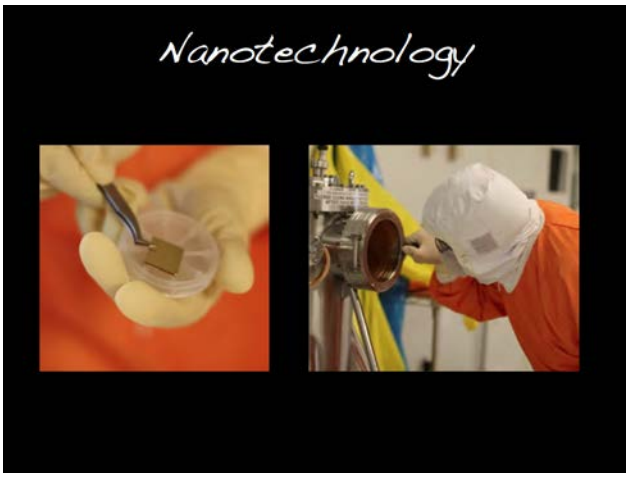
SUPREM (Stanford University PRocEss Modeling) was similar, but it started as a research tool.



Without process simulation and without circuit simulation, all the tools that came from small university groups, we would not have a 300 billion dollar semiconductor industry today; it's completely unfeasible. Without these two pieces of software, this would not have happened.

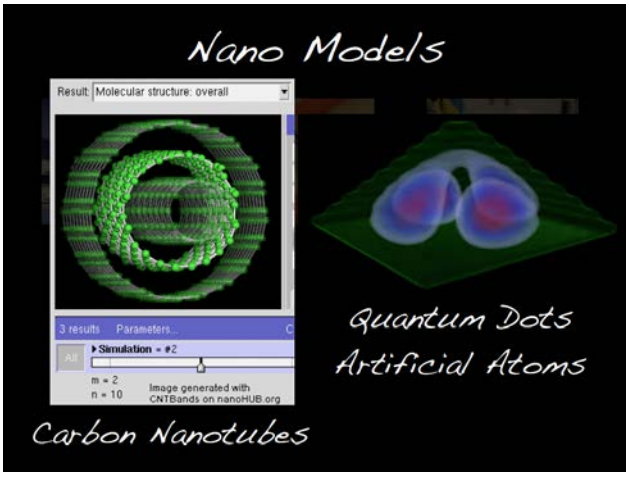


What's next? We have about two billion dollars' worth of research in the United States in nano. There's about eight billion dollars in the world invested into nano. We have a whole lot of research that sits in our research shelves, but can we put this into the real world? That is what we focus on in the team that I lead. What we really want to do is have these tools, reap them from research, and put them into the real world, making them useful to others.

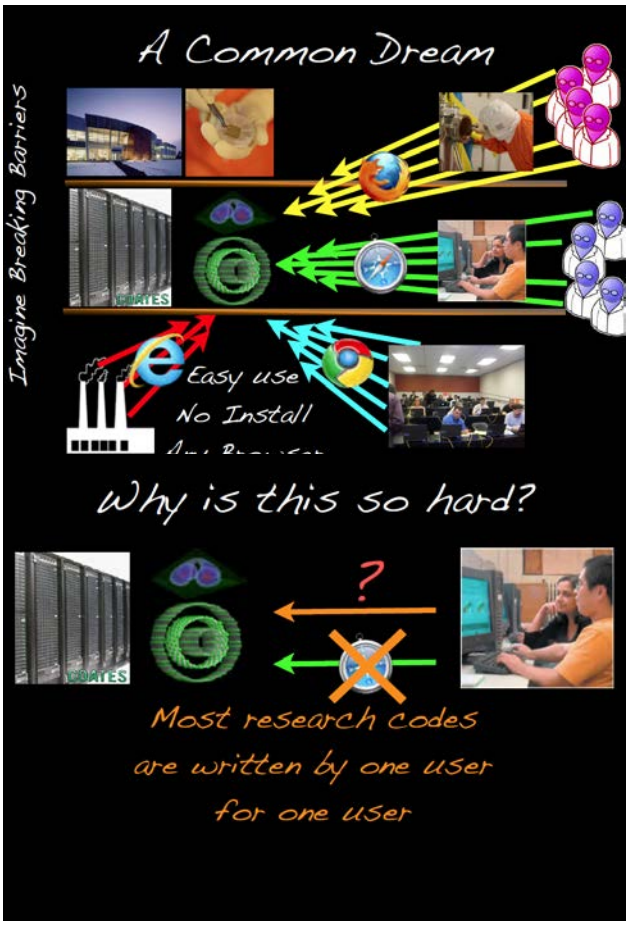


You guys are material scientists. I don't have to tell you about nanotechnology, right? It's people in bunny suits.

Nano facilities are really expensive.



You typically think about models like these. Here on the slide is a visualization of carbon nanotubes and quantum dots that are artificial atoms. These are models…

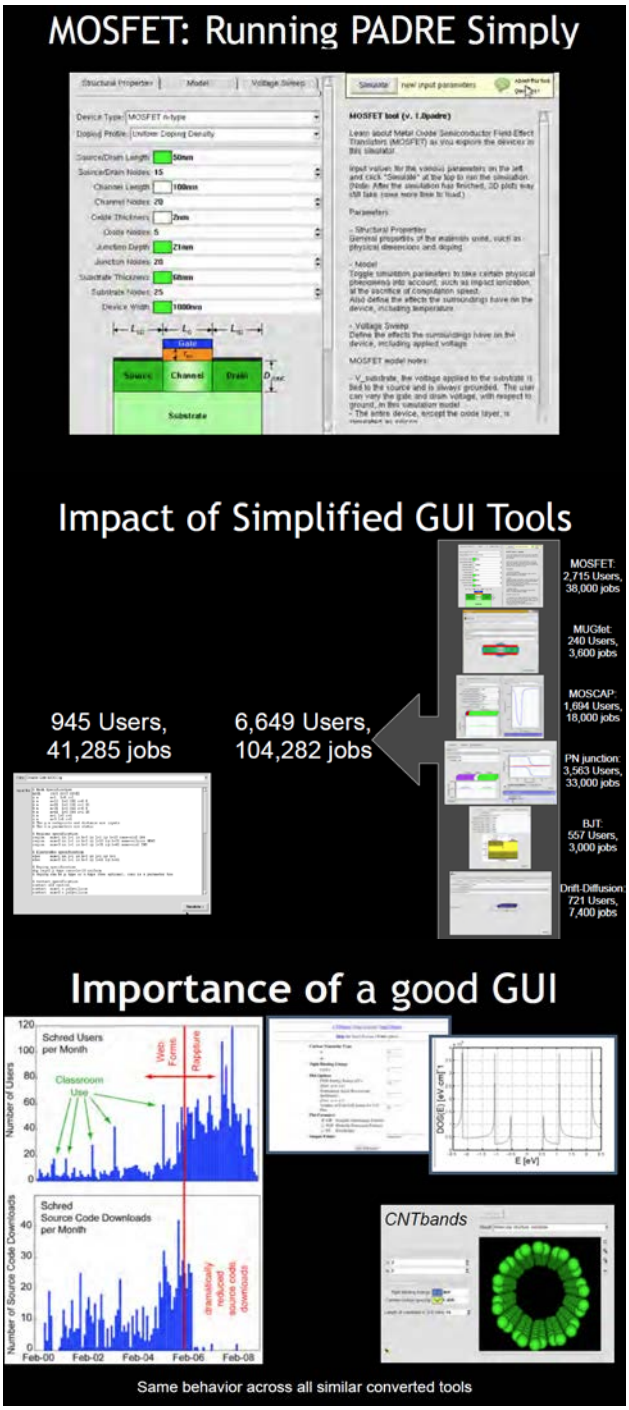


That run on computers, which is why we are here. They are operated by geeks like me. And you probably realize that there is a huge barrier between the experiment and the theory. The idea now is connecting these experimentalists into these models. We would probably have to start with putting these models on the web and making them available. Maybe there needs to be many of these models, not just one, with a whole community working on them. Maybe then there would be lots of other people showing up. But there are other things, right? You could conceive of using this in a classroom, and actually teaching people how to use it and transition at it. You could have economic impact as well. So, all of that should be easy to use, easy to install, and should run into any browser. This is a common dream. You could replace nano and put in materials, or fluid dynamics, whatever you wish. This is a common portal dream. But there are really only few portals that were successful, and nanoHUB was one of them. It has been very hard to implement them, and the question remains: why is this so hard?

This here on the slide is what the key element is: we have got to put this stuff on the web. But most recent codes are written by one person for one person, and that's a major element. That is how we, most of the time, value a PhD. It looks pretty structured like this, but there might be completely garbled nonsense in the input deck where you can't read it and you don't know what it is. You really have to go from user-hostile to something that is more user-friendly, like what is seen here on the slide, where you have a user interface that is actually usable for a broad set of users. So we develop technology that can do that, to get to a user-friendly state. But you also have to be more than that; you have to be developer-friendly. It has to be easy to develop in order to put it onto the web. So we developed Rapture to build user interfaces and we built HUBzero to deploy this stuff on the web. While this has been hard, and many people have tried to build portals, they have also received, basically, an understanding of what's possible and not possible. I call those myths.

On the user side, people think you cannot use research codes for education. They think that you have to specialize your code in order to use it for education, because it's otherwise too hard. Or they think that you must write your own code to do quick research, or to do any sort of research. They also believe that an experimentalist will never touch research codes. Then on the developer side, they believe that building user interfaces is too hard, and the code has to be rewritten to put it on the web anyhow. They think to themselves "Why would I do it? I can write papers myself, thank you very much." On the accessibility side, there was no end-to-end science cloud that existed before we started that. I'm going to pick up some of these topics, but before I dive in you can think of the user-friendly aspect as customers that actually need to use the stuff, the developer-friendly aspect as suppliers that provide it to the customers, and the accessibility aspect as the marketplace where they can meet. You have to analyze these three different stakeholders to do this work.

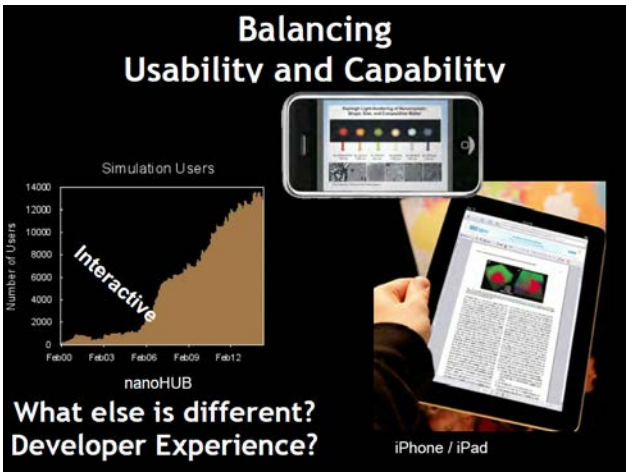
We're going to dive in and look at the user-friendly issue. This here on the slide might be a garbled nonsense input deck. We can have things like that, but it's still hard to use, and a visualization would be nice. That's the first, simplest way of getting access to an industrial strength tool like PADRE. But we can do much better.



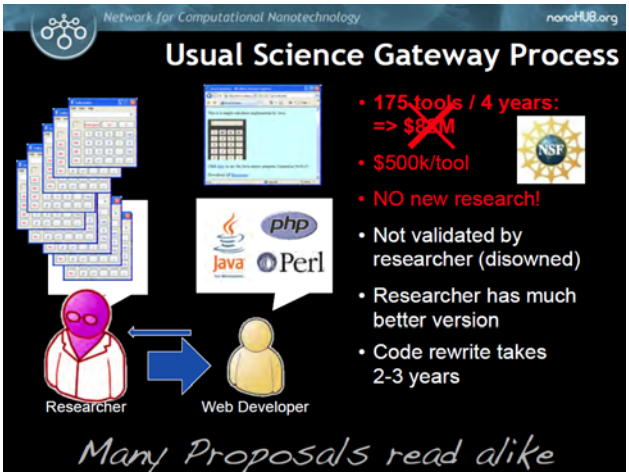
We can wrap PADRE with the help of an undergraduate in two or three weeks over the summer, with a toolset like this, with a GUI, and suddenly you can run the same PADRE tool very easily, and it actually makes sense. It's restrictive, in that it can't do everything PADRE can do, but it can do these MOS devices really well.

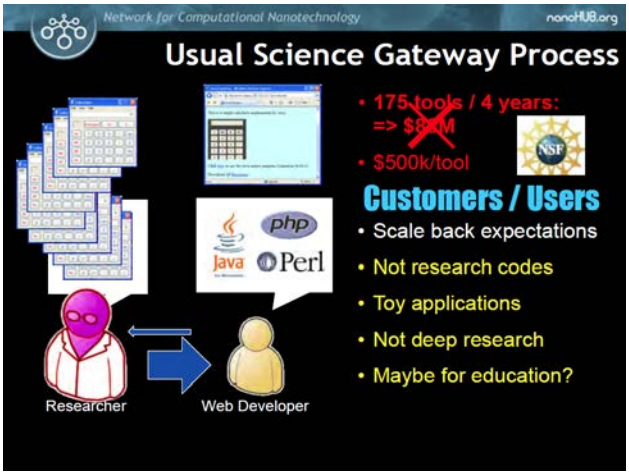
We've done this for six tools. In this PADRE case, we get six thousand users and 100,000 runs versus just 945 users obtained without a simplified GUI. Evidently, usability is a key element.

This here on the slide is what web forums used to look like. You have some form, you fill it out, and you click run. On the left of the slide are some user numbers of this tool up until it was changed. Here, roughly eight users per month, a couple of poor suckers in their classroom, used it. We then changed it to something that's more interactive, and since then we've seen a little bit of growth in these user numbers, not by advertising it, however: it just happened. What's even more exciting is that this tool is one of the open source tools. Not every tool is open source, and you don't have to make it open source. But we tracked the downloads of the source. Those eight users that used it also downloaded it because the service was subpar. But once the service was actually nice, they didn't download it anymore because they didn't have to. They preferred not to download and install the whole thing, find the right FORTRAN compiler, etcetera. Is this new? No. The user interface is not a new thing.

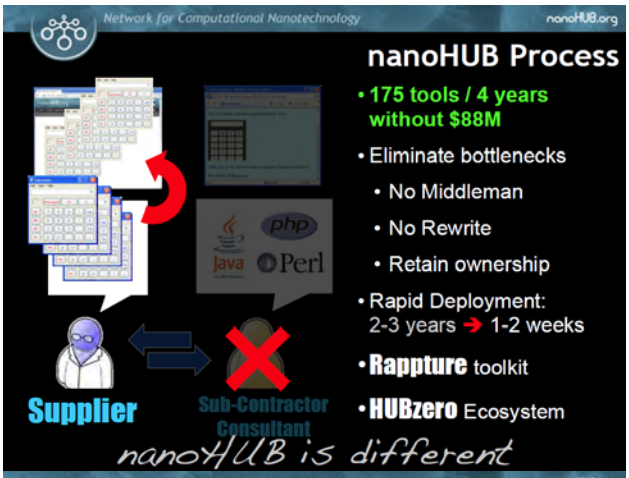


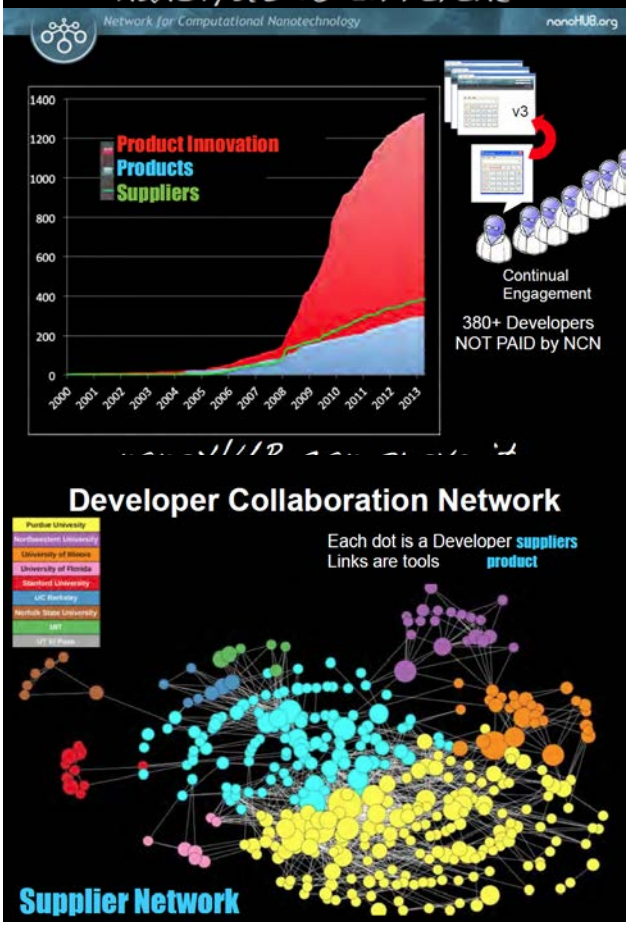
You have seen this all the time. People trade usability for capability. Remember when the first iPhones came out? They had crappy CPUs in them, and they were all so slow. But nobody had to read the 200 page manual you had to read for the Windows-based phone. These thing have changed how we access computers because they're usable. No one needs to read a 200 page manual. Are the processors inferior to what you can buy with Microsoft? Yes. Could you do well if you read the 200 page manual? Yes. Will you read the 200 page manual? No. Thus, you get something usable that's maybe not quite as capable. In science, we have to get used to that. Not everyone wants to be compiling tools and installing things. They want to use it. At nanoHUB, that's what we see. Coming from kind of usable to really usable, you see a little bit of growth. What is different in nanoHUB? Why could we do that?





Let's look at the developer-friendly piece. Normally, a proposal about putting a tool on the web reads like this. I'm going to have to get some money. I'm going to hire a web developer. I'm going to take the software and convert it for web use. Basically, I end up having to rewrite the software. That's the standard model of a portal from the early 2000s. There's a lot of information flowing one way. It takes two to three years; it's kind of slow. The tool here keeps developing. The researcher gets ticked off, because the web developer knows nothing about the science here. This tool keeps evolving, yet the web developer says "Don't give me a new version, I'm recoding everything". In other words, no new science is being done, and it takes a couple of years, 500 K maybe per tool. The researcher gets upset because you can't do that; it does not scale. You can't take 175 tools at that budget in four years and ask for 88 million dollars. What happens then is people scale back their expectations. They say "These are toy tools" because they has nothing to do with the original ongoing research piece. Maybe you can use it for education. But in the end, no deep research is being done. That builds a bad reputation for this whole portal business.

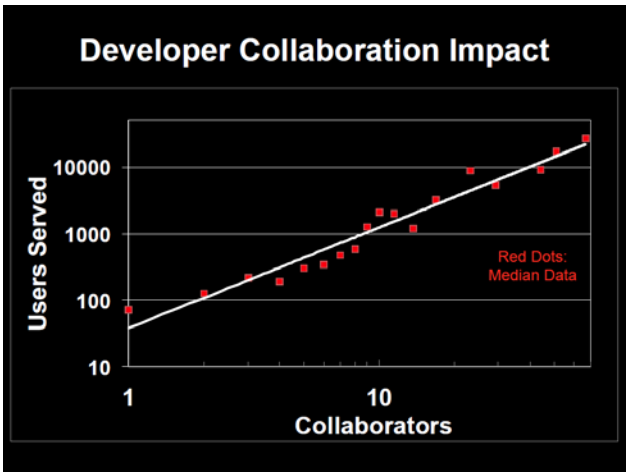


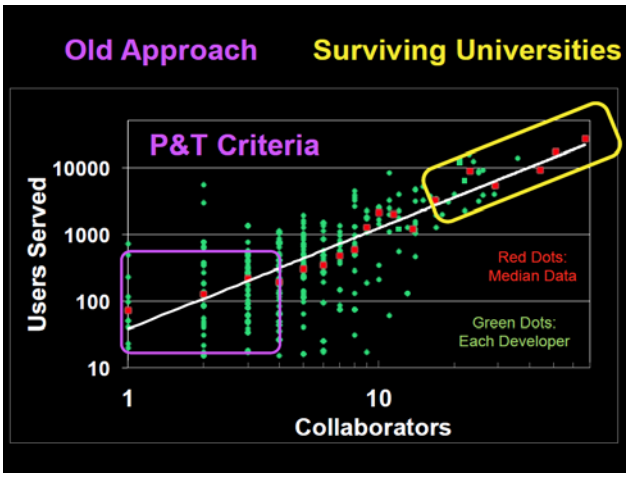


The nanoHUB way is a little bit different. We did deploy 175 tools without 88 million dollars. We basically took away the middle man. We empowered our researchers to deploy their own tools in nanoHUB. Problem solved: they take ownership. If you have a tool with a user interface, it takes one to two weeks, that's it. If you don't have a user interface, it takes a little longer, and maybe an undergraduate student to help with that. That is the difference.

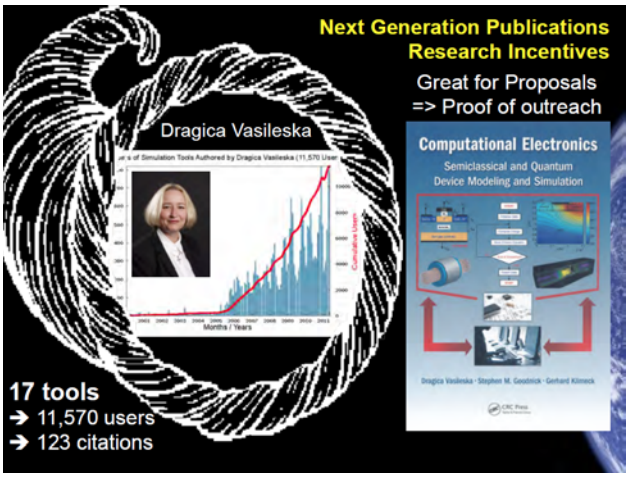
I want to prove to you with numbers that this actually works. On this graph featured on this slide, in blue, is the number of tools. The green line represents the number of active developers. They have new versions. And now, in this statistic in red from two years ago, we have some 1400 tool versions. These tools are no longer the static thing that you throw all over the wall as tar balls and forget about. These things evolve. In fact, they have publications now. Web of Science is now listing these tools as publications.

This method works. Here on the slide is an example of a collaboration network. Each dot is a person developing software. They are linked by common tools: yellow is Purdue, orange is Illinois, purple is Northwestern. Those are part of the original NCM. Turquoise is really interesting. Those are the people that we formally have no connection to. Now, I want to do something with this beyond showing a pretty picture. I want to measure the impact of collaboration.

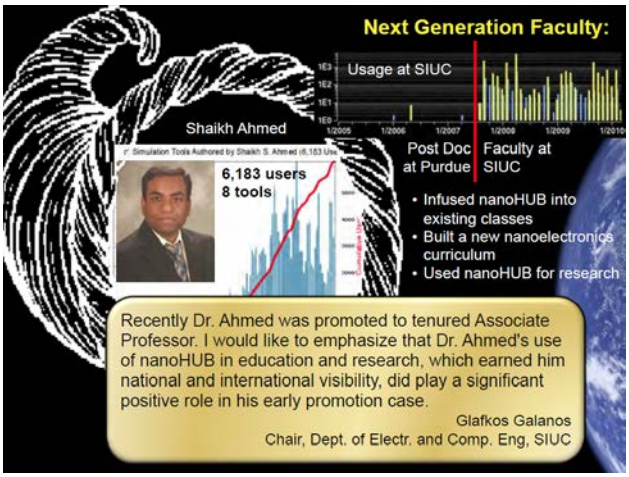




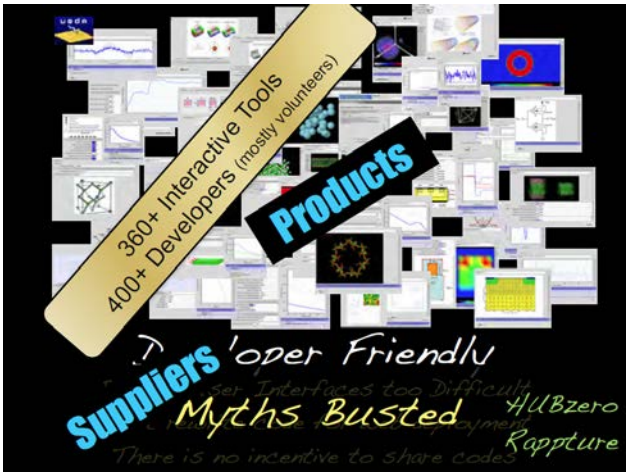
I measure from each dot on this graph the number of lines that go away, that's the number of collaborations you have, versus the number of users you ultimately serve with that tool. What you see here is a curve that is not linear, but it goes up. I showed you the median data first, just to prove it's not just a cumulative effect. I need to put it on a log log scale; this is an exponential impact. If you collaborate, you have impact. This is median data. If I put in all 285 developers I have in this particular data set, you see something very interesting. Each dot is, again, a person, measuring their collaboration strength versus the users they serve. What does that say? If you have a lot of collaboration, you're working up here at the high end of the graph. If you have few collaborations, your success is basically a crapshoot. You could be either very successful or you could be very unsuccessful, statistically speaking. The point is, normally, we work at the lower end of the graph. In academia, we worked at the low end. What's even worse is promotion and tenure criteria are also down there. We should be working at the high end of the graph. We should value collaboration and build it into the tenure process, and not restrict ourselves to the low end of the graph.



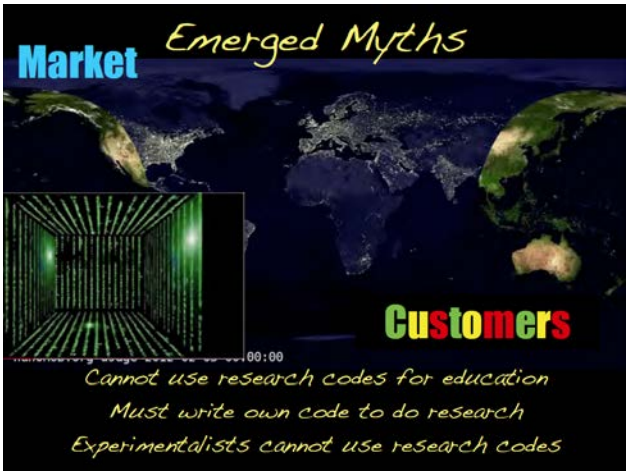
We need to build incentives. Dragica Vasileska, a colleague of mine at ASU, had the shred tool early on. She served some 11 thousand users. Every contributor gets a graph like the one depicted on the slide. It looks really good on a NSF proposal that you don't promise that eventually or maybe in the future you will put your stuff on the web. But you actually measure the impact now. We also write books; these tools will soon be listed by Webble Science as real, proper publications.



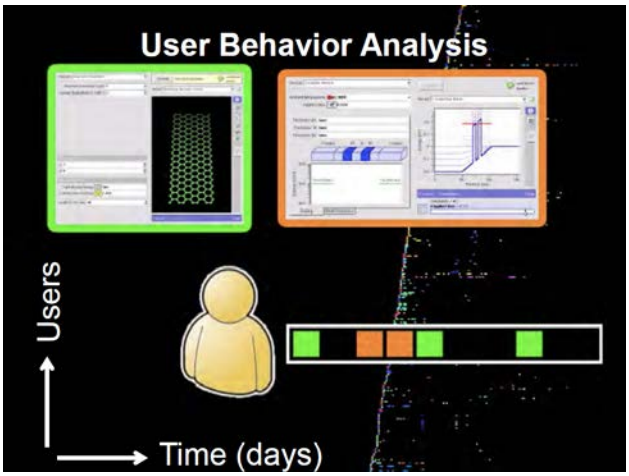
Here is an even more interesting story. A former post-doc in my group who published a bunch of tools went to Southern Illinois. There was no activity at Illinois there before. He arrives, and suddenly there's activity, as seen on this graph of usage at the university. Now there's activity there. He uses existing classes. He builds new classes. And he gets promoted with tenure in within two years.



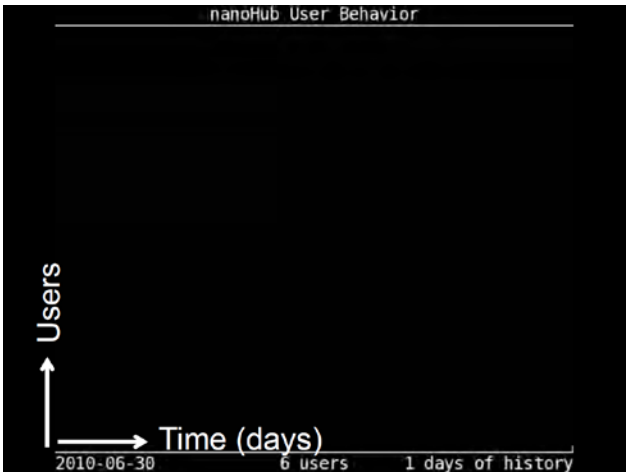
We have some 350 tools now. You can host them. You can run them. They are small tools in general, but there are often big tools as well. So we have the suppliers. We have the products. Now the question is what in the world are they doing.



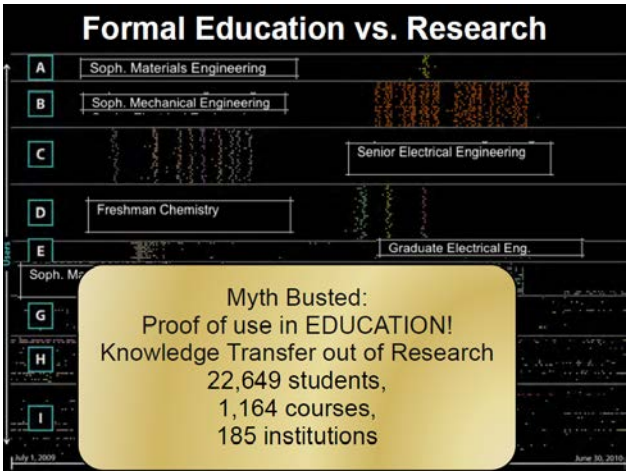
First of all, we operate the thing. But what are these customers doing? The rumors are that you can't use research codes for education, that you can't use somebody else's code to do research, and that experimentalists won't touch it. What we have then is our own matrix. We have to stare into this thing and figure out what they do.



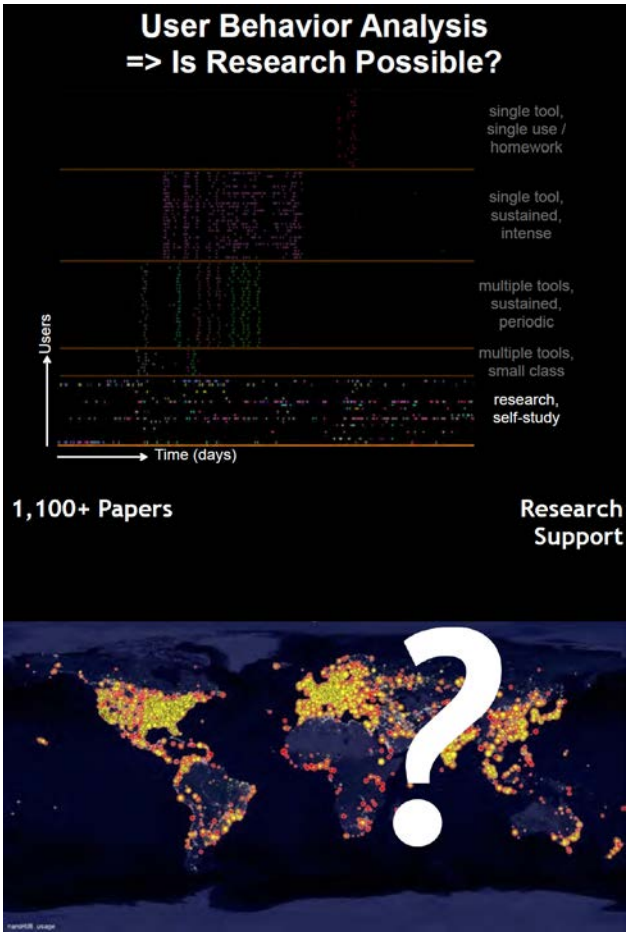
This here on the slide is my view of our matrix. There's a user using the green tool on one day, not coming back the next day, then using the orange tool the third and fourth day, and returning to the green tool on the fifth day. Then I stack these users up from there in sections based on when they first show up on nanoHUB.

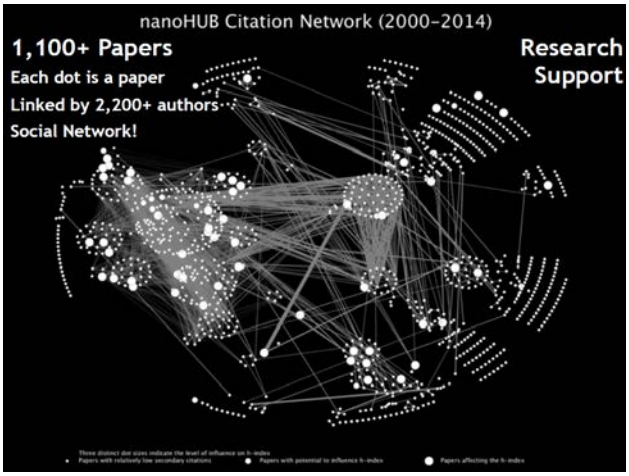


Here's our matrix on the slide. We are looking back at data from 2010; it looks pretty scattered. But then you start staring in it, and you see some little patterns here. You see a stripe there. More patterns: little chunks, big chunks. You start to wonder what is going on. There must be correlated behavior.



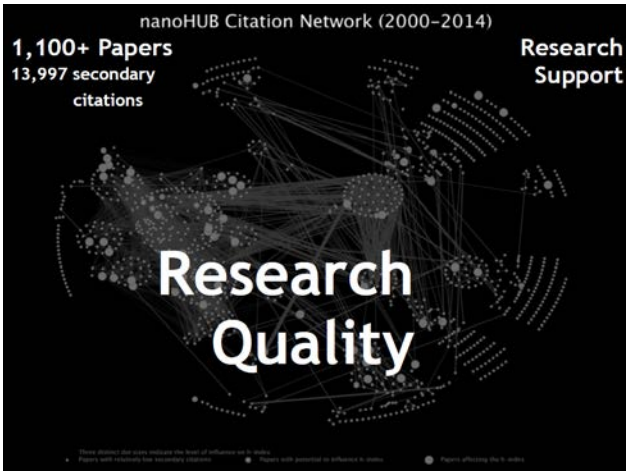
We analyzed user-to-user correlations. There are groups like the one in row A that basically show up once, using one tool. There's another group that might look like this in row B, coming in periodic patterns. Here in row C there are six different tools. These are classes. We can measure the classroom size by peoples' behavior. Nobody would give us in the feedback upfront why they signed up on nanoHUB, since most students would say that their professor told them to. Which professor? For what class? We would know. Thus, we can positively identify them. We have other ways too. I can show you that these groups are experimental researchers, these are computational researchers, and these are self-study users. What's really amazing is that these are research groups. There are 22,000 students in over 1,100 classes, all over the globe, in 185 institutions. It's viral. It's taking off.



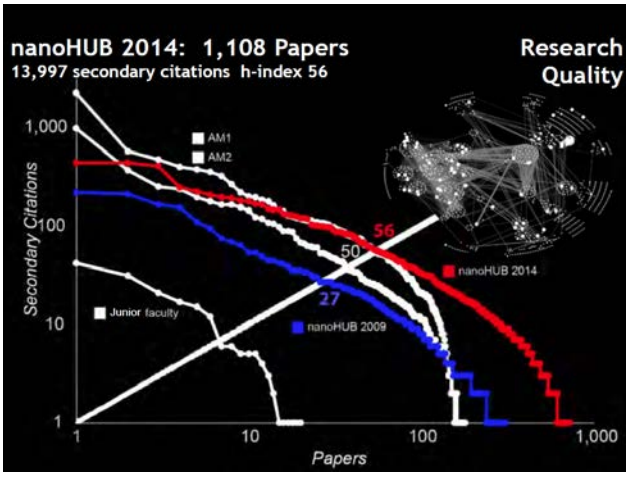


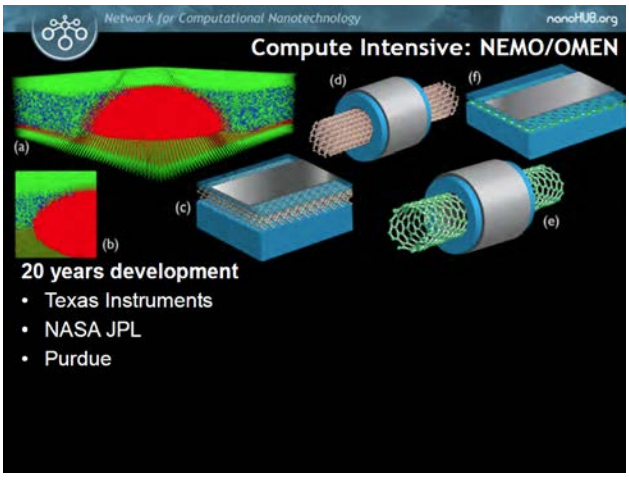
They use some two hundred and ten tools. So what's with this stuff here? What is being shown in the area denoted "Research Activity".

Can we support research? How do you measure research? You look in the literature and develop a process. We found over 1200 papers that cite nanoHUB. We can build social network charts, as depicted on the slide. They are linked by 2,200 authors. We can delineate; do we know these people? Do we not know these people? 64 percent are outside of the original network. We can ask "What are they doing? Are they doing nano-research?" There are actually papers that describe how you use nanoHUB in the education realm. There are a bunch of cyber infrastructure papers. We can look at the nano-papers, and over 50 percent of them support experiments. But is this good research? To determine this, we look at secondary citations. Is this good research?

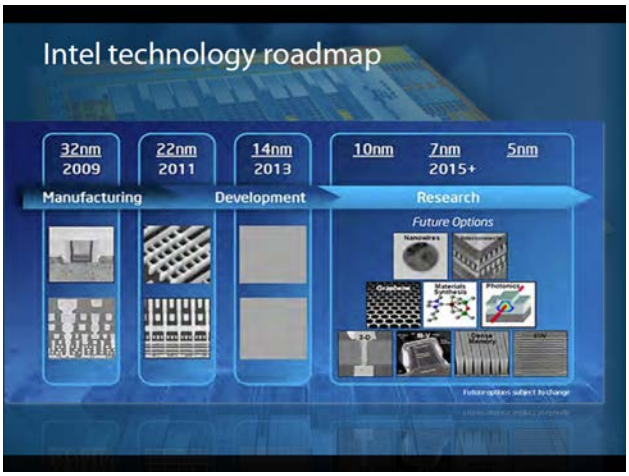


You have all heard of the h-index. Basically, the nanoHUB index is 57 or 58 right now after being about ten years in business.

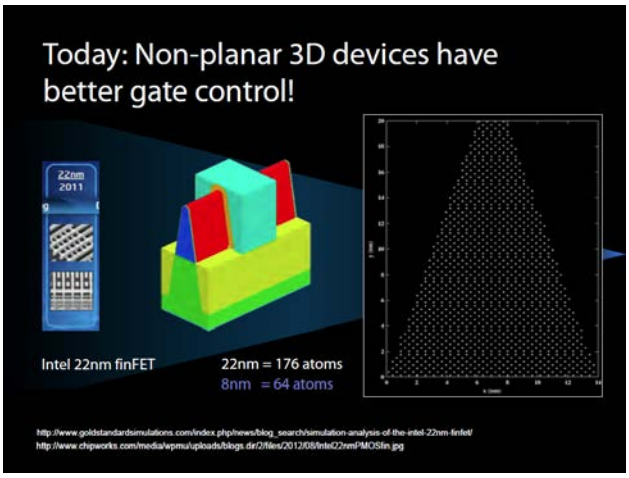




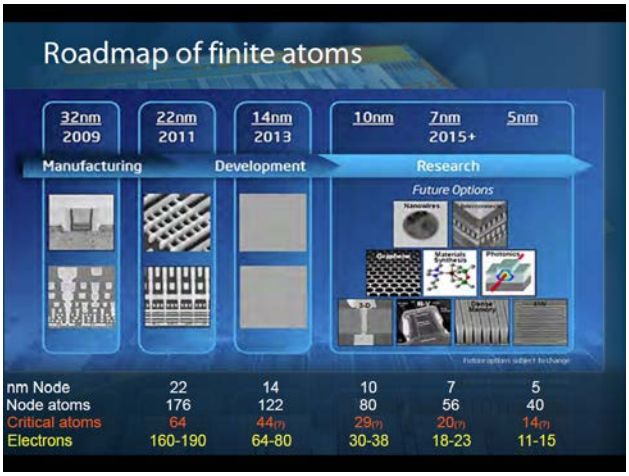
Not only can you do research, it actually seems to be good research. Then there's the perception that these are toy applications, since they are kind of small and don't run big jobs. That's my pet peeve here. That was going to be my talk that I was going to give on the development of NEMO. It has been about 20 years of development. I used to be at Texas Instruments. I used to be at NASA JPL. Now I'm back at Purdue, because I love them more than the ocean.



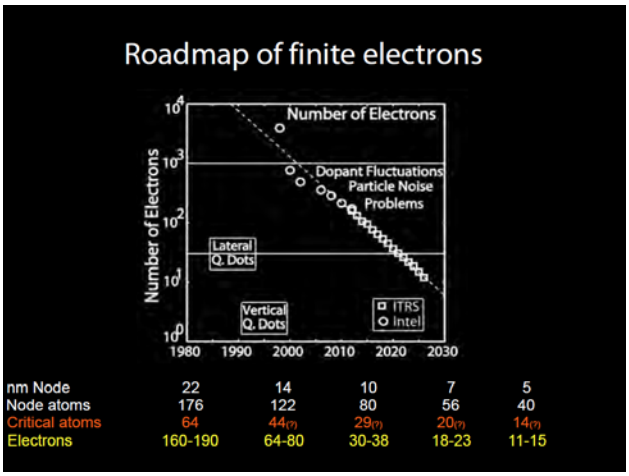
I'm going to look real quick here at the Intel Road Map. We've heard of nodes before.



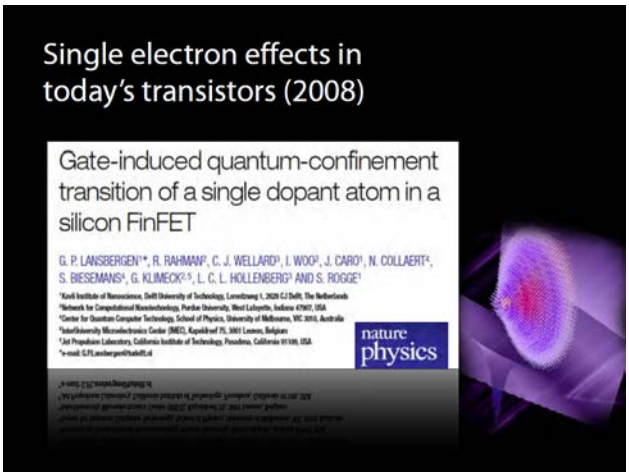
This here in the middle of the slide is today's FinFET that you can buy in devices. The i7 that's in there has FinFETs in there I believe. FinFET in a nice model looks like this solid model. This is the fin, this is the gate, and this is the electric valve that squeezes a flow. Now if you look at that more realistically, and with a SEM, this fin is eight nanometers wide. It might be 22 nanometers long on the gate. Eight nanometers is 64 atoms. It's a small number. If I put this into my model, it will look more like this here on the right; it's about a thousand atoms in the cross-section.



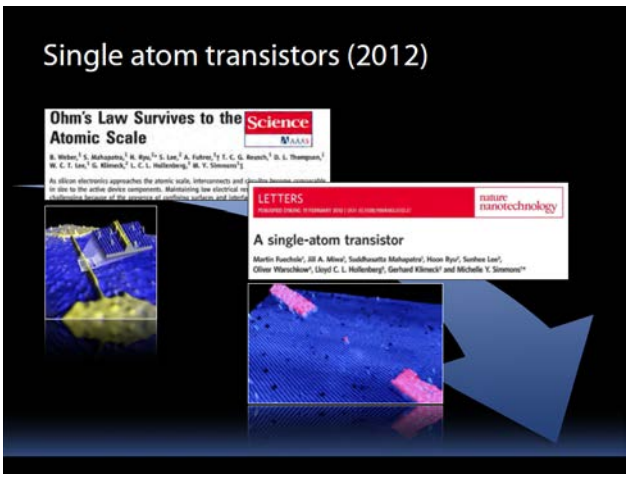
Let's go back to the technology roadmap. If you go down the line here, you can buy a 22 nanometer node today. You can probably buy a 14 nanometer node this year or at the end of next year. 22 nanometers translates to 176 atoms. But if I look at the critical atoms, what's in the width, a 22 nanometer node is 64 atoms. This number goes to tens of atoms in the future; it's very small. I don't think continuum theory will work at that point. I know it doesn't work here in the present, and that's why I work with Intel. What's even more interesting is how many electrons are sitting under the gate that actually control that valve: hundreds in the year 2011, down to tens after the year 2015. That's why we do this atomistic modeling in my group.



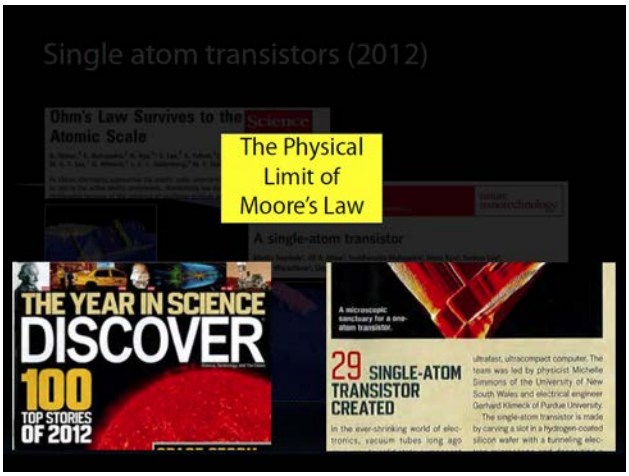
We've worked in quantum dot type things with single electronics.



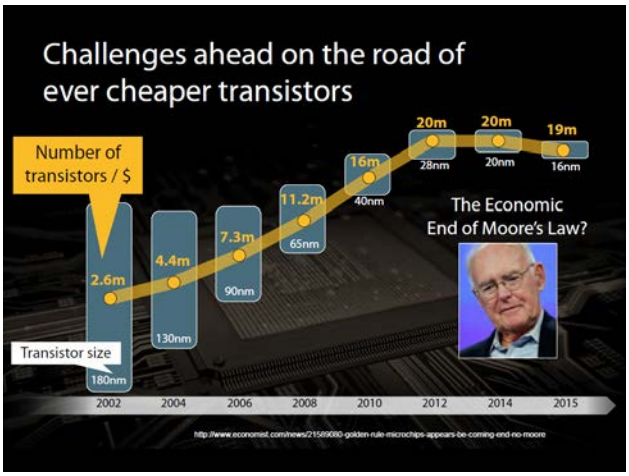
You can see single electron effects in FinFET. You can actually identify what the impurity is with an atomistic simulation using NEMO.



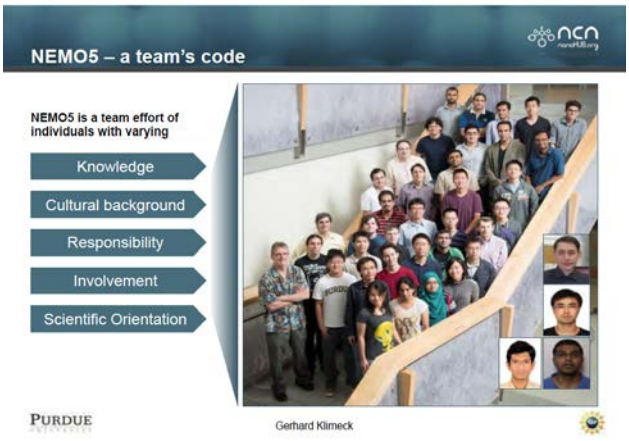
We work with the Michelle Simmons group in Sydney where we can make a predictive single impurity being connected to wires that are one atom tall and four atoms wide. That's about as small as you can make nanoelectronics.



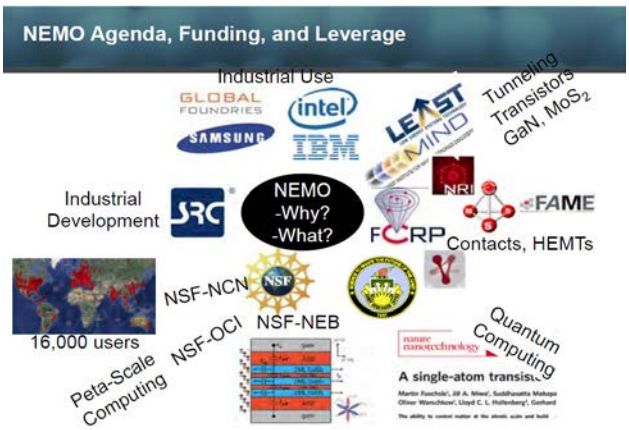
That's the end of Moore's Law in a sense: the physical limit.



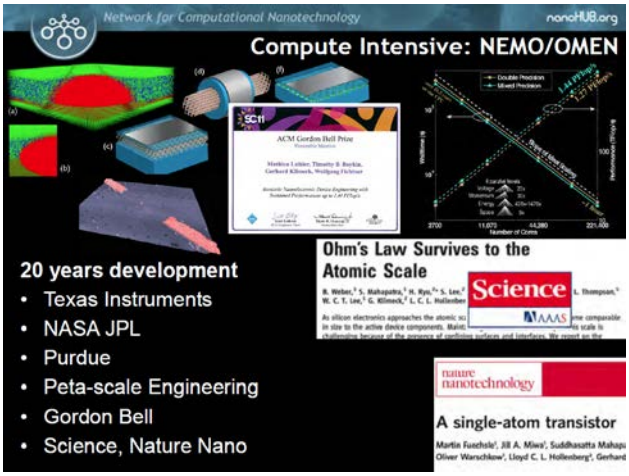
The economic limit of Moore's Law has hit us already. We cannot buy more transistors for the same amount of money anymore as we move forward. Moore's original economic law has stopped. This is the first time in over 40 years that we cannot buy more for the same amount of money. That really calls into question why are we trying to pursue that. It's not clear that there will even be a five nanometer node. People say that they want to have one. They will have to pay for it then. That's ultimately what it comes down to.



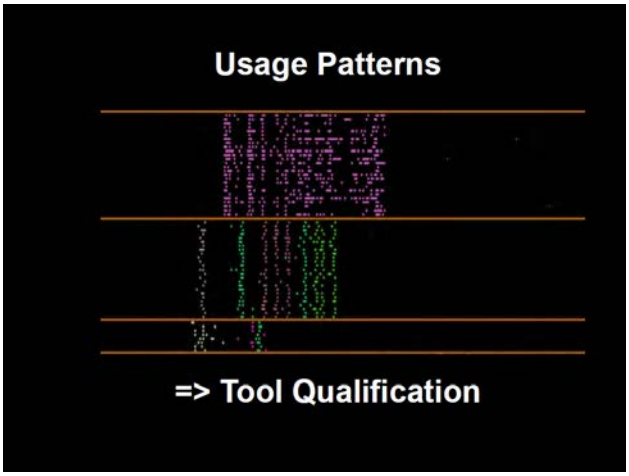
It's done with a group of people.



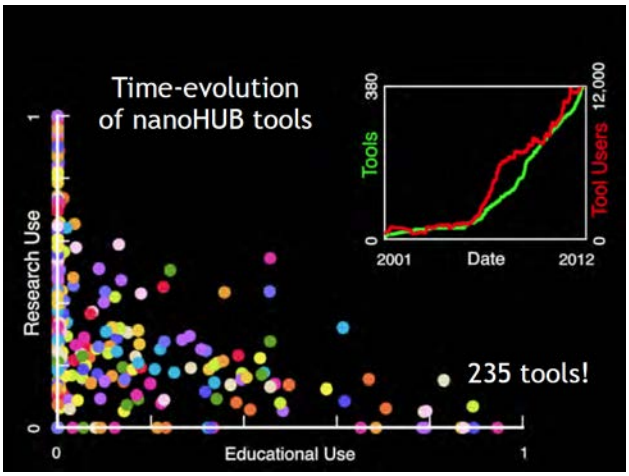
The only way I can work with a group of about 30 people to build this NEMO software is to have a very diverse setup portfolio. I do not have a single funding agency that would fund it, so it has to come from small pockets of money from all over the place.



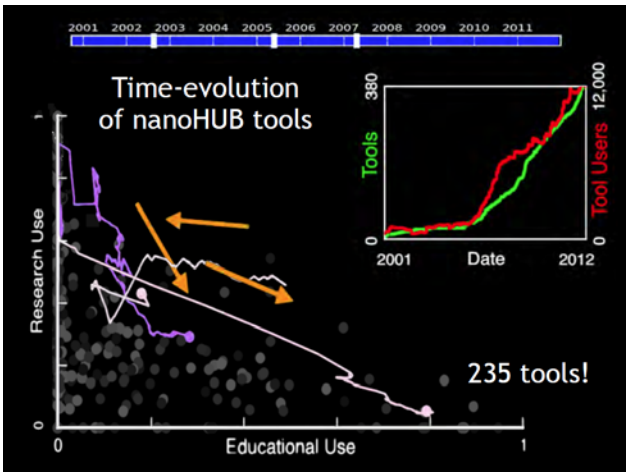
This code depicted on the right of the slide is the first engineering peta-scale code. It runs at a peta-scale at 200,000 cores in cranks of 1.4 petaflops. It has the same code basis as these entities on the slide, but what's even more exciting to me is something I could never have done at JPL: I can let that code loose, with interfaces, and get 18,000 users all over the world to use it with real simulations jobs. And I would never have imagined that they would actually use it in systemic classes that I can now track. You can do computation intensive stuff. NEMO's not the only big thing. All these codes appearing on the slide should mean something to you. They also run in nanoHUB. That is a possibility for deployment for a whole community rather than just throwing tar balls of my software over the wall.

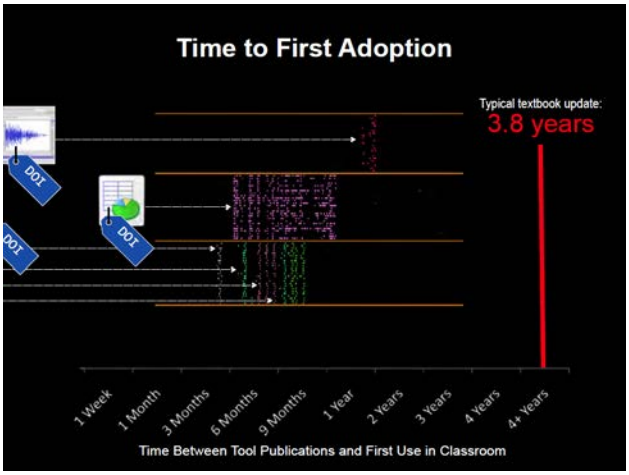


We have these usage patterns that close back onto tool qualification. I can now say that they are used in classrooms. I can rank these things from zero to one in terms of intensity of classroom use. I showed you the social network chart of research use of these tools; I can put that on another axis. Five years ago, I was asked the following question: "You have 120 tools, which ones are research and which ones are educational?" Back then, I thought "I'm not the author, how would I know?" But now I can tell you, with quantitative data, that many tools have bridged that gap. It's not just education. It's not just research. It's dual use. And that's where we want it to be; we know that already. SPICE came from teaching, and moved into the realm of research.

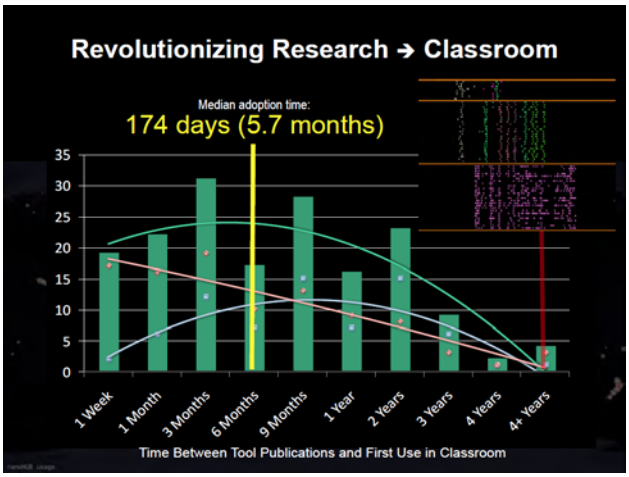


I can look back at 12 years of data, depicted on the time-evolution graph on the slide. Tools are born down here at the origin, and I don't know what they are going to be. In time they evolve, typically coming from the research side, trickling towards education. Releasing programs such as Rapture accelerates the process. Post-Rapture, you see a whole lot more tools popping in there. But in general you see the trend. To me, that's the essence of a research university.

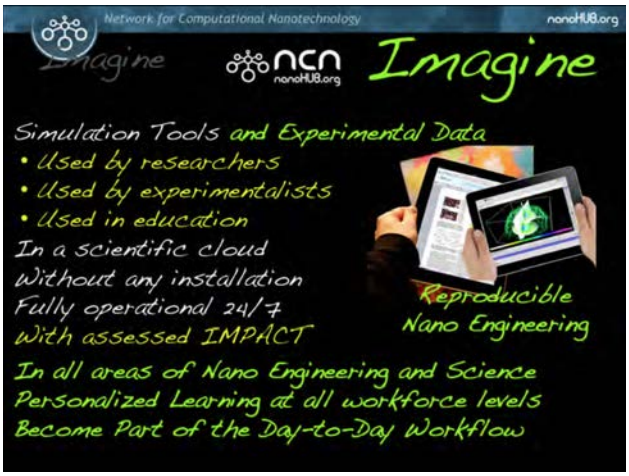




I can now measure time from market into the classroom. Each tool has a digital optic identifier that's essentially its birthday; I can measure the first time they shows up in a classroom. Writing a textbook normally takes about four years on average...



Now we have tools that show up in a week, a month, or three months. That is a very rapid infusion of new research into classrooms.



We had this dream of doing this. Now in the future we're going to do experimental data. But really, the big dream is now changing publication processes. I want to be able to read my paper on whatever device I have. I want to click on the data, and I want to link back to the tool or the data, and I want to be able to compare it. We're driving that right now with a couple of publishers: Springer, IEEE, and IOP. That's where we're going to go in the future, and we're going to be part of it. We're aiming to make this sustainable. We are dreaming of a sort of professional society that fosters the sustainability of a hub, so we don't have to go back to the government to ask for more money.

Thank you.

Simulation of Materials Properties Using the Tight-Binding Method

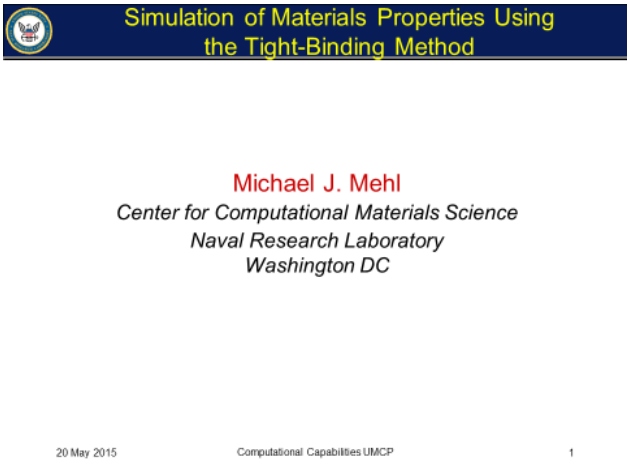
Dr. Michael J. Mehl
Center for Computational Materials Science, Naval Research Laboratory
Washington DC

Abstract

Over the last two decades computational power has increased enough so that quantum-mechanically accurate simulations of systems containing hundreds or even thousands of atoms are routine. Even so, many calculations, *e.g.* grain boundaries or surface reconstruction, can require even more atoms for an accurate description. It is desirable to have a method that can accurately describe these systems while maintaining quantum-mechanical accuracy.
The NRL Tight-Binding method was developed to handle these calculations. An extension of the Slater-Koster formalism, the tight-binding parameters are chosen to reproduce the total energy and electronic structure for a series of relatively small first-principles calculations approximating the systems to be studied. This reduces a calculation requiring 100 or so basis functions per atom to one requiring only 9-16 basis functions per atom, giving an immense speedup in the calculation and allowing much larger systems to be studied.
This talk will describe the development of the NRL-TB, its successes, and some of the problems encountered along the way. Plans for improvements in the method will also be discussed.

Biography

Michael J. Mehl graduated from the University of Kansas in 1973. He received his M.S. and Ph.D. degrees from Indiana University in 1975 and 1980, respectively. From 1979-1981 Dr. Mehl was a Postdoctoral Fellow at Rutgers University, working with Prof. David Langreth. He held a similar position at the University of Maryland from 1981-1983, working with Prof. Ted Einstein. Dr. Mehl came to NRL in 1983 as a contractor, working for Sachs-Freeman Associates. He joined the Condensed Matter Theory Branch as a full-time employee in 1986. In 2006 he was named the Head of the Center for Computational Materials Science. Dr. Mehl was named a Fellow of the American Physical Society in 1999.



The subject of the talk is not exactly a success story, but a story still in development. I've enjoyed this, and I'm not going to say we have thousands of users or anything but we have gotten quite a bit of interest in it over the years.

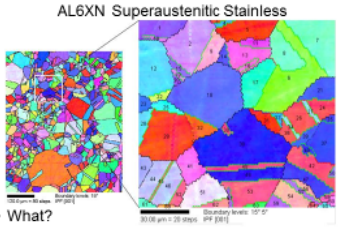
As I was introduced, I am now the head of the Center for Computational Material Science, which has about twelve scientists, and sometime in the near future I'll be the head of the Center for Science and Materials Technology which has about thirty people.

These are the collaborators, past and present. This project was started back in the 1990's by Dimitris Papaconstantopoulos and Ron Cohen when we were all at NRL. Ron has since moved to Carnegie Geophysical Laboratory and Dimitris is incorrectly labeled on the slide and is at George Mason University. There are various other collaborators (indicating the rest of the names on the slide) including Noam Bernstein who is still a major developer and user on this project.

This project received sponsorship from multiple sources including ONR (Office of Naval Research) back in the 90's as part of the Design of Naval Steels project, we receive a lot of our computing capability from the HPC project that Doug talked about earlier today, and we received funding originally from a project called Common High Performance Computing Software Support Initiative. Essentially what CHSSI is, is that in the 90's the DOD funded HPC, providing a lot of parallel computers, and what they realized is that there was no software to run on parallel computers. So we had these old programs written in FORTRAN77, I hate to say it my first FORTRAN books were WATFOR and WATFIVE, and we inserted MPI calls along with other parallelization enhancements.

Finding Materials Properties

AL6XN Superaustenitic Stainless



- Why?
  - Improvements: Ductility, Strength, Weldability, Melting Temperature, Magnetization
  - Determine Failure Modes
  - Decrease Total Ownership Costs
- How?
  - First-principles calculations
  - Tight-Binding and atomistic potentials
  - Finite Strain Models
  - Thermochemical Modeling
- What?
  - Structural Energy of Phases
  - Dynamic Response (Phonons, C_{ij})
  - Grain Boundary Energies
  - Grain Boundary Segregation
  - Dislocation Energies and Motion
  - Crack Formation and Propagation

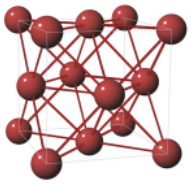
20 May 2015 Computational Capabilities UMCP 4

So what we wanted to do was to devise an atomistic basis for complicated structures like this (indicating the material image in graphic 1.b). If you notice the scale of the graphic you will see that there are a lot of atoms in the grain boundaries. As a result you cannot do atomistic simulations, as described earlier, because with first principles code such as MD you're lucky to get 500 or 1000 atoms and maybe 10,000 by next decade, however you will always want to simulate more atoms than you have capability to do.

Real First Principles Calculations

Example: Bulk Modulus of Copper

- fcc unit cell, 29 electrons/atom
- Need about 10^9 atoms to describe a "bulk material"
  $\Rightarrow 2.9\times10^{10}$ electrons
- Want to solve with Quantum Mechanical Accuracy



Solve Schrödinger's equation:

$$\left[-\frac{\hbar^2}{2m}\sum_i \nabla_i^2 + \frac{1}{2}\sum_{i\neq j}\frac{e^2}{|\mathbf{r}_i-\mathbf{r}_j|} + \sum_{\alpha i}\frac{Z_\alpha e^2}{|\mathbf{r}_i-\mathbf{R}_\alpha|}\right]\psi(\mathbf{r}_1,\mathbf{r}_2,\mathbf{r}_3,\cdots,\mathbf{r}_N) = E\,\psi(\mathbf{r}_1,\mathbf{r}_2,\mathbf{r}_3,\cdots,\mathbf{r}_N)$$

for E as a function of atomic positions $\mathbf{R}_\alpha$, bulk modulus is

B(V) = V E'' (V)

Equation has $\approx 10^{11}$ dimensions, excluding spin

20 May 2015 Computational Capabilities UMCP 5

If you are doing real first principles calculations for instance let's say you have copper, you'd need about a million atoms to get something you could call copper bulk. That translates to about 29 billion electrons each with three degrees of freedom, not accounting for spin, leaving you with a 100 billion dimension equation and I don't think we will have the capability to do that any time soon.

Simplification: Density Functional Theory

- Density Functional Theorem:
  The energy of the system is a function of its electronic density
  $$E = E[n(\mathbf{r})]$$
  (Hohenberg and Kohn, 1964)
- Single Particle Approximation:
  Pretend that the electrons are independent of one another (Slater, 1930s; Kohn and Sham, 1965):
  $$-\frac{\hbar^2}{2m}\nabla^2\psi_n(\mathbf{r}) + v_{KS}(\mathbf{r})\,\psi_n(\mathbf{r}) = \varepsilon_n\,\psi_n(\mathbf{r})$$
  Replaced 10^{11} dimensional equation by 3×10^{10} three dimensional equations

In a periodic crystal we can reduce this to < 100 equations per electron in the primitive cell

20 May 2015 Computational Capabilities UMCP 6

So the simplification came from Hohenberg and Kohn, who developed Density Functional Theory (DFT) Speaker indicates Eqn. 1) as physicists, although Walter Kohn was eventually given the Nobel Prize for Chemistry. Then Kohn and Sham reduced the 100 billion dimensional problem down to 3 billion three dimensional equations which is actually much simpler. Furthermore, if we are dealing with a periodic crystal we can reduce this problem down to about 100 equations, or wavefunctions, per atom in a primitive cell which is ultimately what we really need to calculate.

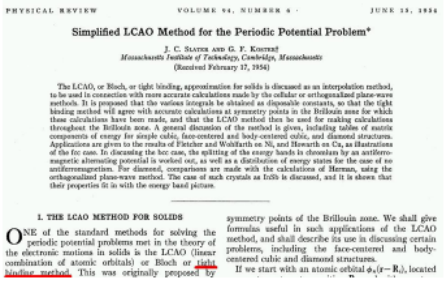
Why Not Use DFT for Everything?

- Relatively Slow
  - Computation time to solve Schrödinger's equation scales with the *cube* of the number of atoms (O[N^3] problem)
    - O[N] methods have been developed for insulators, but are much more difficult to implement for metals
  - Memory use scales as N^2
  - Iterating to Self-Consistency can take 5-50 steps, depending on the complexity of the system
    - Acceleration schemes sometimes fail
  - Practical limit of a few hundred atoms for systems containing transition metals.
- For larger calculations, we want something that is faster, yet preserves Quantum Mechanical properties
- Dynamics requires multiple solutions as the system evolves
- There is always a bigger system

However, this solution technique (DFT) is still relatively slow as the solution time goes as the number of atoms cubed, and to the fourth if you're doing Hartree-Fock. Furthermore, memory scales as square in the number of atoms due to the matrices which must be stored. Finally, all of this must be done self consistently requiring several iterations. There are schemes which may reduce the number of iterations by increasing response, however the practical limit is a few hundred atoms for transition metals.

For some perspective, this technique is considered to be "second principles" which comes from Alex Zunger whom some people may know, who is now at Boulder. He in the 90's considering first principles, took tight binding, which was based on first principles, and called that technique second principles, and we'll call atomistic potentials no principles. The main focus of our work with tight binding was to preserve the quantum mechanical properties, i.e. keep it second principles. Furthermore, there will always be a bigger system people will want simulate. If you produce a version of VASP that can solve a 10,000 atom problem in 20 seconds someone will want to solve a 100,000 atom problem. Thus there is always a bigger system and we haven't systemized that issue.

This problem was faced before



PHYSICAL REVIEW VOLUME 94, NUMBER 6 JUNE 15, 1954

Simplified LCAO Method for the Periodic Potential Problem*

J. C. SLATER AND G. F. KOSTER†
Massachusetts Institute of Technology, Cambridge, Massachusetts
(Received February 17, 1954)

The LCAO, or Bloch, or tight binding, approximation for solids is discussed as an interpolation method, to be used in connection with more accurate calculations made by the cellular or orthogonalized plane-wave methods. It is proposed that the various integrals be obtained as disposable constants, so that the tight binding method will agree with accurate calculations at symmetry points in the Brillouin zone for which these calculations have been made, and that the LCAO method then be used for making calculations throughout the Brillouin zone. A general discussion of the method is given, including tables of matrix components of energy for simple cubic, face-centered and body-centered cubic, and diamond structures. Applications are given to the results of Fletcher and Wohlfarth on Ni, and Howarth on Cu, as illustrations of the fcc case. In discussing the bcc case, the splitting of the energy bands in chromium by an antiferromagnetic alternating potential is worked out, as well as a distribution of energy states for the case of no antiferromagnetism. For diamond, comparisons are made with the calculations of Herman, using the orthogonalized plane-wave method. The case of such crystals as InSb is discussed, and it is shown that their properties fit in with the energy band picture.

I. THE LCAO METHOD FOR SOLIDS

ONE of the standard methods for solving the periodic potential problems met in the theory of the electronic motions in solids is the LCAO (linear combination of atomic orbitals) or Bloch or tight binding method. This was originally proposed by

symmetry points of the Brillouin zone. We shall give formulas useful in such applications of the LCAO method, and shall describe its use in discussing certain problems, including the face-centered and body-centered cubic and diamond structures.

If we start with an atomic orbital $\phi_n(\mathbf{r}-\mathbf{R}_l)$, located

- 1954: difficult to do even a very simple electronic band structure calculation
- Slater and Koster proposed using the tight-binding formalism to develop an interpolation scheme

Well we faced this problem way back in the 1950's when Slater and Koster proposed a tight binding model (Slater was mentioned previously on slide 6).

Solving Schrödinger's Equation

$$H\,\psi_n(\mathbf{r}) = -\frac{\hbar^2}{2m}\nabla^2\psi_n(\mathbf{r}) + v_{KS}(\mathbf{r})\,\psi_n(\mathbf{r}) = \varepsilon_n\,\psi_n(\mathbf{r})$$

- Use variational theorem. For any function $\chi(\mathbf{r})$:

$$\varepsilon \le \langle \chi(\mathbf{r}) \mid H \mid \chi(\mathbf{r}) \rangle$$

- Given basis functions $\varphi_m(\mathbf{r})$, write

$$\chi(\mathbf{r}) = \Sigma_m\, c_m\, \varphi_m(\mathbf{r}-\mathbf{R}_m)$$

- Find stationary solutions of $\langle \chi(\mathbf{r}) \mid H \mid \chi(\mathbf{r}) \rangle$ as a function of the c_m

$$\Sigma_j\, H_{ij}\, c_j = \varepsilon\, \Sigma_j\, S_{ij}\, c_j$$

$$H_{ij} = \langle \varphi_i(\mathbf{r}-\mathbf{R}_i) \mid H \mid \varphi_j(\mathbf{r}-\mathbf{R}_j) \rangle$$

$$S_{ij} = \langle \varphi_i(\mathbf{r}-\mathbf{R}_i) \mid \varphi_j(\mathbf{r}-\mathbf{R}_j) \rangle$$

- Solutions for ε are the approximate eigenvalues
- In DFT we need to construct H_{ij} and S_{ij} at every step
- Slater and Koster chose to *parametrize* H_{ij} and S_{ij} from certain calculations

20 May 2015 Computational Capabilities UMCP 9

What that tight binding model does is put in a very small set of wavefunctions and use that to diagonalize the system, where if the wavefunctions are chosen appropriately you obtain reasonably good solutions. Note that a tight binding wavefunction is a wavefunction that is centered on an atom.

Divide the World Into 4 Types of Integrals

- Calculate various terms for

$$\langle \varphi_i(\mathbf{r}-\mathbf{R}_i) \mid H_k(\mathbf{r}-\mathbf{R}_k) \mid \varphi_j(\mathbf{r}-\mathbf{R}_j) \rangle$$

- Onsite Integrals:

$$\langle \varphi_i(\mathbf{r}-\mathbf{R}_i) \mid H_i(\mathbf{r}-\mathbf{R}_i) \mid \varphi_j(\mathbf{r}-\mathbf{R}_i) \rangle$$

- Crystal Field Integrals:

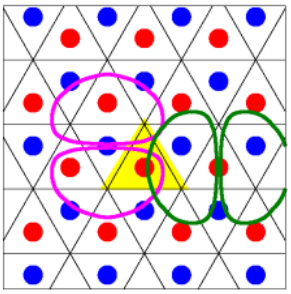
$$\langle \varphi_i(\mathbf{r}-\mathbf{R}_i) \mid H_k(\mathbf{r}-\mathbf{R}_k) \mid \varphi_j(\mathbf{r}-\mathbf{R}_i) \rangle$$

- Two Center Integrals:

$$\langle \varphi_i(\mathbf{r}-\mathbf{R}_i) \mid H_j(\mathbf{r}-\mathbf{R}_j) \mid \varphi_j(\mathbf{r}-\mathbf{R}_i) \rangle$$

- Three Center Integrals:

$$\langle \varphi_i(\mathbf{r}-\mathbf{R}_i) \mid H_k(\mathbf{r}-\mathbf{R}_k) \mid \varphi_j(\mathbf{r}-\mathbf{R}_i) \rangle$$



So Far There Are No Approximations

20 May 2015 Computational Capabilities UMCP 10

What they found (Slater and Koster) is if you divide a crystal up you could have interactions between two different atoms due to the potential produced by a third interacting atom, and you would have to account for all possible interactions. Then any possible interaction would be represented by a matrix element. As a result there are a huge number of possible interactions.

Slater-Koster Approximations

- Combine on-site and crystal field terms into one integral

$$\langle \varphi_i(\mathbf{r}-\mathbf{R}_i) \mid H_k(\mathbf{r}-\mathbf{R}_k) \mid \varphi_j(\mathbf{r}-\mathbf{R}_j) \rangle$$

- ~~Ignore the three-center integrals:~~

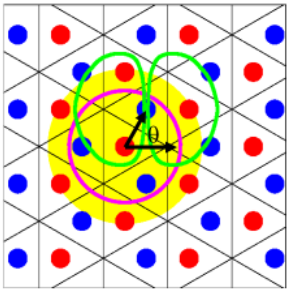
~~$\langle \varphi_i(\mathbf{r}-\mathbf{R}_i) \mid H_k(\mathbf{r}-\mathbf{R}_k) \mid \varphi_j(\mathbf{r}-\mathbf{R}_i) \rangle$~~

- Approximate two Center Integrals with a spherical potential:

$$\langle \varphi_i(\mathbf{r}-\mathbf{R}_i) \mid H_j(|\mathbf{r}-\mathbf{R}_j|) \mid \varphi_j(\mathbf{r}-\mathbf{R}_i) \rangle$$

- Reduces calculations to a very few integrals: on-sites + 10 two-center (*hopping*) integrals, ssσ, spσ, ppσ, ppπ, sdσ, pdσ, pdπ, ddσ, ddπ, and ddδ
- Two-center integrals depend on angle and an irreducible matrix element

This approximate Hamiltonian still has the symmetry of the crystal



$$\langle \varphi | H | \varphi \rangle = \cos(\theta)\, H_{sp\sigma}(R)$$

20 May 2015 Computational Capabilities UMCP 11

What Slater and Koster ended up doing was throw out a lot of those interactions by assuming only a spherical potential centered on an atomic site. This assumption preserves the quantum mechanical basis.

Slater-Koster Approximations

- Given an *spd* basis for the valence bands, need only diagonalize a 9×9 matrix at an **k**-point for a one atom system
- For multiple atom systems it' s a 9N×9N diagonalization
- Comparable DFT calculations require diagonalization of 100N × 100N matrices
- Example from the Slater-Koster 1954 paper for band structure of Cu: only circled points were calculated

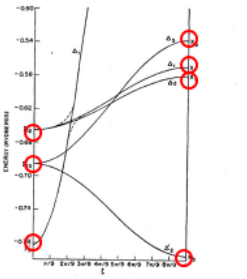


FIG. 1. Variation of energy with wave vector in the 001 direction for Cu calculated using Howarth's energy values at the center and boundary of the zone. (Symmetry symbols are taken from reference 4.)

20 May 2015 Computational Capabilities UMCP 12

What one can do then is using only a simple set of parameters you can do your calculations at the end points of a certain direction in the Brillouin zone. The end points are very small unit cells you can fit easily and then one can interpolate between the endpoints. As a result of the quantum mechanical symmetry, you generally get good fits.

Advantages and Disadvantages of Tight-Binding

Advantages

- Small basis set size (*spd*):
  - Fast enough for Molecular Dynamics
  - Can handle thousands of transition metal atoms
- Maintains Quantum Mechanical nature of bonding
- Transferable to structures not included in the fit, allowing calculation of
  - Elastic Constants
  - Surface and Vacancy energies
  - Phonons
- Can include spin-polarization

Disadvantages

- Parameters must be fit to first-principles calculations or experiment
- Difficult to develop parameters for ternary and higher systems
- This an *interpolation*, not an extrapolation: "Interesting" results should be checked by first-principles calculations and/or experiment

20 May 2015 Computational Capabilities UMCP 13

So if we are using tight-binding, and can neglect transition metals, we are able to use an spd basis set and therefore only have 9 bases per atom, where compared to VASP we might have 100 basis states per atom. Furthermore, we can preserve the quantum mechanical nature of the binding and if we're lucky our results will be transferable to structures we did not fit.

Disadvantages of tight binding is that it is difficult to develop, you must have a first principles calculation. And fitting is very much an art, not a science. Also interpolation is not guaranteed to yield correct results, for example if you have two phases of ice, i.e. ice 2 and ice 9, you are not guaranteed to get ice 5 in the middle as the bonding may differ. As a result one must be careful to check any assumptions.

NRL Tight-Binding

- Derive an approximate Hamiltonian that can be fitted to first principles results
- Express DFT Total Energy as a sum of shifted eigenvalues:

$$E = \Sigma_i \varepsilon_i + F[n(\mathbf{r})] = \Sigma_i \varepsilon_i'$$

- Two-Center Non-Orthogonal Slater-Koster Parametrization:

Hopping: $H_{il\mu}(R) = (a + b\,R + c\,R^2)\exp(-\lambda^2 R)$

Overlap: $S_{il\mu}(R) = (d + e\,R + f\,R^2)\exp(-\kappa^2 R)$

- Onsite terms depend on the density of surrounding atoms:

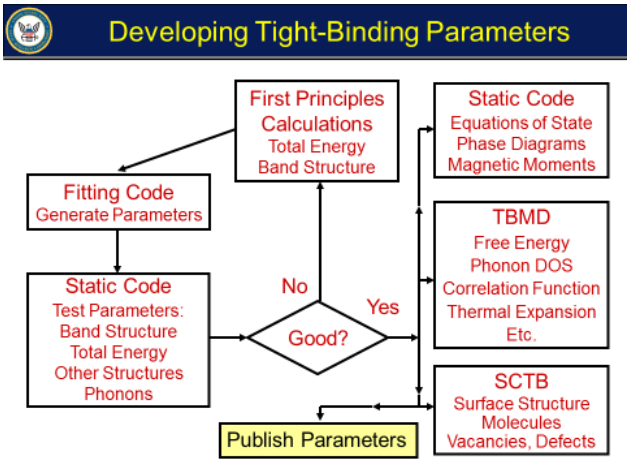
$$\rho_{i\alpha} = \Sigma_{j \subset \beta} \exp(-\zeta_\beta^2 R)$$

$$h_i(\alpha) = \varepsilon_\alpha + \Sigma\beta\,(s_{i\alpha\beta}\,\rho_{i\beta}^{2/3} + t_{i\alpha\beta}\,\rho_{i\beta}^{4/3} + u_{i\alpha\beta}\,\rho_{i\beta}^2)$$

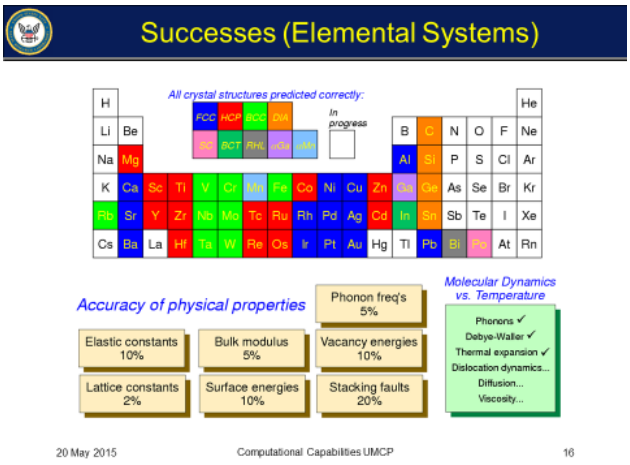
- Single species calculation requires 93 parameters
- Two species calculations require ~ 300 parameters (A-A, B-B, and A-B)
- Fit these parameters to first-principles eigenvalues and total energies

20 May 2015 Computational Capabilities UMCP 14

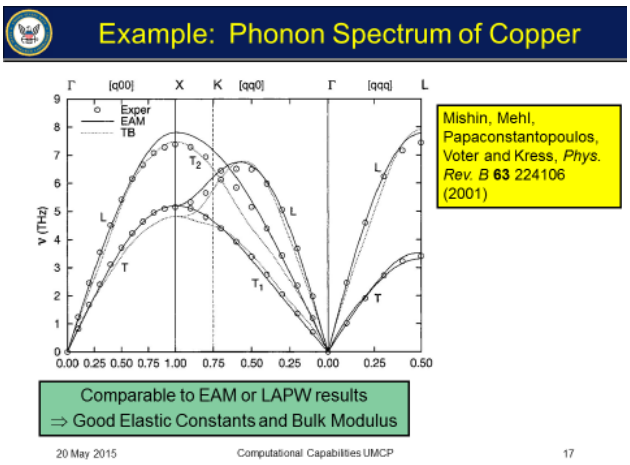
So what we have done is parameterize the tight binding scheme, so if you have two species you will need 300 parameters, or on that order.



Then one would go through the first principles calculations fitting the parameters, which Dimitris was good at, and I would then identify the bugs and we would repeat. Then from these parameters we can calculate the equations of state, magnetic moments if we have spin polarization tight binding, surfaces, and many more things which we can then publish.

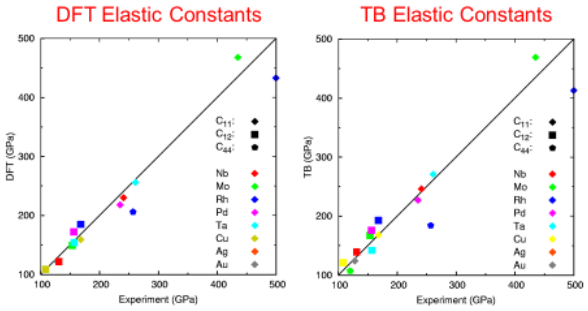


This slide indicates which elements we have successfully predicted material properties for, and actually the slide is a little old and we have added to it since. The nice thing is that for the transition metals if you feed our model a couple structures you can get most of the other structures, at least in the right order. For example manganese, which has a very complicated crystal structure, we fit the FCC and BCC crystal structures and we obtained a Manganese structure which has 29 atoms per unit cell and lo and behold we obtained the ground state, which was quite impressive. Then we were able to show that the two elements directly below Manganese, Technetium and Rhenium, are closely related to Manganese and have the Manganese low lying structure, which actually comes out of first principles calculations.



In addition one can calculate phonon spectra.

TB Elastic Constants

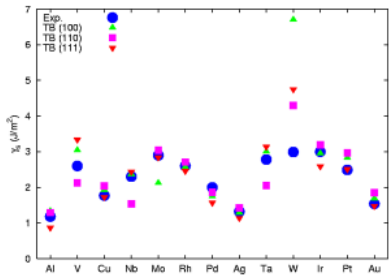


FCC and BCC Transition Elements

Cohen, Mehl, and Papaconstantopoulos, *Phys. Rev.* **50**, 14694 (1994)

We can also calculate elastic constants where on the slide you can see that both our DFT and tight binding predictions, for cubic materials, match experiments quite well.

Beyond 1 atom unit cells: Surface Energies

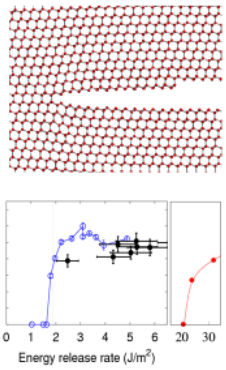


- Computed from 20-25 layer unit cells
- Accuracy better than Embedded Atom Potentials

You can calculate surface energies. Here on the slide we are calculating surface energies for each face orientation and we can see that our results closely match experiments for which you generally can only obtain a single data point. Furthermore, we did not fit anything on surfaces for those.

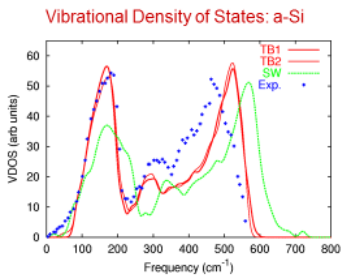
Fracture in Silicon

- Atomistic potentials make Si ductile (red line) -- must include QM bonds
- "Combination of Length Scales" -- (non-NRL) TB in region around crack, EAM elsewhere
- Experiment (black) from Hauch *et. al*, *Phys. Rev. Lett.* **82**, 3823 (1999)
- Vertical Line is Griffith criterion for brittle fracture (based on energy conservation)
- From Bernstein and Hess, *Phys. Rev. Lett.* **91**, 025501 (2003)



In addition we can do fracture in silicon, which is possible via first principles calculations but often takes a large amount of time. But less expensive methods using atomistic potentials will not capture the bond breaking. In the study shown on this slide we are using tight binding near the crack location and atomistic potentials away from it. Using this method you can obtain the correct rate of energy release. This study was conducted by Noam Bernstein roughly ten years ago.

Phonons in Amorphous Silicon

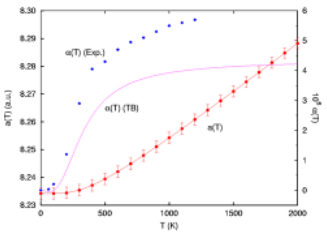


- Used TBMD code to compute forces on 216 atom amorphous silicon unit cell
- Direct calculation of the dynamical matrix
- Much better agreement with experiment than using Stillinger Weber
- Feldman *et al.*, *Phys. Rev. B* **70**, 165201 (2004)
- Now working on 1024 atom cell

20 May 2015 Computational Capabilities UMCP 21

We can also look at amorphous systems, where recall my comment on interpolation on slide 13, however in amorphous silicon 99% of the bonds are tetrahedral, they are just bonded to the next atom at different angles. So we get a very good agreement between our results and experimental results for vibrational states and density of states.

Thermodynamics of SiC



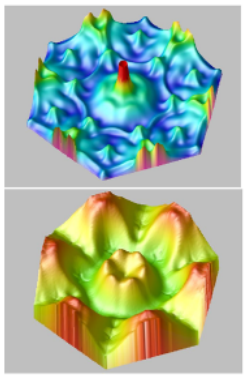
- Compute phonon frequencies as a function of volume in irreducible Brillouin zone
- Determine thermal expansion via quasiharmonic approximation
- Good low temperature agreement with experiment [Karch *et al.*, PRB **50**, 17054 (1994)]

20 May 2015 Computational Capabilities UMCP 22

We can calculate thermodynamic properties such as thermal expansion, where our results deviate at high temperatures but at low temperature are in good agreement because we can obtain a full quantum mechanical quasi-harmonic calculation.

Tight-Binding for Superconductivity

- Odd-gap Superconductor $Na_xCoO_2{\cdot}yH_2O$
- Fit TB parameters for CoO_2
  - Na_x only used as source of electrons
  - H_2O only used to maintain lattice spacing
- Determine susceptibility from one-electron TB spectrum
- Double-humped peaks on zone boundary indicates nesting
- Johannes, *et al.*, *Europhysics Letters* **68**, 433 (2004)



$lim_{\omega \to 0} \chi_0''(\mathbf{q},\omega)/\omega$

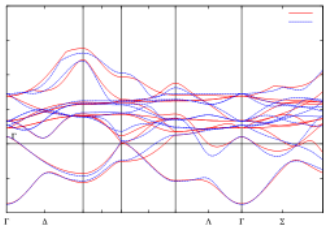
$\chi'_0(\mathbf{q},0)$

20 May 2015 Computational Capabilities UMCP 23

If you are willing to do some work and specialize to a very complicated system, such as the one shown on the slide, once you fit your parameters you can get a very good look at the shape of the Fermi surface and susceptibility. To do the fitting we required a large number of k points. So in reality it was easier to fit parameters and then we could query as many k points as we wanted.

Can even add f-states

- Hamiltonian size is now 16Nx16N, but still much smaller than the first-principles matrices
- Note that we've fit band structure, but not the energy versus volume curve

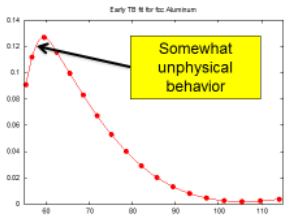


20 May 2015 Computational Capabilities UMCP 24

We have also been working on adding f-bands although the graphic displayed is not a very good picture.

Problems and Pitfalls

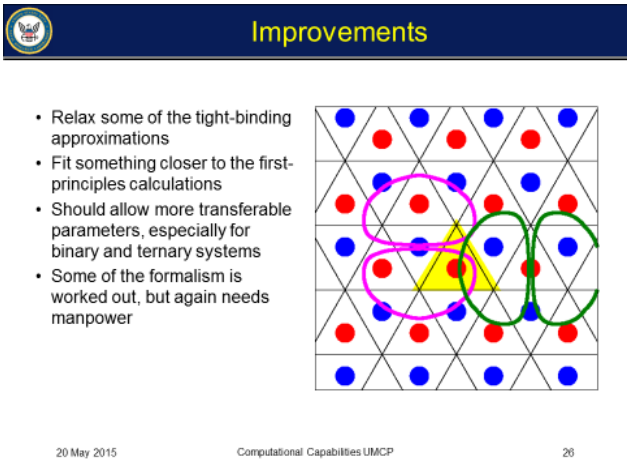
- Technical problems: it is easy to generate an unphysical fit, difficult to detect it. Also difficult to generate transferable parameters.
  - Cu parameters in Cu-Au should be the same as in Cu-Cl.
- Distribution: Although codes are approved for public distribution, web pages do not meet NRL security standards
- Problems can be fixed, but need manpower



20 May 2015 Computational Capabilities UMCP 25

Now, there are some problems. As I mentioned this is an art not a science, consider the displayed graphic which was an early fit for aluminum and you get a nice equilibrium solution. As pressure is added we expect the curve to diverge to infinity, however at some point the parameters fail and we see unphysical behavior. This is a result of one set of parameters not being in the same form as another set and they therefore do not cancel each other out leading to a decrease in the curve in the limit, again displaying these parameters is an art and not a science.

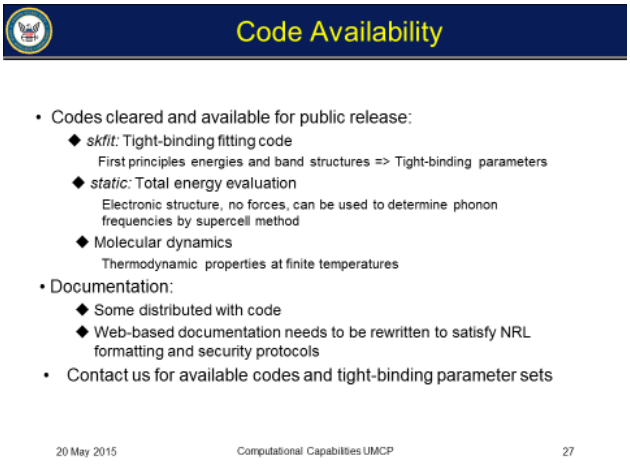
These codes have been approved for public distribution. However, to give you an idea of one of the problems we face, I submitted this talk for security office approval April 17th, and I got the approval for this talk May 19th. So everything we publish has to go through this massive security apparatus. We used to have a nice set of web pages with useful technical information but when we changed the security system at NRL they said we had to redo those web pages, meaning we had to obtain the security approval number for a paper we published in 1996, which is out in the literature and has been since 1996. The reason being that there might be PII (personally identifiable information) issues. To which we respond all those people are still alive you could merely ask them … or they're dead and no longer care. So those are some, among many, of the bureaucratic issues.



We would like to proceed with some improvements to this code, which requires manpower. We can relax some for the tight binding approximations, such as including some of the terms we had previously ignored when considering the interactions of two atom centered wavefunctions and potentials (slides 10-11). In fact my coauthor, Dimitris Papaconstantopoulos, has recently published another book for how to calculate these interactions for specific cases. His previous book had gone out of print and is only available on amazon for several hundred dollars so he decided it was time to publish a new book.

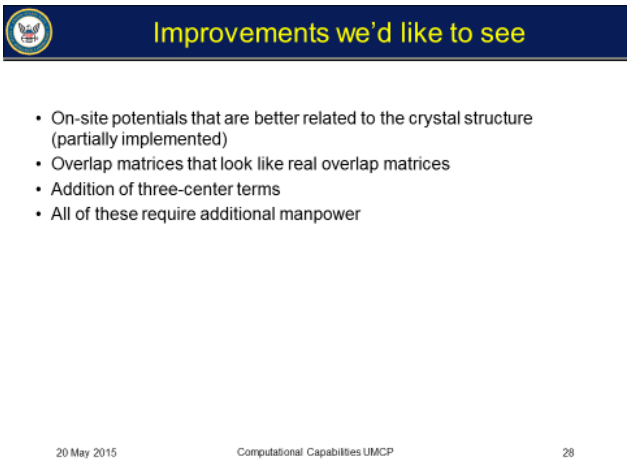
Relaxing these assumptions should allow for transferability. For example if you wanted to fit a ternary system like lithium, potassium, and oxygen for a battery calculation you would have to go through a multistep process. You'd have to determine your lithium parameters, your potassium parameters, and your oxygen parameters, then you'd have to determine your lithium-potassium parameters, your lithium-oxygen parameters, and your oxygen-potassium parameters. Then, because you are only considering two atomic sites you would be done, however your results would still not be very good yet.

So some of these issues can be remedied but this requires manpower.



Another aspect of the problem is how the code can be made available to the public. Originally when the CHSSI program was in operation we were able to release the code as it was government work, not subject to copyright and had been freely discussed in the literature. For a while we were able to distribute publicly via a website where all that was required was some personal information so that use of the code could be tracked. However, this eventually stopped as software codes in the 1990's were classified as munitions and were thus subject to export control. SO we eventually were able to get past that but we no longer could provide the codes via webpage. Rather, I receive an e-mail from someone requesting the code, about once or twice a month, and I send it to them. However, due to the lack of a feedback mechanism there is no further communication with the end user.

So some documentation was distributed with the code. The web based documentation is waiting for any industrious person to go out and track down all those references and get it through security. However, anyone can contact us for the codes and the parameter sets.



Improvements I would like to see is the implementation of onsite potentials that are better related to the crystal structure, this would alleviate the problem we observe of the parameter set failing at higher temperatures (recall the curve dipping back down, unphysically, in slide 25). In addition we need to improve the overlap matrices (which relate how all wavefunctions interact with all other wavefunctions), which in principle we should have but right now we do not because of the way we are fitting, and this can lead to problems at times. Finally I would also like to improve the three center terms I alluded to earlier.

All of these things require additional manpower, and this is one of my problems. I work at NRL which the Navy calls the "Navy's Corporate Laboratory", this is meant to conjure images of Bell Labs in the old days. What the Navy thinks of it as is a warfare center, where there are two types of naval labs NRL and the warfare centers, which has implications for security information, getting people access to the laboratory, and funding. As a result this puts restrictions on the number of post-docs I can hire. I certainly can't go out and get graduate students, Dimitris can but their foreign national status severely restricts their access to the NRL facility. SO all of these factors present significant challenges.

In addition getting funding from outside agencies is extremely difficult for NRL. NSF is a non-starter completely, with DOE it depends on the project, NASA possibly if they have something you're interested in, and ONR is a possibility but the program must fit in with what their project managers want to do. So we do get some funding from NRL but the issue is it is allocated in five year chunks and they generally want your proposed project to be completed within that span of time. If you require more time it becomes more difficult to receive funding. However, this situation arises in the first place due to the lack of infrastructure (alluding to Dr. Klimeck's online infrastructure for code distribution) to make the work publicly available which would have accelerated the work in turn. This lack of infrastructure is really one of the central disadvantages of doing this type of work in the government as it severely hampers progress. Despite all this people are, to my surprise, still interested in this work even after twenty years, and they still come to us asking about doing these types of simulations.

So I will stop here and take questions. Thank you.

Density Functional Theory and Related Modeling for Simulation of Wide Bandgap Semiconductors

Dr. Neil Goldsman
Department of Electrical and Computer Engineering, University of Maryland
College Park, MD

Abstract

Wide bandgap semiconductors represent a next generation in the science, engineering, design and fabrication of power electronic devices. However, these materials are difficult to process and therefore often contain numerous defects. To investigate these electrical defects at the atomic level, and how the material's electrical properties depend on its basic atomic structure, we apply Density Functional Theory (DFT) to SiC structures and their interfaces. DFT is a very powerful theoretical tool that, when used appropriately, can provide the band structure and density of states, as well as the effect of defects in the material on its electronic properties. DFT results can be used as input to Monte Carlo transport simulations to directly relate electron transport to atomic structure. In addition to using DFT to understand the atomic structures that give rise to defects, once we know these structures, we can further use DFT to investigate mechanisms, processes and annealing procedures that will eliminate these defects on the atomic level and thereby improve device characteristics. A brief summary of Density Functional Theory as well as its application to wide bandgap semiconductor materials and devices will be given.

Biography

Neil Goldsman is a professor in the Electrical and Computer Engineering Department at the University of Maryland at College Park where he directs Wide Bandgap Semiconductor Group and the Semiconductor Simulation Laboratory. His recent work has focused on high temperature electronics, UV detectors and semiconductor modeling. He has served as both the conference chair and the symposium chair of the International Semiconductor Device Research Symposium. For the International Conference on Simulation of Semiconductor Processes and Devices, Dr. Goldsman has served as the program chair, and regularly serves on the technical committee. His research has been sponsored by leading governmental and industrial organizations. Dr. Goldsman is the co-recipient of the IEEE Dasher Award; the NSF Research Initiation Award; the University of Maryland's George Corcoran, Invention the Year, and Business Plan Awards; and Cornell University's Post Foundation Scholarship. Goldsman has published more than two hundred peer-reviewed technical papers, and supervised the design of over fifty integrated circuits. He has authored two departmental educational texts in electronics. Dr. Goldsman received his Ph.D from Cornell University with a major in Electrical Engineering and a minor in Applied Physics.

(Slide 1)

Thank you. My name is Dev Ettisserry. I am a graduate student, working with Professor Neil Goldsman at the ECE Department of UMD. I will be explaining how DFT could be used in applications of concern to device physicist and electrical engineers.

(Slide 2)

Here is a brief overview of my talk. I begin with a brief introduction as to why we worry about silicon-carbide electronics. Then, I will discuss a specific case study describing how DFT could be used for high-temperature reliability issues in silicon-carbide MOSFETs. Finally, I will talk about specific details regarding the study of oxygen vacancy-related hole traps and how they impact device reliability.

(Slide 3)

So, why do we worry about silicon carbide? Since it is a wide band gap semiconductor, it can be used for high temperature and high power electronic applications. This is due to its wide band gap structure and high thermal conductivity, high breakdown field, and its ability to grow native oxide.

What are the concerns in silicon carbide MOSFETS? First, we have a poor quality of interface. It has many defects so it can lower channel mobility. Secondly, there is also poor reliability, especially threshold voltage instability. Our goal of research in the community has been to understand these mechanisms for this poor performance and to devise fabrication techniques to mitigate those defects.

(Slide 4)

The main problem is the reliability of the silicon carbon MOSFET, the critical concern being the high-temperature threshold voltage instability. Shown here are two graphs which measure the threshold voltage shift as a function of stress temperature, one at room temperature and one at high temperature. It can be seen that at high temperature, there is excessive aggravation of the threshold voltage shift when the device is stressed beyond 10^-4 seconds. The cause of this excessive aggravation is not known. One reason may be the activation of the original tracking centers.

(Slide 5)

The approach in our lab has been to start integrated modeling. Our idea is to use density functional theory to come up with various properties of material systems, and use it with conventional modeling techniques like rate equations or diffusion equations in order to arrive at a good MOSFET. Once we identify various mechanisms that limit performance and reliability of these MOSFETS, we would want to use DFT again to come up with sortable passivation processes using techniques like molecular dynamics. This is our research approach about which I will elaborate on in the next slides.

(Slide 6)

I will skip this slide because we have already had a previous talk that already covered this.

(Slide 7)

Now, let me give a brief introduction as to how we calculate energy levels of defects with DFT. We model this as a chemical reaction where you have an oxide band. This is an example to show how a defect is formed inside an oxide, silicon dioxide specifically. So, you have an oxide band and you add a defect. You charge the defect to get this as the reaction product, so you have a carbon defect, an interstitial defect. The chemical reaction can be written like this. The feasibility of this chemical reaction is calculated using formation energy. It is the energy of the product minus energy of the reactant with these energies calculated using density function theory. The stability of this defect in its various charge states is arrived at by writing formation level energy as a function of Fermi level. In the case of a MOS system, what really determines the electrical activity of a defect is the band gap alignment of the semiconductor-oxide. So, in short, those defects in the oxide could be active in the MOSFET whose switched charge states when the Fermi level is within the band gap of a substrate. So, this is a band gap alignment. This defect could be called as electrically active because as the Fermi levels sweeps the band gap in response to an applied bias, it could switch the charge state of the defect. It could affect the properties of the device.

(Slide 8)

We studied oxygen-related hole traps and 4H-SiC MOSFETs. Here is our result. I will explain this chart in a moment. The motivation behind studying oxygen vacancies was that recent years' experiments showed signals from E-prime centers. To begin with, we have an oxygen vacancy dimer structure here. Upon hole capture, this dimer converts into a positively charged dimer state. We have found other energetically favorable configurations into which these dimers transform. When the MOSFET is stressed under negative bias and temperature. We also calculated the activation values for all of these transformations using Nudged Elastic Band method, using DFT.

Now let us discuss the electrical activity of these defects. We studied the electrical activity of these defects using the method which I outlined in the previous slides. It turns out that the neutral dimer state, shown by the red line here, has its charge transition level falling outside the silicon carbon band gap. That really means that the neutral dimer isn't going to be active inside the silicon carbide MOSFET. However, as this configuration transforms into higher energy configurations under negative bias and temperature stress, these higher energy configurations tend to be electrically active because they have a charge transition which falls within the band gap. We arrived at the conclusion that under negative bias and temperature stressing, there are certain oxygen vacancies that are initially inactive, but the stress could activate them to form electrically active defects, and that could be the reason that we get the sharp increase in threshold voltage shift.

(Slide 9)

In order to further validate this theory, we did a transient modeling of oxygen vacancy hole trap activation. This is the same chart from the previous slide. I have shown all the activation barriers from DFT. Then, we modelled the formation and loss of each of these configurations using the simple Arrhenius model. We solved a couple of Arrhenius equations for each of these configurations using the values we obtained from DFT. In short, we calculated the concentration of positively charged defects over time. Then, we converted that accumulation of positive charge

into voltage, and we filtered the threshold voltage shift versus stress duration or time. We obtained fairly good agreement with the experimental result. This tunneling process was basically modelled using Shockley-Read Hall tunneling model. In conclusion, the model explains high temperature threshold voltage instability for silicon carbon MOSFETS. We accomplished that by unifying DFT and conventional device modeling. We have also worked on combining DFT with drift diffusion simulations to identify mobility limiting defects in silicon-carbide MOSFETs, but that's a different story. In short, oxygen vacancies are responsible for this high temperature effect that we see in silicon carbide MOSFETs.

(Slide 10)

How can we passivate these defects?

(Slide 11)

Shown here is one case study. We ran a molecular dynamic simulation. This is where the oxygen vacancy is located, in between these two silicon. We modeled an implantation of two fluorine molecules. Towards the end of the simulation, this seems to be a stable configuration. It is clear that fluorine basically passivates the dangling bonds of silicon vacancy. We also studied whether certain complexes were electrically active or electrically inactive using the same technique. We did not find any charge transition level within silicon carbide band gaps on these floating passivate structures. The conclusion of the case study is that fluorine is a good passivate for getting rid of oxygen vacancy effects and limitations in silicon carbide MOSFETs.

(Slide 12)

In summary, the general research approach in our group is to unify density functional theory with conventional device modeling techniques. We have tried to solve some practical problems encountered by devices, especially in the wide band gap industry. We attributed the high temperature reliability effects observed silicon carbide MOSFETs to the activation of switching electrically inactive oxygen vacancies to form electrically active defects under stress, over time. We also concluded that fluorine could be a good passivating agent for getting rid of oxygen vacancy-related effects.

LAMMPS and classical molecular dynamics for materials modeling

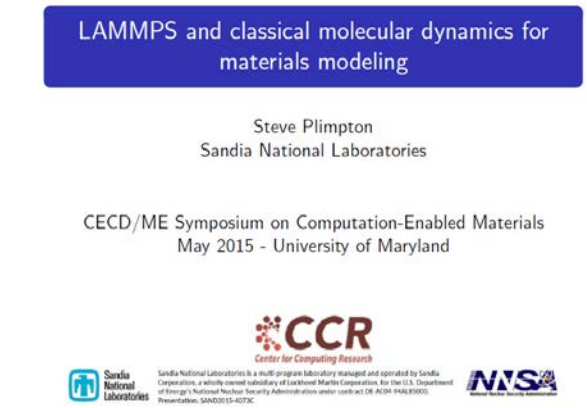
Dr. Steve Plimpton
Sandia National Labs
Albuquerque, NM

Abstract

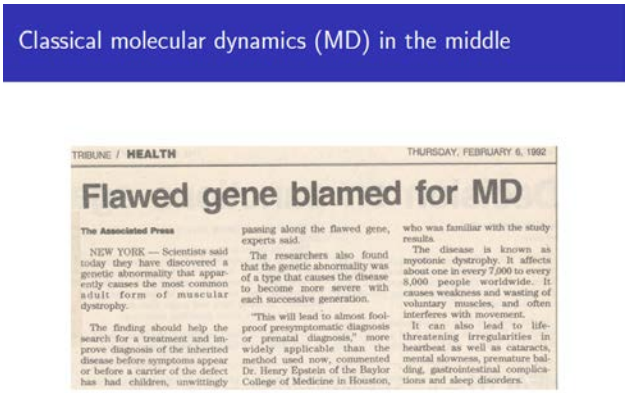
Classical molecular dynamics (MD) and LAMMPS occupy the middle ground in this symposium, between quantum and the meso or continuum scales. I will try to highlight some of the reasons MD as a method and LAMMPS as a software package have become popular tools for materials modeling on high-performance computing (HPC) platforms. I will also discuss areas of active research where many MD codes, including LAMMPS, are working to improve and extend. I will illustrate with some recent successes we have had in developing quantum-accurate potentials and coarse-graining to extend the length and time scales accessible to classical MD. I will also discuss a challenge all materials modeling codes are facing, to adapt to the changing hardware landscape in HPC, due to the end of Moore's Law.

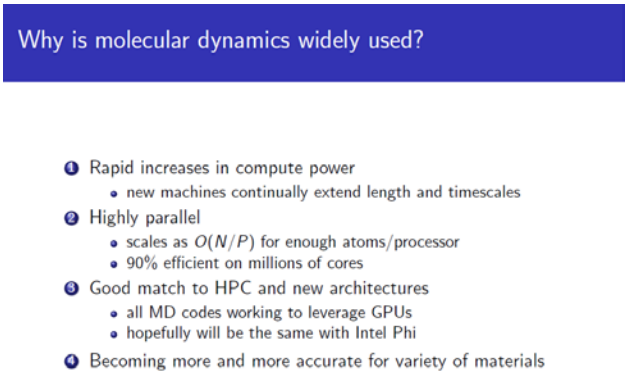
Biography

Steve Plimpton is a staff member at Sandia National Laboratories, in the Multiscale Science Dept of the Center for Computing Research. He received his Ph.D. from Cornell University in 1989 in Applied & Engineering Physics and has been at Sandia ever since. His work involves developing and using scientific simulations designed for large parallel machines, with an emphasis on creating efficient parallel algorithms. LAMMPS is the most well-known code he works on, but he is also a co-author of several open-source software packages which he supports, SPPARKS (kinetic Monte Carlo), SPARTA (Direct Simulation Monte Carlo), ChemCell (stochastic particle modeling of biological cells), and MR-MPI (MapReduce on top of MPI); see www.sandia.gov/~sjplimp for details.



All right, thanks Peter and other organizers for setting up this symposium and inviting me to come. I said in my abstract that I think classical molecular dynamics (MD) is kind of in the middle ground for the symposium between the quantum methods we heard about this morning and continuum scale methods on the schedule this afternoon. Sometimes that middle ground is kind of a no-man's-land. Quantum people think we are too empirical (I like the earlier joke about "no principles"), but the continuum people think that we are still too small scale and can't address the time scales they are really interested in. So thinking about that, plus the fact that I've come here to UMD to visit, I'll show a newspaper headline I saw many years ago (audience laughter). For those of you from Maryland, I apologize. The article is not about Maryland or molecular dynamics, it's actually about muscular dystrophy. But my quantum and continuum friends think this headline might explain why I choose to work in the middle ground.

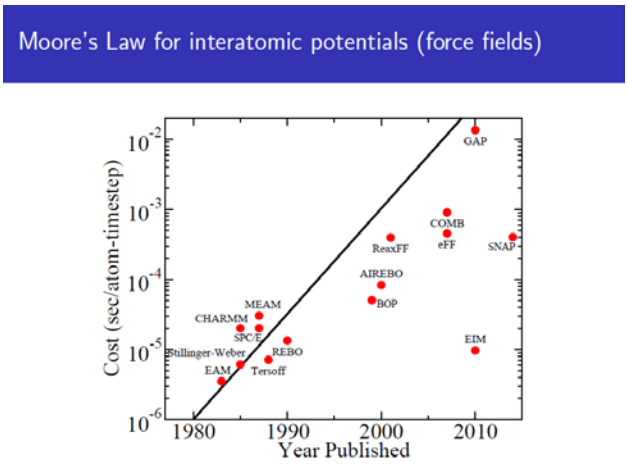




To address some of the themes for the workshop, I tried to first think about this question: why has molecular dynamics or classical MD been so popular over the last few years? And I've come up with four reasons. First, it's really a method that can take advantage of the increasing speed of computers that have increased so dramatically over the last 30 years, especially as parallel computing has become so popular. That means whenever an institution gets a new machine that's faster and better than the last one, you can typically do a bigger and longer simulation and do some new science, get a new paper. So people are always keen to do that. The second reason is that MD is a very parallel simulation method. The majority of the computation involves short-range forces, so that means it scales very nicely to big problems on big machines. So long as you have enough atoms per processor it scales linearly with N, the number of atoms, and inversely with P, the number of processors. So you can run big MD problems now on millions of cores and get 90% efficiency if your problem is big enough. The third reason

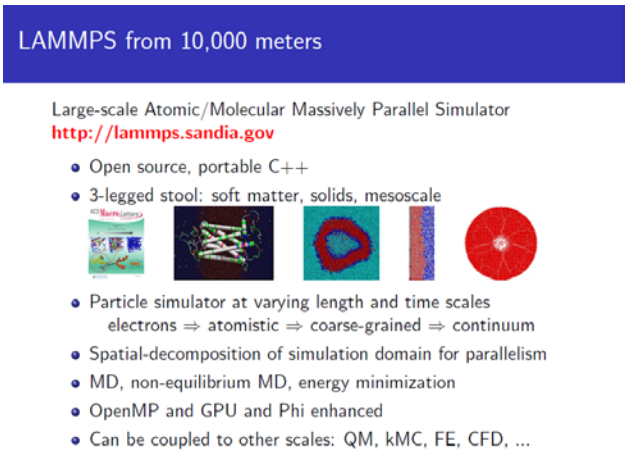
is that MD is good match to all the new chip architectures that are becoming common in High Performance Computing; all MD codes are trying to take advantage of them. For example, the kernels in MD codes are generally well suited to running on GPUs and hopefully soon on the Intel Phi. Finally, and perhaps most importantly in a scientific sense, is that the materials modeling in theory community has been very creative over the last few decades in developing better and better models, more and more accurate interatomic potentials to use with a variety of materials within the classical MD context.

I'll illustrate that last point with a movie analogy. Hopefully all of you have seen some of these movies, do you remember the company responsible for them? Pixar . So Pixar is obviously successful with all their movies because they tell good stories. But if you think in a CGI (Computer Generated Image) sense, they are also very good at developing new models and algorithms that allow them to model new materials in a convincing way. I've listed here new capabilities that at the time were cutting edge in terms of the kinds of things they could animate and model successfully. So they have been able to develop new methods and new models as time has gone by to leverage new computers. I saw a discussion where somebody asked some Pixar people if it took them, say 10 years ago, a certain amount of time to render the images in their movie, if they now make their movies a lot quicker with faster computers. And they said no, actually if it took one hour to render a single image in one of their movies 10 years ago, today, it still takes one hour to render each image. That's because they aren't using faster computers to model old models more quickly, they are using faster computers to develop more complex models and put more complexity in each scene to do things more accurately.

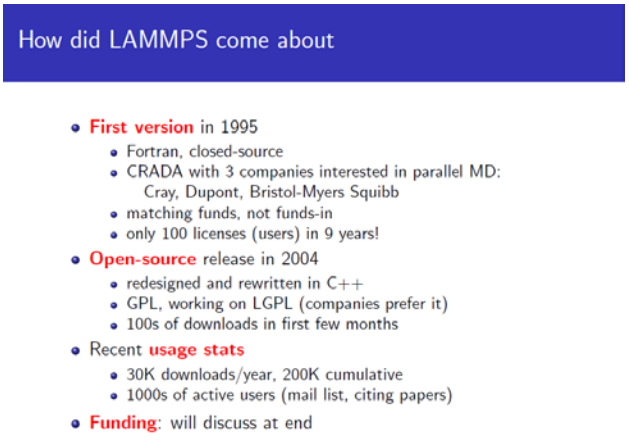


And I think there is an analog to that in classical MD, at least for materials modeling, in terms of what has happened over time. Here is some data to support that. Each point in the plot stands for an interatomic potential, or force field, suitable for a particular class of materials. The X axis is the year that model was originally published as a method. These are all implemented in our LAMMPS code so we can make a fair comparison of their relative costs. The Y axis is the cost per atom per timestep to do a simulation with that particular model. The solid line here is meant to be a Moore's Law kind of doubling in cost every two years. So you can see it does not quite track that, but there have been dramatic increases, several orders of magnitude,

in the complexity of the potentials people are using. Again, people are using faster computers to run simulations based on expensive force fields like ReaxFF, COMB and so on. Not only to just run simpler models like Stillinger-Weber or EAM more quickly, although some people do that as well. I'll note that potentials on the left side of the figure are simpler pair-wise or many-body potentials, while potentials like MEAM and REBO are more complex bond-order potentials, and models like COMB and GAP are many-body plus reactive potentials.

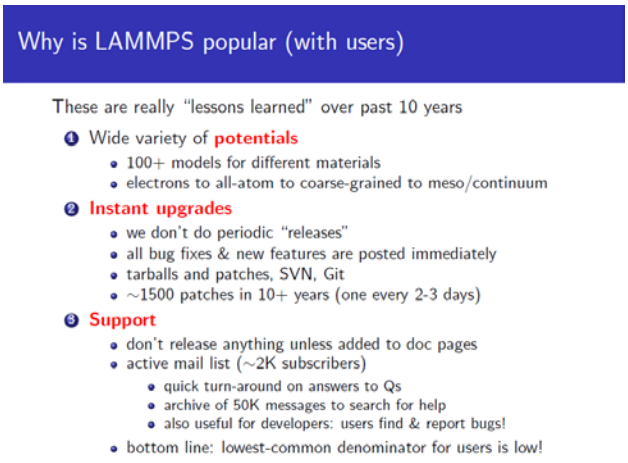


Ok, so let me say a few words about LAMMPS. This is a one slide overview. It's an open-source, C++ code. We've tried to make it have one foot in each of three camps for different classes of materials. Soft matter could be biomolecular systems, polymers or liquids. Solid state systems require different kinds of potentials with different features and options in a code to model correctly. And there are a variety of mesoscale to continuum models in the code, which still essentially a collection of interacting particles which represent materials at different scales. These models cross a wide range of length scales. There are models, which I think quantum people won't like, which model electrons as individual particles with a variable radius that is meant to capture the wave function. And there are all-atom models which people are most familiar with. The various coarse-grained methods and continuum methods go all the way to macroscopic length scales. For example, this image is from a peridynamics model which is like a meshless continuum method to model fracture of various kinds of systems. Only particles are used, no finite element mesh. Spatial decomposition of the simulation domain is what LAMMPS exploits for parallel execution. I'll come back to some of these final bullets later in the talk to describe things we've tried to do with LAMMPS.

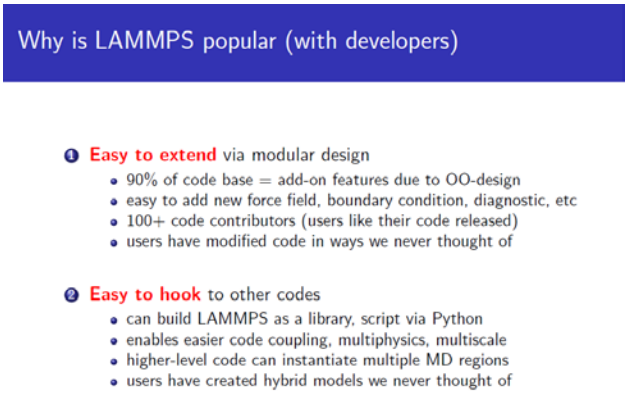


So I though I would say a little about how LAMMPS came about in the context of some lessons learned along the way. The first version was about 20 years ago, and it was actually at the time when DOE was eager to collaborate with industry. We had a CRADA, or cooperate research and development agreement, with three companies who were interested in developing a parallel MD code. So Sandia and LLNL received money from DOE to work on this and the industrial partners put up their own matching funds. That resulted in a closed-source, Fortran code. The companies did not mind if we gave it away for free but they wanted legal protections and so users had to sign some license forms if they wanted to use it. That ended up limiting us to about 100 users in 9 years. And that was because if people made the mistake of

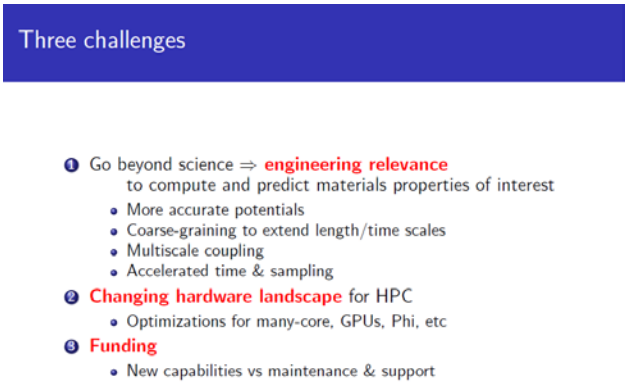
showing that license form to their lawyers, it would really slow down the process. So 10 years later, around 2004, we wanted to make the code more flexible and rewrote it in C++, so that we could do some new things with the code. At that point we got permission from the companies to release it open-source. So we did that and within the first couple of months afterward we had hundreds of downloads and users. So that was the first lesson we learned, Anny barrier, even if it is having to click and filling out a website form or signing some piece of paper, can actually be a big barrier if you really want to give the code out freely. The more open and easier you can make it for people to get their own copy, the better. We also made the decision to license the code as GPL. Now we wish we'd done it LGPL because companies prefer that if they are interested in developing and distributing their own proprietary software that works with or wraps a code like LAMMPS.



Ok, so let me talk about the lessons we have learned over the 10 years as an open-source code. It will sound like we knew all these things advance or were pretty smart 10 years ago, but these are really things we learned along the way, sometimes by hard experience and so we've adapted the code as we went along to meet these standards and goals. I will divide this into two categories. First is what people like about LAMMPS from a user perspective and then from a developer perspective. These are the three items for users. First is that we have tried to provide wide range of potentials so that people can learn our code and how to do things in LAMMPS and can then apply what they've learned to lots of research areas and research topics. We have on the order of 100+ models for different materials, which includes the models at various length scales I talked about earlier. A second idea is that we do not do versions or periodic releases, we simply do instant upgrades and put the latest version on the web. That means whenever we fix a bug or finish a new feature, we just post it immediately. So from our perspective, the current version is the only thing we think about. A user can download tarballs and apply patches, or use SVN or Git and keep up with the code. I did a little count on the patch pages, in the last 10 years, we have released about 1500 bugs fixes or new features in this manner, which is about one patch every 2 or 3 days. I think users like not having to wait for periodic releases. The last idea is that people do like some level of support to make their user experience more pleasant. Part of that is the documentation; we don't release anything until it has been documented, so there is no lag between what is in the code and what is in the documentation. We also have an active mail list where we try to provide quick answers to questions, and that results in an archive of Qs and As that users can search to get help.

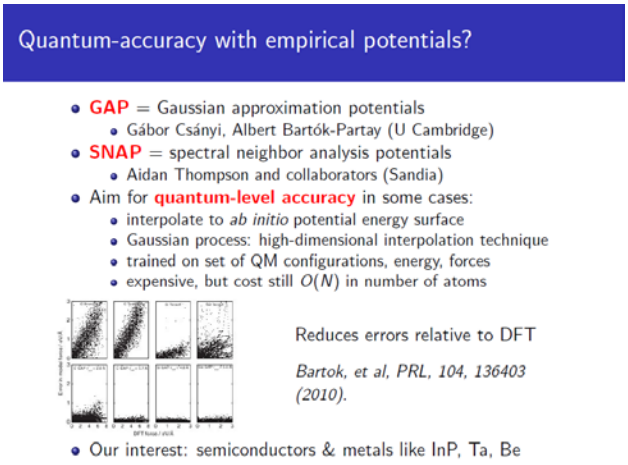


Now let's switch to the developer perspective. Two ideas here. The first is that over time, we've tried to make our code as easy to extend as possible because there are always people, especially with classical MD that want to do things your code does not do. So for people who know programming we try to make it easy for them to adapt the code. That's done through object orientation in C++. Basically if you write a new derived class that implements the new feature, you just put the couple of files into the source directory and re-compile. About 90%-95% of the code base is these kind of add-on features, every force field, every boundary condition, every diagnostic computation is an add-on. This has allowed creative modelers to modify the code in ways we never thought of. The second idea is that we have tried to make it as easy as possible to hook LAMMPS to other codes in a multiscale or multiphysics sense. The way we do that is allowing LAMMPS to be built as a library with a C-style interface that makes it easy to call from other languages or to script from Python. A higher level code, like a continuum code, can thus instantiate different regions of MD within its simulation domain and run multiple LAMMPS simulations if it wants to. And again, users have leveraged this idea to create hybrid models which we never thought about.

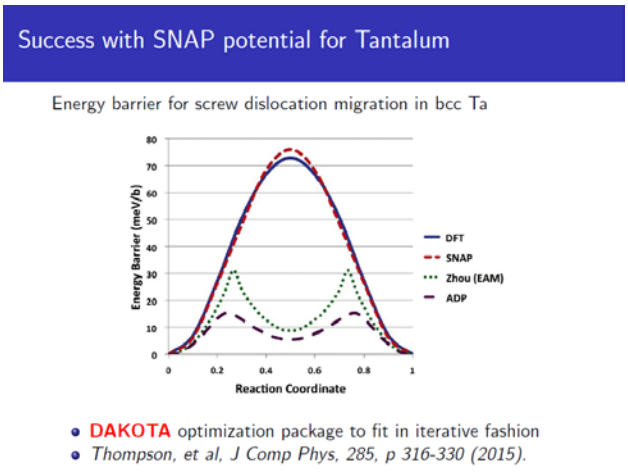


Now let me address three challenges. Most of these issues apply more broadly than to just classical MD and LAMMPS. The first is a technical or science challenge, and that is the goal of going beyond just doing a new science problem and writing a paper to being able to perform calculations that really have an impact in what I'm calling an engineering relevance sense. Meaning you can actually compute and predict properties of materials at the scale engineers care about, for example trying to design some new material with some new properties. So there are at least four aspects underneath that, where I think, not only LAMMPS, but all MD codes are trying to extend their capabilities. One is to have more accurate potentials. Second is coarse-graining to get you to higher length and time scales, trying to approach the scales that engineers care about. Third is simulating in a multiscale context where MD is just one part of a bigger work flow. I will talk about these three points with a couple of slides, with examples of things we are working on. I won't talk about the fourth idea, accelerated time and sampling, though its also an important idea that MD codes are pursuing. The second point is what Robert talked about this morning, how you take an established code like LAMMPS and try to adapt it to the changing hardware landscape that is happening in HPC because of Moore's Law. I will talk at the end about some techniques and libraries we're trying to use to do those optimizations in as painless

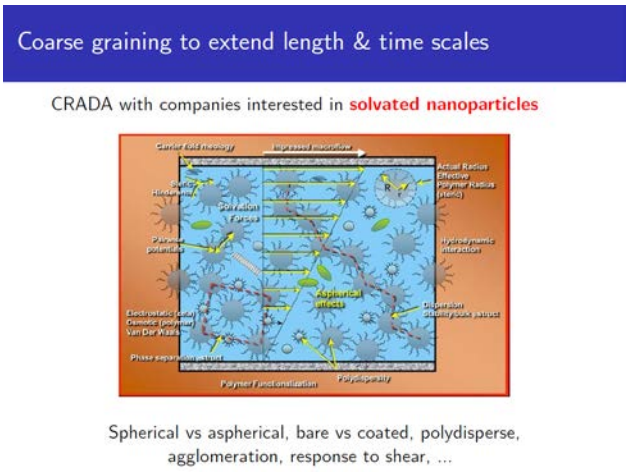
a way as possible. The third point is about funding challenges, which are basically the tension between trying to get funding for new capabilities versus maintenance and support for an existing code.



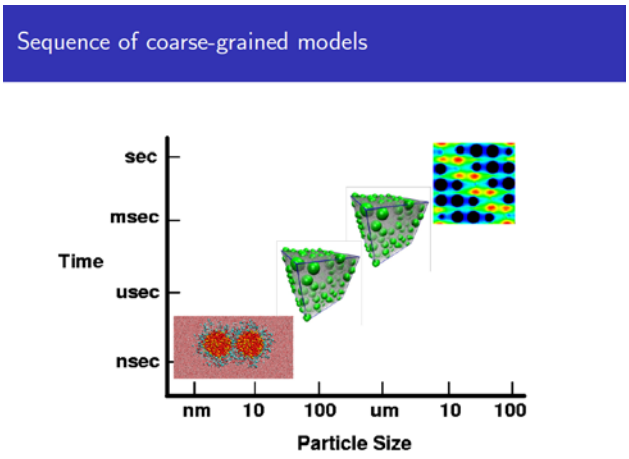
This is the first technical challenge, which is can you use empirical potentials to get near quantum-accuracy in an MD simulation at least in some context. There have been couple of recent developments on this which I think started with the Gaussian Approximation Potentials (GAP) from University of Cambridge a few years ago. And my colleague, Aidan Thompson from Sandia has done some recent work to improve the computational speed and accuracy of the method with potentials he calls SNAP for Spectral Neighbor Analysis Potential. In both cases, the idea is to use a database of pre-calculated DFT results. So for a bunch of DFT simulations, we have different conformations of small groups of atoms, where the total energy and forces on each atom are known. You put that information in a database and then when you run your MD simulation, for each atom, you take its local environment or neighboring atoms, and you lookup that conformation in the database. Since the exact conformation isn't there, you need to interpolate the force on the atoms from nearby conformations. That requires a high-dimensional interpolation technique and an ability to represent these conformations of an atom with its neighbors in a way that allows you to find nearby conformations. All of this is relatively expensive compared to simpler empirical potentials, but still much, much cheaper than DFT calculations. An important point is the computational cost is still linear in the number of atoms, so that you can actually use this kind of potential for systems with millions of atoms. The figures shown here are from the original GAP paper. The Y axis is the relative errors of forces in a MD simulation compared to the "gold" standard DFT calculation. The X axis is the magnitude of force on each atom. The figures on the first row are MD simulations using some common empirical potentials like Tersoff and REBO for carbon and silicon. You can see that the errors relative to DFT are large, and grow larger with the magnitude of the force. The lower plots are for their quantum-fitted GAP potentials for the same elements, and now the errors versus quantum forces are small and don't grow with the magnitude of the force. So for these systems, you can think of this as a quantum-accurate potentials.



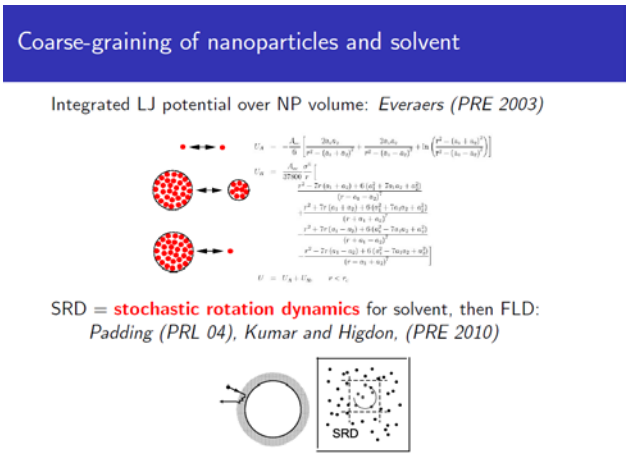
Here is a recent success for Aidan's SNAP potential formulation for Ta. This is a plot of the energy barrier for screw dislocation migration in bcc Ta, which is a key mechanism in plastic deformation of the material. You can see that a couple of commonly used empirical potentials like EAM and ADP are very poor at capturing that barrier, in fact the top of the barrier is actually a local minimum. But SNAP is able, for the first time for an empirical potential to get DFT levels of accuracy for the transition of a dislocation over that barrier. The paper listed at the bottom has more details. The fitting process with SNAP uses an optimization package developed at Sandia, which is called DAKOTA. It's good at taking the quantum database, and performing a fit to get a SNAP potential that can be used in LAMMPS. Then an MD simulation is run and you take MD snapshots and see where the errors are large compared to DFT. Then you can use that info to setup and perform more DFT simulation to add conformations to your database. So you can iterate on that procedure to get a more and more accurate SNAP potentials, that covers a broader range of atomic conformational space.



Now let me talk about the second technical challenge, coarse-graining to extend length and time scales. We recently had a CRADA, or cooperative research and development agreement with 3 companies interested in processing solvated nanoparticles. The idea is you take nanoparticles and put them into a fluid and then coat them over a surface and then evaporate the solvent to leave particles in some ordered or self-assembled state of interests. This cartoon represents what we're trying to simulate for the different kinds of nanoparticles of interest. It's really a zoo of particles with different sizes, different shapes, they can be coated with short polymer chains to inhibit aggregation. We want to be able to measure diffusivities and shear viscosities of that mixture fluid as it coats over a surface, and at the end the evaporation process as well, where solvent is removed.

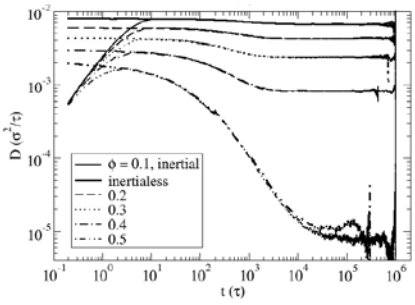


So in a modeling sense, that means is that you want to go from atomic scale to meso or even continuum scales, as shown in this diagram. At the atomic scale, in an all-atom MD simulation, we can only model a couple or few nanoparticles. But that allows you to parameterize forces between particles as a function of separation distance. You can use that information to develop coarse-grained models for both the nanoparticles and the solvent, which I'll highlight on the next slide. And you can work your way up to systems where the particle size is 10s or 100s of microns, really a colloidal scale, where now the solvent can be represented implicitly or as in a CFD sense as a continuum fluid. So that is a span of many orders of magnitude in length and time scales.



I won't talk about the details here, but these are the kind of ideas we try to exploit in an MD code to formulate a coarse-grained model. If you want to treat a big nanoparticle as a single particle, which is effectively a collection of smaller particles, you can do a pre-computation step to essentially sum the pairwise interactions over all the small particles, integrating over the two volumes of the big particles. That gives you an analytic expression for the collective interaction between two big single particles. Its a complex formula but it's still cheap to evaluate in an MD code. Likewise you can coarse-grain in various manners. The SRD model shown here ignores the interactions between solvent particles and just moves them ballistically. They bounce off the large nanoparticles and impart force and torque to them. The background solvent can be given an effective viscosity by performing a rotation operation shown in the diagram on the right. The solvent particles are binned, the net velocity of particles in each bin is subtracted out, and the remaining velocities are rotated in a random manner before the net velocity is added back in. This kind of model turns out to orders of magnitude cheaper than computing all the nano particle-solvent and solvent-solvent interactions.

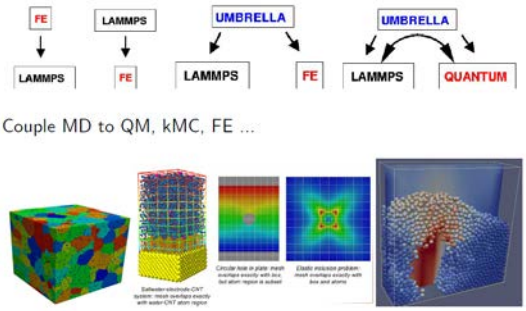
Diffusion across time scales and volume fractions



Bolintineanu, et al, Comp Particle Mechanics, 1, 321-356 (2014).

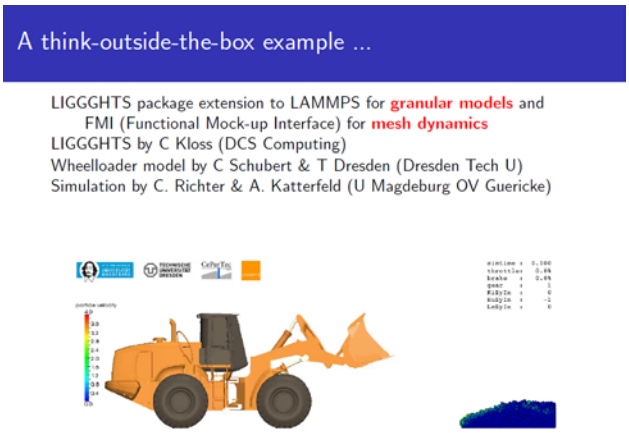
Here's an example of how this approach can be effective. This figure plots the diffusion rate of nanoparticles in solvent across a cascade of length scales with different methods. The different curves are for different volume fractions of nanoparticles from dilute to pretty dense. Experimentally, they can only measure this plateau region on the right side of each curve, where nanoparticles escape the cage of their near neighbors and move over long distances. We were able to compare this plot to an experimental system for 1 micron polystyrene spheres in water, where the diffusive time scale of interest is on the order of seconds. The x-axis here is in dimensionless time, but at the right-hand side it corresponds to seconds for this system. So by coarse-graining we were able to have the MD code model micron-size particles for second of real time, and get diffusivities that matches experiment reasonably well.

Multiscale & multiphysics via coupling to other codes

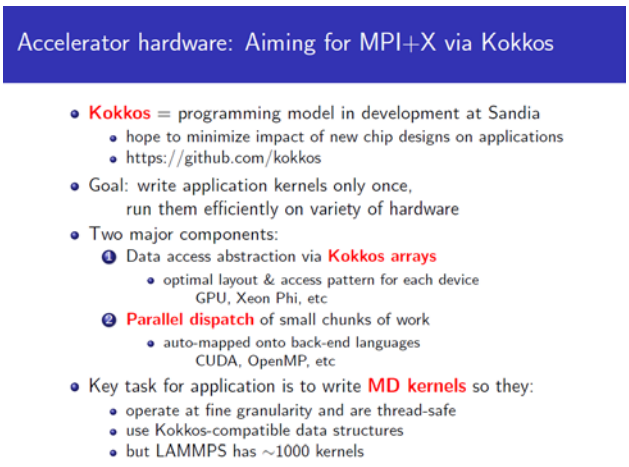


Couple MD to QM, kMC, FE ...

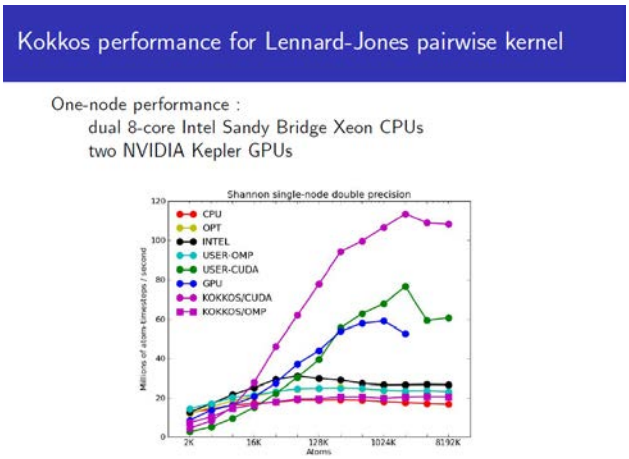
The last technical challenge was running multiscale and multiphysics models by coupling MD to other codes. The diagrams at the top show how that can be done in different ways in a software sense. As I said, we enable use of LAMMPS as a library to make this possible. So you may have a finite element code calling the MD code or vice versa. You may write some umbrella code or Python script over the top that calls the two codes one after the other. The pictures are some examples of users doing this to hook to kinetic Monte Carlo modeling of green growth, or to various finite element calculations for heat transfer, momentum transfer in the context and stress-strain deformation calculations. And a air jet mixing simulation a LAMMPS model of granular particles was coupled with a CFD solver.



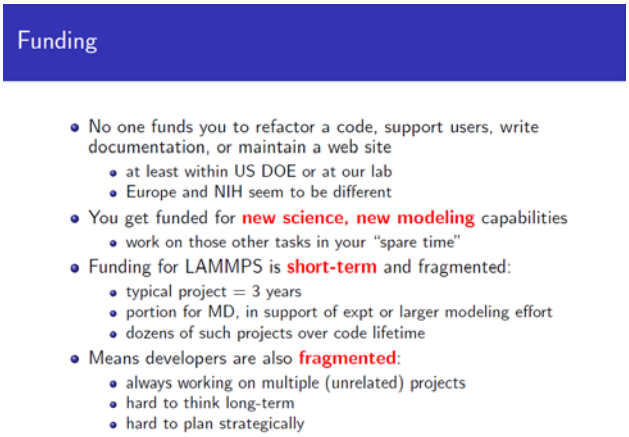
And here is the fun movie I promised, an example of somebody using coupling to do something outside-the-box. They use LAMMPS with extensions in this LIGGGHTS package for granular interactions, to model this pile of rocks interacting with each other. And they have a Functional Mock-up Interface (FMI) for mesh dynamics to model the little bulldozer. So they can model it as shown in this movie. The particles are colored by their kinetic energy. So I do not really know what these results mean (audience laughter), but it is definitely an engineering scale simulation so we're happy LAMMPS can be used for something that is truly at the macroscale.



Ok, so let me say something finally about the challenge of preparing codes like LAMMPS to take advantage of new kinds of accelerated hardware. What we are trying to do is to leverage a software tool developed at Sandia not just for MD but for a variety of other codes as well, which is called Kokkos. It's really a programming model that is trying to minimize impact of new chip designs on existing codes. The goal is that you can write your application kernels once, and the Kokkos interface will convert that code to a format that runs natively on GPUs or via OpenMP on multi-core chips, or on the Intel Phi. So it's trying to insulate the application code from the different kinds of accelerator hardware, including future hardware designs that might come along. Kokkos does that in two ways. The first is that it defines multidimensional Kokkos arrays which the application uses in the same way, no matter what accelerator hardware is being targeted. On the back end, those arrays may be laid out in different ways for different hardware. Say take a 2D array, it might work optimally in column order on GPUs but in row order on a many-core Phi which will optimize the performance of accessing the array on different devices. The second idea, is that you need to break up your code into small chunks of work that can be parallelized at a fine-grain level. So Kokkos provides a parallel dispatch syntax where you define the chunks of work and then Kokkos maps them onto back-end languages like CUDA, OpenMPI and so forth on to make them run well on different kinds of hardware. So that all sounds great, but from the application perspective, like LAMMPS, it means have to rewrite our MD kernels in this Kokkos style. That means we have to use Kokkos-compatible data structures, we have to identify and isolate the fine granularity of parallelism. And unfortunately for a code the size of LAMMPS, we have on the order of a 1000 kernels, so its not a quick or simple task.



I'll show you just one plot of what this can provide for the pairwise Lennard-Jones kernel which is part of many force fields. These are performance curves on a single node with 16 CPUs and 2 GPUs for performance measured in millions of atom moves per timestep per CPU second as a function of the number of atoms simulated. The lower band of curves are using all 16 CPUs, and essentially have perfect scalability. The flat lines are performance independent of the number of atoms as the model size increases. And you can see Kokkos is giving performance competitive with MPI and Intel optimized code. These three higher curves used the GPUs. We had previous packages in LAMMPS that could use GPUs in different ways. You can see Kokkos actually outperformance those, at least for large problems. The overall speed-up over all the CPU cores is about 3 or 4x for the larger problem sizes. Again, the real benefit here is that you can write the same code in the application and compile either for the GPUs or CPUs, you get the same kernels running on both efficiently.



So this is my last slide, about funding issues. One observation is that nobody will fund you to refactor a code, maintain a website, or support users. I think that has been true for us at least within DOE or Sandia. But Europe seems different, they seem to provide more basic support for codes over longer periods of time. And I think the biomolecular MD field, as opposed to materials, is kind of like that as well which provides longer-term support for institutions to maintain a code. What we do get funded for is new science and new modeling methodologies. So that means we can kind of work on other code tasks in our "spare time" once we're funded. It also means that LAMMPS funding is often short-term and fragmented. A typical project might run 3 years, and often we get funded in partnership with other people where there is a portion for MD work. That can be in support of experiments or some other larger modeling strategies. So we've had dozens of such projects over the code lifetime. That means from a developer standing point, we can also feel fragmented. We are often working on two or three different projects, some of which may or may not be directly related to MD or to LAMMPS. That makes it a little hard for us to think about the long-term or plan strategically.

Thanks and Links

- LAMMPS: **http://lammps.sandia.gov**
- **Funding**:
  - DOE (BES,BER), Sandia (ASC,LDRD)
  - NINE (university), CRADA with Corning, 3M, BASF
- **Joint work** with:
  - LAMMPS: Aidan Thompson, Paul Crozier, Stan Moore, Ray Shan, Axel Kohlmeyer (Temple U)
  - Kokkos: Carter Edwards & Christian Trott
- Two papers with more info:
  - *S. J. Plimpton and A. P. Thompson, "Computational Aspects of Many-body Potentials", MRS Bulletin, 37, 513-521 (2012).*
  - *S. J. Plimpton and J. D. Gale, "Developing community codes for materials modeling", Current Opinion in Solid State and Materials Science, 17, 271276 (2013).*

Ok, so I'll close with some thanks. These are places we've gotten funding for LAMMPS over the years. And these are LAMMPS developers I work with and have presented some of their work here. The first paper is about many-body potentials in LAMMPS for materials modeling. I also wrote a paper with Julian Gale discussing some community codes and the pros and cons of being part of a community effort. That paper covers some of the ideas I presented today in more detail.

Computation-enabled Materials Discovery: Addressing Grand Challenges in Energy Storage and Materials Design

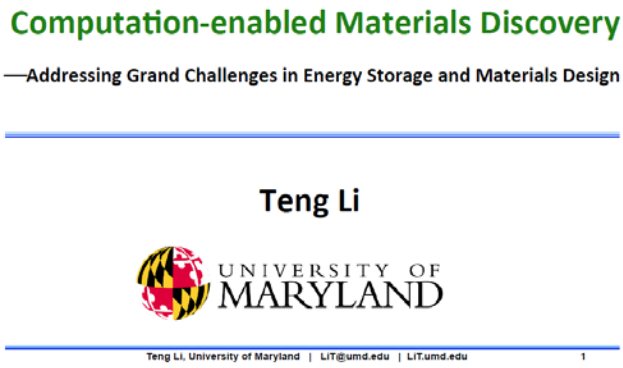
Dr. Teng Li
Department of Mechanical Engineering, University of Maryland
College Park, MD

Abstract

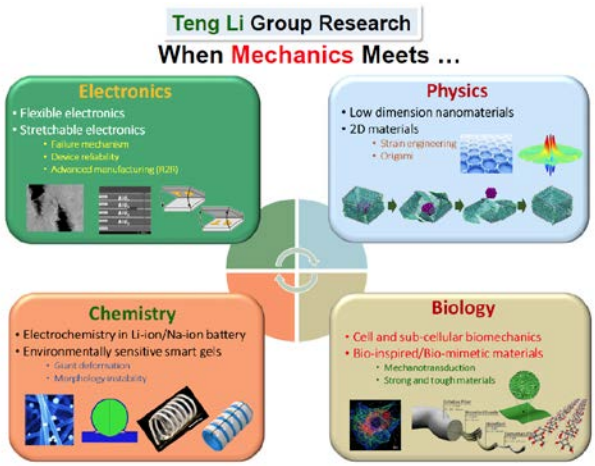
In this talk, I plan to demonstrate two case studies of computation-enabled materials discovery using molecular dynamics modeling. The first case is graphene origami enabled high density hydrogen storage. The malleable nature of atomically thin graphene makes it a potential candidate material for nanoscale origami. Enthusiasm aside, the success of graphene origami hinges upon precise and facile control of graphene morphology, which still remains as a significant challenge. Inspired by recent progresses on functionalization and patterning of graphene, we demonstrate hydrogenation assisted graphene origami (HAGO), a feasible and robust approach to enabling the formation of unconventional carbon nanostructures. For example, we show controllable and reversible opening and closing of HAGO-enabled graphene nanocage, a mechanism that is crucial to achieve molecular mass uptake, storage and release. We further demonstrate HAGO-enabled high-density hydrogen storage with a weighted percentage of 9.7%, exceeding the US Department of Energy target of 5.5% for the year 2017 and the ultimate goal of 7.5%.
The second case is designing materials that are both strong and tough using wood fibers. The quest for both strength and toughness is perpetual in advanced material design; unfortunately, these two mechanical properties are generally mutually exclusive. So far there exists only limited success of attaining both strength and toughness, which often needs material-specific, complicated or expensive synthesis processes and thus can hardly be applicable to other materials. A general mechanism to address the conflict between strength and toughness still remains elusive. Here we report a first-of-its-kind study of the dependence of strength and toughness of cellulose nanopaper on the size of the constituent cellulose fibers. Surprisingly, we find that both the strength and toughness of cellulose nanopaper increase simultaneously (40 & 130 times, respectively) as the size of the constituent cellulose fibers decreases (from a diameter of 27 microns to 10 nanometers), revealing an anomalous but highly desirable scaling law of the mechanical properties of cellulose nanopaper: The smaller, the stronger AND the tougher.

Biography

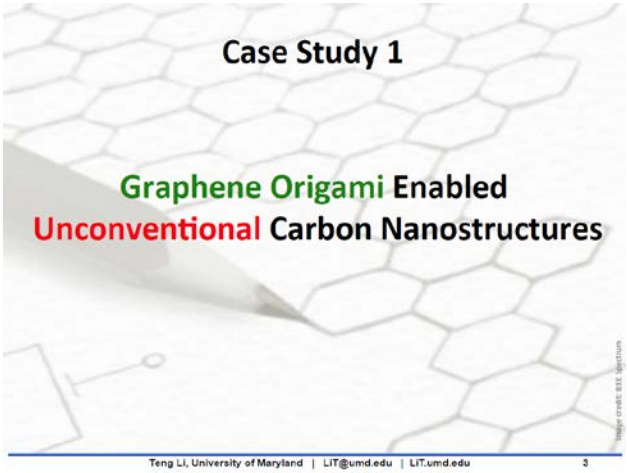
Teng Li received his Ph.D. degree in Engineering Science from Harvard University in 2006 (advised by Zhigang Suo), after earlier study in Princeton University (advised by Anthony G. Evans) and Tsinghua University (advised by Wei Yang). He is currently an Associate Professor of Mechanical Engineering and the Keystone Professor in the Clark School of Engineering at UMD. He is also an affiliated faculty of Maryland NanoCenter and University of Maryland Energy Research Center. His research interests include mechanics of flexible electronics and nanoelectronics, mechanics of low dimensional carbon nanomaterials, and mechanics of energy systems. Among his awards includes US National Committee of Theoretical and Applied Mechanics Fellowship in 2012, E. Robert Kent Outstanding Teaching Award in 2012, University of Maryland GRB Research Award in 2009 and RASA Research Award in 2014, Ralph E. Powe Jr. Faculty Award in 2007. He has been a member of the Technical Committee of Integrated Structures in ASME Applied Mechanics Division since 2006 and served as the Chair of the Committee during 2008-2012. He currently serves as an Associate Editor of *Extreme Mechanics Letters* and a member of the Editorial Board of *International Journal of Computational Materials Science and Engineering*. He is the co-founder (with Zhigang Suo) of iMechanica.org, the world's largest online community of mechanics with ~78,000 registered users as of May 2015.



As one of the thousands of users of LAMMPS, my goal here in the next 15 minutes is to give you two cases where we have been using LAMMPS for computation-enabled materials discovery. One is on energy storage and the other is on material design.

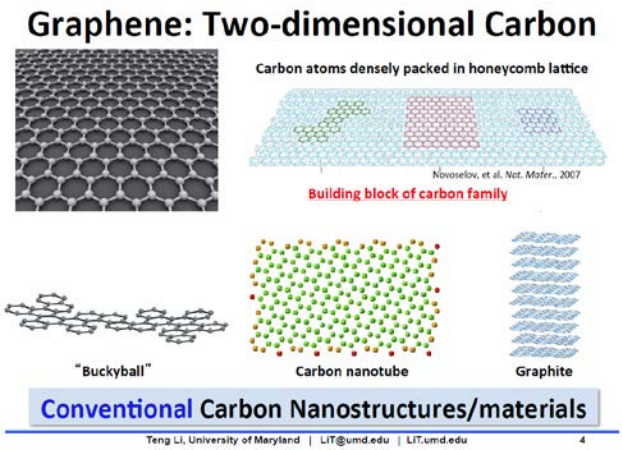


So a few words on what we are doing in my group. My training background is in solid mechanics and material science. We are particularly interested in using multiscale simulation strategies going from atomistic to coarse-grain to continuum along the interfaces between mechanics and different fields: with electronics, we work on flexible electronics and stretchable electronics of interfaces; with physics, we look into the low dimensional materials like 2-D crystals; with electrochemistry of the energy systems and also the soft materials along with bio-inspired materials.



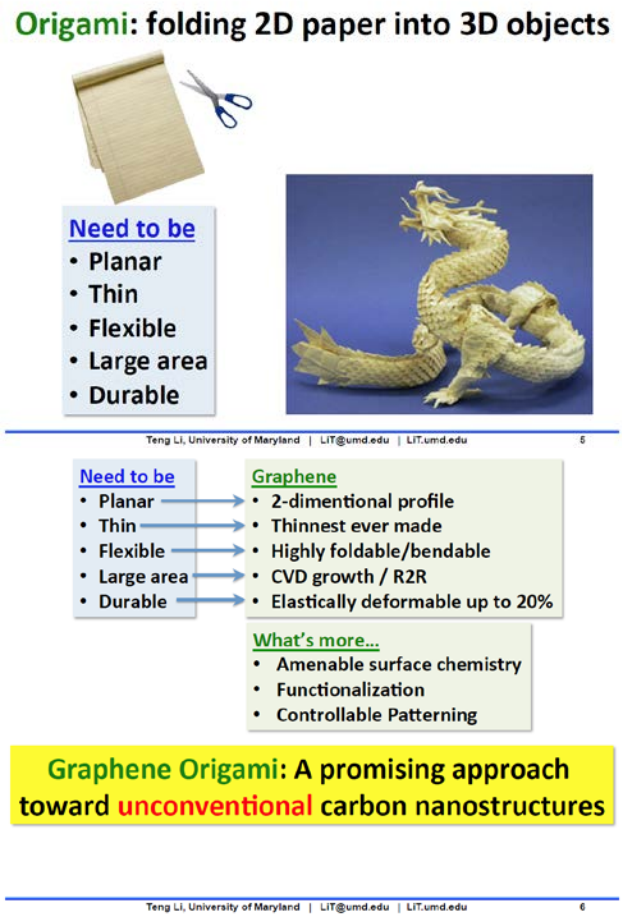
In this talk I will go over two examples among these research fronts.

The first one is Graphene Origami Enabled Unconventional Carbon Nanostructures.



Graphene is a two-dimensional carbon material, densely packed in honeycomb lattice. It is a building block of the carbon family. The experimental discovery of graphene led to the Nobel Prize in physics in 2010. And it is one of the candidate materials which hopefully can replace silicon in the near future to allow us to have better performance because it remains stable down to the six-atom realm.

If you cut graphene into certain shapes, it can form the buckyball. If you cut it into a rectangular sheet and bond it covalently, it forms the carbon nanotube. And if you stack it up, that is graphite. These are all the conventional carbon nanostructures and materials.



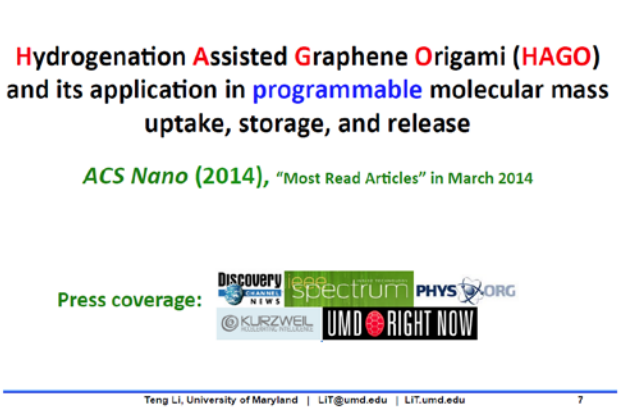
What is origami?

Let us start with 2-D paper. You cut it, you fold it, you can make it a 3D object which can be very complicated. To make origami successful, you need to have the paper to be planar, very thin, flexible, with large area and also durable.

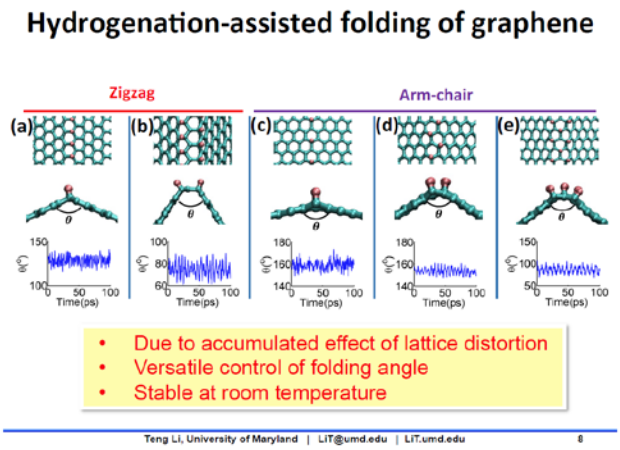
Now if you apply this to graphene, you will find that graphene is the perfect candidate.

It is two dimensional, and the thinnest material ever made with just one atomic layer thin. Also there are actually more desirable features for graphene because all the atoms in graphene are on the surface which makes it amendable in surface chemistry for functionalization and for controllable patterning.

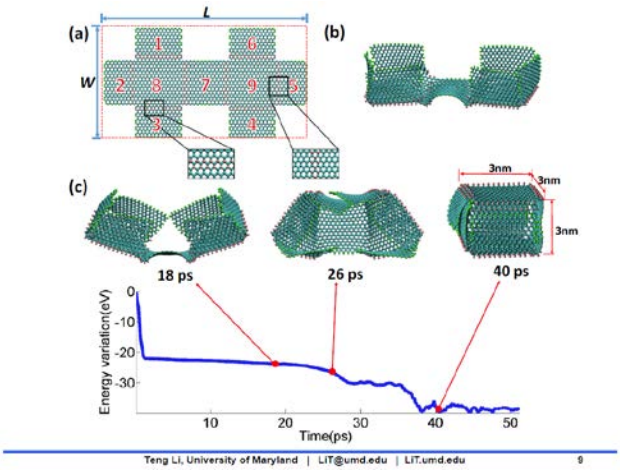
So graphene origami has been seen as a promising approach towards unconventional carbon nanostructures.



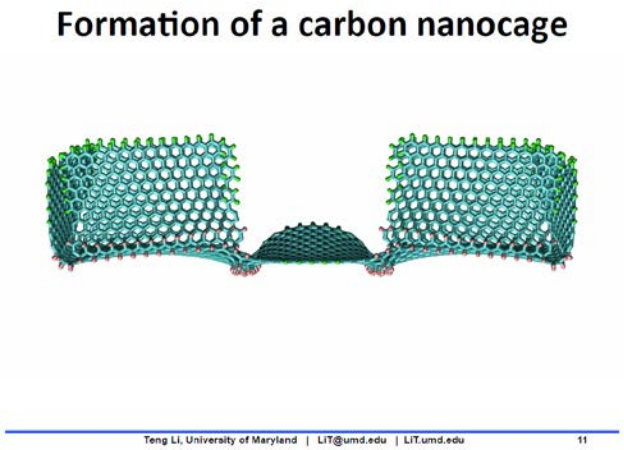
To this end, we looked into the Hydrogenation Assisted Graphene Origami (HAGO) and tried to demonstrate the application of this HAGO process in programmable molecular mass uptake, storage and release.



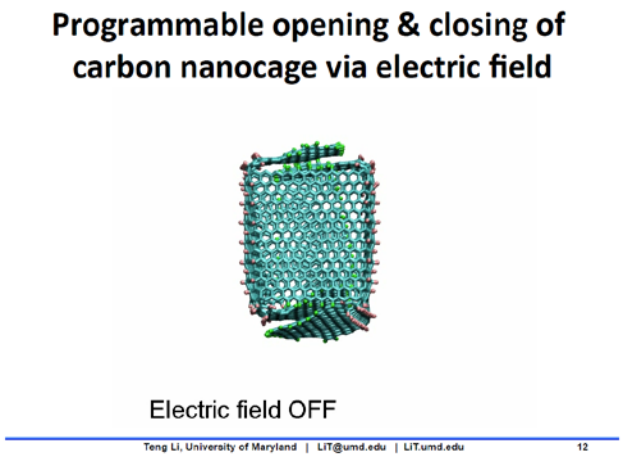
So, graphene is a 2D crystal with all the atoms on the surface which makes it very feasible to be functionalized. For example, if you attach hydrogen with carbon atoms in a controlled way, then the local distortion of the lattice due to the hydrogen can accumulate which can make the folding angle in a more programmable way. So you can have control over the folding angle which leads to the origami process. And this is stable at room temperature.



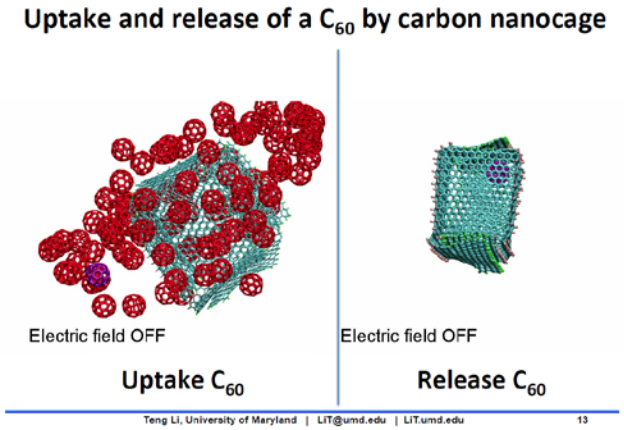
Inspired by this, we designed this initially planar structure, double crossed, and introduced the hydrogenation along the edges we wanted to fold. Then you minimize the system energy, and the initial planar structure can eventually fold up into a carbon nanocage during which energy decreases all along the way, meaning that it is favorable.



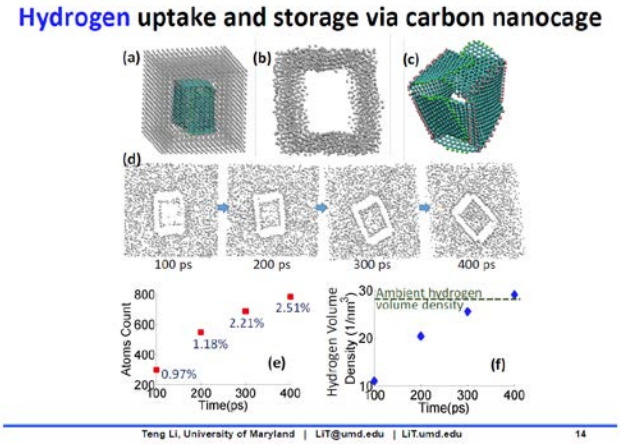
The video I will show you here is the initial folding process of the carbon nanocage. Of course, this is only demonstration, a first step. To make it useful, you want to have controlled opening and closing of the cage.



To this end we actually demonstrated that by using the external electrical field, you can control the opening and closing of the carbon nanocage. Because such a tiny structure here is hard to manipulate. Applying electric field will open up the cage and turning off the field will close the cage.



One application of the programmable opening and closing of the cage is the uptake and release of molecular mass. As demonstration, for example, if you immerse the cage into C_{60} buckyballs. And by opening and closing the cage, one buckyball highlightened by purple is uptaken by it. It is sort of an uptaking process. If you move the cage to the destination and then release the C_{60} by turning on the electronic field, then the buckyball will escape.



And this holds for many other materials, one demonstration is that it can be used for high density hydrogen storage. If you immerse the carbon nanocage in a bath of hydrogen, eventually it can reach a very high hydrogen storage density.

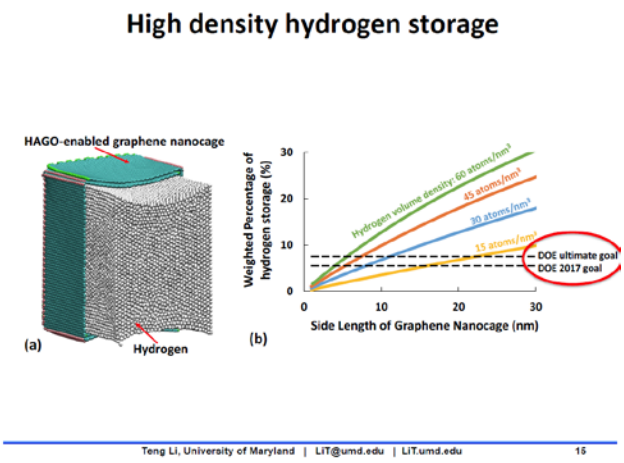
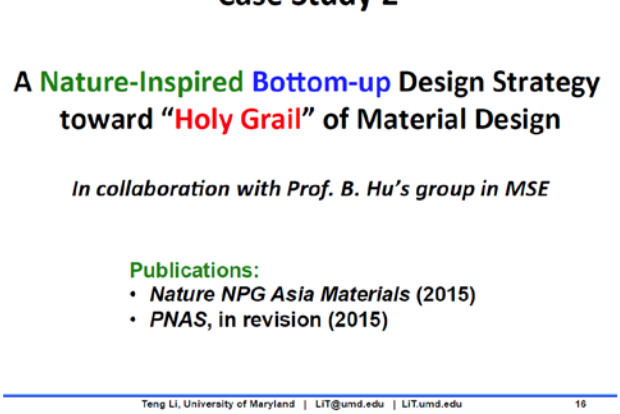
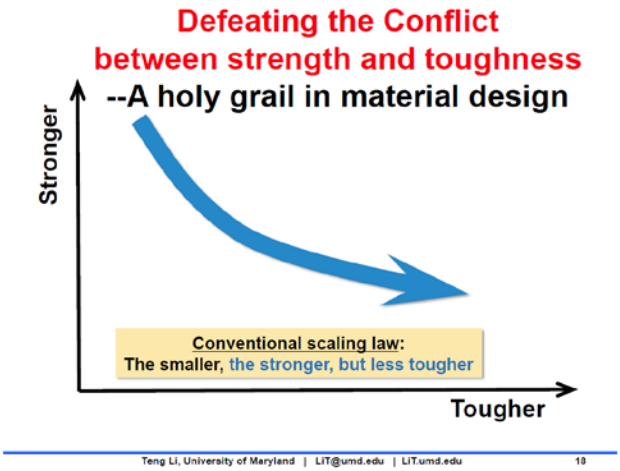


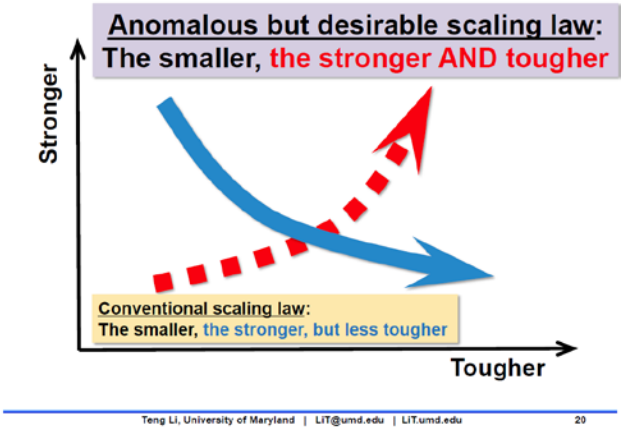
Figure on the left is schematic diagram for the nanocage storing hydrogen. And the right figure is the plot of hydrogen storage vs side length of nanocage. The Department of Energy (DOE) set the on board hydrogen storage density to be 5.5 and the ultimate goal to be 7.5 in 2017. Based on the curve of our prediction here, we can achieve ultimate goal set by DOE by using the graphene nanocage with side length less than 10 nm which is quite feasible as demonstrated in simulation.



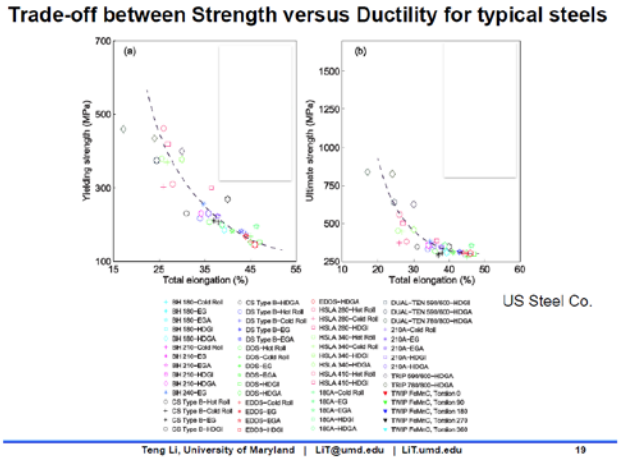
Let me switch to the second case then. We want to use natural materials to come up with bottom-up design strategy to solve a grand challenge in material design. This is collaboration with Prof. B. Hu's group in Material Science department in UMD. In next few slides I will show you some recent results we found very exciting which are currently under review at PNAS.



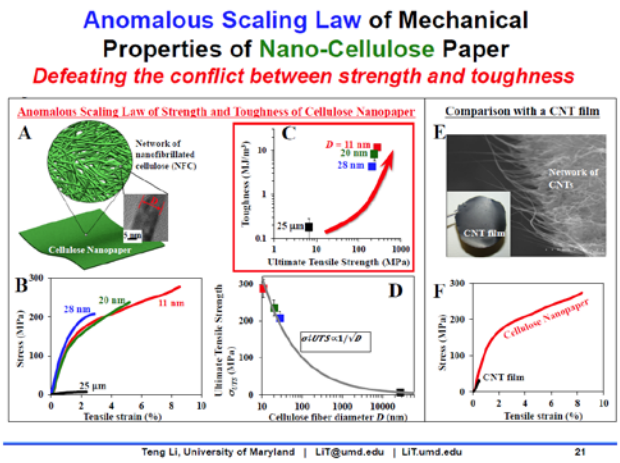
Here is the challenge in material design. For engineering design, you want the material to be both strong and tough. There are many ways to make materials stronger. For example, for metallic material, you can decrease the grain size to make it stronger. The figure shows a conventional scaling law: the smaller, the stronger. But it comes with a price, the stronger you make the material, less tougher it gets.



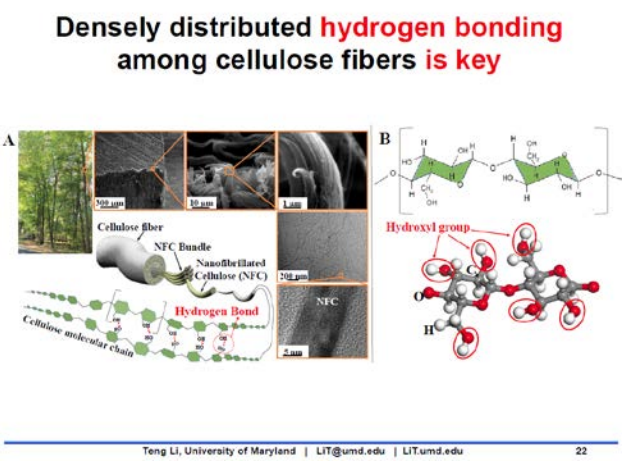
Here is an example. The two figures are yield strength and ultimate strength vs ductility for typical steel. If you increase the strength, you sacrifice the elongation, the ductility.



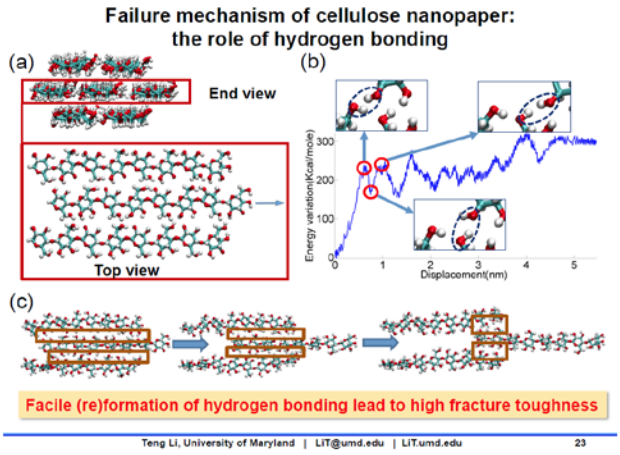
So, indeed, what you have typically is the smaller, the stronger, but less tougher. And this actually holds for many engineering materials. The desirable trend here, will be the smaller, the stronger and the tougher. So we try to offer a possible solution for this trend.



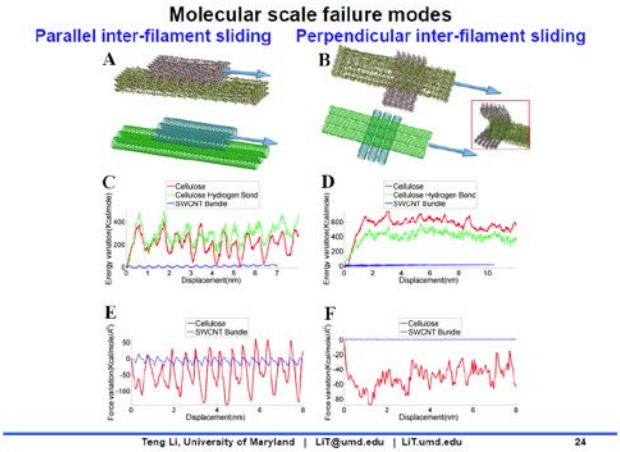
To this end, we look into wood. The paper we typically use is made of wood fiber. You can make the fiber much thinner than typical size we use in paper. In typical paper, the size of the fiber is of similar order as human hair-a few tens of micrometers. We decrease the diameter of the cellulose into nano regime down to a few tenth of nanometers to make the nano-cellulose paper and test its mechanical behaviour. Figure B shows the stress-strain curve of the paper. The black line stands for typical paper and others stand for the nano-cellulose paper. You can see that when you decrease the nano-cellulose fiber size, going from micro to nanometer, both the strength and ductility increase. If you calculate the toughness and strength of the nano-cellulose, you will see the desirable trend we just discussed as shown in figure C. When the material gets smaller, it gets stronger and tougher.



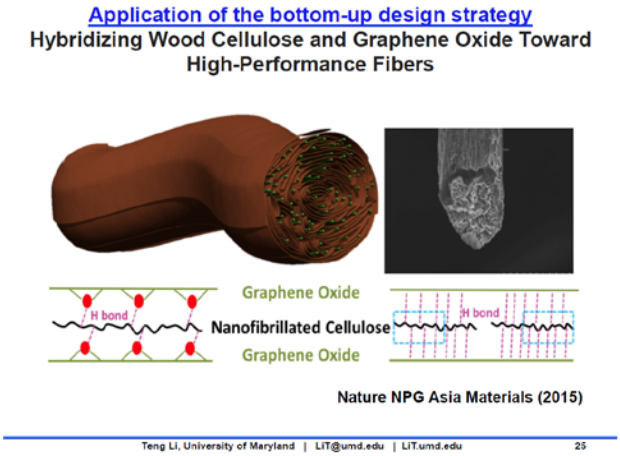
To understand the reason for this, you need to look into the hierarchical structure of the wood fibers. Indeed, it has multiscale features as we see from figure A. The building block of this is the nano-cellulose molecular chain. The much thicker fiber is the one we use in our regular paper, and the much thinner one is the one we use in our nano-cellulose paper. And the feature here is it has a lot of hydroxyl groups, meaning that among the nano-cellulose molecular chains, the hydrogen bond can easily form which is quite strong. It is the reason for nano-cellulose paper being strong and tough.



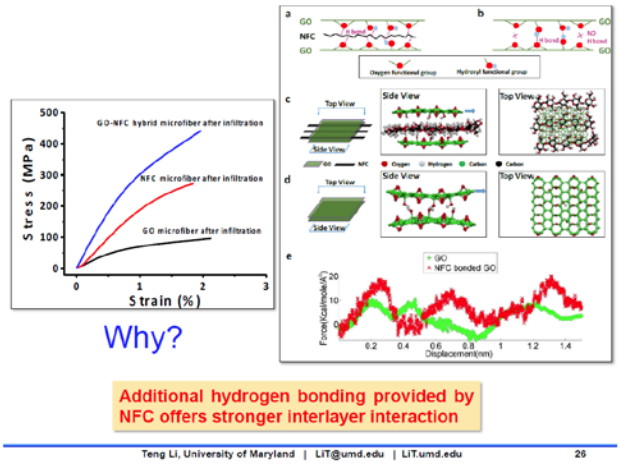
To this end, we studied the failure mechanism of cellulose nanopaper. We modeled seven molecular chains here as shown in figure (a). You can see the energy profile goes up and down in figure (b). This feature captures the breaking and formation of the hydrogen bonds. The process dissipates a lot of energy, leading to high fracture toughness.



For a larger scale of modeling, we modeled the stack of the molecular chains in two different deformation modes. If you compare the results with carbon nanotube bundle, you can see the difference here like in figure C. Some of the peaks will give you much higher toughness. If you do not have the mechanism, for example, CNTs between which you only have Van der Waals interactions, you have much lower performance.



This suggests a bottom-up design strategy and we applied it to different material systems. Here we hybridized wood cellulose fibers which are the green fibers in the figure with graphene oxide which are brown flakes here to make the high performance fibers.



You can show that by hybridizing these two materials, you can have both stronger and tougher microfibers. And the underlying mechanism is essentially the same. You have the additional hydrogen bond formation and reformation during the fracture process.

So these are the two case studies as a demonstration of application of LAMMPS in material design and discovery process. And I hope you enjoy the symposium here, and (HAGO=) have a good one.

The development of ParaDiS: Parallel Dislocation Simulator

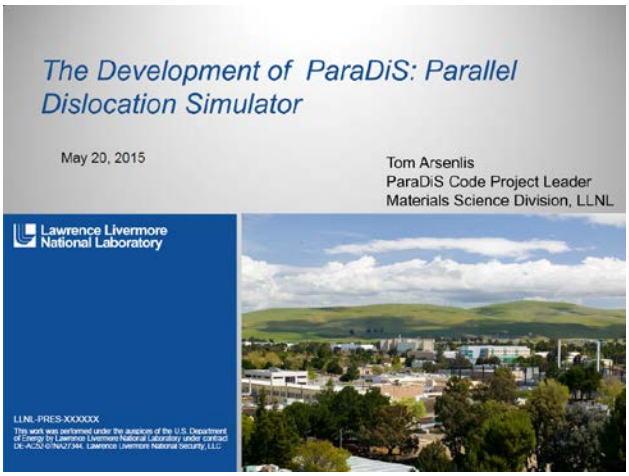
Dr. Tom Arsenlis
Materials Science Division, Lawrence Livermore National Laboratory
Livermore, CA

Abstract

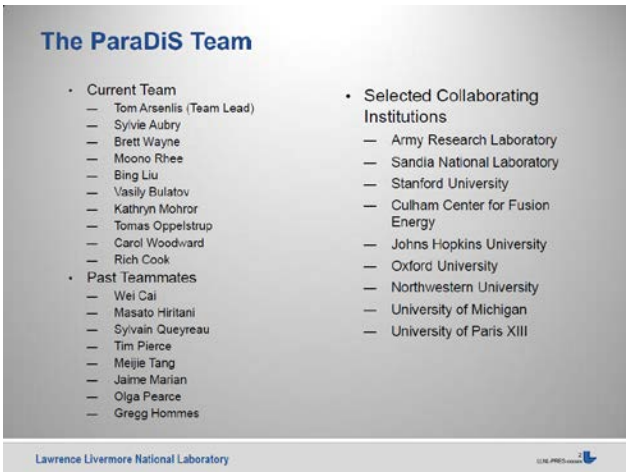
The ParaDiS project began at LLNL in the early 2000's to build a scalable massively parallel code for the purpose of predicting evolution of strength and strain hardening and crystalline materials under dynamic loading conditions by integrating the elements of dislocation physics on an unprecedented scale. The code was first released as open source to the public in 2007 after being deployed on the IBM BlueGene/L machine at LLNL, and extensively used by researchers at LLNL and around the world to simulate the behavior of dislocation networks in a wide variety of applications, from high temperature structural materials, to nuclear materials, to armor materials, to photovoltaic systems. It has been used to elucidate new mechanisms in the plasticity of crystals much in the same way that the transmission electron microscope did in the 1950's. The code currently has over 200+ registered users around the world modifying it as needed to suit their particular needs, and the LLNL based team is looking toward a future of being able to simulate polycrystalline response with the code in the future.

Biography

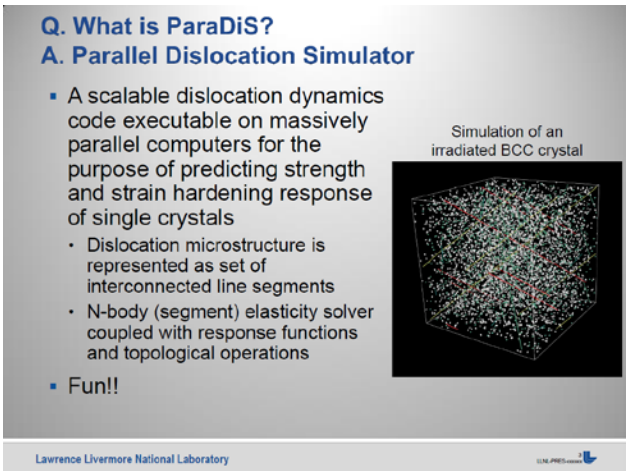
Dr. Tom Arsenlis is currently the High Energy Density Materials Integrated Experimental Team Leader at Lawrence Livermore National Laboratory responsible for executing laser driven experiments at multi-Mbar pressures to understand the equation of state, strength, and phase stability of materials under dynamic loading conditions. Along with this role, Dr. Arsenlis continues to lead the ParaDiS dislocation dynamics code development team at LLNL with the goal of providing a tool capable of predicting the strength of crystalline materials from first principles dislocation physics. Dr. Arsenlis graduated Summa Cum Laude from Cornell University in 1997 with a B.S degree and majors in both Mechanical and Aerospace Engineering and Materials Science and Engineering. He attended the Massachusetts Institute of Technology as a National Defense Science and Engineering Graduate Fellow and received his Ph.D. in Mechanical Engineering in 2001. Upon graduation he joined the technical stat at LLNL. Dr. Arsenlis' research interests are mainly in the area of the mechanical properties of materials focused on the development of physics-based models of the ductility and strength of materials in dynamic and irradiation environments with 40+ publications in the field.



Hi, it's my pleasure to be here. I am now talking about what has become a hobby for me, as I have moved on to become an experimentalist in some capacity, which is very strange. Here I am talking about ParaDiS (Parallel Dislocation Simulator) which is a code we developed at Livermore (Lawrence Livermore National Laboratory) in essence from scratch to do dislocation modeling.



This is the current team, there is a long list there but there are in essence four primary developers: myself, Sylvie, Brett, and Moono, who essentially form the core of the team. Then we have people who come in and contribute on a periodic basis. In addition listed are past teammates as well as institutions with whom we are strongly collaborating with at the present time. So we are not a large team and this is really a cottage code, perhaps only two or three heads over about a decade, during which there has been some cycling of team members.



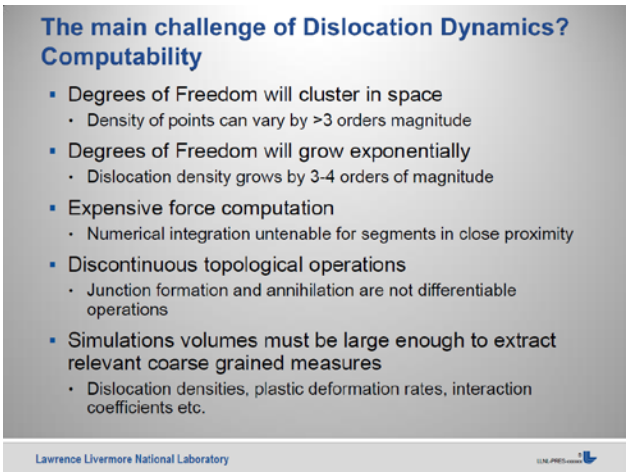
So what is ParaDiS? ParaDiS is our flagship dislocation dynamics code. It is designed to run on massively parallel computers and was built essentially around the time that Blue Gene L came online at Livermore. The purpose of the code at Livermore was to predict strength and strain hardening response, and I want to focus on the fact that we are focused on strain hardening. Essentially our goal was to take simulations out to much larger times compared to what had been done in the past with the purpose of predicting strength and the evolution of strength.

So this (gesturing to the simulation of irradiated BCC crystal) is what a simulation box looks like, essentially a large number of interconnected line segments. If you think about the problem we are solving it is analogous to an N-body problem, similar to something you would see in gravitational mechanics, where the body is now a segment and every segment interacts with every other segment through elasticity. Then through the use of a mobility law and updating in time you may obtain plastic strain as a result of this model. Along with all of that this work is a lot of fun.

Compared to other codes that we have talked about I think this is really the youngest simulation method which is out there in terms of material science. This code started in roughly the 2000 time frame and precursors to this were serial codes.

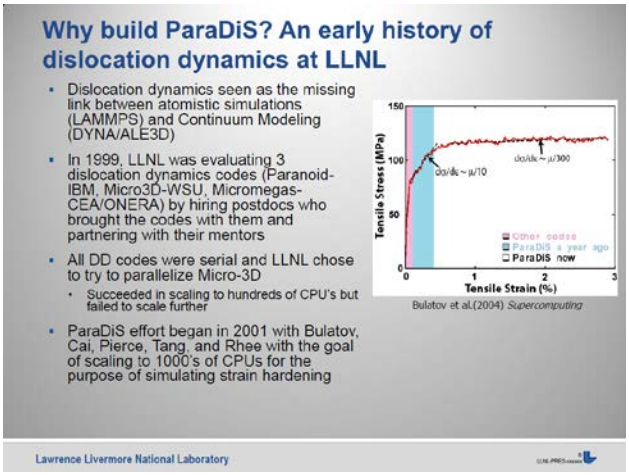
So the question is why was Livermore interested in this type of technology, and I think Doug (Douglass Post) can probably talk better about this than I can. However, as nuclear testing went away in 1992 there was a movement towards this process of certifying the (nuclear) stockpile through science. Meaning the US was going to build very big computers to perform very high fidelity, high resolution simulations and then validate those simulations with very detailed physics experiments. The place where I am currently working, the NIF (National Ignition Facility), is really intended to go to high energies, high pressures, high intensities, and reach conditions similar to nuclear weapons without actually detonating a nuclear weapon. So the challenge is can you design a model which is predictive and valuable at those conditions without actually getting to those conditions.

Therefore there was a premium placed on this predictive model development, and when it came to strength there was in essence this multiscale modeling paradigm that had existed since the 1950's where you could connect the length scales: atoms, to defects, to microstructure, to mechanical properties. So you could see these pictures going back to classical textbooks such as McClintock and Argon (F.A. McClintock and A.S. Argon *Mechanical Behavior of Material*) which had schematic illustrations of defects. So from that time period we knew defects mattered and we can integrate defects to get answers, but we did not have computers capable of doing that at the time. So with Blue Gene L we finally had the ability to connect these wide range of scales and actually simulate the response of an engineering material, with simulation alone. Thus truly predictive modeling.



So the main challenge of dislocation dynamics was computability. Where unlike Lammps, or these other codes, because we are evolving our microstructure and evolving our strain, our degrees of freedom in essence grow exponentially. We are getting many, many order of magnitude increases as the simulation proceeds, and what would happen is we would run to a point, say where we had doubled the number of degrees of freedom in our simulation, and we would then ask to double the number of our CPU's so we could continue on with our simulation. We would continue to do this until we ran out of computer, and once this occurred that was as far as we could go and as long as we could simulate in time. So we would continue to do that, we would start small then grow, and grow, and grow until we ran out of computer and then wait until the next one came along (more powerful supercomputer). So this is hard, the other thing that is hard is that the degrees of freedom would cluster, meaning we would get tight bundles and then areas of sparsity. Therefore, one needs to have a spatial domain decomposition that is able to handle say three order of magnitude in density of points, which is also a very challenging. This is actually very similar to modelling foams in molecular dynamics calculations.

Force computation is expensive. If you want to numerically integrate in essence the error increases very quickly as two dislocation segments come into close proximity , so you have to resort to alternative methods. You also have discontinuous topological operations, so these lines are in space and they will intersect and you have to resolve that intersection somehow. Then once you connect your lines, or do some similar operations, you no longer have something you can differentiate over, which introduces a stop in your time step. Finally we also want to obtain average quantities so we have to run sufficiently long enough to obtain statistically significant results.

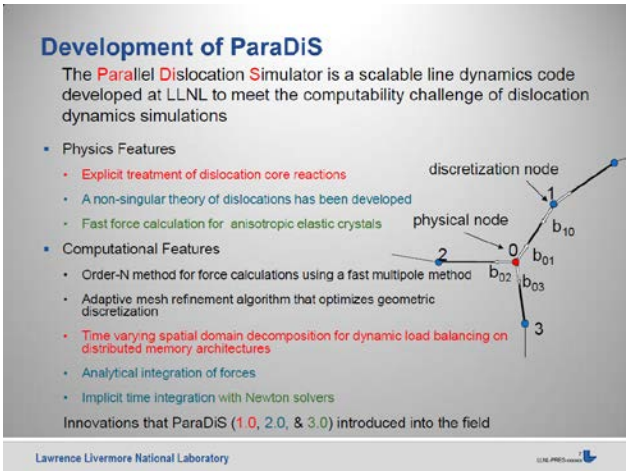


So initially ParaDiS, and dislocation dynamics, was seen as this link between Molecular Dynamics, where we could simulate an individual dislocation or an individual defect associated with strength, and continuum modeling where we are building constitutive models to close field equations of mechanical momentum, energy ,and density.

Initially, we wanted to look first for existing capabilities, which at the time, at the turn of the century, there were four codes in existence. Of those, three were at Livermore. The way we got them to Livermore was basically through hiring post-docs and students who were working on those

codes at various places. We brought them to Livermore saying "ok we are interested in doing this let's see what we can do." So we had three post-docs, one from each of these groups, and we started evaluating these codes. We picked one of those codes, MICRO-3D, and we decided to parallelize the code due to it having a good internal structure. We spent about a year working on that and got it to scale to about 100 CPU's which wasn't enough for our purposes. So that was a success, again that was really the first parallelization of dislocation dynamics, but it did not get us to the point we wanted to get to in terms of scaling to reach large strains.

So we began this effort on ParaDiS in 2001, initially without me, and within three years we were able to scale to thousands of CPU's (indicated by the blue line in the graphic), but again we weren't able to extend out to large enough strains with the first version of the code (indicated by asymptote in the figure). However, progress had been made.



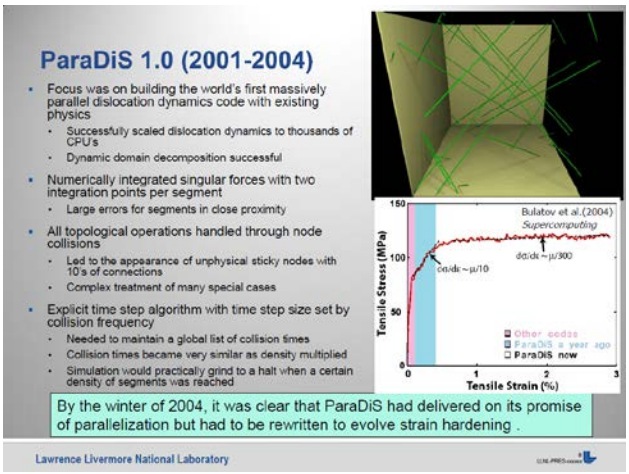
ParaDiS had three major releases (1.0, 2.0, 3.0), where each release is marked by updated physics and an extension in the code capability, all with the singular goal of delivering on the promise of strength and strain hardening modeling.

The first version of the code (1.0) treated dislocations as connected line segments and performed explicit core reactions where it could resolve the intersection of three lines segments (see graphic). Also, from the beginning we had to have a time varying spatial domain decomposition as a result of the clustering of these points. That meant a regular space partition would not scale. So in essence we took the codes that existed and applied this dislocation treatment and irregular spatial decomposition to achieve scaling to thousands of CPU's. However, we were not able to model a long period of time.

In order to model this, in version 2.0 we had to implement implicit time integration, we had to evolve the forces so we had to change the theory of elastic defects and dislocations so that we could do some regularization of the singularity that was there in the classical theory. In addition we also included analytical time integration.

Then in version 3.0 we were looking to augment the physics and extend out the time scale. For example, in version 2.0 all elasticity dealt with elastic crystals, then in 3.0 we extended this to anisotropic elastic crystals. This allowed us to model arbitrary elasticity, arbitrary crystal structure, and all of a sudden you can calculate forces between these defects. In addition we implemented Newton Solvers which allowed us to reach larger times scales.

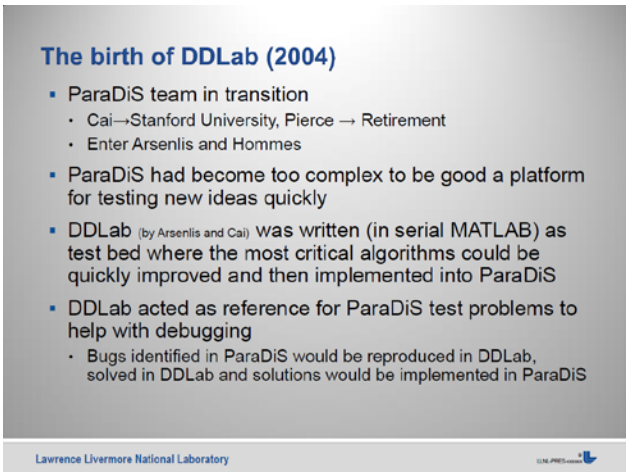
In summary, this is a problem of strong scaling on a large enough box to allow us to obtain a statistically meaningful result to be used in a large scale continuum simulation.



So in the top right we can see a simulation created in ParaDiS 1.0. Just to recap, we were able to fully scale to thousands of CPU's. The issue we had with forces was due to the presence of an elastic singularity as dislocations came into an intersection where the theory broke down, so we could not describe what the force was as the intersection of those elastic bodies. As a result we resorted to line tension, or lower order, models to resolve that intersection, or we coarse grained the calculation of the force such that we no longer had an overlapping point, and both of these methods would introduce error.

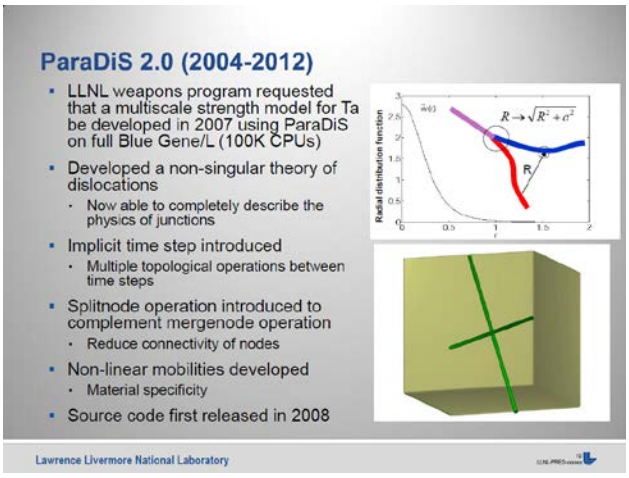
Topological operations were all handled through collisions, so lines would collide, we would perform a reconnection, and through multiple reconnections we would get line segments to form new junction, and this process would repeat. The issue that arose, was as lines kept colliding the code had no simple method for breaking the lines apart. As a result a common occurrence was having a node with 80 or more line segments emanating from them. We referred to these as "Shivas." We knew that these Shiva's were not real but we had no way of relaxing them within the code to something that was more realistic. Finally, we were inexperienced, at this point, with how to handle the time step so we naively set the time step to be the time until the next collision to ensure that all collisions were adequately resolved. For small problems that worked well because the time rate of collisions was relatively small. However, the number of line segments increases with time, and this leads to increasingly smaller time steps as the frequency of collisions increases with line segment density. So as a result you will reach a certain level of strain at which the code will just arrest due to this choice of how to update time. Therefore, we had to break some of our previous features to enable the code to bypass its current capability of plastic deformation, and reach 5 or so percent. The other thing you will noticed in our stress-strain plot is that our stress-strain curve is in fact rather noisy despite the fact that we are working with a rather large volume. This is primarily to do with stored elastic energy located in tight elastic clusters with many line segments.

So come 2004, this was the state of the code and it had become clear that we had to move on in order to get to larger strains even though we were able to scale thousands of degrees of freedom in the current code, which was the first hurdle.



In 2004 there was a transition in the team, Wei Cai who had been team leader left for Stanford, Tim Pierce retired and Greg Hommes and I came in and took over the main core development of the code. In that time ParaDiS had become very complex, as you can imagine the spatial domain decomposition, the ghost nodes and all the memory management. As a result it had become very hard to work with in a development environment, so we needed something simpler to test algorithms on and then port them to what we referred to as our production code.

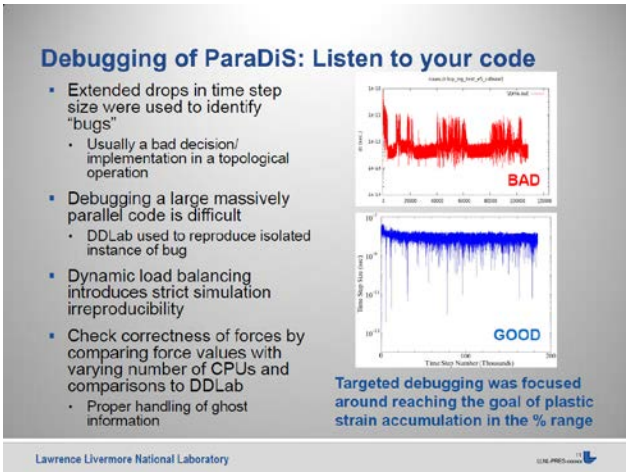
So we built DDLab, which was in essence a serial MatLab code where we could test the most critical algorithms, such as the force calculations, the mobilities, the topological operations in a simple-to-program and fast-to-iterate environment. Then once we had proven the algorithms were stable, we could port them to ParaDiS and run a large scale simulation to reproduce what we had created in DDLab. The other thing we did with DDLab was to use it as a reference, so if we ran a ParaDiS simulation and found a bug, we could reproduce that bug in DDLab, find the solution, and port that back to ParaDiS. So in essence we were not developing with the production-version of the code, and we had a separate code to verify our test problems as we were building.



Then in ParaDiS 2.0, which spanned the next 8 years, we introduced a regularization of our dislocation core. This allowed us to completely describe the physics of the junctions, so in essence dislocation physics was stuck back in the 50's were one could accurately describe the interactions between two well separated dislocations, but as they came into contact the theory broke down. We had to repair that as our interest in material strength meant that we were not satisfied with modeling elastic field problems for a general case. Therefore, we repaired the theory so that we could accurately describe what happened at a point of intersection, then we could perform junction simulations with full physics, instead of a reduced physics model.

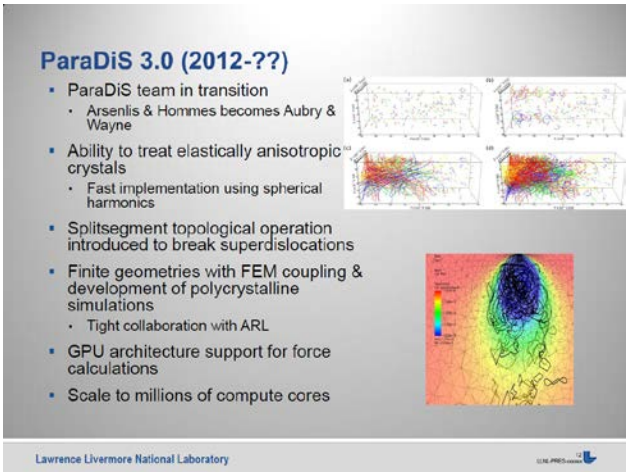
We introduced an implicit time step, where the time step was set by a stability criterion and we would perform many topological operations between time steps, so remeshing would occur, coarsening or refining of the mesh, or even reconnections (as seen in the lower graphic) where we were actually changing the topology of the network. So this would all occur at a fixed time where you would leave the geometry fixed and modify the network, and then evolve in time.

We also introduced non-linear mobility to enable material specificity. The other thing that happened in 2004 was that there was an intense pressure and focus put on the strength modeling team at Livermore. It had been about ten years since the multiscale modeling effort had begun, associated with the ending of nuclear weapons testing, so our funders wanted to point to a strength model product resulting from a decade's worth of investment. Furthermore, they needed the code to evaluate whether the project had potential or if a new direction for strength modeling, based on empirical data, should be used. So over the course of 2-3 years there was intense pressure to get the code working on Blue Gene/L and get a strength model from it that could be used by engineers. This pressure was a source of motivation.

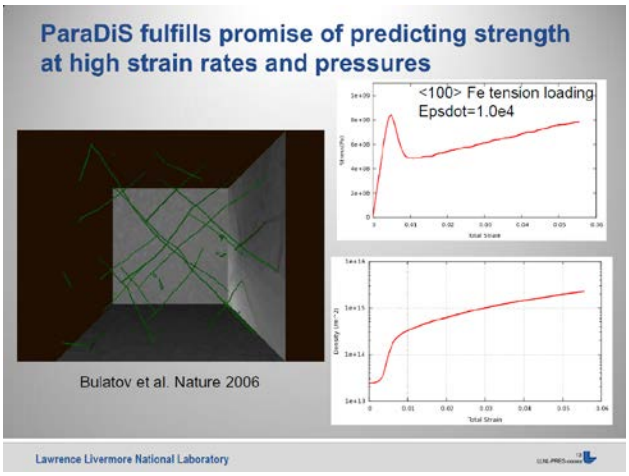


Debugging ParaDiS presented several challenges, for which we employed many solutions. We used the implicit time evolution as a method for debugging the code. Our mission was to get to large strains, so anything that reduced the time step was considered a bug. So this could be a discontinuity in our forces which would show up as a time signature in the code, we would find it, reproduce it on a small scale, solve it, and port it back to the code, and move on. The majority of the bugs dealt with topological operations in the code, so we would do reconnections in the code, and if you were not careful in how you did things, you would end up with a connection and reconnection of the same line segments occurring every time step leading to very low time signatures. So initially we were losing 3-4 orders of magnitude in time step (in the top graphic the loss is roughly 1 order of magnitude). As you solved one problem, the next one would show up, maybe with a smaller loss in time step, and you would continue to remove these things and it would incrementally improve the time step until you reached a code that behaves similar to the bottom graphic. Here, a brief drop in time step size, due to a topological operation, might be followed by very quick recovery. This resulted in less time wasted due to redundant topological operations.

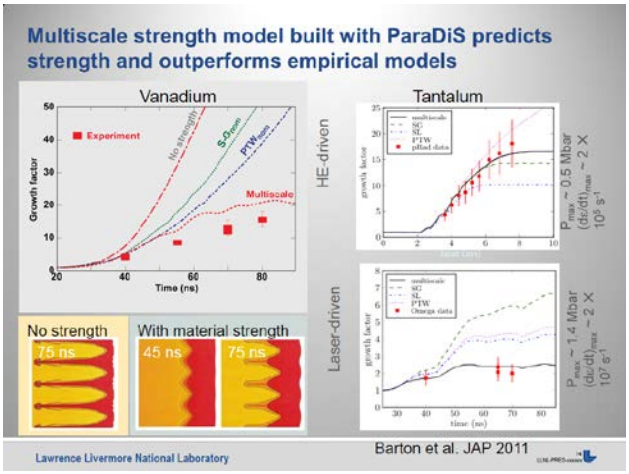
So in essence we would listen to the code, spot bad signatures, reproduce it on the small scale version, fix it there, and then port the fix to the large scale code and move on. This allowed us to reach large strains.



So in 2012, we transitioned from ParaDiS 2.0 to 3.0, then over the past few years Gregg Hommes and I transitioned out, while Sylvie Aubry and Brett Wayne transitioned in. I am still involved but I am not the lead programmer as much as Sylvie and Brett are these days. Now the main physics feature is to treat these anisotropic crystals which is a great improvement in terms of the physics fidelity of simulation we are able to do. Also, we are interested in finite domains and are working with the Army Research Lab to implement new features into the code. In addition we are increasing the number of types of topological operations to improve the physics we are able to perform. Finally we have crossed the threshold of scaling to thousands/ hundreds of thousands of cores to the realm of millions of CPU's/GPU's and are evolving the code for the future. This involves as others have seen a type of hierarchal parallelism where you have threading underneath and mpi to leverage the core structures that are available.

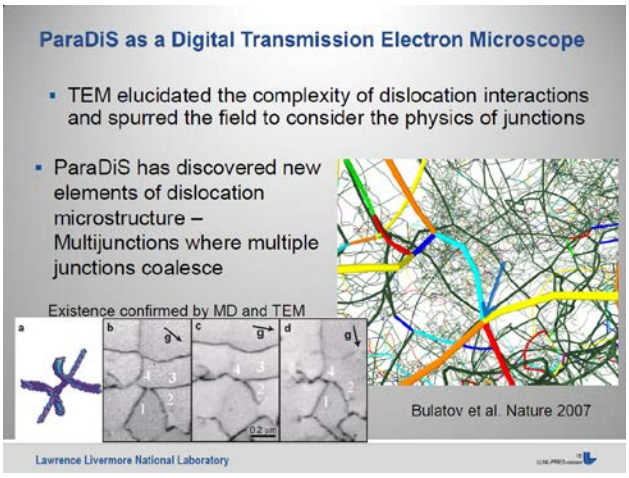


The graphic displayed here is ParaDiS 2.0, so you can get an idea of what it looked like. This is one of our large science simulations, but you can see the noise in the simulation has gone away and we observe very smooth behavior. This is a result of running a large volume and not storing elastic energy and taking care of the topological operation that did before. So in the video we note a smooth evolution in dislocation density, and here note that we are increasing dislocation density by 2 orders of magnitude but in other cases we can increase by more than that. This resulted in a Nature publication.



By 2007-2008, we were finally able to run the code and execute it effectively to model several material orientations under variable conditions to produce a continuum strength model based on simulation alone. This involved a multiscale strategy. We obtained elastic constants with ab initio codes, mobilities with molecular dynamics codes, then took those mobilities and elastic constants and integrated them into dislocation dynamics and ran sufficiently large problem sizes to output a stress-strain curve, density evolution, plastic activity, and then fit that to a continuum model for a material point calculation at continuum. We then used this to make predictions. One such prediction we made was for the National Ignition Facility, where they were interested in the growth of Rayleigh-Taylor instabilities of a solid liquid interface where essentially the low density material is being pushed in the high density material. Where even though the material is solid and not liquid, it retains strength and you will observe elongation and growth of that elongation under acceleration. We had other models at the lab for those conditions, where in the top right graphic you observe a material strength plot under explosive conditions (strain rate 10^5), here all the models agreed. These same material properties are then investigated in the bottom right plot under laser loading, 10^7 strain rate, here you see the multiscale model has much better agreement with experiment than the fitted models.

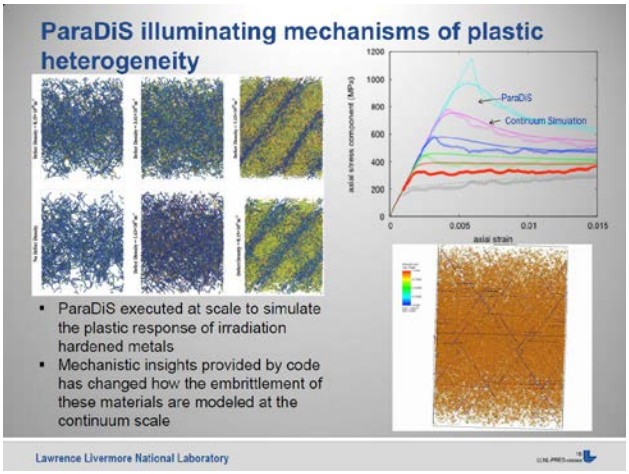
The point was that with simulation we could achieve much better results for two materials (Tantalum and Vanadium), than extrapolated models that had been fit to experiment. This was a validation of our effort and allowed us to continue on our present pace.



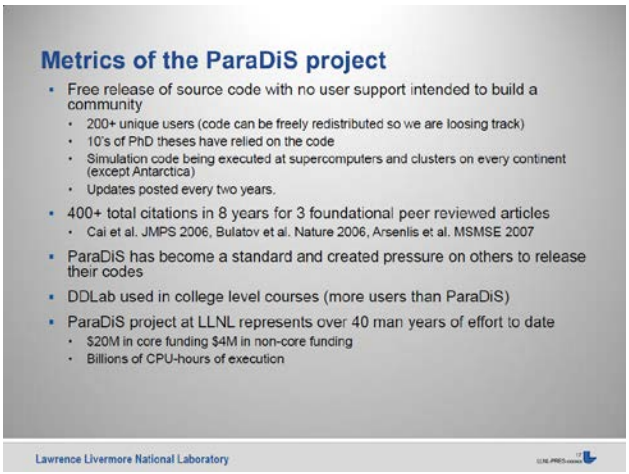
ParaDiS also became essentially a digital TEM. When TEM's were first created people examined crystals and realized there were dislocations and interactions were more complicated than previously theorized due to networks and crossings. This brought on a revolution in the description of plasticity and strength. Similarly with ParaDiS, we have all this information about dislocation evolution (pointing to graphic on RHS), and the only thing that is really governing this is elastic interactions, so ParaDiS is very fundamental in the way it is treating the defect physics. So the questions is: "Can I interrogate the network, learn something about its properties, and then propagate up and say something about strength." What we found in our simulations is that we were getting objects of network connectivity (indicating the multi colored line segments in RHS graphic) in which 3, 4, or more lines would come in and form a stable node which would endure over long periods of time. This phenomena had never been

described in the literature, however it was very stable and repeatable in our simulations. So believing we had found something new we asked our Molecular Dynamics colleagues to reproduce the same kind of configuration and verify that it was stable. In addition, knowing what to look for we could suggest to our experimentalists to strain a material in a certain direction and look at certain planes, and speculate that we should be able to find many of this certain type of defect as it should not be rare due to their stability. Then lo and behold when they did this they were able to tilt the foil in such a way that they could index the dislocations as they were coming in, and they were able to find things that were not simple binary dislocations, which might have been mistakenly identified previously with TEM alone.

So the point is, is that now we have this tool where we can really interrogate what is happening at the defect level, and then promote that forward.

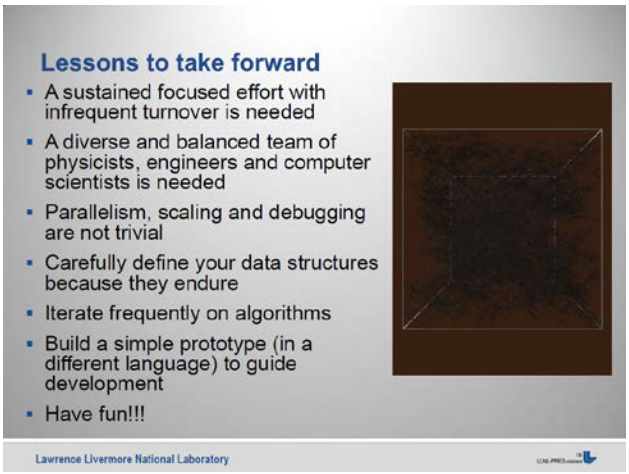


More recently, we've played with larger scale simulations, so here (graphic on LHS) we're interested in a simulation of an irradiated material. There is a classic observation that as a material is in a nuclear reactor, let's say some type of cladding or structural material, its strength is increased while its ductility is decreased due to the appearance of defects that accumulate over time. So we can numerically create these structure, place these defects, resolve them fine enough, and finally do a plasticity simulation with that structure. What we were able to show (upper RHS graphic) is that we could smoothly go from this non-irradiated behavior to highly irradiated behavior by augmenting the simulation with the appropriate number of defects associated with the radiation damage, revealing the fact that we can go from stable plastic behavior to unstable behavior as the material is augmented with enough dislocations. In addition we could show that as you augmented the simulation with radiation damage (shown in banded yellow portions of the visual), you would transition from a material that deformed homogeneously to something that deformed heterogeneously, leading to local failure and ductile to brittle transitions with increasing damage. The nice thing is that once you understand this mechanism and understand the mechanics, you can create a continuum model and promote it forward to do engineering calculations. So people are now taking this on and moving forward with more fundamental models for describing irradiated materials.



So now let's talk about metrics. The first version of ParaDiS did not draw significant interest due to the limitation in time scale. Version 2 was very stable, worked on many platforms, and was released with no restrictions. Therefore, anyone had full access to the source code, allowing them to remove elements or wrap their own code around it. This was in an effort to build a community, as this was really the first parallelized code of it's kind and only serial versions existed previously. As a result of filling that void we became the standard. Since that time we've had about 200+ unique users, however that could well be an underestimate as you can freely reproduce and redistribute this code. So as a result ParaDiS is being run everywhere, on every type of supercomputer, on every continent except Antarctica. Even in Africa, where we have users in South Africa running ParaDiS on whatever computing systems they can get their hands on. In addition, we post updates about every 2 years, so we maintain a very close control over updates because we are so new we want to keep close track of any issues people are having with the code. This is partly because, like others, we do not have a user support model funded in any way, so we want to ensure that we are providing a robust, bug-free code.

If we look at the three seminal publications for the code, they have about 400 citations, and actually the most cited is the algorithm paper that described the topological operations, and provided the foundations for our modified elastic model for dislocations (the algorithm paper is listed). Another way the code has had an effect is that since we have released the code, others have wanted to produce their own versions. So now there are various developments, based on ParaDiS, from other countries such as France and Germany. Additionally DDLab, which is distributed along with ParaDiS, is used a large amount due to its value as a teaching tool because of its ease to understand (MatLab) and well documented source code. So looking at what this has meant at Livermore, this is about 40+ man years, about 20 million $ in core funding going back to 2001, 4 million $ in non-core funding, and billions of hours in CPU execution time as a result of the scale of problems we are now able to run.



So in conclusion, this project required a sustained focused effort and in addition pressure to deliver a scientific result on top of a working code useful to weapons development to sustain the level of funding. The team included chemical engineers, nuclear engineers, applied mathematicians, computer scientists, and mechanical engineers. It was important to have people with real computer science credentials as they know how to structure the code well, but it was also important to have domain experts to know what the mechanics are and how the algorithms should work. Parallelism was hard in this problem due to the spatial non-uniformity, and debugging also become hard as you have these dynamic load balancing issues with the domain boundaries moving, you end up with a problem that is not strictly repeatable. This means that as you move the boundary and perform a topological operation in one step, and then later want to perform the simulation again from a restart file, the boundaries may be slightly different because the load on the CPU's may be different due to load balancing. So all of a sudden the topological operation does not occur in the exact same way the second time, and drift over time ends up occurring. Even though the stress strain behavior and coarse metrics look the same, the fine details are not exactly the same.

So I think it is very important how we defined our data structures, and you want to do that because that is something that endures. In fact the core of the data structure hasn't changed from ParaDiS 1.0, however we do frequently change algorithms, and a good place to do that is in a simple prototype, aka Matlab.

And remember to have fun! Thank you

Dislocation dynamics simulations with microstructure

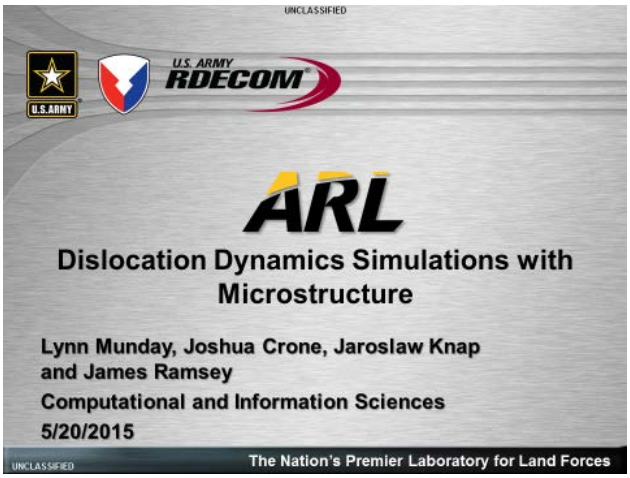
Dr. Lynn B. Munday
Computational Sciences Division – Computation and Information Sciences Directorate
Army Research Laboratory
Aberdeen Proving Ground, MD

Abstract

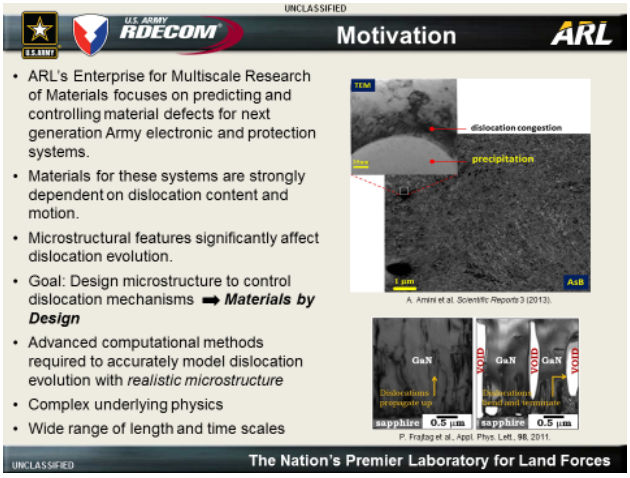
Plastic deformation near lattice defects is primarily dependent on locally activated modes of dislocation motion including cross-slip and nucleation. These locally active dislocation mechanisms are promoted by stress concentrations arising from the lattice defects and lead to the formation of prismatic dislocation loops around defects such as voids and misfit particles. In this work we will show how these locally activated dislocation processes affect the overall stress-strain response of the crystal. In the present work, we use a coupled finite element – discrete dislocation dynamics - (FED3) code to model the evolution of dislocations interacting with material heterogeneities. Dislocations in an infinite bulk crystal are modeled with the ParaDis DDD code and the correction fields produced by the heterogeneous material properties are determined with a parallel finite element code. The two codes are coupled through a scalable data transfer module allowing independent domain decomposition and computational resource allocation. The long term goal of this work is to model polycrystalline plasticity within the dislocation dynamics framework.

Biography

Lynn B. Munday has been a mechanical engineer in the Simulation Sciences Branch of the Computational Sciences Division within the Computation and Information Sciences Directorate (CISD), since joining ARL in 2011. His research interest is in solid mechanics, especially computational models for inelastic deformation. He is currently working on a coupled finite element-discrete dislocation dynamics simulation tool to model dislocation induced plasticity of heterogeneous materials. His doctoral research was on atomistic models of dislocation motion and nucleation in the energetic molecular crystal RDX. He received a Ph.D. (2011) from the University Of Maryland, an M.S. (2005) from the University of New Mexico, and B.S. (2002) from Montana State University, all in mechanical engineering.



I work at the Army Research Lab in the Computational and Information Sciences Directorate. Listed are some of the members of our multi-scale modeling team. This presentation will be building on the previous discussion of dislocation dynamics simulations through ParaDis.

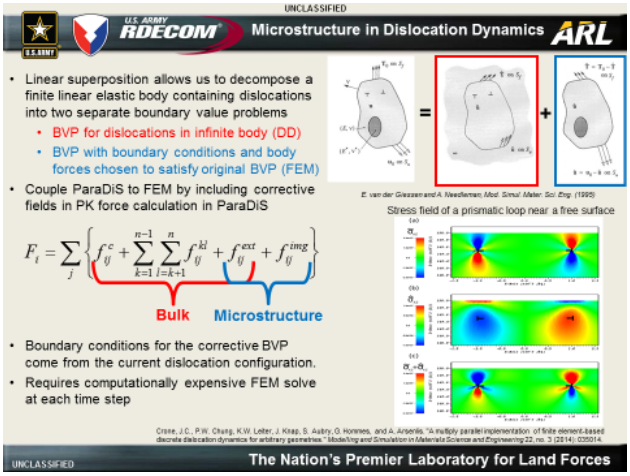


The motivation for this work is the Army Research Lab's enterprise for multiscale research of materials. The primary objective is developing the ability to predict and control defect structures for next generation army electronic and protection systems. Each system is strongly dependent on dislocation content and structure. It is well known that micro-structure can influence the way in which dislocations evolve in these materials. Our goal is to design the micro-structure to control the dislocation mechanisms.

In order to design materials to control dislocation dynamics, we need to include micro-structure effects in dislocation dynamics simulations. Current modeling software such as ParaDis are for a single crystal plasticity, and do not include the effects of grain boundaries or free surfaces on dislocation evolution. We would like to develop a set of simulation tools for dislocations in realistic microstructures that include factors such as grain boundaries, free surfaces, and curvature in polycrystalline materials.

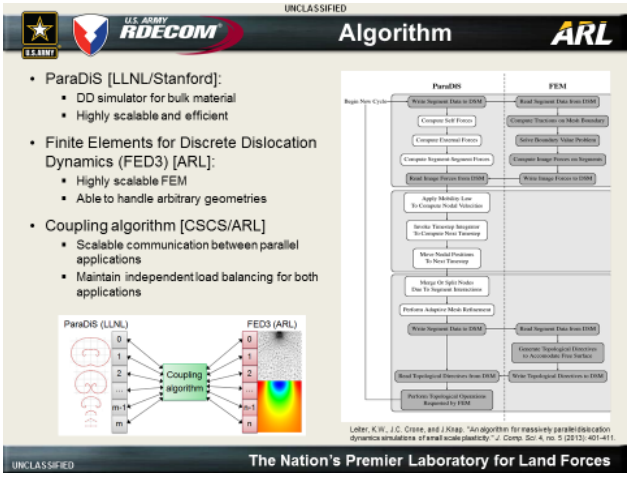
This tool would be crucial to our materials by design initiative. For instance, defects and precipitates could be used to strengthen armor materials, but we currently do not understand how these imperfections will alter the plastic response at high strain rates and it is important not to reduce the toughness. As a second example, adding dislocations to electronic materials relaxes the mismatched strain at interfaces. Introducing voids into the material drives dislocations towards the voids instead of towards the free surfaces where they degrade the electronic properties.

There is complex underlying physics in this problem in a wide range of length and time scales.



Our approach is relatively simple. Dislocation dynamics is based on linear elasticity, enabling us to decompose our problem into two simpler problems using linear superposition. When we wish to model a material that has free surfaces, dislocations, and material heterogeneity, we can break the problem into multiple parts. Our dislocation dynamics problem can be solved in an infinite bulk material, enabling the use of ParaDis. We have a separate boundary value problem in which boundary conditions arise from dislocation structures. We solve this second problem numerically using Finite Element Method. We couple ParaDis to our finite element code through the interaction forces between dislocations.

Normally dislocations interact through the bulk portion of the equation shown above, resulting in n-squared dislocation to dislocation interactions. For our calculation, we introduce new forces due to the microstructure and elastic fields produced by free surfaces. To give you an idea of how this works, we begin with a semi-infinite plane containing two dislocations and a free surface on top (as seen in the top image on the lower right-hand side of the slide). The elastic field associated with the dislocation will not give you a traction-free surface, so a corrective boundary condition must be superimposed to produce a corrective stress field. Only once the two stress fields (shown in the second picture on the lower right) are added, is the correct result obtained. As seen in the third image, the surface is now traction-free, and a free surface has been introduced to the model.



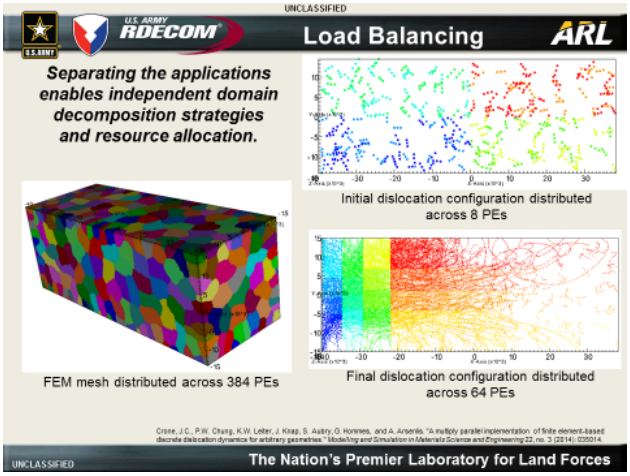
The idea is simple if the dislocation configuration is static, but when the problem is expanded to three dimensional dislocation evolution, the moving dislocations result in changing boundary conditions, and the boundary value problem must be updated and solved every time step.

We need a fast boundary value problem solver, but we also need to couple these two codes in a way that is not going to degrade ParaDis's performance too much so that we can still get to that stress/strain response out to the percent strain levels. We are working with Lawrence Livermore using their ParaDis code, and we have developed our own finite element code to solve that boundary value problem. We use a coupled, distributed shared memory to transfer data between the two sets of code. The code has been written as two completely separate executables with their own parallelization strategies.

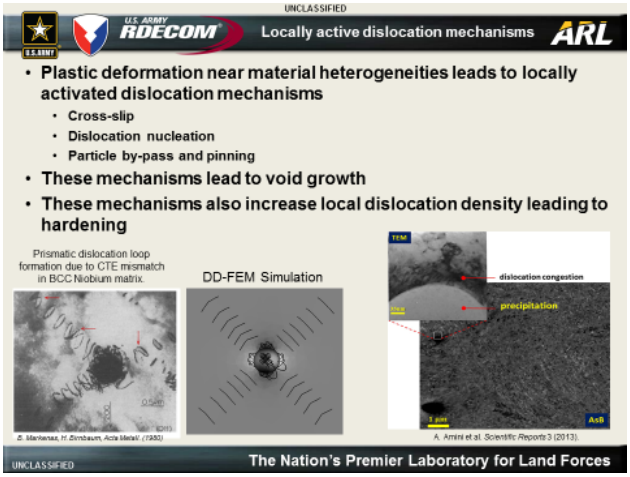
The distributed shared memory is used to communicate between the two programs. Out finite element code needs to know the locations of the dislocations to calculate the boundary conditions

and ParaDis needs to know the forces from our FEM code. The data is transferred back and forth between the programs as solutions are generated. A significant amount of data is generated requiring the processes to be run in parallel.

This approach gives us two advantages. The first advantage is that our two separate codes can be developed in parallel. Lawrence Livermore is developing ParaDis and we are developing the FEM code, allowing us to work independently and collaborate on a few key items. The second advantage is that the calculations can be run in parallel. ParaDis can compute the n-squared calculation at the same time that the FEM code is solving the boundary value problem. These are the two most computationally expensive portions of the code, and running them in parallel increases efficiency.

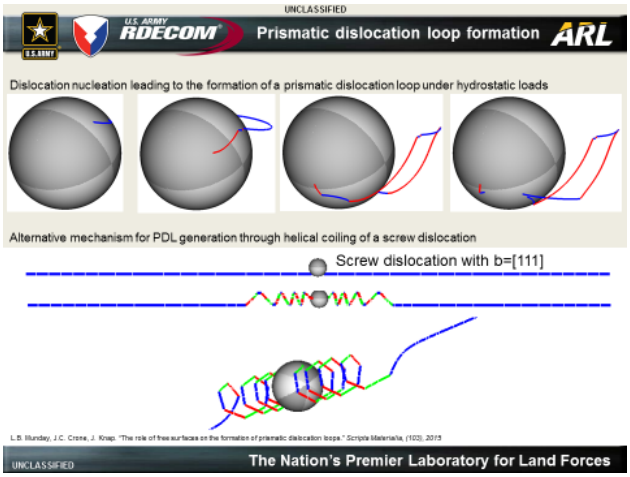


Displayed is a test simulation we ran on a cantilever beam with the dislocation distribution seen in the top right figure. We applied a force on the end of the beam and studied how the dislocations evolved. The beam is seen in the image on the left. The colors represent different processors used to calculate the boundary value problem. The initial dislocation density can be seen in the graph on the top right of the slide. As we apply a load to one end of the beam, bending it, we get unique high stress on the top and bottom, but no stress along the neutral axis. The dislocations hit the stress-free region and stop moving, resulting in a high dislocation density on the neutral axis (as seen in the lower right image). As the dislocation density increases, the number of processors that ParaDis is using must also increase, but the FEM processors remain constant throughout the calculation.



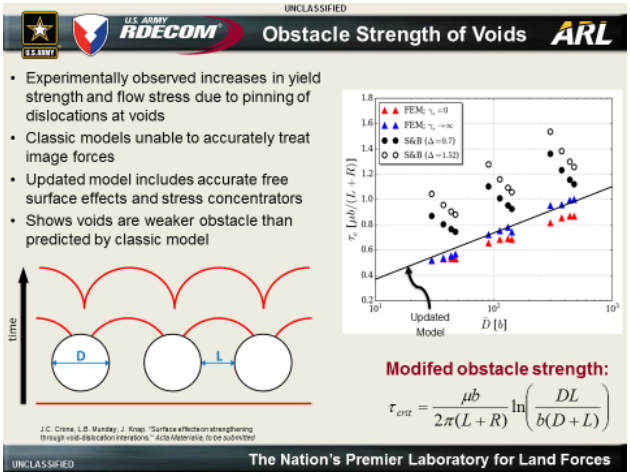
Our key interest is studying the plastic deformation occurring in armor materials. We examine the plastic deformation near material heterogeneities, which lead to locally activated dislocation mechanisms such as cross-slip, dislocation nucleation, and particle bypass. We would like to learn how these single mechanisms influence the overall stress/strain response of the large-scale simulation. Our ability to model these single mechanisms is critical to the study.

Cross-slip and some of the other dislocation mechanisms lead to void growth. If a void exists within a material, it emits prismatic loops (as seen in the central image). These loops cause voids to grow, eventually resulting in ductile fracture. A second area of interest is the accumulation of various dislocations around particles. The dislocations increase the effective size of the particle.

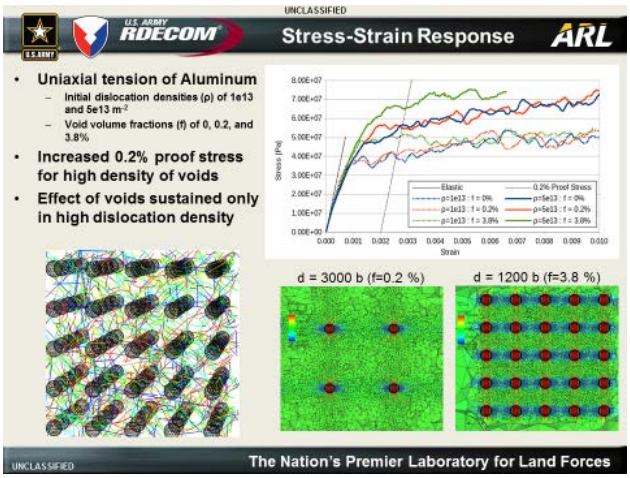


Here we have studied the formation of a prismatic loop around a void. The top images show a void, or essentially a free surface within a material. We want to look at how prismatic loops form as a dislocation peels off of the void (as seen in the first image) and cross-loops around it (as seen in the second). This entire study is enabled by our ability to accurately model free surfaces, and would not be possible on regular dislocation dynamics code. From these high resolution simulations we are able to find a barrier to prismatic loop formation. It is well known that dislocations are naturally attracted to voids, but we have discovered another mechanism, the curvature of the free surface, which delays the formation of prismatic loops. This mechanism was not considered in the development of classical models.

In addition to studying dislocation formation, we also examined the evolution of dislocations already present in materials. As seen on the slide, screw dislocations around voids will start coiling and form helical coils. These helical coils have been previously observed in some materials, and were assumed to be diffusion-based mechanisms. Through our simulation, we discovered a stress driven mechanism. Some have speculated that the helical coils will lead to whisker formation, signifying that we may have found a stress driven formation of whiskers.

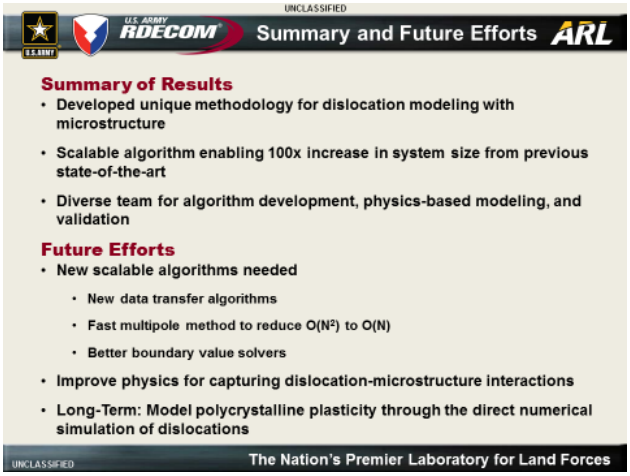


In a third study, we looked at obstacle strengthening. If a dislocation is driven against an obstacle, it hits the obstacle and a higher shear stress must be applied for the dislocation to bypass that obstacle. We want to know the critical stress required, because it determines the overall stress/strain response of a material. Particle separation and particle size are two factors impacting the critical stress. Through high resolution image stresses calculated by our FEM code, we found a lowering of the effect of particles. Prior to this study, only the segments of the dislocation intersecting the voids were modeled, but we found modeling the entire dislocation was important in getting the correct image stresses. Based on our study, we discovered a modified obstacle strengthening model, which is parameterized differently than the original model.



We have all of these single mechanisms, and would like to combine them to discover how each one affects the overall stress/strain response of a material containing multiple voids and dislocations. As a first step, we calculated the stress/strain curves for materials with different void densities and different initial dislocation densities. By turning the mechanisms off and on and comparing the stress/strain curves to the perfect crystal, we can look at the effect of each mechanism. We can then build more detailed atomistic models around the dominant mechanisms to better represent the physics. If we find that certain mechanisms or combinations of mechanisms do not affect the material response, we do not need to expend the effort on atomistic calculations. The tool can be used to drive future developmental work.

As seen in the graph, blue depicts the response of a perfect crystal. The red curve is the response of the crystal seen in the lower center image. Two dislocation densities can be seen in the chart differentiated by the dotted and solid lines. We saw that a lower dislocation density results in an increase in yield strength if there is a high void density, but the overall flow strength remains constant. A high void density results in a large increase in the yield strength. These were the results of some preliminary calculations, but we intend to include more physics in future models.



Right now we have a unique methodology for modeling dislocations and some types of microstructures. We are looking at misfit particles and free surfaces, which is why we modeled voids. We will expand our code to model materials with different properties so that we can move towards a polycrystalline dislocation dynamics code. We have a scalable algorithm able to model a 100 times increase in the size of our finite element model through some of the coupling methodologies I have discussed.

We have a team for algorithm development as well as physic-based modeling and validation teams. Experimentalists are currently validating the code through electronic film studies since the dislocation density is low and they are able to track the evolution of individual dislocations. We can then reproduce the results with our code.

Our future efforts are new scalable algorithms for the data transfer because that is one expensive part of the code. A fast multipole method is already implemented into the ParaDis code as a coarse graining technique, but we do not currently take advantage of this application. We plan to develop

a better boundary value solver since that is another expensive portion of our code. Furthermore, we intend to improve the physics for different microstructure interactions with the long-term goal of being able to model polycrystalline plasticity through the direct numerical simulation of dislocations with strengthening and inverse strengthening effects.

Phase Field Simulations of Structural Transformations in Crystals

Dr. John D. Clayton
Impact Physics, US Army Research Laboratory
Aberdeen Proving Ground, MD
Office of Advanced Engineering Education, University of Maryland
College Park, MD

Abstract

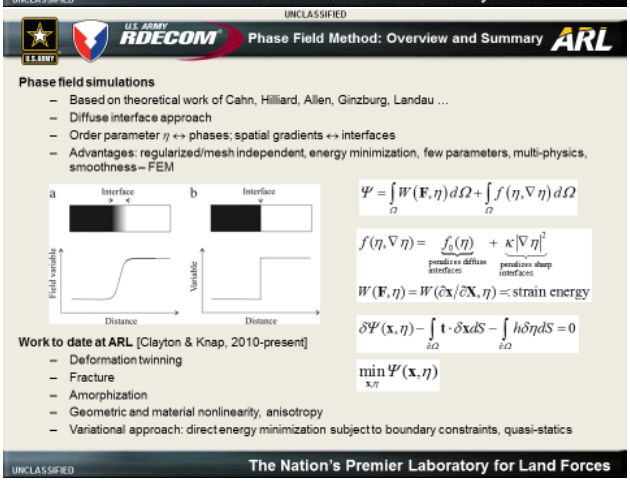
Over the past six years, phase field theory and computational software have been advanced by ongoing fundamental scientific research programs at the US Army Research Laboratory (ARL). Specifically, a novel nonlinear theory based on incremental energy minimization for deformation twinning and/or fracture in anisotropic single crystals and polycrystals has been implemented in a parallel finite element code. An overview of model/code capabilities and key results to date will be given. The latter include validation studies, prediction of twinning in nano-indentation and at crack tips, and prediction of size or scale effects in fracture of metallic and ceramic polycrystals. Results suggest the potential for discovery or design of engineered materials with intra- and inter-granular microstructures—e.g., grain size distributions, lattice arrangements, and secondary grain boundary phases—tailored for globally optimum mechanical properties such as maximum strength or ductility. Issues encountered to date regarding software development and maintenance will be noted and opportunities for future collaboration will be proposed.

In collaboration with Dr. Jarek Knap, Computational Sciences, Army Research Laboratory.

Biography

John Clayton has been a member of the technical staff at ARL since 2003 and is presently leader of the Multi-scale Mechanics Team in the Impact Physics Branch. His research interests include theoretical, computational, and analytical modeling of behaviors of materials, with a focus on mechanics of crystalline solids. He holds a Ph.D. from Georgia Institute of Technology in Mechanical Engineering (major: solid mechanics; minor: applied mathematics); he was an NSF Graduate Fellow and a graduate intern at Sandia National Laboratories (2000). He was awarded the NRC Post-Doctoral Fellowship in 2003. He has authored about 55 journal papers (25 sole-authored) and two sole-authored books (Nonlinear Mechanics of Crystals, Springer 2011; Differential Geometry and Kinematics of Continua, World Scientific 2014) and given numerous invited lectures in international conferences and at various universities and national laboratories. He presently serves on the editorial boards of four technical journals, has guest edited two other journals, and has been a reviewer for around 50 other journals in engineering and physical sciences. He has co-advised several Ph.D. students at universities as well as post-doctoral scholars under the Davies Fellowship program at USMA (West Point, NY) and the ORAU program. He received the

ARL Award for Laboratory Publication of the year (2011), the Army Special Act Award (2014), and five ARL Director's Research Initiative Awards (2005, 2006, 2010, and 2013-2014). He has been an active member of the American Academy of Mechanics, American Physical Society, and American Society of Mechanical Engineers. He also presently serves as an adjunct professor in the A. James Clark School of Engineering at the University of Maryland where he teaches a graduate course in finite element analysis (summer 2015).



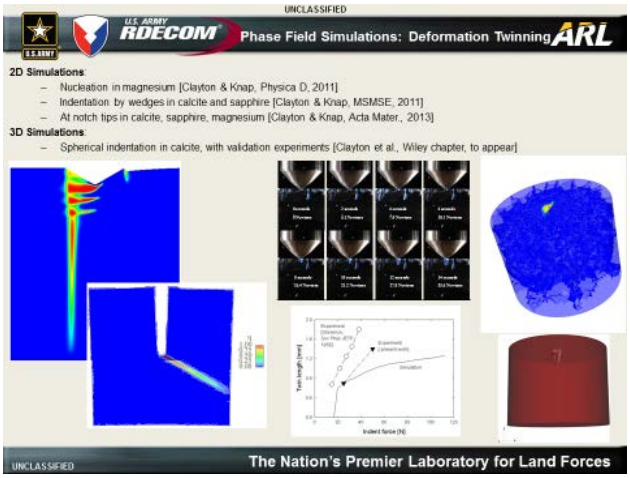
I will be talking about phase field simulation today. I'd like to acknowledge my collaborator Jarek Knap, his name has been on a few of the previous talks as well he is a great person to work with at the Army Research Lab (ARL).

For those not familiar with the phase field method, it is a continuum method based on theoretical work going back to Cahn and Hillard, and perhaps even earlier dating back to the 1950's, and it is known as a diffuse interface approach.

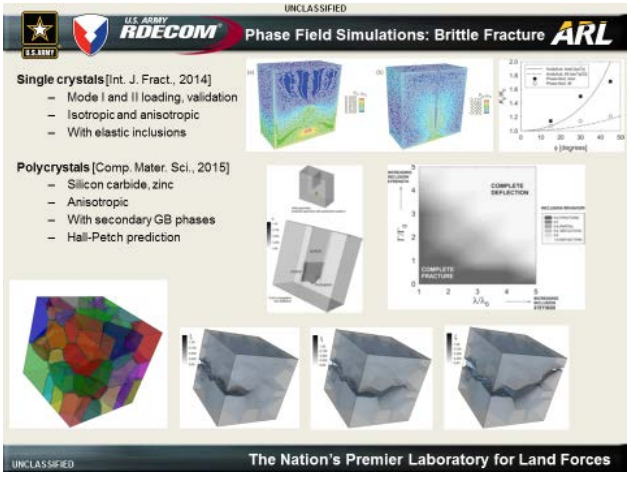
So say you want to simulate two different phases of a material, say a liquid and a solid (these two phases are represented in graphic 1). In the diffuse interface approach we smear out the interface and we have some gradient as well as what is called an order parameter that distinguishes between the two phases. Whereas, in a sharp interface model you would just have no gradient. The advantage of using the phase field model is that you can get a somewhat regularized and mesh independent result, also you can construct a model with relatively few parameters based on energy minimization principles from material science. It is good for handling multiphysics problems and the smoothness of the gradient lends itself well to finite element methods.

So what is shown in equation 2 is an energy functional, where the total energy is the sum of the integral over the body of some elastic strain energy (first term on R.H.S. of equation 2) plus some interfacial energy (second term on R.H.S. of equation 2). The interfacial energy will at a minimum consist of two terms (shown in equation 3.a), one is a function of just the order parameter which will tend to cause your interfaces to shrink, and another term is the gradient term which penalizes the sharp interfaces. It is the competition of these two terms that yields a prediction of microstructure and a prediction of the width of an interfacial zone. Here (referring to the elastic strain energy used in the functional shown in equation 2, whose functional dependency is then expanded upon in equation 3.b) we are actually using a large deformation theory where **F** (appearing as an argument to the strain energy function) is the deformation gradient, for those familiar with continuum mechanics. We also consider in these problems a variational approach for quasistatic where we minimize the energy functional subject to certain boundary constraints.

Work to date, with Jarek, over the past 6 years, we have looked at deformation twinning in crystals, fracture mechanics, amorphization in crystals. In addition we've considered geometric and material nonlinearity and anisotropy, combinations of these things we think are new and have not really been addressed by other groups in phase field modeling for mechanics problems.



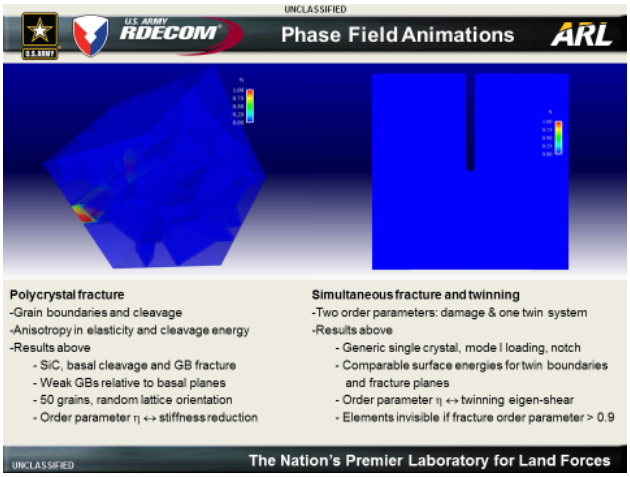
So next I am going to go over a few example problems in more detail to give an idea of what we've done. We began with a 2-D code and 2-D simulations first looking at twin nucleation in pure magnesium. The next problem we looked at was indentation by wedges in transparent materials (graphic 1), calcite and sapphire. The interesting thing is that we have experiments done and there has also been a good amount of work done in the literature on the indentation in calcite which is a softer transparent mineral, and what you see there is that if you do an indentation and then remove the indenter the twin will pop in and then remove back out, which is kind of neat as it is a reversible twinning. We also looked at twinning induced by stress concentrations at notch tips in calcite, sapphire and magnesium (graphic 2), where the twin plane is oriented in the direction of maximum shear stress and you have your notched body loaded in pure K1-loading. We did some complementary validation experiments using the calcite with spherical indentation (graphic 3), where the tip in graphic 3 looks conical, but when you get close to the tip it is a spherical indenter. We have also done some fairly large 3-D simulations (graphics 5 and 6?) using the phase field model and obtained reasonable validation, at least for the initiation of twinning (graphic 4). As the force became larger our simulation tended to under predict the length of the twin relative to experiment.



The second problem we looked at was brittle fracture. We began by considering single crystals or homogeneous bodies, first validation of the model for pure mode 1 and mode 2 loading (graphic 2). We looked also at anisotropic materials, so say we have a cleavage plane where a material tends to fracture in a single crystal that might be misoriented relative to your notch. Then in graphic 8 we have a validation result for this situation where we compare with an analytical linear elastic solution for a misoriented boundary, so you might have a crack going in one direction but it wants to meander on the cleavage plane.

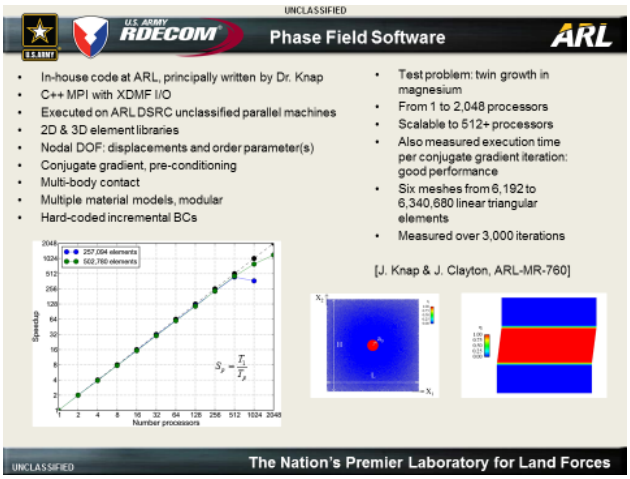
We also looked at what happens when you put an inclusion, or a second phase particle in front of your notch (graphic 3), and whether the crack will deflect around that inclusion or cut through. So we observed the crack going around the inclusion and rejoining on the other side, which is called crack bridging. A common method of strengthening ceramics is to place secondary phase particles within them to arrest the growth of cracks and prevent cleavage cracks.

We were able to parameterize what would happen if you were to increase the stiffness of the inclusion or the strength of the inclusion relative to the surrounding material, and whether or not you get deflection or fracture cutting through that second phase. Another problem we looked at was fracture in polycrystals (graphic 1) so we have meshes of dozens, or even hundreds, of grains that may be anisotropic both in terms of the cleavage planes and the elasticity. We can put layers of secondary phases in between these grains to account for interfacial impurities or amorphous zones, both of which are common in the Army's armor ceramics such as silicon carbide. We were also able to make a Hall-Petch prediction by shrinking the grain size of the ceramics and observing the strength of the material increase. This was qualitatively in agreement with some experiments which have been done. So graphics 4-7 are just a time sequence of a polycrystal being pulled in tension where you can observe the crack propagating through the grain and grain boundaries.

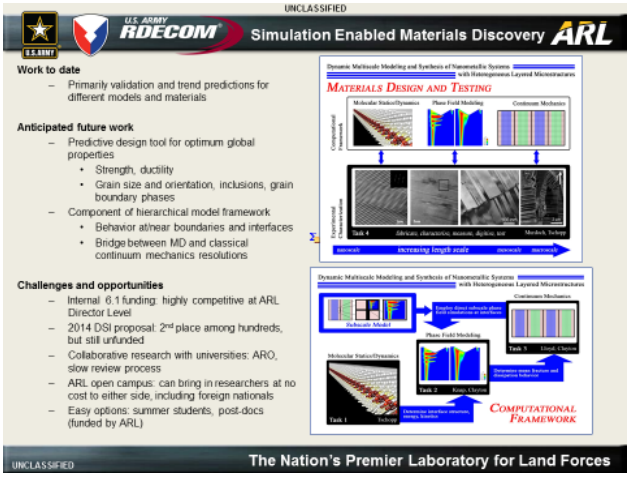


So I will show a few animations as well to give you an idea of what the simulation looks like (graphic 1). On the left is the animation of fracture in the polycrystal, this being silicon carbide, with weak grain boundaries, and we actually seed the material at one edge to induce cracking and as the simulation proceeds you can see the damage start to develop. As you are pulling apart the damage zone starts to grow and the material weakens.

The simulation on the right is something we have been working on recently where we have two order parameters at once, we have a damage and one twin system in this 2-D simulation. So the twin system here is oriented at angle relative to the notch (as shown before in slide 2 graphic 2) and the fracture model is actually isotropic so that the fracture can occur anywhere. We also use comparable surface energy for the twin boundaries and the fracture planes, and what the color represents is the twinning transformation. For the simulation graphic the elements (twins) from the visualization are deleted when a fracture forms so when an elements goes away it is a fracture and when you see the color that is indicative of twinning. So (the simulation video begins running) we see a twin start to form as you pull this notch apart, and eventually it will also start to fracture. The interesting thing about this model is we can try to learn a little bit more about the interplay between the two different mechanisms, whether twinning is promoted by the fracture or vice-versa. Eventually, once the twinning saturates, a mode one crack will form and as that crack propagates a twin is formed at the tip of the crack. There also occurs some cracking between the original twin and domain.



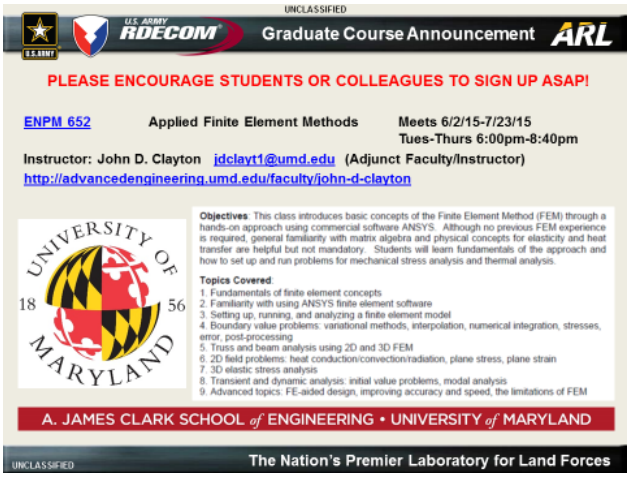
So I thought I'd talk a bit more about the software that we use. We have an in house code developed primarily by Jarek, originally by Jarek at least. It is written in C++ where we run on parallel machines, we have a library of 2-D and 3-D elements. It is not a standard mechanics finite elements program because in addition to your displacement degrees of freedom, you also have your order parameter as a degree of freedom to your nodes. We use conjugate gradient energy minimization and we have the ability to account for multi-body contact, multiple materials at once (which is somewhat modular). Unfortunately right now the boundary conditions are hard-coded but that is something we want to generalize in the future. We have seen good scaling up to 512 or more processers (graphic 1) running a twin nucleation problem in magnesium (graphics 2-3), where starting with a circular inclusion and you shear it, at some point you are going to get a twin to cut across your domain.



So the work to date is primarily focused on validation and trend prediction for multiple material models and multiple materials. What we'd like to do in the future is expand the use of this model to more practical application at the Army as far as optimizing global properties such as strength and ductility, safer armor and projectile applications, in addition looking at different effects of grain size, orientation, inclusions, grain boundary phases. What we have proposed doing to our management last year was a hierarchical framework (graphic 2) were we use MD or some finer scale model to feed into phase field and give us energy potentials as well as surface energy and other similar properties. In addition, use phase field to represent boundaries (graphic 1), were the application of the simulation displayed was a layered nanometallic system, also continuum mechanics and crystal plasticity code for doing larger scale things which are beyond the reach of the phase field model.

Some of the challenges we have include funding, for example we received second place in a director's level proposal, making it all the way to the end and getting nothing because only the top proposal received funding but that's the way it goes at ARL. However, there are other opportunities such as collaborative research, which has historically been done through the Army Research Office but unfortunately this tends to be a very slow review process and people tend to move on by the time things get funded. Recently ARL has started the open campus initiative where researchers can come in at no cost to either side, including foreign nationals, it has also become a bit easier now than previously to get guest researchers to come and visit ARL. Probably the easiest and best

option for collaboration with universities is the summer students and post-docs which can be funded directly by ARL rather than going through the Army Research Office.



So as my final slide, I am teaching a course here this summer in applied finite elements for anyone who is interested, or has students who might be interested please encourage them to sign up so that they do not close the course as there are only 5 students currently registered and the minimum is 7. I was told that students tend to wait until the last minute so it may not be anything to worry about but I'd like see it go through so if there is any interest please register.

And that's all I have.

Materials cartography

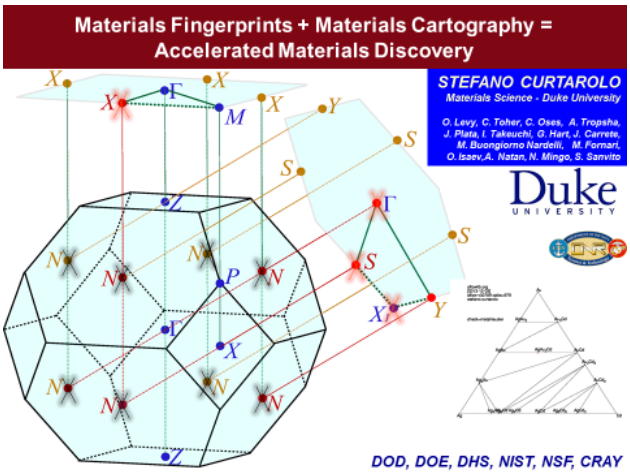
Dr. Stefano Curtarolo
Department of Mechanical Engineering and Materials Science, Duke University
Durham, NC

Abstract

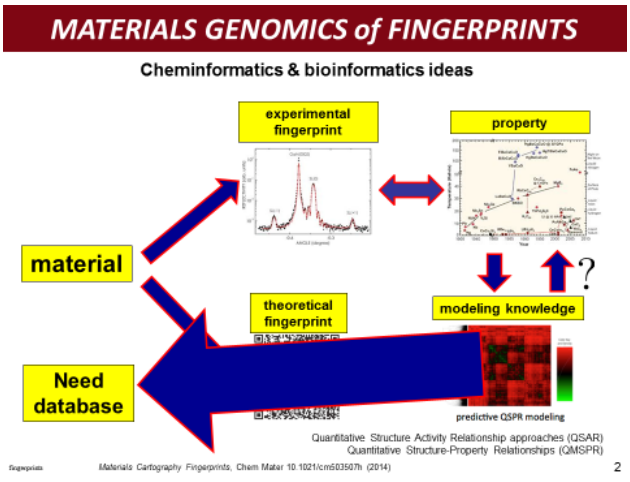
In this presentation, we will introduce electronic and structural fingerprints for representing and mining the material space offered by online repositories. Examples will be given to assess the challenge of accelerated materials development.

Biography

After studying Electrical Engineering and Physics in Padova, Italy, Stefano received his PhD in Materials Science from MIT in 2003. Since then, he was faculty of Materials Science and Physics at Duke University. During his time at Duke, SC received the ONR-Young-Investigator, the NSF-Career, the Presidential PECASE Awards, the International Union of Pure and Applied Physics - Young Scientist Prize in Computational Physics, the Stansell Research Award and the 2013 MURI Award for strategies in element replacement. SC was promoted to Associate in Oct.2008 and to Full Professor in Feb. 2012. Currently he has more than 100 refereed publications and more than 160 invited departmental seminars and talks in national and international conferences. At Duke University, the SC's group started and mantains the ``on-line ab-initio aflowlib.org <http://aflowlib.org/>" consortium containing free energy information and electronic characterization of more than 700,000 entries/compounds.



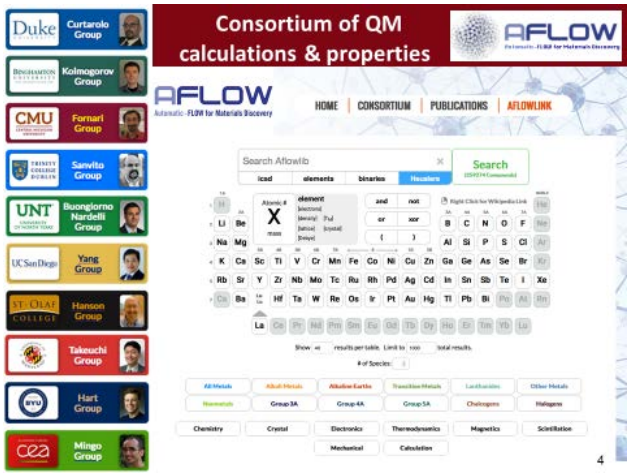
Today I will talk about accelerated material discovery through material fingerprints and material cartography. More specifically, how we use these databases and techniques to look for correlations and find new and better materials.



What is material genomics of fingerprints? First, let us talk about how experimentalists look for new materials. They perform experiments, parameterize the materials, and get the fingerprints like those shown in the top left graph. They perform a lot of experiments and put these properties in a table or picture to see the development of the materials, how to look for the directions, how to tune these materials to pursue new directions, and so on.

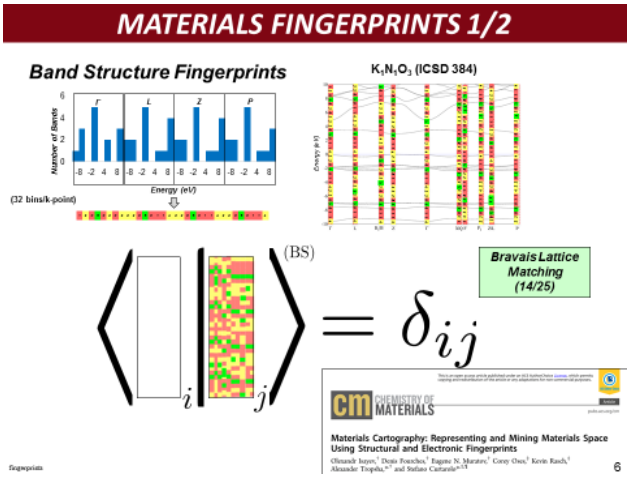
We would like to get the same information, but through our computational methods. How do we want to do this? We want to create the fingerprints, which are exactly the same as the experimental approach, but represented in a computational format. The image on the bottom left is an example of a Q-R barcode that we see all the time, and I will show you how it is created. Once we are able to look into the fingerprints, we try to make tables of the properties with respect to the fingerprints, and look inside the region where there are properties of interest. Once we are able to do this, we go back to the experimental properties and we are able to predict materials and inform experimentalists what to test. The idea is that instead of going clockwise, we go counter-clockwise, and we might be quicker, better, and able to explore systems that are hazardous to experimentalists such as toxic elements or explosive combinations. To do this, to create the fingerprints, we need a database.

Here is the database that we have been developing for the last ten years. This is the AFlow consortium. AFlow stands for Automatic Flow for materials discovery. It is a sort of Google of materials. You can search for a combination of elements, identifiers, or ICSD, and you can get applications such as phase diagrams. It is getting more complicated, but also more powerful. AFlow is a big consortium in which there are 9 universities including Duke. Listed on the left are my collaborators, including the cofounders.

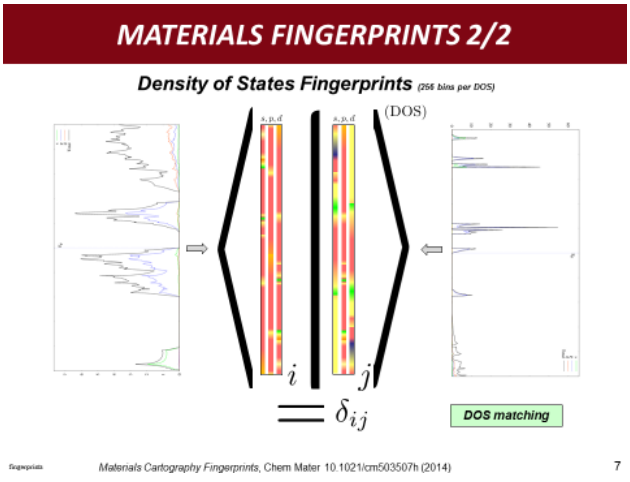


The image displayed is a periodic table of materials. You can click the elements and it displays the properties, or you can search for particular properties. Below the table, you can search for descriptive properties, and below that you can search for functional properties.

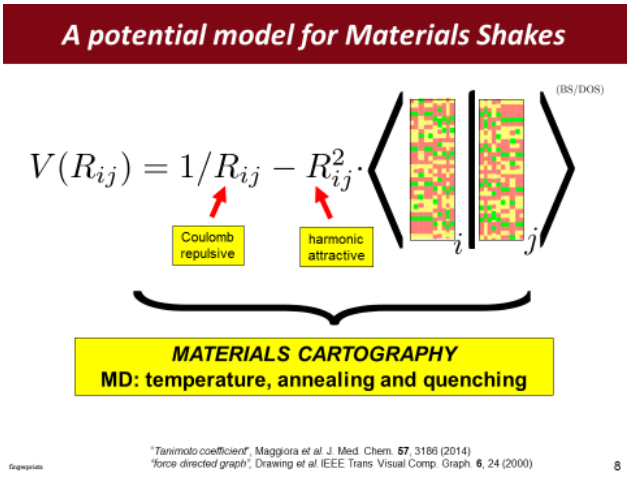
Now that we have our database, let's discuss the materials. How can you compare two materials? It is difficult to compare two materials because they are completely different. It is like comparing apples with animals. To compare materials we do the following trick, suppose that your material is a periodic, crystalline material. The crystalline material is described in reciprocal space. In reciprocal space, every material has a set of band structures, the solutions to the eigenvalue equations, forming the levels of all electrons. We can discretize these lines, and create a histogram of how many times a line crosses each of the points in reciprocal space. Once we have discretize the band structure, we associate colors to the histogram quantities. You can change the calculation by altering the number of degrees, the distribution, and the number of points. If materials are different crystal structures and not compatible, for example one is BCC and one is FCC, they have at least the origin in common. There are always tricks to compare materials that have different configurations.



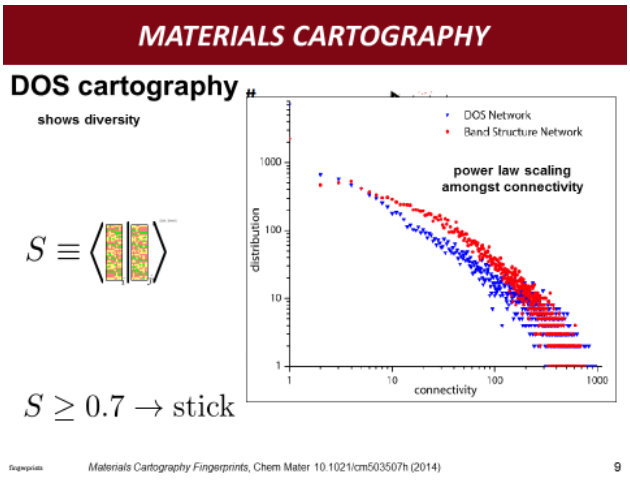
Once your discretized electronic structures are complete, you can generate a picture in which every compound can be visually represented by a series of colors, such as the images displayed in the lower left-hand side of the slide. When we have multiple pictures, we can use the same techniques that computational scientists have been using for the last 30 years to compare pictures. We create a metric or norm, and are able to say how one picture relates to a second picture pixel by pixel. For instance, a red pixel in the first picture and a yellow in the second, could equal a black in the resultant picture, or red and red results in a red, and so on.



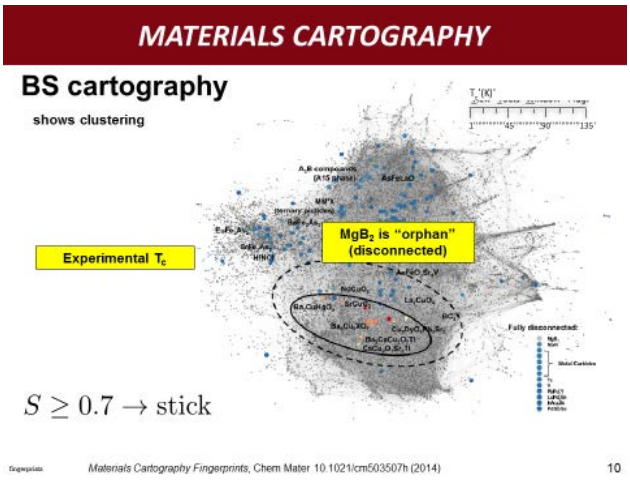
Once you have the resultant matrix, you know the distance between the two materials. When you have the distance between the two materials, you can calculate a force. For instance, materials that are similar have a small distance and will attract each other. So once you have a distance between the two materials, you can create a simple potential comprised of a Columbic repulsion plus a harmonic part, proportional to the relative proximity of the materials.



Thus far we have calculated fingerprints, norms, and potentials. With the potentials, you can now run molecular dynamics within the space of all possible materials. In these calculations, each material is a point leading to a calculation with hundreds of thousands of dimensions. The calculation is performed by first annealing and then quenching the system. Upon quenching the materials, we look for trends of materials aggregating in similar regions. This process is called materials cartography.



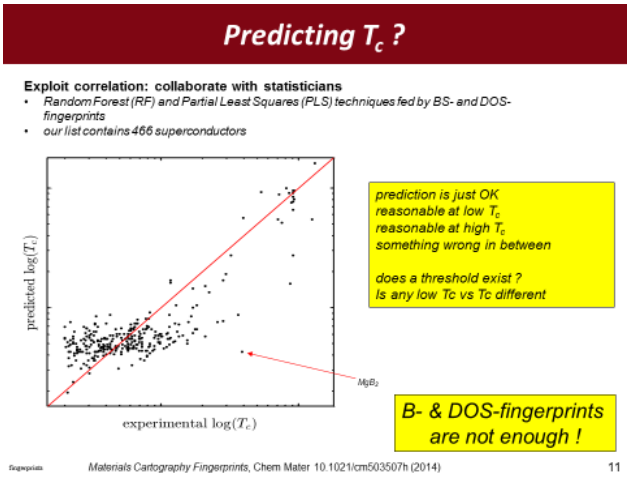
For example, in this plot that looks like a 'V' we see that the thickness of this 'V' is the diversity of electronic structures. It is wider with more diverse materials as a function of how many elements there are. This is intuitive, in that if there is only one element, there will be very small diversity, but if the number of species is increased, then the electronic structure gets more diverse. Since we can parameterize the angle, we know the diversity with respect to number of species. Depending on the materials, the calculation may have enough diversity with tertiaries and quaternaries might be unnecessary to the calculation.



Moving on to band structure cartography, we see that upon annealing all the materials in our library (approximately 50,000 at the time of this experiment) they quench and they create this picture resembling France. You can see that similar materials tend to aggregate or coagulate in similar regions. For instance, the circled portions on the chart display areas with metallic compounds with non-metallic atoms, ceramics, etc. More importantly, we can include experimental data and see if material properties are grouped together within material type.

This plot displays the experimental superconductors in our library, color-coded by critical temperature. We see that the good superconductors tend to aggregate in the circled region. The graph indicates that materials far away from the circled region are unlikely to have high critical temperatures. Whereas new materials inside the circled region have a favorable chance of high critical temperatures, and would be valuable experiments. This coagulation based on material properties gives us the potential to accelerate the discovery of new, better super-conductors.

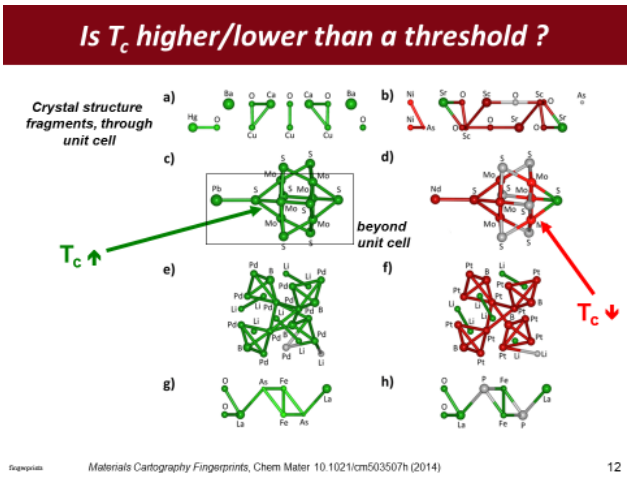
Of course there are always exceptions. Some materials are outliers, or orphans, and are different from all other materials. For example, magnesium-diboride is unique, and known to be dissimilar from other super-conductors. It is located in the lower right-hand portion of the chart, separate from all other materials.



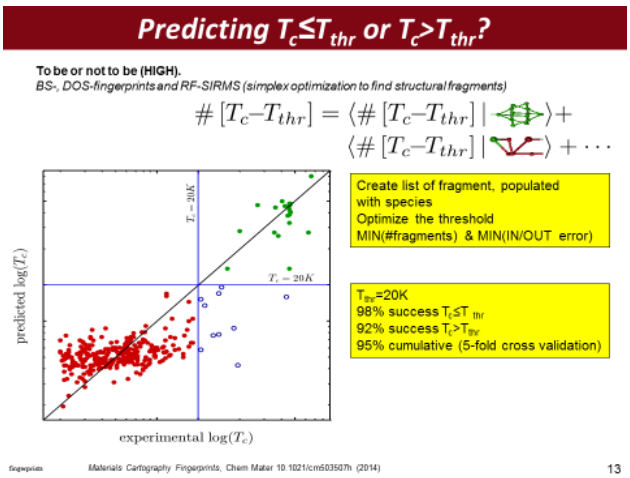
Could we exploit this correlation to predict critical temperatures? Here we have plotted the predicted critical temperature based on the plots on the previous slide. We have used a model based on random forest, and matched fingerprints to predict critical temperatures.

As can be seen in the graph, the model works well at low and high critical temperatures, but does not work in between. This suggests that the physical mechanisms of low critical temperature and high critical temperature must be different. If the physics remained constant, there would be no threshold between temperatures, and the ability to form predictions at low temperatures could be extrapolated to high temperatures, which is not the case.

Instead of using the model to predict an exact critical temperature, we use it to see if a material is likely to have a high or low critical temperature.



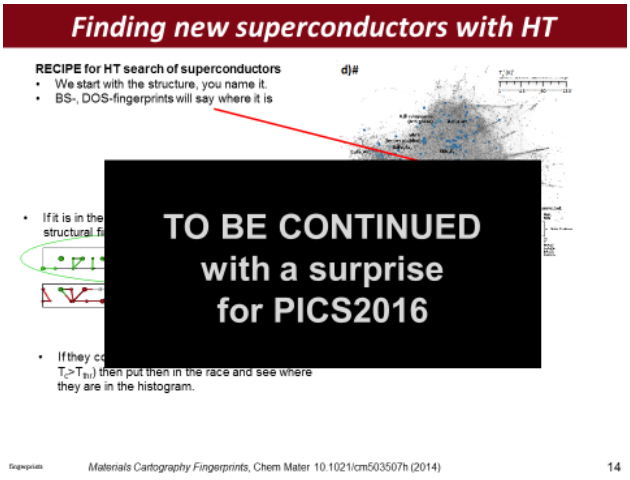
In addition, we can also add structural fingerprints such as connectivity. The colors here indicate the geometry is expected to increase or decrease the critical temperature. This kind of geometrical fingerprints are commonly used in the medical industry to make drugs with specific properties.



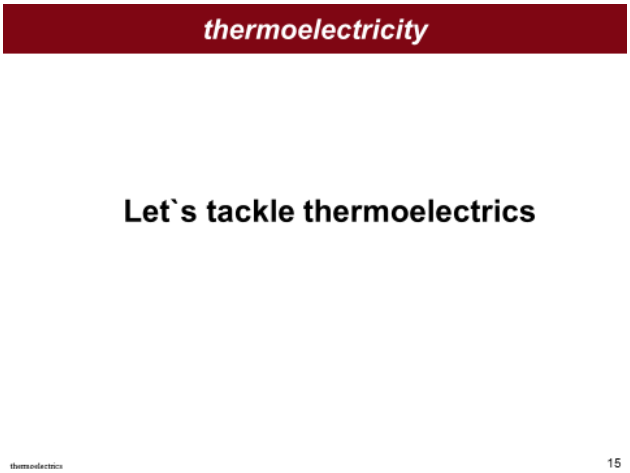
Assuming that there is a threshold between low and high critical temperature, we can calculate it by maximizing the correct predictions that we have below and above the threshold. We can move the threshold and minimize the number of false positives and true negatives until we find the correct value. Through this process, we found that the optimal threshold temperature is 20 K.

At 20 K, the rate of success in predicting if the critical temperature is below the threshold is 98%, and above the threshold it is 92%. Overall, this gives a success rate of 95%. What do these number mean? If an experimentalist

comes, and asks about a specific compound, we can simply find the band structure, taking half a day instead of two months of experiments, and know with 95% probability if the material is a good superconductor. We are not able to tell the exact critical temperature, but we can decide if it is worth testing.

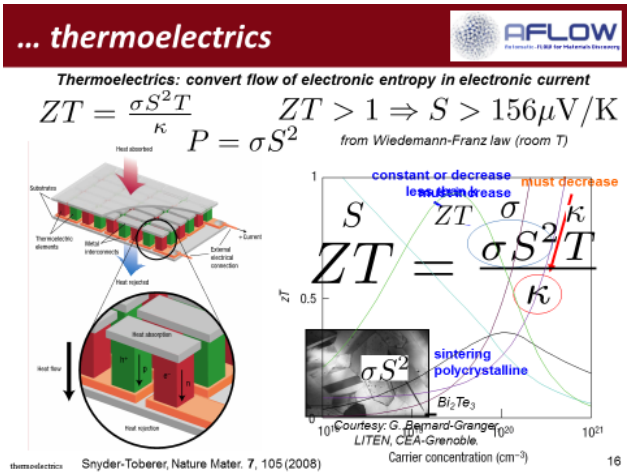


Our new process becomes the first find all of the compounds that are inside the circled region on the chart. Next we can add the geometric factors to the fingerprints, and the ones that have all green structural fingerprints. Finally, we can test these materials. The results from these experiments will be reported after all experiments are complete.



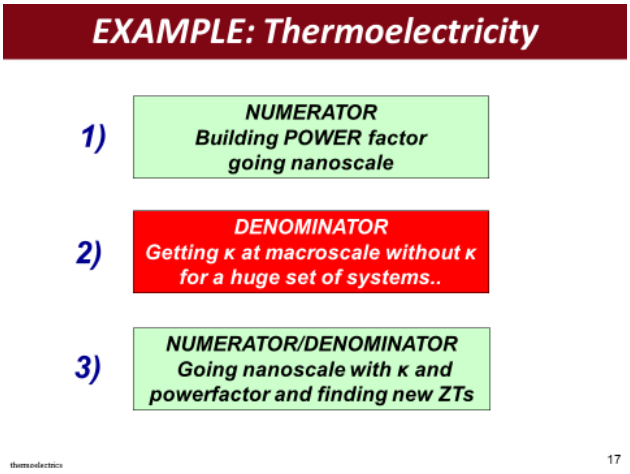
For a second example, instead of looking at accelerated fingerprints of materials, we can take a formula-based approach. This example will involve thermoelectric devices.

First, what is a thermoelectric device? A thermoelectric device converts the flow of electronic entropy into electronic current. Entropy is disorder so they create a flow between disorder and order. Disorder is imminent, and is everywhere. By the flow of disorder, you have a flow of current because each particle is charged. If you follow the calculations and models, you find that the thermoelectric figure of merit, which expresses the quality of the thermoelectric device, is the ratio on the top left.



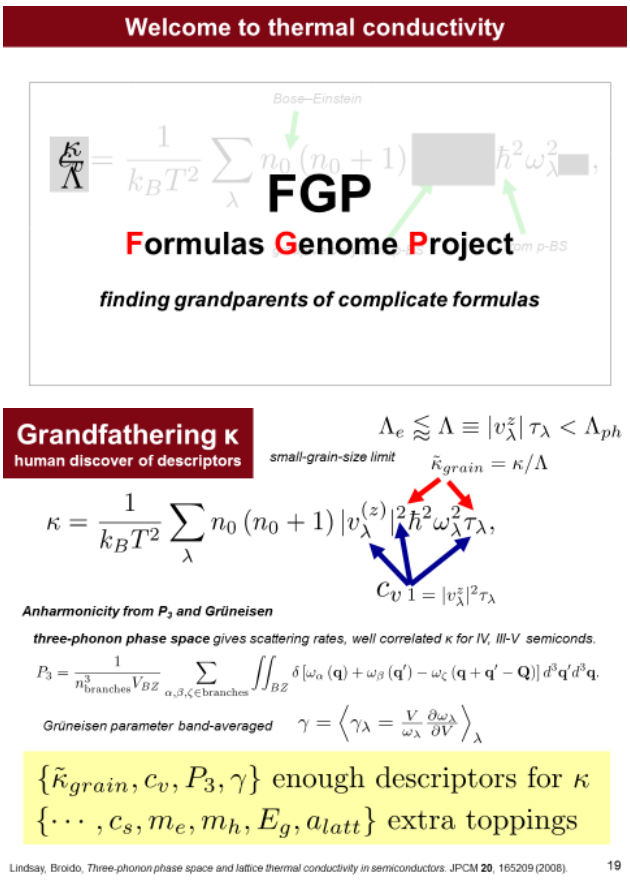
The figure of merit is the ratio between the power factor times the temperature and the heat conductivity. The power factor is relatively simple to calculate, but the heat conductivity is very difficult. We will discuss today how to approximate the figure of merit less expensively. The idea is that to find better

thermoelectric devices, we need to deal with the heat conductivity.



To deal with the heat conductivity, we need to find a way to get the conductivity without having to calculate it.

If we look at the definition of heat conductivity, it has the Stefan-Boltzman constant, the temperature, the sum over the phonon branches (the vibration degrees of freedom of the crystal) of the Bose-Einstein distribution, the group velocity (the gradient of the phonon band structure), the energy of the phonon, and tau. Tau represents how long two phonons can coexist before colliding, and the anharmonic force constants are necessary to calculate this value. We want to minimize the number of times we calculate conductivity because it takes a few weeks for each material.



We introduce the Formula Genome Project (FGP). We want to find the simple grandparents of complicated formulas. Instead of solving a formula that is too complicated, we find its DNA, or grandparent formulas, and calculate those instead.

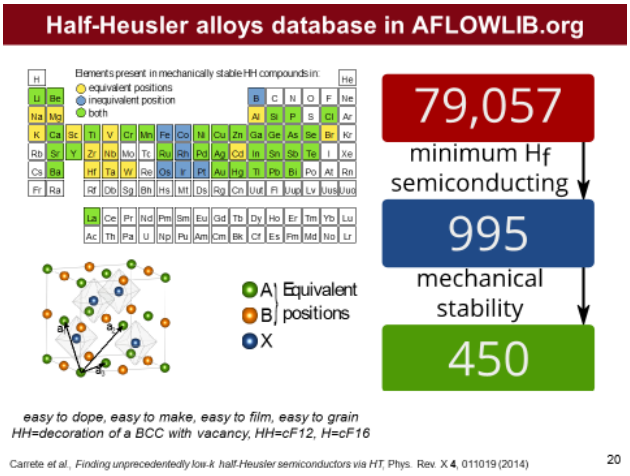
Coming back to the heat conductivity formula, we would like to take a razor and delete the parts that are difficult to calculate. For instance, tau is the lifetime of each phonon, and the group velocity is the speed. Speed times time is distance, so we delete the tau and the square of the velocity and instead calculate the heat conductivity of a single grain instead of overall heat conductivity of the material. Now, lambda is the size of a grain. In a generic sinterized system the grains are a few nanometers.

We can take the razor again and delete the speed. If we delete the speed, we get the specific heat at constant volume. We like the specific heat at constant volume because it is simple, but the specific heat at constant pressure generally more useful. To go from constant volume to constant pressure we need to add thermal expansion, but thermal expansion is related to the Gruneisen parameters, essentially the compressibility of the phonon spectrum. We now have

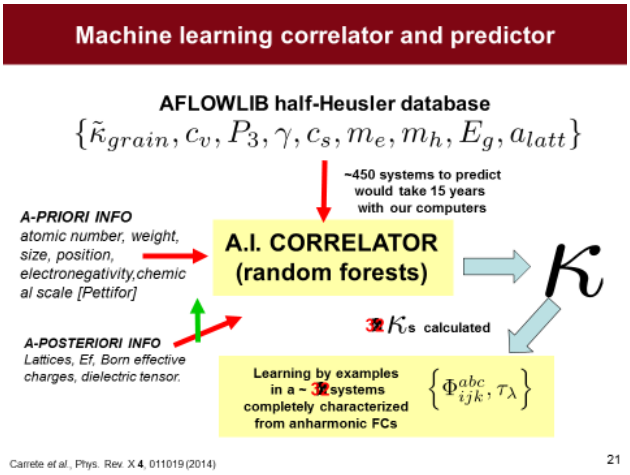
three of the grandparent equations for heat conductivity: conductivity per grain, specific heat at constant volume, and the Gruneisen parameters.

The fourth grandparent gives the scattering rate, how well the phonons can connect to each other. The three phonon phase space, P3, is nothing more than the integral in the Brilluoin zones of the conservation of momentum and conservation of energy.

My claim is that these four grandparents are enough to calculate, or to predict thermal conductivity. Perhaps there are other factors involved in the calculation such as the speed of sound, effective masses, gap, etc. However, from my work, I have seen that the four descriptors of the grandparent equations are enough.

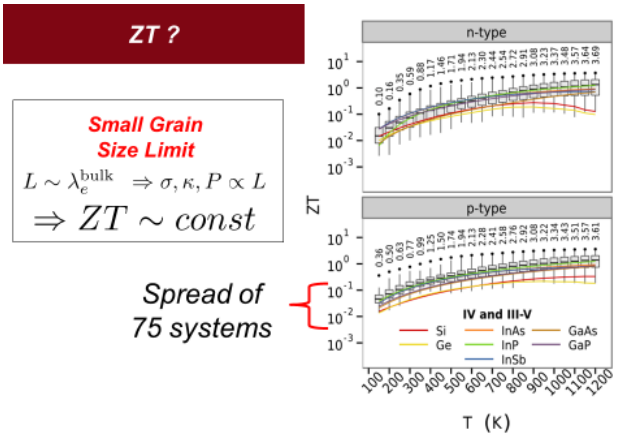


We take, for example, our database of half-Heusler alloys. The database contains a total of 80,000. Eliminating the alloys that are mechanically unstable or are not semiconductors, we are left with 450 alloys. We cannot calculate the heat conductivity of all 450 because it would take too much time.

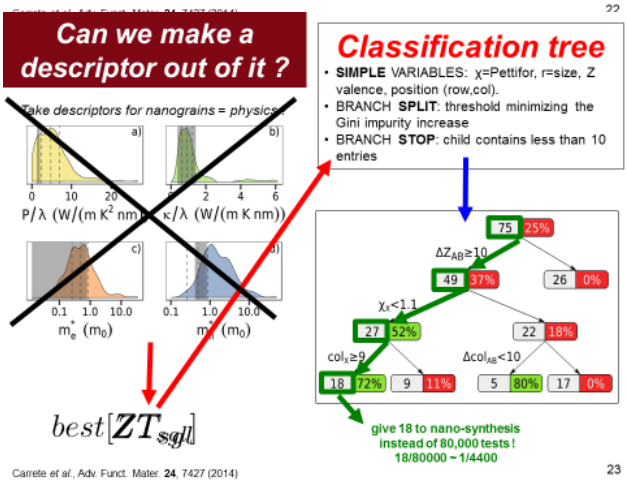


We want to come up with a minimum amount of alloys to calculate. How do we do this? You take one alloy out of the 450. We calculate the heat conductivity of this one alloy. We obtain the atomic force constants, solve the Boltzman Transport Equation, and get tau. From tau, we calculate the heat conductivity. We put this calculated value inside a random forest calculator and try to predict the function shape of thermal conductivity with respect to the four descriptors. Choosing a second material, we calculate the accuracy of the model, and repeat the calculation and model calibration if the error is greater than 1%. Eventually the cross-validation is within 1% of the calculations.

Through this process we have found that calculating 32 materials, provides a model that predicts the other 418 within 1%. The model was formed based on 31 random materials, and then the 32nd was used to verify the 1% accuracy. No matter which materials of the 450 are chosen, 32 calculations yields the best possible prediction for all of the other materials. The group calculation time was accelerated by a large factor.



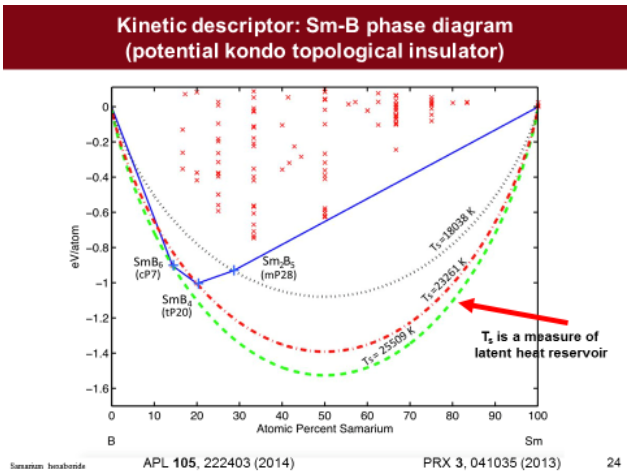
In addition to calculating thermal conductivity, we have also calculated sigma and P in several previous articles. With these three values we can calculate the figure of merit for all half-Heusler alloys, and then plot the figure of merit with respect to temperature for p and n types. Here we display the spread of the 75 systems of the 450 that were possible to make.



Even better, now that we have the answer we can reverse engineer the problem. We know which materials are the top 25% and try to find simple descriptors to locate these materials. We introduce four very simple descriptors electro-negativity (chi), the atom size (psi), balance or number of branches, and row and column in the periodic table. We make a decision tree based on these four simple descriptors, and identify a combination of rules to isolate the best materials.

If we start from 75 systems and eliminate the ones with a difference in balance of less than 10 between ‘a’ and ‘b’ atoms, we are left with the best 49. If we make a statement about the electro-negativity of the third element, the best 27 systems remain. Finally, if we make a statement about the position in the periodic table, only the best 18 are left. The best of the p or n channels could have been predicted with these four descriptors.

We started from 80,000 total materials and applied very simple rules about electro-negativity, size, position, etc. and end up with the best 18. This is a total acceleration of 4,400 times, and an example of accelerated materials design. As a next step, we can give these 18 materials to experimentalists. More detailed calculations are too complicated and would have required experimental validation.



As a final example, we will look at samarium boride. We tried to make samarium boride because it is a potential kondo topological insulator. Looking at the phase diagram there are several different kinds: samarium boride 66, samarium hexaboride, samarium tetra-boride, samarium di-boride, etc. However, experimentalists were consistently obtaining samarium hexaboride instead of other phases.

In order to describe this phenomenon we introduced a new descriptor, entropic temperature, essentially the curvature of free energy. The descriptor factors in the reservoir entropy so that we can predict the different compounds that will form coming from reservoirs of high or low entropy. High entropy (high temperature) and low entropy (low temperature) reservoirs have different kinetics mechanisms. Low temperature reservoirs will push a system with low formation energy. High temperature reservoirs will form a compound of higher formation energy because the entropic temperature measures the latent heat reservoir of a material. It quantifies the amount of entropy a material can absorb during solidification. Using this descriptor, we found that samarium hexaboride was the expected phase. This was confirmed by many experiments.

In summary, these three examples about materials, formulas, and descriptors, are examples accelerated materials development. In each case we needed scientific courage to go beyond our comfortable niche. For instance, I was born a thermodynamics analyst, but I had to become an electronic structure analyst. It takes doing a little bit of everything to not only get results, but useful results.

Doing these calculations requires an understanding of physics, because from physics you get descriptors. You need to have standardized databases written in a common vernacular. You must invest in methods to interrogate large amounts of data from multiple databases and get material properties. This requires a lot of framework development, which takes a long time, demanding a commitment from sponsors. It is not the quantity of money, but the stability of flow. The most important thing is to have fun.

I'll stop here. Thank you.

Integrated materials discovery engine

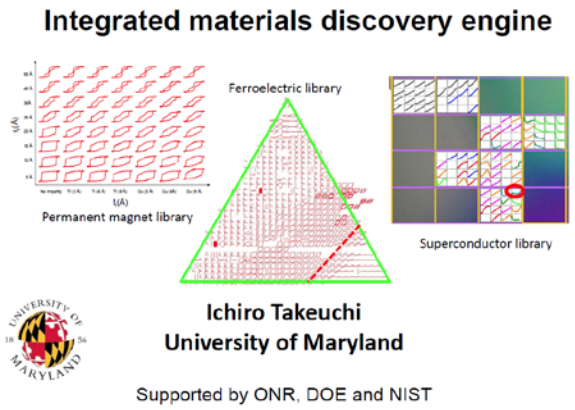
Dr. Ichiro Takeuchi
Department of Materials Science and Engineering, University of Maryland
Department of Physics, University of Maryland
College Park, MD

Abstract

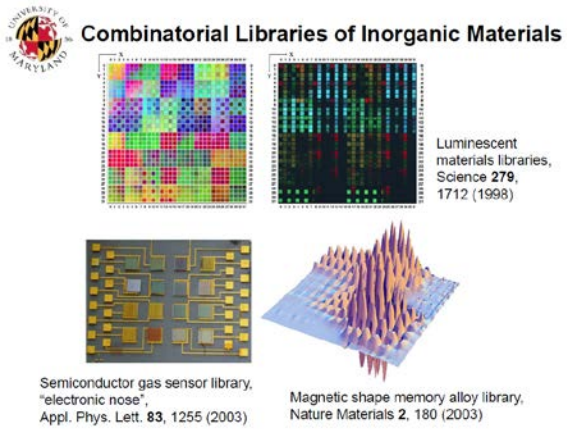
We are developing techniques to effectively integrate combinatorial experimentation with high-throughput computational approaches. Theory guided combinatorial experiments can be particularly effective, but data from combinatorial libraries can also be used to help guide computations. I will address the experimental side of this integrated approach. As a key component, techniques to rapidly analyze a large amount of data from combinatorial libraries and methods to interface experimental data with existing databases are being developed. Some data formats are particularly challenging because of their spectral or higher dimensional nature. We map the structural properties across libraries and quickly cross-reference them with known phases in crystallographic databases and AFLOWLIB, a computational database. Examples of materials topics include rare-earth free permanent magnets and superconductors. This work is carried out in collaboration with A. G. Kusne, S. Curtarolo, M. Nardelli, M. Fornari, and G. Hart. This work is funded by ONR and NIST.

Biography

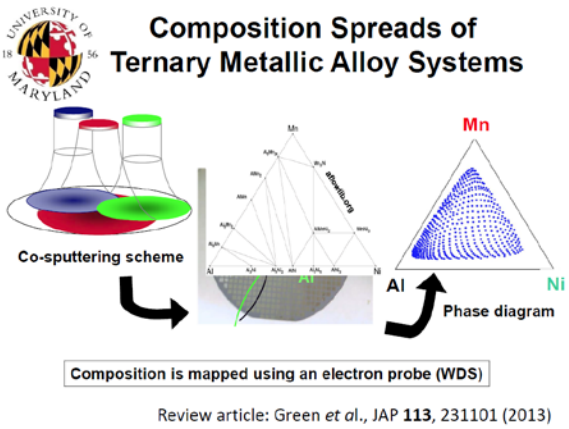
Ichiro Takeuchi is a Professor of Materials Science and Engineering and an Affiliate Professor of Physics at the University of Maryland. He obtained his PhD in Physics from the University of Maryland in 1996. He was a Postdoctoral Fellow at Lawrence Berkeley National Laboratory from 1996-1999. He joined the University of Maryland faculty in 1999. Takeuchi has served as a Visiting Professor at the University of Tokyo, Tokyo Institute of Technology, and Ruhr University, Bochum. He was a fellow of the Japan Science and Technology Agency from 2007 to 2008. Takeuchi won the Invention of the Year Award from the University of Maryland in 2010 for his invention of thermoelastic cooling, a novel alternative cooling technology. Takeuchi is a fellow of the American Physical Society.



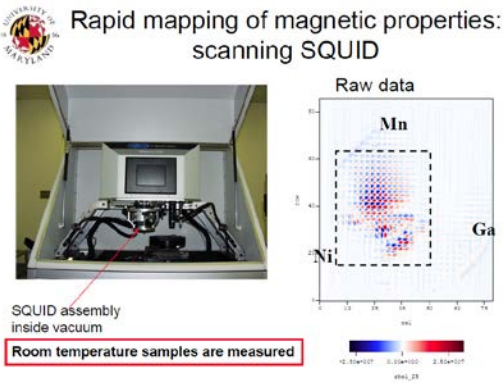
What I'm going to talk about today is how, with all the efforts in predicting materials, you still need to close the loop by actually making the materials. What we do in my lab is rapid syntheses, high-throughput techniques to make lots and lots of materials with all different compositions all at once. We do experiments like those shown on the slide: new superconductors, new ferroelectric materials, and new permanent magnets, etcetera. The whole idea, with the advent of this material genome initiative, is how to go about integrating experimental efforts with theoretical efforts. That's what I'm going to talk about today. Our efforts are funded by ONR, DOE, and NIST.

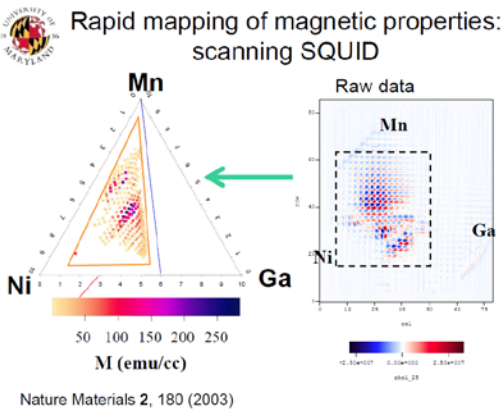


The whole idea for this high-throughput experimentation is that when you're out there working with materials and you decide you need to improve its physical properties by changing the composition, you will be much better off if you can make hundreds of thousands of samples all at once and rapidly screen them rather than if you did the traditional one-by-one Edisonian method. It's an idea that started in the pharmaceutical industry back in the late-80s, and then we adopted it in the material sciences. In the early days, we used to do a lot of luminescent materials because it's easy to do the initial screen. All you have to do is look at the chip and excite it, and then you see which types light back at you. Over the years we've moved on to more difficult topics, a few of which are displayed on the slide. Sensors are actually not too bad, because you just do the sensing of the arrays. We do a lot of magnetic materials, shape-memory alloys, and ferroelectrics. Nowadays everything is an energy related topic; we do fuel cell materials, battery materials, etcetera. We make arrays of materials and we try to rapidly screen them. This something we've been doing for about 25 years or so.

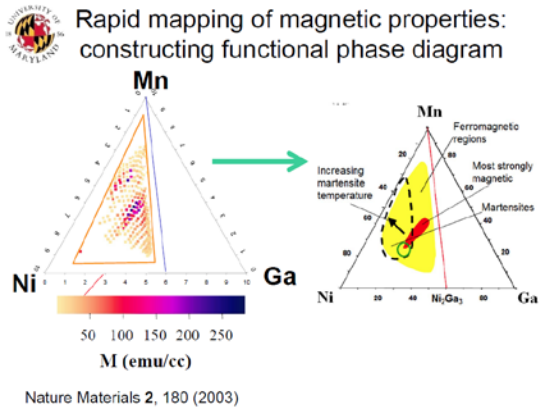


Let me walk you through how we do this kind of an experiment. We try to do as much as possible using think down techniques, because everything that exists in this plane, everything that exists in microelectronics, lends themselves to allowing us to make the libraries, characterizing them, and rapidly interrogating them, etcetera. We would make what we would call composition spreads. Blue, red, and green would be three different elements, and we want to study all the different combinations of those three elements all at once on a single wafer. We make our thin film samples in such a way that we deliberately create maximum composition variations. This is the opposite of usual microelectronics, where the goal is to try to make everything uniform on a single wafer. Here we want to make maximum composition variations. Then we will go ahead and map each composition from the wafer; we typically do this on a three inch one. We can map all the points from the three inch wafer onto a ternary phase diagram, and at this point we would start characterizing the properties and the distribution of different structures. Crystal structure information is definitely also entering the computational arenas as well, and is one of the most important pieces of information. If you can map the distribution of materials across the computational phase diagrams, what different structures exist and in which areas, your task is more than halfway done. These days, we talk about actively incorporating the results from Aflowlib, the computational database, so we can do rapid cross-referencing and rapid validation of our computer predicted results.



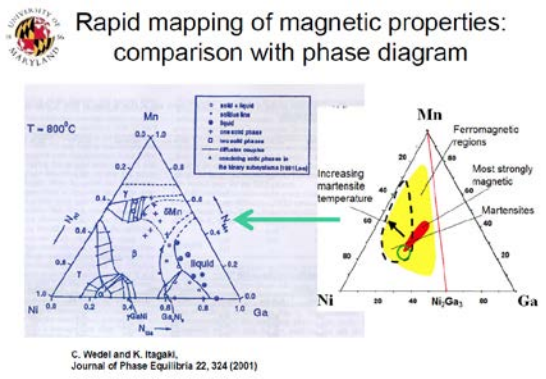


What this means is that we've had to develop many unique measurements tools that hadn't existed previously, because now we're talking about measuring a whole lot of samples all at once, each one of which is really tiny. It's a different mode of doing experiments. We've had to develop new methods of measurements over the years. This particular method on the slide, known as SQUID, gives us information about the particular composition on a wafer, such as whether or not a particular region is magnetic.



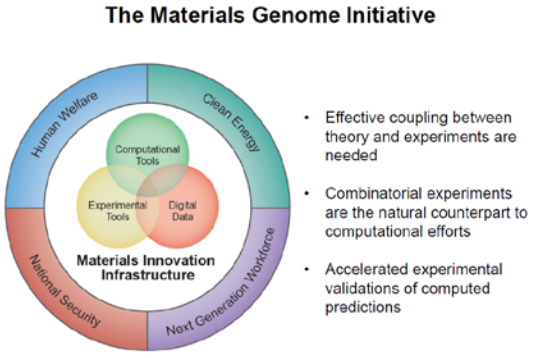
Nature Materials **2**, 180 (2003)

We do lots of computations so we can back calculate and get quantitative information. As seen on this diagram, magnetization M gives you the strength of the permanent magnet. We can combine this with other pieces of information. On a single wafer, we can map what we call the functional phase diagram. Here on the slide is a nickel, manganese, and gallium ternary.

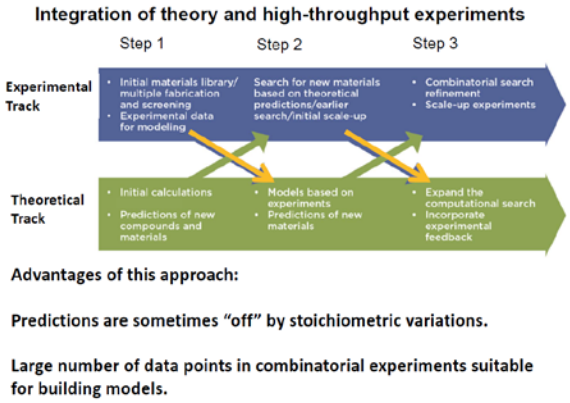


Nature Materials **2**, 180 (2003)

Sometimes we find out that there are actually computed phase diagrams, computed like a thermocal kind of things, together with a few experimental points. These are all individual bulk experiments that somebody had carried out, and constructed this high temperature phase diagram. Compared to that, on a single wafer, we can map the complete functional phase diagram. In this instance, we can say what was calculated to be the beta phase at 800 degrees at low temperature becomes this phase change material. This is the typical way in which we do these experiments.

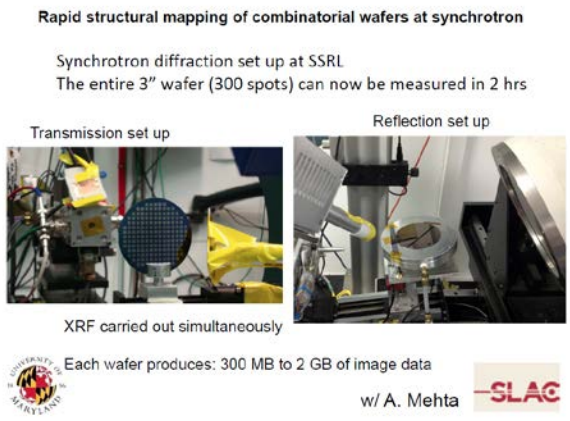


This here on the slide is the guiding light for us: the Materials Genome Initiative. We want to accelerate materials discovery and increase manufacturing so that we can create lots of jobs, etcetera. The key components are computational tools together with experimental tools and digital data. We've been staring at this for the last four years, so recently we decided to take this apart.



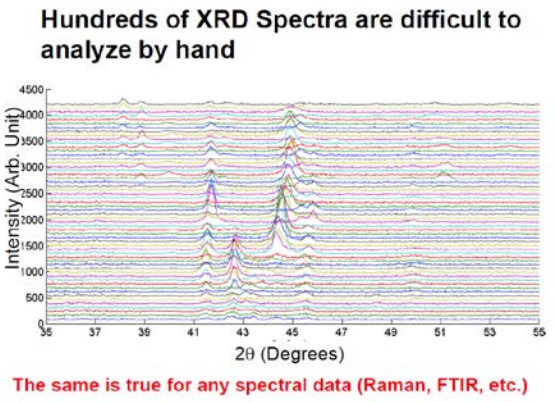
We had our tools of high-throughput experimentation, and we'd also heard about high-throughput computation, which can create a database like Aflowlib. What if we came up with a new way to do experimentation where you run the high-throughput experiment and the high-throughput computation in parallel so it leads to the goal that you want? This could accelerate by the speed in which you discover new materials at least by a factor of ten. The idea is that this whole platform lives in an environment where you're constantly sharing data back and forth between theory and experiment. In reality it's a little bit tricky. There's always meta-data issues. But this is something that we need to contend with because we would like to get this to work. The key is to have frequent feedback between theory and experiment. We call this platform the Integrated Materials Discovery Engine.

The advantage to this approach is pretty clear. The typical way in which experimentalists like myself used to work with theorists is we would talk to them, give them a call or they would call me, and the theorist would tell me: "Hey, I think I discovered, or predicted, a new thermoelectric material, can you see if you can validate this?" This would be a one-off experiment. And what happens is that the theory is often more or less correct; it has captured the essence of the phenomenon. But the exact composition is tricky. Thus, it makes sense to widen the region in which you're going to be doing the validation and the search. That's how we can expand the compositional phase space that you're going to be doing the exploration in, and that would be the combinatorial library. So it makes sense to do things in this way. The other way actually makes sense as well, because to build an accurate model, which computational material scientists do, the data is taking from experimental values, so the more data you have, the higher accuracy with which you can build your model. This is the whole point of the combinatorial experimentation: you get lots and lots of data.

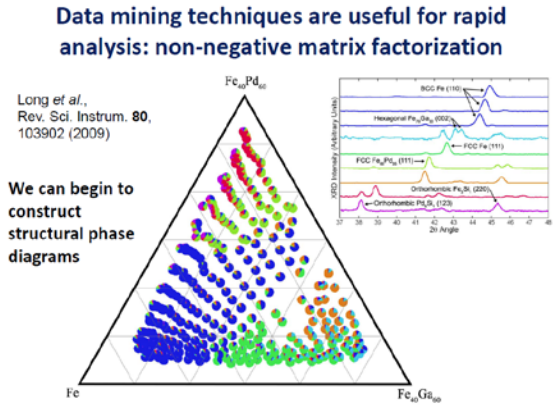


One experiment that is common to all of our topics, whether it is energy, materials, topological insulators, etcetera, is the need to do rapid x-ray diffraction experiments so that we can rapidly map structures across the phase diagrams. To this end, we've set up a couple of beam lines at SLAC , and it took us a long time but we finally set it up so that we can show up with our library wafers and within a day we can map five to six wafers. We're beginning to talk about a large amount of data. But right now, it's actually still not too bad: two gigabytes of

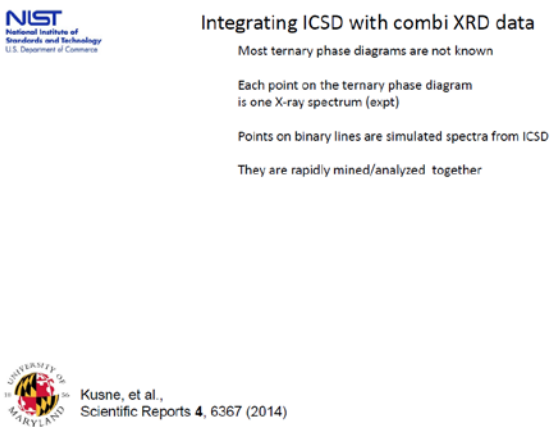
image data per wafer. After a week, we get tens of gigabytes. Another topic that often comes up in parallel with the kinds of things we talked about today is the rise of big data.



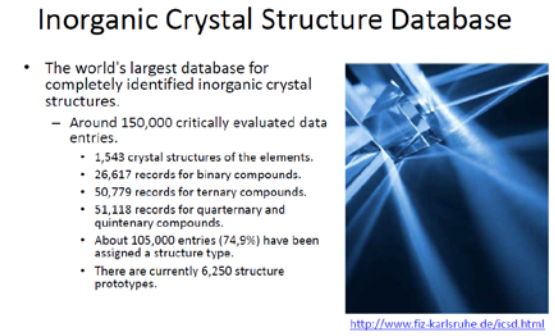
What do you do when you have a large amount of data? This is what we are doing. We take a lot of diffraction data, as seen on the slide. The goal here, in the orthodox material science, is to just look at one of them at a time, and then study each peak to death. Then you write a one PhD thesis. That's how material science used to be. But what we're doing here is different. Instead of one at a time, we take 500 diffraction patterns at once. We need to still fight this urge to study every single peak; you simply cannot do that, because you're talking about hundreds of diffraction patterns.



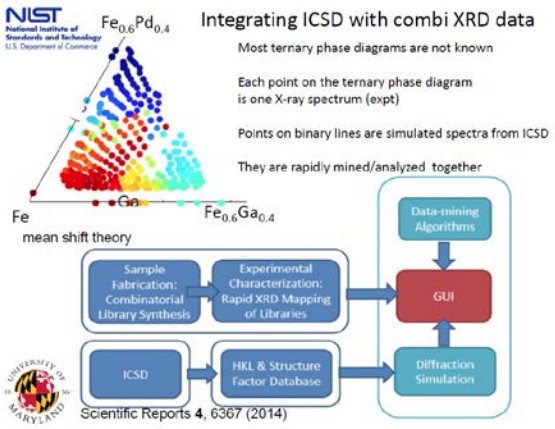
We started incorporating different data-mining techniques. We're actively working with people in the computer science department to do machine learning, machine vision, machine reading, and the like. These techniques have been around for a long time, but as applications to material science, especially experimental material science, it's still relatively new. So the idea is to take all the large amounts of data, let the machine learning techniques run its course, and come up with ways in which you can group data together. This clustering method is one way in which we do these things. We do a lot of this with my colleagues at NIST. For example, on this ternary chart, each point corresponds to each composition and its logistic diffraction pattern. By writing this particular machine learning technique, in this case non-negative matrix factorization, we were able to separate the details of roughly 200 diffraction patterns into five or six different patterns. At that point, it's not so bad to actually go back and look at and try to decipher each technique. This is the bread and butter of what we do.



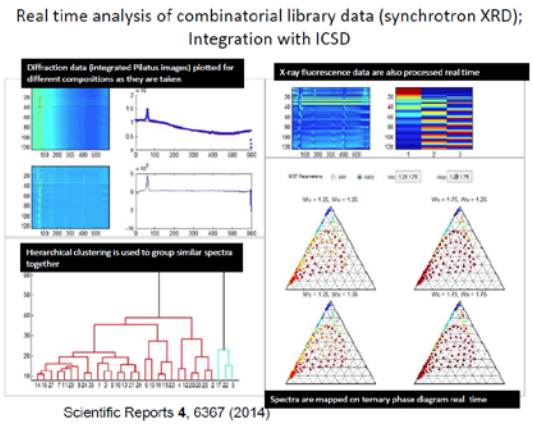
We started thinking that because we're doing this purely with experimental data, it would be interesting if we could use theoretical data or data that's already been vetted. The Inorganic Crystal Structure Database (ICSD) is the one of the largest materials database of studied materials, with complete information about lattice constants and many other material properties. The materials that we are trying to discover and study have structures that are close to the materials in the database that are known already.

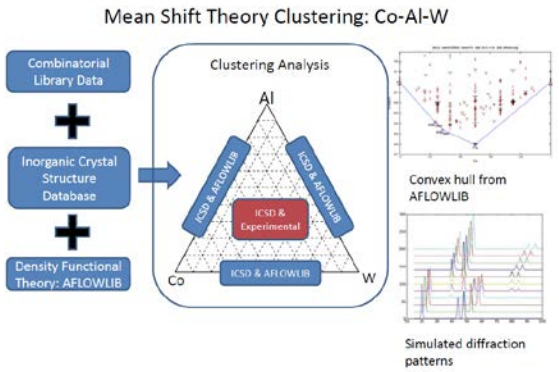


We asked ourselves: "When we run this machine learning techniques algorithm, why don't we mix the experimental data together with the vetted data taken from ICSD?" And that's what we're doing; we've written our GUI, and this works really well.

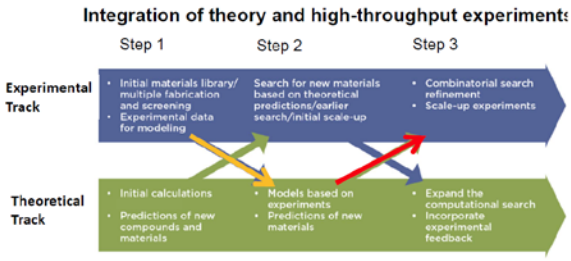


The next thing we decided is that since we're already taking information from a database, why don't we take data from a predicted, computed result like Aflowlib? In this way, not only are we doing the clustering and the rapid identification of structural phrases across the phase diagrams, we can actually do rapid validation of computed results all at the same time.





That's one of the next steps that we're beginning to do: mixing experimental data together with data from the ICSD and the computational tools.



Example:
Synthesis of SmB_6 topological Kondo insulator

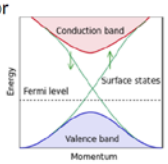
To finish up, let me give you an example of how this Integrated Materials Discovery Engine flows works.

Combinatorial Growth and the Robust Topological Surface State of Kondo insulator SmB_6 Thin Films

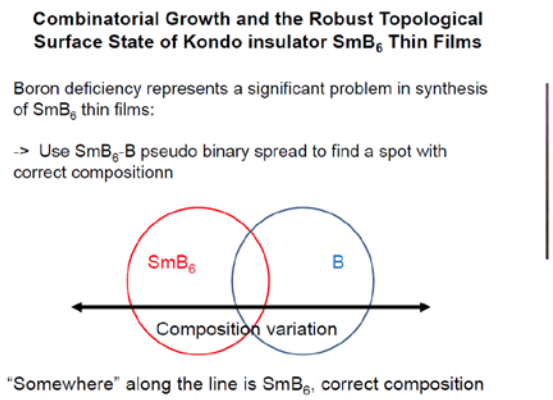
Topological insulators: insulating bulk but protected conducting surfaces due to spin-orbit interactions and time-reversal symmetry

There has been single crystals of SmB_6, but to date no report of thin film

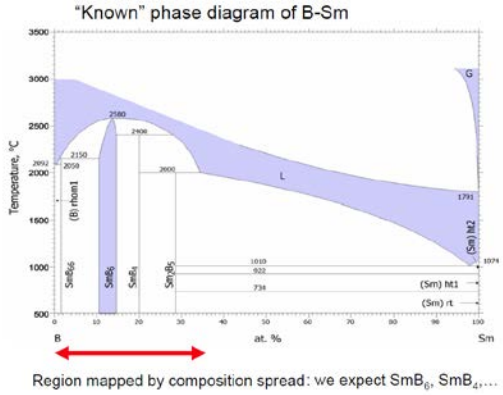
Thin films with smooth surfaces are needed for devices and investigation of surface states



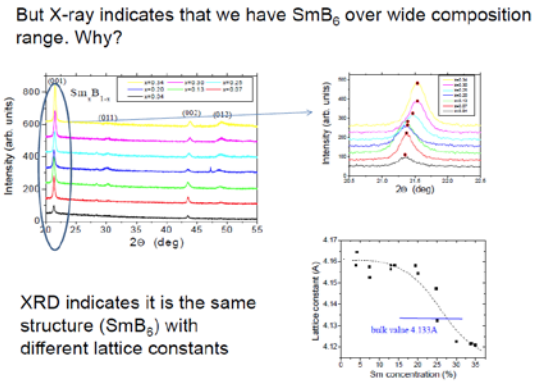
Topological insulators are really interesting, exotic materials, and this is all the rage in condensed matter and physics. Everyone is studying topological insulators. It could lead to informed quantum devices or the basis of a quantum computer. It turns out materials are really tricky. Can you even make a good topological insulator? One material that we looked at that's predicted to be a good topological insulator is Samarium hexaboride.



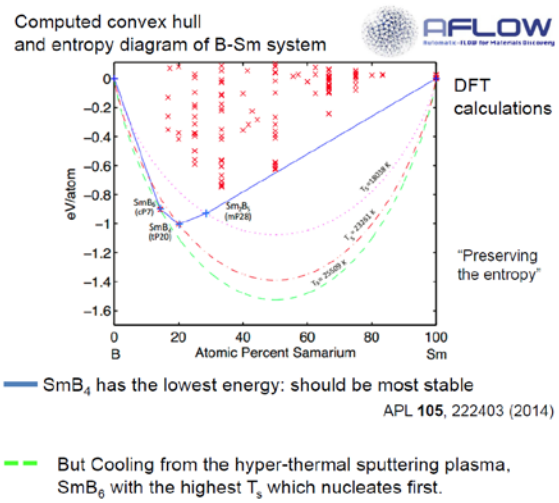
This particular combinatorial experiment was kind of backwards in terms of the goal because we already knew the composition that we were looking for ahead of time. But the challenge was that to actually make the synthesis of the material can be really difficult. Here, the difficulty arises from the fact that the boron has a completely different kind of vapor pressure compared to that of samarium. You end up mostly with a severe boron deficiency. To remedy this, we set up our experiment in such a way so that somewhere on the wafer there's going to be a composition with the correct stoichiometry. Again, we know the composition in this case already; we're looking for the optimum position on the wafer.



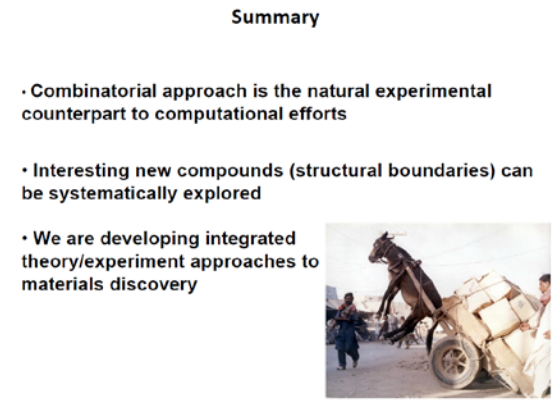
This is the phase diagram of Boron-Samarium. There are a couple known phases already; this one here on the left of the diagram is the one we want to make, and we were able to make it. This region towards the bottom left is the one that we mapped on the composition spread, so whatever results we get should mirror this known phase diagram. We expected different phases of SmB_6, SmB_4.



But what was interesting was that no matter how many times we'd make this, we'd only end up with one structural phase across the entire phase diagram. This didn't make sense. They said there's something wrong with this equilibrium phase diagram. So what's going on? This is when we exchange insights with Stefano's computations.



We went ahead and calculated the convex hull, which gives you information about the relative stability of the phases, which is the most important property; if the phase is not stable, you cannot make it. What you want to look for in a curve like the one on the slide is the lowest point, which in this case is Samarium tetraboride. That is the most stable phase. So what this says, in contrary to our experiment, is that this is the phase of the structure that we should see everywhere across the composition spread. But instead, we saw Samarium hexaboride. So what is going on? That's just looking at the blue curve, the relative stability, the formation energy, which is, arguably, the most important parameter. But there is this other parameter, represented by the green curve, that's called the entropic temperature curve. It turns out that this parameter could be equally as important as the formation energy; it depends on through which process you're synthesizing the material. In our particular case we get this sputtering, which starts off with this hyperthermal plasma, tens of thousands of degrees in temperature. To make the films, you're cooling down from that temperature to the room temperature, and in doing that, it turns out that the really important parameter is this entropic temperature, which is a measure of the material's observed entropy competing with the formation energy. It's coming from tens of thousands of degrees, and you're cooling it down, quenching it. Evidently, if there's a compound with a higher entropic temperature, that's the one that will dominate the formation. If you look along this green line, it turns out hexaboride indeed sits on the curve that's at a higher entropic temperature than the red curve. There are two opposite trends. In terms of formation energy, it's the tetraboride. But looking at the entropic temperature, it's the hexaboride. This is why we found across the composition spread the structure that dominates the formation is Samarium hexaboride. Now we're using this information to study other material that have similar compositions. We're looking for new topological insulators and we're looking for new superconductors.



In summary, I wanted to give you a snapshot of the type of things you have to do. You have to do all the computations and you still need to do rapid validation. This high-throughput experimental technique is the natural counterpart to the computational efforts, and we are actively developing ways to integrate theory and experiment: a challenging task.

Thank you

Evolution of ALE3D, challenges, enabled science, return on investment, and a vision for a path forward

Dr. Rose McCallen
Lawrence Livermore National Laboratory
Livermore, CA
University of California Davis
Davis, CA

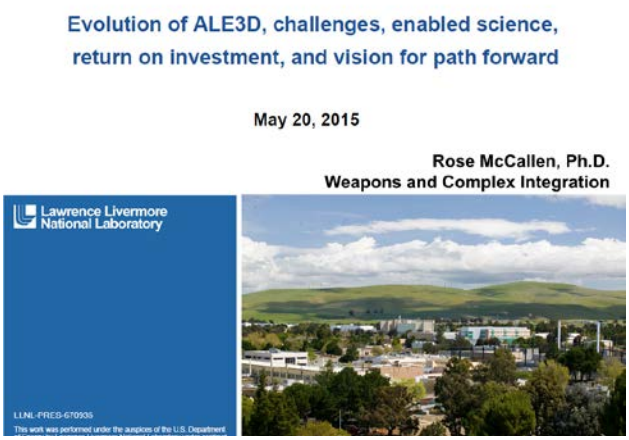
Abstract

As a Department of Energy (DOE) National Nuclear Security Administration (NNSA) Laboratory, the primary mission of Lawrence Livermore National Laboratory (LLNL) is National Security, as well as supporting the DOE's energy and environmental missions. In support of their mission objectives, the DOE Laboratories have been at the forefront of scientific computing and this has taken them to the cutting edge of computational multi-physics modeling. Dr. McCallen will present an overview of the evolution, challenges, enabled science, and return on investment for LLNL's Arbitrary Lagrangian-Eulerian (ALE) multi-physics simulation code called ALE3D. The history starts with the codes inception in 1989, through Dr. McCallen's years as a developer on the team in the early 90's, and her leadership over the last decade in the development of one of the nation's recognized codes for problems at extreme ranges of solid/fluid behavior. The challenges of a diverse funding portfolio and large user base will be discussed, as well as the requirements and retention of a multi-discipline high performing team. Included is a brief discussion on specific material and chemical explosive modeling challenges. The benefit of ALE methods for complex problems is the achievement of solution accuracy for problems with extreme displacements and deformations. One challenge is that material failure and fracture models are typically developed for Lagrange simulations and the corresponding models are computationally mesh dependent. In addition, Chemical reactive flow models are computationally intensive and approximate methodologies with demonstrable accuracy are necessary for complex applications. Dr. McCallen will also share a vision and outline an approach for a path forward to address the goal of achieving and maintaining wide adoption of cutting-edge science and engineering software and codes.

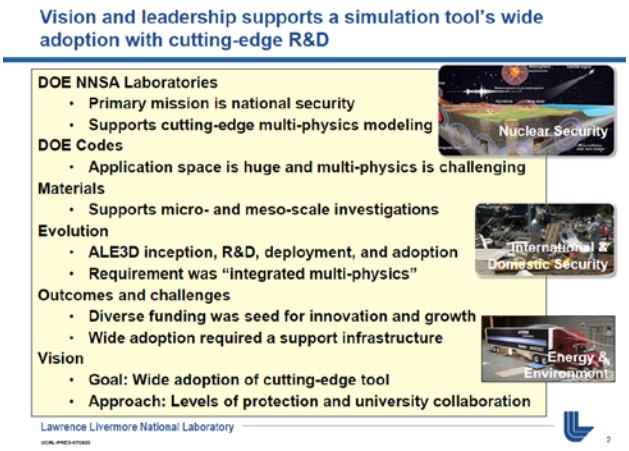
This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.
LLNL-ABS-670213

Biography

Dr. Rose McCallen is currently on sabbatical at UC Davis as a Research Associate and returning to Lawrence Livermore National Laboratory (LLNL) this fall. In her 30-year career at LLNL, Dr. McCallen has held several leadership and technical R&D positions. For the past 7 years, Dr. McCallen led the ALE3D Code Research and Development Team, successfully doubling the user base and budget to over $10 million with expanded interest and support from DOE, DoD, DHS, and NASA, as well as their industry contractors. She was also the Lead on the Rocket Motor Warhead Impact Modeling Project for the DoD and the Multiphase Reactive Flow Subgroup Lead for the Joint US/UK Insensitive Munitions Project Agreement. She organized and led a multi-lab project investigating the terrorist threat to subways for the DHS, and the DOE multi-lab, multi-university consortium for the reduction of heavy vehicle aerodynamic drag. She has also held administrative roles as the Lead for the Applied Mathematics Group in the Center for Applied Scientific Computing. Dr. McCallen's research interests include a range of multi-physics computational applications. She received her Ph.D. in 1993 and a MS in 1984 in Mechanical Engineering (ME) from the UC Davis, a BS in ME from CSU Chico in 1980, and a BS in Mathematics from Saint Mary's College in 1977. She received a 2009 DOE NNSA Recognition of Excellence Award, 2004 LLNL Computation's Leader Recognition Award, 2000 LLNL Engineering's Emerging Leader Recognition Award, and 1998 DOE/NNSA Weapons Recognition of Excellence Award. In 2010, Dr. McCallen completed a leadership program at the UC Berkeley Hass Business School.

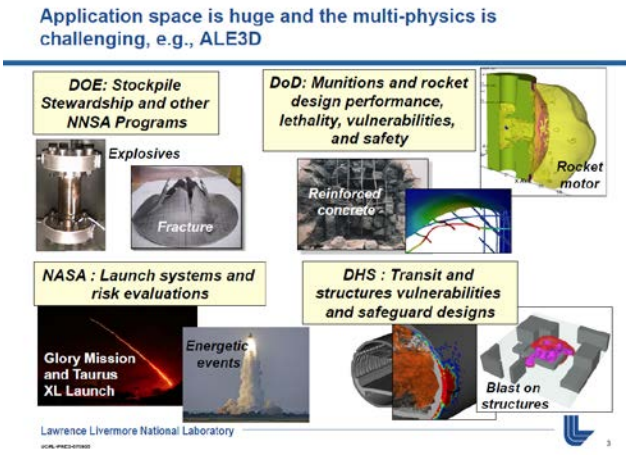


The title of this talk is "Evolution of ALE3D." I will be describing the challenges, enabled science, return on investment, and vision for path forward.

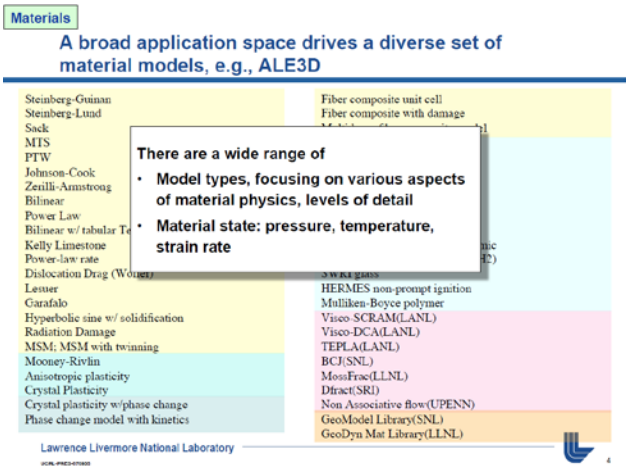


This is the outline of the talk. I want to start with the mission of the Department of Energy National Nuclear Security Administration Laboratory. The mission of the DOE NNSA is national security, and that includes nuclear security and also international and domestic security. Part of that mission is energy and environment. First of all, nuclear security: that picture shows that a blast has gone off. It's in the air, it's on the ground, or something has happened. Here we have some type of seismic event, and what the Department of Energy wants to know is what happened. Was it something that naturally occurred, was it a terrorist event? What was it? So that's what we mean by national security.

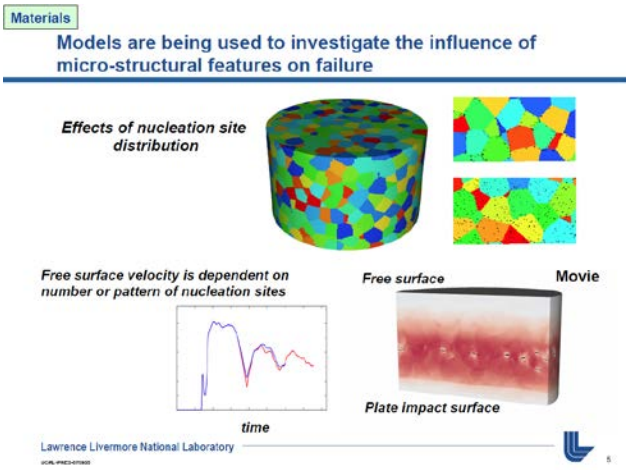
When we talk about international and domestic security, that's a picture of a hazmat team and something has happened, and we want to know if it's a biological release or if it is a blast that's gone off. Is it the result of a terrorist event? Is it something that's naturally occurring? Did an earthquake happen? In energy and environment, this is actually a full sized truck in the NASA Ames wind tunnel. That's their 80 foot by 120 foot wind tunnel, and I used to work on truck aerodynamics. People in the national laboratory work on all kinds of things. So, with all of these applications, we support cutting-edge multi-physics modeling. That's all I'm going to talk about regarding the general mission of the DOE and the NNSA labs. What I'm going to do now is show what the application space is. It's huge and the multi-physics is challenging. I know most of you are material scientists, I'm a mechanical engineer, but I'm going to show how this supports micro- and meso-scale modeling. Then I'm going to get into the heart of it, and talk about the evolution of ALE-3D, and that one of the major requirements is integrated multi-physics. Then I will address some outcomes and challenges with an addition in direction where we looked at diverse funding, which ended up being a seed for innovation and growth. Then, I will discuss how we accomplished wide adoption with the code using a support infrastructure. Finally, I want to share a vision for the future: wide adoption of a cutting edge tool. The basic bottom line approach for that goal is levels of protection and university collaboration. The bottom line is that I believe vision and leadership support a simulation tool's wide adoption with cutting-edge R&D. Hopefully the presentation will guide you to that or a similar conclusion as well.



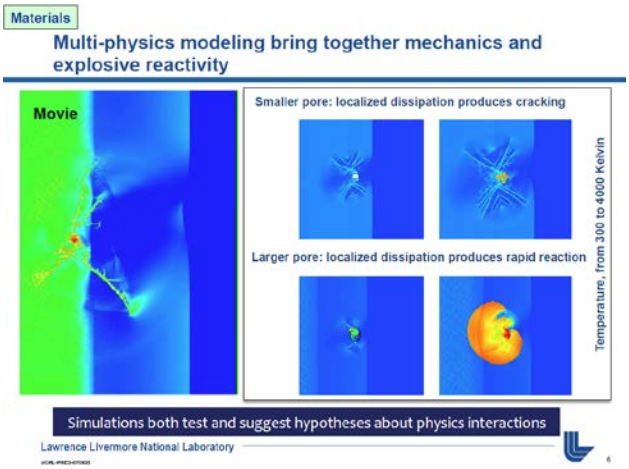
Let's talk about the application space. I'm going to go clockwise here. Again, here my example is ALE3D. ALE stands for Arbitrary Lagrangian-Eulerian. The Department of Energy cares about stockpile stewardship and other NNSA programs. One of those things is explosives. In that metal container is explosive, and we heat it up over long periods of time, and we want to see how energetic it is and when it explodes. Fracture and failure is very important to the Department of Energy. For the Department of Defense, the code is used for munitions and rocket motor design performance, lethality, vulnerabilities, and safety. For example, with this rocket motor, this little blue dot is a projectile hitting the rocket motor and exploding. You don't want a rocket motor to explode when a projectile hits it. For reinforced concrete, if you set off an explosion, you want the munition to perform such that it cuts through the rebar, because you won't have access if you just tear off the concrete. For the department of homeland security, you care about transit and structure vulnerabilities and safeguards. Here is an internal explosion. These codes have to handle an internal explosion and you want to know when the structure fails, shown here in red. You also want to know about blast in the open atmosphere – this would represent buildings. Finally, we also support NASA. That's for launch system and risk evaluations. NASA Kennedy Space Center had quite a few failures because the last stage did not separate, so that was one thing we helped them with. We tried to understand what happened and what the requirements should be. We also collaborate with NASA Ames looking at energetic events. They want to make sure the platforms and personnel are protected in the case of an energetic event. So that gives you an idea of the wide application space.



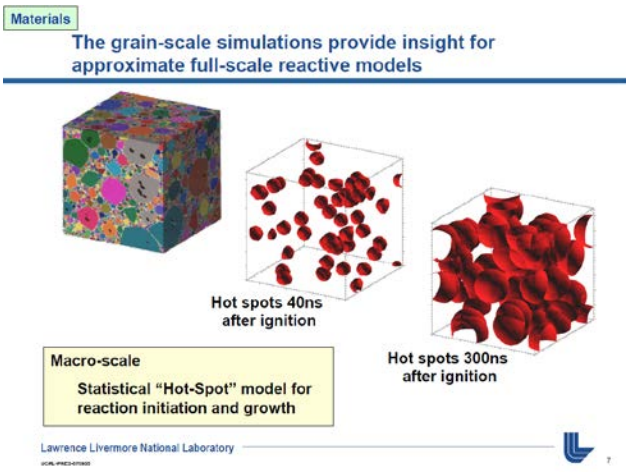
Let me switch and talk a little about materials. I borrowed these slides from Nathan Barton of Lawrence Livermore Lab. There's a huge amount of materials models in these codes, at least 50-100. We have a wide range because we need different model types focusing on material physics, levels of detail, and you want to be able to capture the material state (temperature, pressure, strain rate).



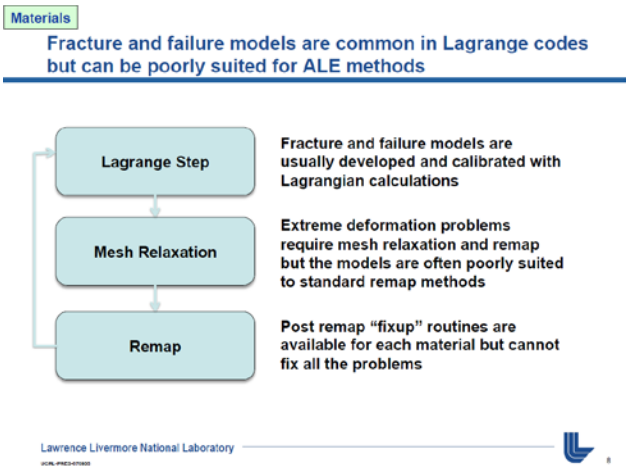
I have a few slides that I'm going to go quickly through. Models are being used to investigate the influence of micro-structural features. For example, here is a grain-scale model. Nathan is looking at the effect of nucleation sites, whether there's a few nucleation sites or a lot of them, or if they're placed on the grain boundaries. What he does is look at plate impact on the bottom of the cylindrical shape, and he follows the motion of the free surface and plots the motion of the free surface based on the number of nucleation sites. That's one application.



Very important to us is the behavior of explosives. We want to understand the ignition of explosives, so we do studies on explosive reactivity. This movie shows a pore in an explosive being hit by a shock moving from left to right. The pore collapses. What we find is that if we have small pores, you get localized dissipation and this fracture pattern, but if the pore is really big, it quickly reacts. This is used to test and suggest hypotheses. What we want to do is develop continuum models.

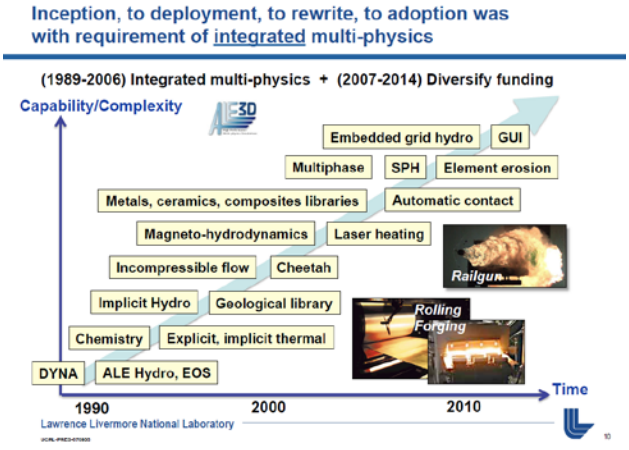


We use these grain-scale simulations to develop these full-scale models. This is an example of running many simulations over tens to hundreds of nanoseconds to develop statistical hot-spot modeling.



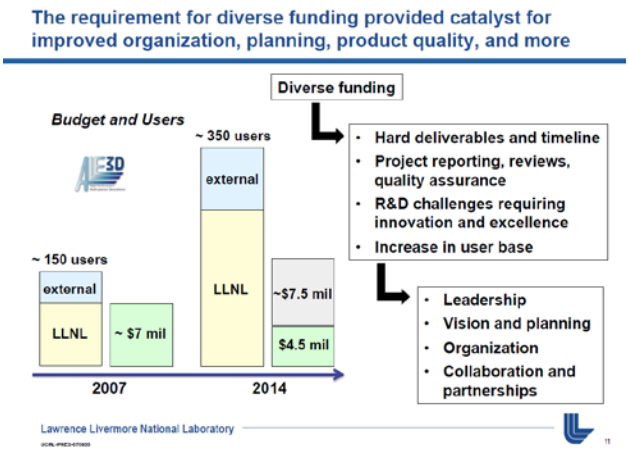
Here's a challenge. Fracture and failure models are common in Lagrange codes. We are an Arbitrary Lagrangian-Eulerian code. What that means is that we do a Lagrange step, we relax the mesh, and we remap the results. That is a challenge because these Lagrange material models have been calibrated with Lagrange codes, so we have to fix them and that continues to be a challenge, to fix them for our ALE codes.

I've told you about the DOE and what NNSA labs are. I've told you a little bit about the application space and how the materials modeling people are using this. Now, I want to jump into the evolution of ALE3D.



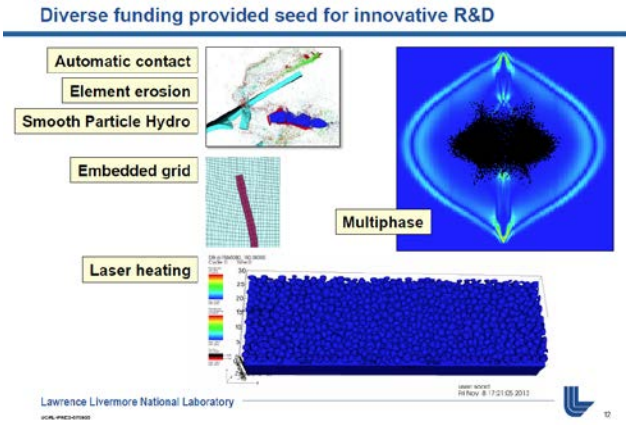
From inception to deployment to rewrite to adoption, there was a major requirement. That requirement was that it was an integrated code. This is a plot of capability versus time. You might also see it as complexity versus time. In 1989, ALE3D started as the DYNA code. The DYNA code was a pure Lagrange code, added to it was the Lagrange with mesh relaxation with remap. The developer, Richard Sharp was his name, and the leader of the effort, there were about 3 or 4 people on the team, cared mostly about ALE techniques and the equation of state. At that time, somebody mentioned earlier that CRADAs was a real push. Well ALCOA came to us and said can you do rolling and forging? That was a totally different problem. In 1993, I joined the team and added an explicit/implicit thermal capability. Then along came somebody else and added implicit solid mechanics. By the way, the chemistry guy liked that thermal was in there and wanted to do explosives so he added chemistry in. Later on, I was doing truck aerodynamics and I convinced Richard Sharp that we should put in an incompressible model. Then we started to do railgun work

so we needed to add an MHD model. You can see there wasn't a whole lot of planning, right? It was just that people came along and he was okay with that, as long as it was integrated. For example, the incompressible model had to have a Lagrange interface to the implicit solid mechanics. Then 2007 came along, and I became lead on the code. At that time, I believe management thought, "Why are we paying so much for this code? Let's get others to pay." Diversification of funding became a push and that was my job. At that time, things changed a little bit, I had to do planning. Let me jump to the next slide, we can come back to this if anyone has questions.



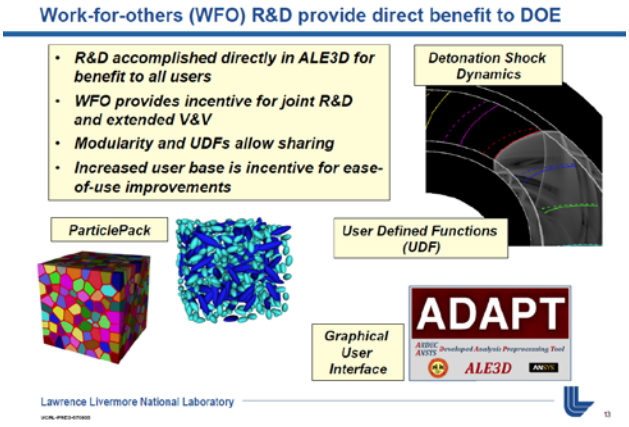
So the requirement for diverse funding, I believe, provided a catalyst for improved organization, planning, product quality, and more. So the plot over here on the left gives you an idea of what happened from 2007 until 2014. In 2007, I shadowed the number of LLNL users in yellow. We didn't track it that well, but there were about 100 users at most. There were about 50 external users, and all the money came from the ASC DOE pot of money. There might have been little bits and pieces from elsewhere, but not a whole heck of a lot. So, I did what I was told: I dropped… Well I didn't drop, it was taken away, this funding gradually declined and I fought as hard as I could to get it to increase, and had to bring in other money. There was some DOE money in there, DOD, and other pots of money. Look at the LLNL user base; it jumped. It's about 250 right now, we think. There could even be more users. We think, externally, there are about a hundred users. There may be more and I may be underestimating that number, but it's at least 350.

Diverse funding led to hard deliverables and timelines. When people pay you for it, they want their deliverables. You had to do project reporting, reviews, and quality assurance. Then, there were all these new, innovative things we had to do and we had to be excellent at it in order to get more funding. This user base just took off. So there had to be leadership, there had to be vision and planning; you couldn't just add in whatever you wanted. There had to be organization, collaboration, and partnerships.



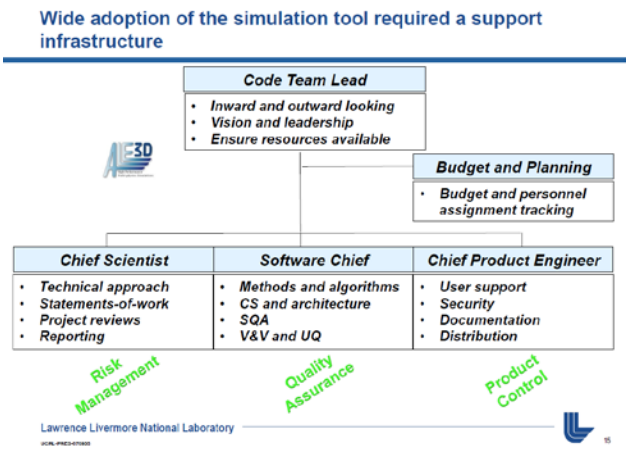
Let me just give you an example of some of the things that happened. We met with the Department of Defense and they said automatic contact is our first priority, automatic contact of materials, and then came element erosion, and then came smooth particle hydrodynamics (SPH). An example of this is you have a plate and a projectile hitting it. You can see those dots. There is element death and then an SPH particle is born and you have to capture all this fracture. Then, the other thing was, we'd like to generate these grids a little more easily and do more complicated geometry, then along came these embedded grids where you have a solid body that's Lagrange and an ALE background, say with a blast. You end up with cut cells. We had to have this very sophisticated interface method. People wanted us to look at explosives with particulates. We started by treating the particulates as a continuum in a multiphase modeling capability. Since then, we've added Lagrange particles to multiphase.

I have a movie here: laser heating. Laser heating started with our national ignition facility, where sometimes our optics have flaws and we want to heal the optic by heating it with a laser, so that's how it started. Then people came to us for additive manufacturing. This is a particle bed for additive manufacturing where you want to heat it, melt it, and re-solidify it. Guess what? You can use that same capability for laser damage. So, we had multiple customers.

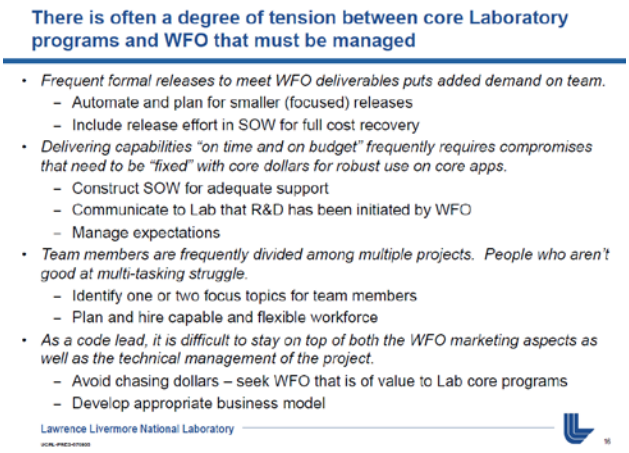


We call it work for others; I like to call it work with others. I believe it provided a direct benefit to the Department of Energy. We accomplished all of this work in a DOE code directly in ALE3D. It was incentive for joint R&D, and people keep talking about when you have so many users, you get extended V&V, and I think that's true. The other thing is that there was a push for modularity in sharing packages as well as user defined functions. A lot of you are familiar with UMATs. There was a huge push for material models to be user defined functions, as well as our package for detonation shock dynamics to be modular. We have a separate capability called ParticlePack where you want to pack shapes into a confined space or you want grain-scale where you have, say, explosive binder and you want to control that. The other thing, which was a big deal, was ease of use. So somebody said this earlier, where the Department of Energy said we can be tough and we don't need a fancy GUI. The Army said we do want a GUI, and they teamed with ANSYS. This was without our intervention, although we were supportiive. There is now a GUI for ALE3D.

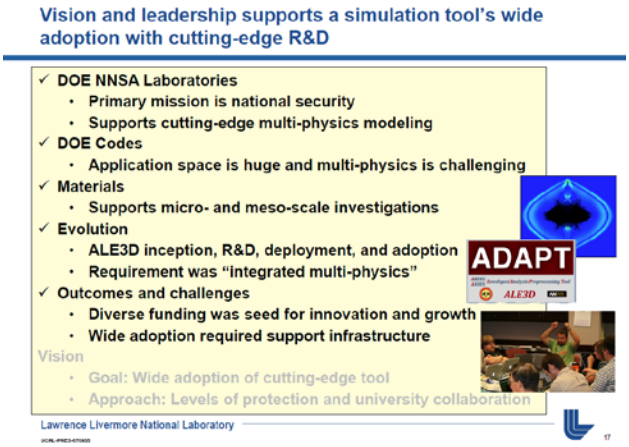
This I think is really fun. We are in the Bay Area. We have to compete with the .com industry for computer scientists and others, but we have to form these multidiscipline teams of chemical physicists, mathematicians, computer scientists. The team is about 25 people. You also have to form small groups with all of the multidisciplines to maybe attack a project. And then, you have to have fun. We started to have these bug fix days about every year and then it was 6 months, and they love it. I would provide all these pizzas, and there is this one guy who found the first bug, like "Woohoo!" You can see we have students in the room who are helping us. You can see how much fun this is, it really is a blast and they have a good time doing this. You have to have fun. It should be challenging, as well as fun.



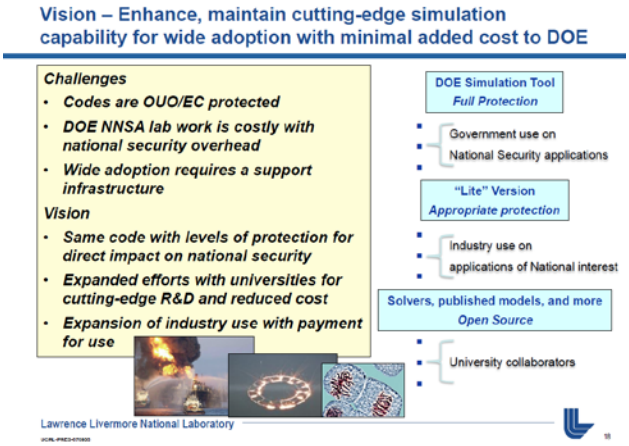
Infrastructure. I found that things were going so fast that I needed a better infrastructure than just me. I had to look inward and outward, share that vision, communicate it, and be sure that resources were available. We had so much funding coming in, I needed a person who tracked it, and people's assignments so we knew if we could hire or not. Then I did these three pillars. Before I stopped being the lead, I had these two jobs filled (Software Chief and Chief Product Engineer) and I was still doing this one, the Chief Scientist. You have to have somebody that decides what the technical approach would be if somebody comes to you and say I need "Auto Contact". Develop a statement of work; make sure you do project reviews, reporting. So I called that risk management. Then you want quality assurance so I had a wonderful chemical engineer who was a fantastic computational scientist who worked with me to make sure I wasn't promising something we couldn't deliver with methods and algorithms, worried about computer science and architecture, software quality assurance, V&V. We had just started to do uncertainty quantification. This guy (Chief Product Engineer) was key. This was the person we ultimately let say "Yes it's time to distribute the code" or "No it's not." So there was a nice balance and I called that product control. This included training. We went from at most, teaching a class once a year, to it being 3 or 4 times a year because of the demand from users.



I apologize for all the words. In my opinion, there is often a degree of tension between core Laboratory programs and work for others that you must manage. I'm not claiming I did the best job at this. I believe, the words in italics would be the comment from the core Laboratory's management, and then what I did to mitigate those situations. So, their comment might be frequent formal releases to meet work for others deliverables puts added demand on team. ALE3D used to be released at most once a year, if not once every two to three years. I went to a six month release. We had to automate and plan for focused, smaller deliverables, again because the statements of work said the deliverable was the release of the code. Somebody said DOE should not support things like code release. I had to include the cost of code release in the statement of work so that not only DOE was paying for that, but some of our other customers were paying. The next statement: delivering capabilities "on time and on budget" frequently requires compromises that need to be "fixed" with core dollars for robust use on core laboratory apps. What I think are they talking about? Well, I believe the expectation is that if you did this for one customer, the core laboratory's users could just pick it up and use it. It was important that the statement of work included that the capability had to be robust, everyone wants robustness. Communicate to the lab people that work is being done with Work For Other funds with that application in mind. Manage expectation; tell them it might not work. Team members are frequently divided among multiple projects. People who aren't good at multitasking struggle. This was a major complaint among the developers. So what we did was pick one or two focus topics for team members. They might work on different projects but at least they have one or two focus topics. Plan and hire capable and flexible workforce. Flexibility was really the key. As the code lead, it is difficult to stay on top of both the WFO marketing aspects as well as the technical management of the project. My comment is to avoid chasing dollars and seek Work For Others that is of value to your core programs as well so they see the benefit. Then, develop appropriate business model. That's why there was that team infrastructure.

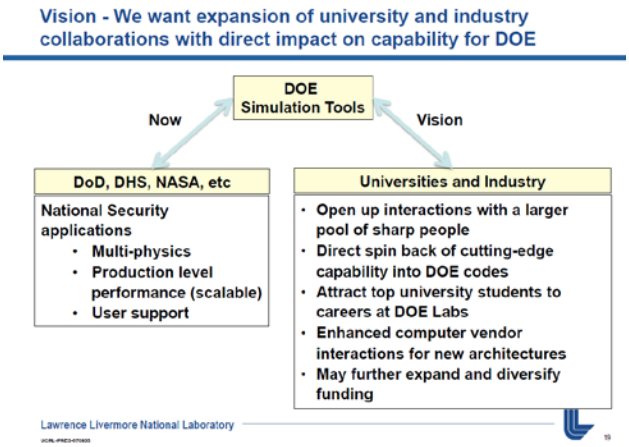


So I've told you all about these. I told you all about the evolution, the outcomes, and the challenges. What I want to share quickly is the vision for the future.

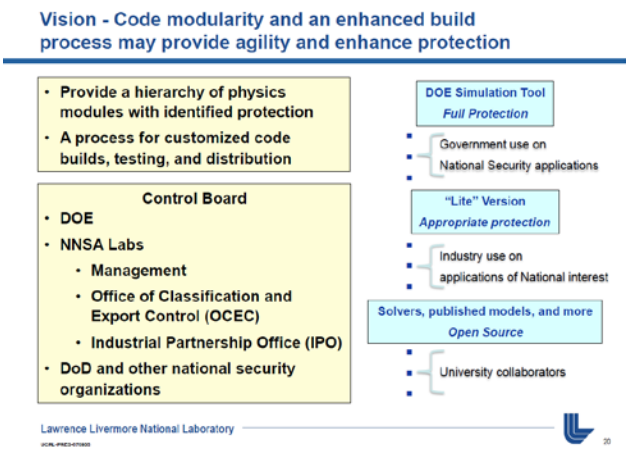


Here is what I think is the vision for the future: you want to enhance and maintain cutting-edge simulation capability for wide adoption with minimal added cost to DOE. That's what they want. Here are the challenges. These codes are Official-Use only and Export Controlled which means they must be used for national security applications. We cost a lot. NNSA lab personnel have huge costs. To have wide adoption, you need this infrastructure to support those complexities.

The first idea is have the same code so DOE gets a benefit, but have different levels of protection or direct impact on national security. Here's where ALE3D would be. It would have full protection and only used for national security. Have a light version with the appropriate protection and the applications are of national interest. At the next level, you have open source parts to this and that's where you have university collaborations. So expand efforts with universities because it not only provides cutting edge R&D, which you guys are doing, but it's also a lot cheaper. Expand to industry use and have industry pay for that use. I think that's what DOD, DOE, and NASA would like to see for things like the oil companies, additive manufacturing, or even bio applications.



We want expansion of university and industry capabilities with a direct impact on DOE capability. We're doing this now but we want interactions with sharp people. We want the spin back to the DOE codes. What's really cool about this too is if you establish those university connections, hopefully they'll come back to work at the labs as well. Because you'll have this light version or the open source, you can interact with vendors and not have that high level of protection, and you might expand your funding.



Department of Defense would have to decide.

How do you do this? You provide a hierarchy of physics modules with identified protection. You probably have to have a customized code build process for testing and distribution so you probably need a lot of computer science with that. You also have to have a control board. The Department of Energy has to decide if what's in this light version is ok, as well as the management. In the case of Lawrence Livermore, our office of classification and export control, this becomes your intellectual property, and other organizations like the

I've told you all these things as well as my vision with these different levels of protection. That's it. Thank you.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

LLNL-PRES-676426-DRAFT

An Overview of DYSMAS Coupled Code Development and Application

Dr. Thomas McGrath & Mr. Gregory Harris
NSWC Indian Head Explosive Ordnance Disposal Technology Division
Indian Head, MD

Abstract

The Dynamic System Mechanics Advanced Simulation (DYSMAS) software is a fully-coupled, government-owned hydrocode for predicting weapon effects and structural response. The software has been jointly developed over a 25 year time period under a series of International Project Agreements between the United States and Federal Republic of Germany. Its development fills a technology gap for massively-parallel, fluid-structure interaction software supporting Department of Defense (DoD) weapon effects and target response analyses. This talk will present an overview of the DYSMAS software and a review of its development history. Major application areas will be discussed along with highlights and impacts. Challenges to the code development and deployment process will also be discussed, with emphasis on challenges unique to DoD.

Biography

Dr. McGrath is the Senior Energetics Modeling and Simulation (M&S) Developer for the Naval Surface Warfare Center Indian Head Explosive Ordnance Disposal Technology Division. In this position, he leads several major research programs focused on the development of energetic and M&S technologies for the defense community. He is a lead code developer for DYSMAS, a massively-parallel, fully-coupled Eulerian-Lagrangian hydrocode for simulating explosively-driven fluid-structure interaction. His technical expertise includes multiphase/multi-scale physics, code development, combustion and detonation science, fluid dynamics, and advanced warhead technologies. Dr. McGrath received a Ph.D. in Mechanical Engineering from the University of Maryland, College Park while developing a novel numerical model for simulating multiphase explosion phenomena.

Due to security requirements, Dr. McGrath's presentation and transcript are not a part of this report.

New Advances in Direct Numerical Simulation of Multiphase Flow

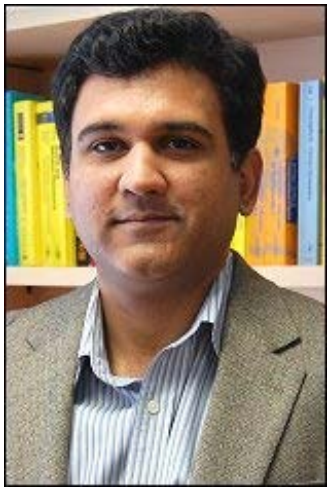
Dr. Amir Riaz
Department of Mechanical Engineering, University of Maryland
College Park, MD

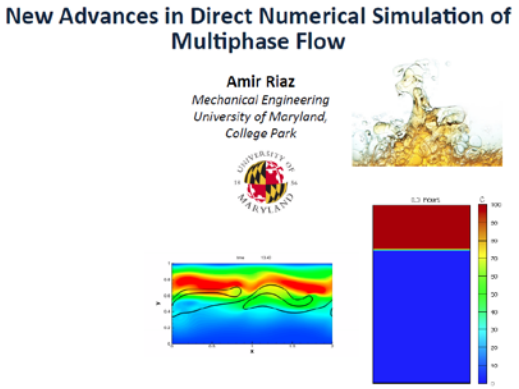
Abstract

Direct numerical simulation of multiphase flow is of interest for characterizing a wide range of physical phenomena occurring in both natural and engineering processes. Our new methods significantly extend the range of multiphase flow phenomena that can be represented accurately with direct numerical simulation. The main features of the new method are preserving the topological integrity of interfaces during advection while maintaining mass conservation and the implementation of second order interfacial jump-conditions as well as capturing the dynamic contact line motion.

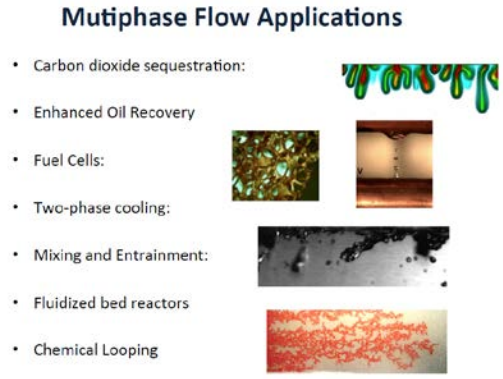
Biography

Dr. Amir Riaz is currently an Assistant Professor of Mechanical Engineering at the University of Maryland, with a research program focusing on numerical simulations of multiphase flow for energy and environmental problems including flows in micro-channels and porous media, as well as carbon dioxide sequestration and enhanced oil recovery.

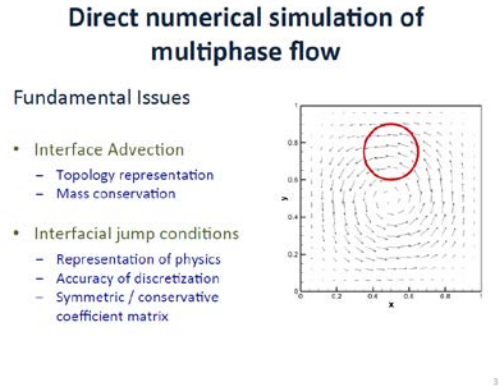




Thank you very much for the kind introduction. I'm going to talk about new advances in the direct numerical simulation of multiphase flow. What we do is we look at fluid fields and other phases within solid-fluid interactions and fluid-fluid interactions. Here on the slide is a turbulent flow field with some particles, some bubbles and droplets, and some carbon dioxide sequestration. Our applications run the range of these systems.

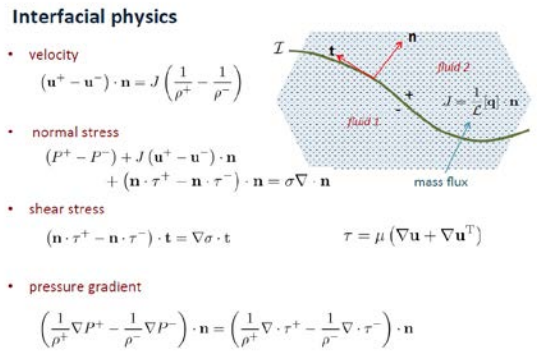


Multiphase flow applications are of great interest; you can find them everywhere. You have carbon dioxide sequestration on the subsurface. There is enhanced oil recovery, with oil pushing water out of the reservoir, fuel cells, two-phase flow cooling, mixing and entrainment in turbulent boundary layers, fluidized bed reactors, and chemical looping. There is a wide range of applications for multiphase flow, and the effort to numerically simulate these types of flows has been going on for a number of years. At this time, the effort is still in its developmental stages, like most other efforts in terms of modeling of physical phenomenon. However, this particular area of simulating two-phase flow directly with numerical simulation has reached a long-raised goal.

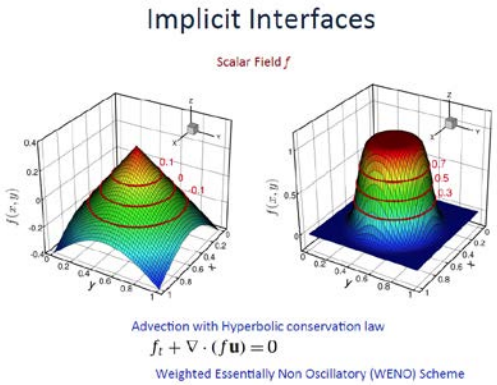


I will tell you how this goal has been reached. There are essentially two fundamental issues in multiphase flow simulation. You need to represent the advection of the interface very accurately with two fluids: for example, salt and water or air and water. Then you need to see how those fluids move around with respect to each other. One needs to be cognizant of the topological representation and the interface-defined properties so you can have the normal, the curvature, and all of the topological properties of the interface that go along with it. In order to illustrate this behavior, I have a simulation of these particles within the flow field on the slide. The flow field is stationary at this time; we are looking solely at the behavior of the particles that represent an object. These particles enclose an object, and they represent how the

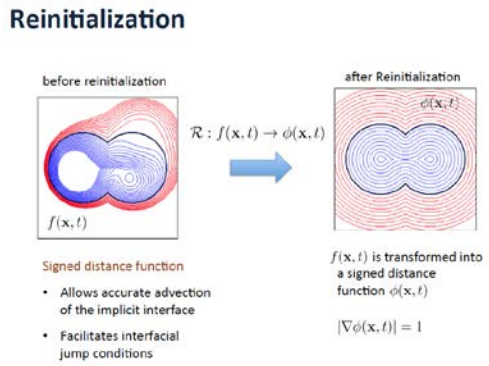
object would move within the flow field. This is an example of a single vortex flow field. The object gets stretched; it will break, and then it will get reconnected. The breaking up and reconnection needs to be represented on the grid. Therefore, we do not look at grids that conform to the object. We look at interfaces that are embedded within the grids. We look at implicit interfaces, which require mass conservation. When these things are moving and you are tracking them numerically with an implicit interface, for example level-sets, they lose mass. The second most important thing is that we need to represent physics very accurately at the interface. If two fluids have jumps in pressure and normal stresses, those need to be represented accurately. They need to be discretized accurately; the formal accuracy of discretization and the convergence properties are important. You also have to have a system of equations that you can solve very easily, so they have to be symmetric coefficient matrices that you can use like black boxes in order to solve them.



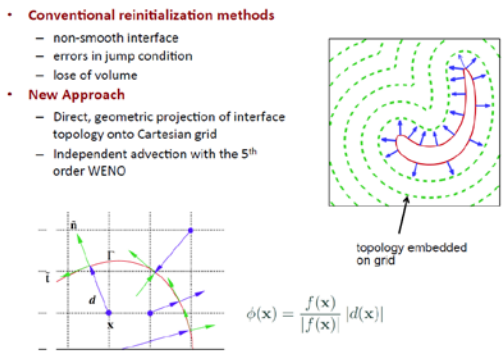
Let us discuss the types of interfacial physics that's involved in looking at fluid-fluid systems. We could have jumps in the velocities on either side of the interface due to the interface exchange of mass, for example due to evaporation and condensation caused by phase change in general. That would lead to a difference in the velocity. It could have normal stresses across the interface produced by differences in surface tension or jumps in mass across the interface. We could have a shear stress discontinuity that could be produced by the spatial variation of surface tension which would have led to a collection of mass transfer called the Marangoni effect. Then we have a pressure gradient at this contact. For all of this, the physics needs to be incorporated into any numerical code that is going to represent the flow situation. Most of the codes exist. But at this time, they only implement the normal stress jump condition without taking into account the jump and the normal stresses. Those are the volume of fluid type codes.



I am going to talk about implicit interfaces and the implementation of interfacial physics, as well as what we have done in order to advance interfacial physics. Let us discuss implicit interfaces on the Cartesian grid. We have solved everything on the regular, uniform Cartesian grid, since that's the best way of solving the Navier-Stokes equations. Now, we do the interfaces implicitly. We handle the interfaces implicitly using level-set functions, which can be defined with a scalar field. The scalar field could be in any form; you could take one contour of that scalar field and call it the interface.

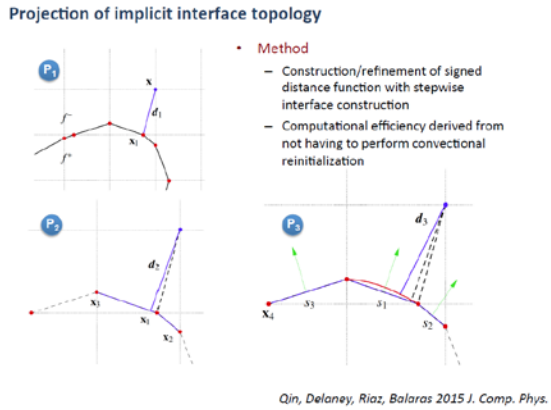


The most challenging aspect of representing interfaces on uniform Cartesian grids is reinitialization. You may have heard about reinitialization before, but what it actually means is that when you advect an embedded interface, a scalar function, the contours of the scalar function get distorted. If you begin before reinitialization, you start with the state which happened when you advected that scalar field. Then what you have to do is map that scalar field back into a special form of a field known as a signed distance function. A signed distance function is such that every contour in the field is equidistant from the interface. For example, with the interface represented by the black curve here on the right of the slide, all the points on these contours are projected onto the interface. In the past, there has been a big effort in terms of supporting reinitialization. There are various reasons why we want to do reinitialization: it allows accurate implementation of jump condition, it regularizes the scalar field, and it defines the interface.

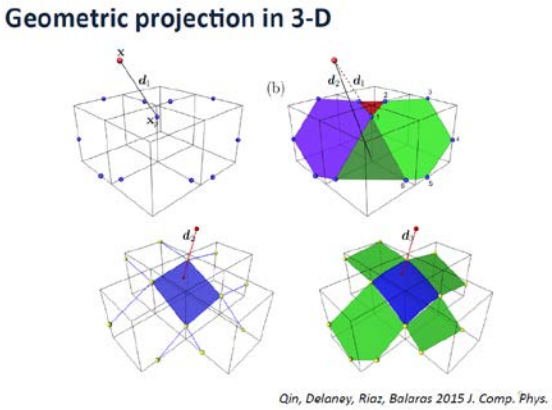


Previously, methods used to reinitialize and advect the scalar field have suffered from non-smooth interfaces, errors in jump conditions, losses in volume, and so on. Our new approach is based on a very simple idea: a direct, geometric projection. What we do is we represent the interface and we localize it on the grid. Since its implicit it needs to be localized and represented by the intersections of the grid. Then we project normals onto the interface from each point and construct our signed distance function.

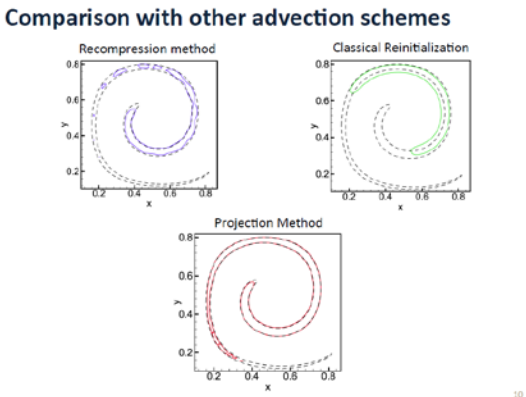
There are some aspects of how to do it in practice, and we have recently shown them here in the Journal of Computational Physics. We have these three methods of carrying out the projection by successively refining the interface.



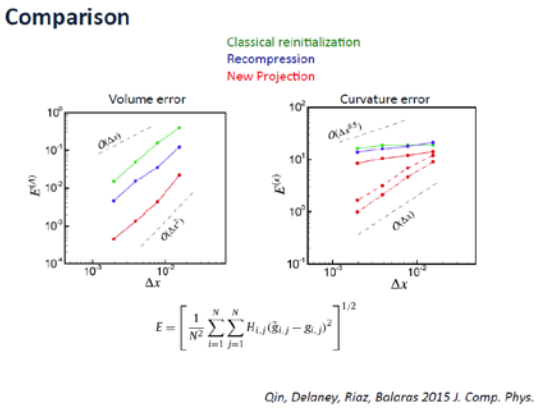
We could do that in 3-D. It becomes a little bit more challenging, but we are still able to do it with good accuracy.



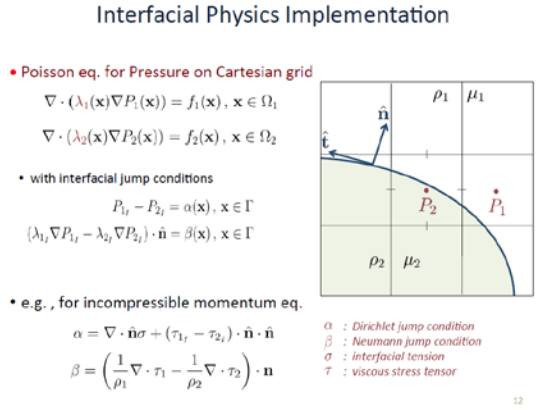
We compared it with other methods in the past that do reconstruction and then advection. However, advection and reinitialization is coupled together. If you do level-set advection and then reinitialize with the Hamilton-Jacobi iteration, what you end up with is distortion of the interface, loss of mass, and loss of topology. We compared it with some of the methods that have been used in the past. One of them is called the recompression method, as depicted in the top left of the slide. There is also the classical reinitialization method proposed by Osher and his group displayed at the top right of the slide. Looking at our projection method, it is significantly more accurate than methods used in the past. These dashed lines here on the Projection Method grid at the bottom of the slide show the exact solution. On a particular grid, we are looking for solutions, or methods, that are most accurate. If I increased my resolution and make my grid more refined, I should be able to get the exact solution. But we are looking at, on a particular grid, which method is able to capture the behavior most accurately.



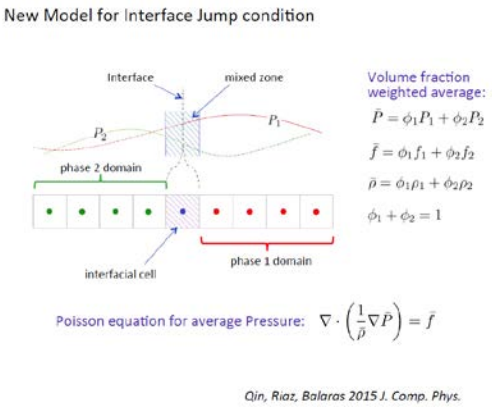
We look at the comparison with previous methods and it shows that our method in terms of the volume error is second order accurate, so we refined the grid and see second order accuracy. We look at the curvature error, which is the most important factor to look at because that's what represents the jumps in pressure between two fluids as well as the jump in the viscous stress. The curvature is the most important quality, the second order divergence of the normal, so we capture that with first order. In the past, all of the volume of fluid codes out there couldn't even converge. If you refine the grid, they kind of bombed. Really,



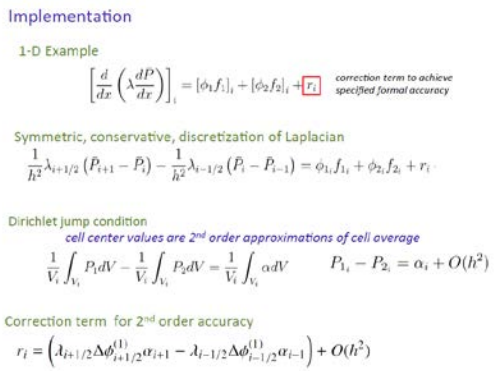
you could only run them at the course grid. That was the volume of fluid method. Recently, the volume of fluid method has been combined with level-set approaches and some sort of hybrid methods have been used.



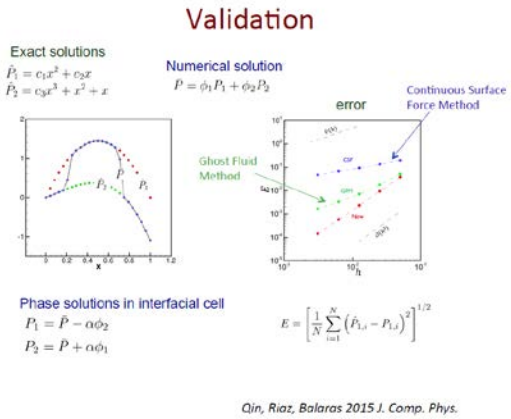
That was the interface advection. We did some new things there, and they seemed to have worked, but the door has yet to be closed on the advances that could still be made. The implementation of interfacial physics is very important. Previously, what we wanted to do was implement the interfacial jump conditions as a jump in the pressure and as a jump in the derivative of the pressure. When you are solving incompressible Navier-Stokes equations, these jump initials need to be implemented at the level of solving the Poisson equation for pressure. We solve incompressible flows, and these jump conditions need to be imposed.



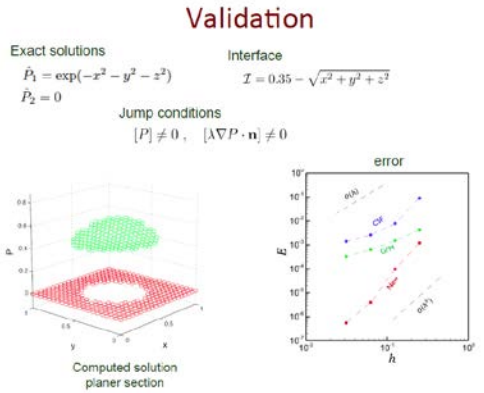
We came up with a new model for interfacial jump condition implementation that has been reported in the Journal of Computational Physics recently. We define a mix zone, and this is different from the continuous surface force model that has been used thus far. We proposed a new model for the pressure Poison equation which was based on the volume of a fraction weighted average of the pressure.



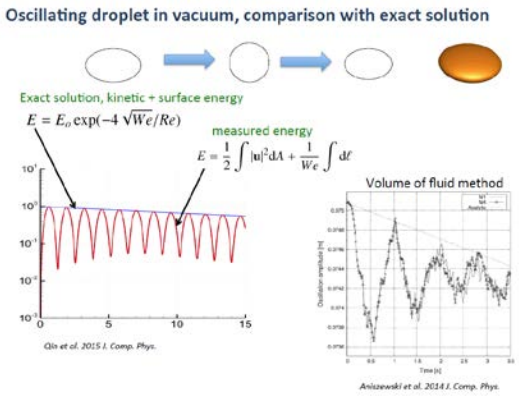
This allowed us to introduce a correction term. We did a symmetric, conservative discretization of a Laplacian, and that led to the second order correction term to drop out.



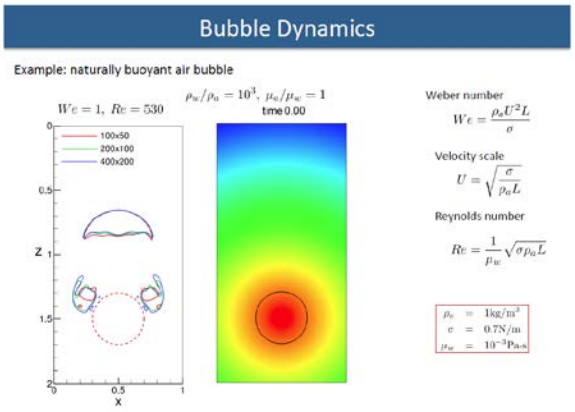
We looked at validation. We took exact solutions and then found a numerical solution and compared them with the results of previous methods that are available out there: the continuous surface force method and the ghost fluid method. You can see that the continuous surface force model, which is invariably used in volume of fluid equations, is not even first order. The ghost fluid method does a little better, and again that was introduced by Osher's students. The new method that we have here gives us good results with second order accuracy.



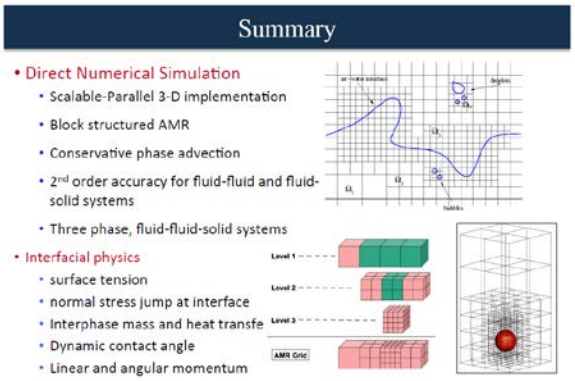
We did validation for 3-D problems, again compared with continuous surface force and ghost fluid methods. Here on the slide is a slice from a planar section in 3-D. We are getting some good results here.



Here is another form of validation where we simulate a droplet in space. This is a droplet that moves under the surface tension effect. We stretch it initially and let it oscillate. The oscillations were computed by Lamb a long time ago. This equation on the left of the slide is the exact solution of Lamb. Here in red is the numerical solution that you measure. In the end, it seems to be reasonably accurate. No other codes have actually been able to get to that level of accuracy for this very simple validation problem. The volume of fluid method with the hybridization approach does something like this but it doesn't look too good.



If we could look at bubble dynamics with our codes, we could simulate the rising of bubbles. This is one of the important challenges in this area. How does the bubble break up? How do the lobes detach from the main bubble? What's the flow field? What does the wake look like? All of these prototypical problems are related to questions of mixing, questions of coalescence of bubbles, and different types of fluids



What we have is a direct numerical simulation of two-phase flow. It's a scalable-parallel 3-D implementation. It's a block-structured AMR code with a conservative phase advection. It has a second order accuracy of a representation of jump conditions, and can handle three phase fluid-fluid-solid systems. In terms of interfacial physics, you can have surface tensions, normal stress jump conditions, interface mass transfers, etcetera. We also can do dynamic contact angle representation. If we have a fluid on a substrate and it moves, the contact angle changes with the Capurro number, it is possible for us to implement that contact angle given some model for the contact angle speed. We can also do linear and angular momentum of solid particles in turbulent flow fields.

I will stop here. Thank you.