Multigrid Methods for Drift-Kinetic Calculations in Stellarators and Rippled Tokamaks

Matt Landreman, University of Maryland,   Håkan Smith, Max Planck Institute for Plasma Physics

Overview

- Several important phenomena – the bootstrap current and collisional transport in stellarators, and neoclassical toroidal viscosity (NTV) in tokamaks – must be computed by numerical solution of the drift-kinetic equation (DKE) in nonaxisymmetric geometry.
- Need time-independent (steady) solution: like an implicit solve.
- High resolution is required in at least 3 dimensions (poloidal angle θ, toroidal angle ζ, pitch angle ξ) due to internal boundary layers.
- Existing continuum solvers (e.g. DKES [1] and SFINCS [2]) have used a direct solver for at least these 3 dimensions, which scales poorly with resolution: large memory and time required for high resolution.
- Multigrid methods [3,4,5] are state-of-the-art for solving PDEs and can have optimal scaling. However multigrid is not straightforward for advection-dominated problems like the DKE.
- Here we develop a multigrid solver for the DKE.

[1] van Rij & Hirshman, Phys Fluids B 1, 563 (1989).

[2] Landreman et al, Phys Plasmas 21, 042503 (2014).

[3] Trottenberg et al, "Multigrid", Academic Press (2001).

[4] Briggs et al, "A Multigrid Tutorial", 2nd ed., SIAM (2000).

[5] Brandt, "The Multigrid Guide", http://www.wisdom.weizmann.ac.il/~achi/classics.pdf

PDEs

Variants with differing complexity:

simpler

Eventual goal: (solved now in SFINCS code [2])

$$\dot\theta\frac{\partial f_{a1}}{\partial\theta}+\dot\zeta\frac{\partial f_{a1}}{\partial\zeta}+\dot\xi\frac{\partial f_{a1}}{\partial\xi}+\dot\upsilon\frac{\partial f_{a1}}{\partial\upsilon}-\sum_b C_{ab} =-\left[\left(\mathbf{v}_{ma}+\mathbf{v}_E\right)\cdot\nabla\psi\right]f_{a0}\left[\frac{d\ln n_a}{d\psi}+\frac{q_a}{T_a}\frac{d\Phi}{d\psi}+\left(\frac{m_a\upsilon^2}{2T_a}-\frac{3}{2}+\frac{q_a\Phi_1}{T_a}\right)\frac{d\ln T_a}{d\psi}\right],$$

$$\sum_a q_a\int d^3\upsilon\left(f_{a0}+f_{a1}\right)=0\ \text{(Quasi-neutrality)},\quad f_{a0}=f_{aM}\exp\left(-q_a\Phi_1/T_a\right)$$

$$C_{ab}=C_{ab}\left(f_{a1},f_{b0}\right)+C_{ab}\left(f_{a0},f_{b1}\right)=\text{linear Landau operator}$$

Unknowns: $f_a=f_a\left(\theta,\zeta,\xi,\upsilon\right)$ and $\Phi_1=\Phi_1\left(\theta,\zeta\right)$.

θ = poloidal angle, ζ = toroidal angle, $\xi=\upsilon_{||}/\upsilon$, a = species

Minimal accurate version:

$$\dot\theta\frac{\partial f_{a1}}{\partial\theta}+\dot\zeta\frac{\partial f_{a1}}{\partial\zeta}+\dot\xi\frac{\partial f_{a1}}{\partial\xi}+\dot\upsilon\frac{\partial f_{a1}}{\partial\upsilon}-\sum_b C_{ab}=-\left(\mathbf{v}_{ma}\cdot\nabla\psi\right)\frac{\partial f_{Ma}}{\partial\psi}$$

Unknown: $f_a=f_a\left(\theta,\zeta,\xi,\upsilon\right)$

Monoenergetic, 1 species, E_r=0:

$$\xi B\left[\iota\frac{\partial f}{\partial\theta}+\frac{\partial f}{\partial\zeta}\right]-\frac{1-\xi^2}{2}\left[\iota\frac{\partial B}{\partial\theta}+\frac{\partial B}{\partial\zeta}\right]\frac{\partial f}{\partial\xi}-\frac{\nu}{2}\frac{\partial}{\partial\xi}\left[\left(1-\xi^2\right)\frac{\partial f}{\partial\xi}\right] =\frac{1+\xi^2}{B}\left[G\frac{\partial B}{\partial\theta}-I\frac{\partial B}{\partial\zeta}\right]$$

Unknown: $f=f\left(\theta,\zeta,\xi\right)$.

Specified: $B=B\left(\theta,\zeta\right)$ and constants ι, G, I, ν.

Axisymmetry:

$$\xi B\iota\frac{\partial f}{\partial\theta}-\frac{1-\xi^2}{2}\iota\frac{\partial B}{\partial\theta}\frac{\partial f}{\partial\xi}-\frac{\nu}{2}\frac{\partial}{\partial\xi}\left[\left(1-\xi^2\right)\frac{\partial f}{\partial\xi}\right]=\frac{1+\xi^2}{B}G\frac{\partial B}{\partial\theta}$$

Unknown: $f=f\left(\theta,\zeta\right)$

PDE properties

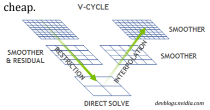
- Eventual goal includes nonlinear terms, but all reduced forms of the PDE are linear.
- Simplest versions are 2D or 3D (monoenergetic tokamak or stellarator). Eventual goal is 5D.
- Steady (time-independent).
- Inhomogeneous.
- Advection-diffusion equation.
- Advection-dominated. In fact, no physical diffusion at all in 2 coordinates (θ and ζ.)
- 2 coordinates (θ and ζ) are periodic on [0,2π).
- In the 3rd coordinate (pitch angle ξ), domain is [-1,1], and there is no boundary condition – only regularity is required. The PDE changes order at the boundaries.
- Recirculating flows (closed characteristics).
- Solution has internal boundary layers.
- Small divisor problem (rational ι)

Multigrid methods

[3-5]

Geometric Multigrid:

- Gauss-Seidel smoothing quickly reduces short-wavelength errors but is ineffective for long-wavelength errors.
- On coarser grids, long-wavelength errors are more quickly corrected.
- Therefore, a combination of Gauss-Seidel smoothing with a coarse-grid solve reduces all wavelengths in the error efficiently.
- Do the coarse-grid solve recursively: "V cycle". On coarsest level, a direct solve is cheap.



devblogs.nvidia.com

Algebraic Multigrid:

- Abstracts ideas from geometric multigrid to yield a 'black box' solver, based on only matrix elements.
- Heuristics for coarsening the degrees of freedom and for restriction/interpolations operators. Coarse-grid matrix is then *(restriction)(fine matrix)(interpolation)*.
- Well-established libraries exist: PETSc-GAMG, Hypre-BoomerAMG, & ML.

Finite difference stencils

We consider finite difference discretizations since:

- Multigrid literature is most developed for finite differences.
- Our coordinate grid is simply a tensor product.
- Conservation errors are no worse than other discretization errors since the problem is time-independent, so no advantage of finite volume.

There are many options for differentiation stencils: $$\left(\frac{\partial f}{\partial x}\right)_j\approx\sum_k\frac{a_k f\left(x_{j+k}\right)}{\Delta x}$$

Name	Order	a_{-3}	a_{-2}	a_{-1}	a_0	a_1	a_2	a_3
D_{1a}	1			-1	1			
D_{1b}	1			-1/2-y	2y	1/2-y		
D_{2a}	2			-1/2	0	1/2		
D_{2b}	2		1/2	-2	3/2			
D_{2c}	2		1/4	-5/4	3/4	1/4		
D_{3a}	3		1/6	-1	1/2	1/3		
D_{3b}	3		1/12+y	-2/3-4y	6y	2/3-4y	-1/12+y	
D_{4a}	4		1/12	-2/3	0	2/3	-1/12	
D_{4b}	4	-1/12	1/2	-3/2	5/6	1/4		
D_{5a}	5	-1/30	1/4	-1	1/3	1/2	-1/20	
D_{5b}	5	-1/60-y	3/20+6y	-3/4-15y	20y	3/4-15y	-3/20+6y	1/60-y

(y = any constant)

For 2nd derivative, we need only consider centered stencils.

$$\left(\frac{\partial^2 f}{\partial x^2}\right)_j\approx\sum_k\frac{b_k f\left(x_{j+k}\right)}{\Delta x^2}$$

Name	Order	b_{-2}	b_{-1}	b_0	b_1	b_2
DD_2	2		1	-2	1	
DD_4	4	-1/12	4/3	-5/2	4/3	-1/12

Null space & solvability

- The 'minimal accurate' PDE has a null space of dimension $2N_{\text{species}}$, spanned by f_{Ma}, $v^2 f_{\text{Ma}}$.
- The monoenergetic PDE has a null space of dimension 1 spanned by 1.
- For each null space dimension, there is a solvability condition of the PDE, corresponding to the density or pressure moment.
- Thus, the discretized PDE yields a rank-deficient matrix, and the algorithm must handle this property.

Augment the system with extra unknowns (sources S) and constraints. For the monoenergetic case,

$$\begin{array}{r} DKE: \\ \left\langle\int_{-1}^{1}d\xi\ f\right\rangle=0: \end{array}\begin{pmatrix} A & B \\ C & 0 \end{pmatrix}\begin{pmatrix} f \\ S \end{pmatrix}=\begin{pmatrix} R \\ 0 \end{pmatrix}$$

where A is the original DKE operator,

$$B=\text{vector of 1's},\qquad C=\left\langle\int_{-1}^{1}d\xi\ (...)\right\rangle.$$

Since $\left\langle\int_{-1}^{1}d\xi\ \left(DKE\right)\right\rangle=0$, then $CA=0$.

Since $\left(DKE\right)\left(1\right)=0$, then $AB=0$.

The formal inverse of $M=\begin{pmatrix} A & B \\ C & 0 \end{pmatrix}$ is

$$M^{-1}=\begin{pmatrix} \left[I-B\left(CB\right)^{-1}C\right]A^{-1}\left[I-B\left(CB\right)^{-1}C\right] & B\left(CB\right)^{-1} \\ \left(CB\right)^{-1}C & 0 \end{pmatrix}.$$

CB is small ($2N_{species}\times 2N_{species}$) so applying $\left(CB\right)^{-1}$ is fast.

Taking A^{-1} to be a multigrid cycle, then M^{-1} is our final preconditioner.

Pitch angle coordinate

$$\frac{\upsilon_{||}}{\upsilon}=\xi=-\cos\alpha,\qquad \xi\in\left[-1,1\right],\qquad \alpha\in\left[0,\pi\right)$$

- Velocity-space grid in (v,ξ) or (v,α) is much better than grid in ($v_{||}$, $v_\perp$) since boundary layer is independent of v.
- A popular choice in previous codes [1-2] has been Legendre polynomials in ξ, since you get both spectral accuracy and sparsity. But here we consider finite differences since multigrid is most likely to work for this choice, and upwinding (needed for smoothing) is tricky in Legendre space.
- Without Legendre polynomials, collision operator field term is dense, so use matrix-free implementation for it.
- We need accurate integrals. A uniform grid in ξ is not very accurate for integration. But a uniform grid in α corresponds to a Chebyshev grid in ξ, allowing very accurate Clenshaw-Curtis integration.
- There is no boundary condition in ξ; only regularity is required. For a grid in ξ, close to the boundaries we would need 1-sided stencils, spoiling upwinding there.
- But for a grid in α, you can extend the α domain and use $f(-\alpha)=f(\alpha)$ to evaluate derivatives near the boundaries using the same stencils as in the interior, preserving upwinding. Seems more elegant.
- For a grid in α that includes points at the boundaries, the pitch angle scattering operator is singular there, and the DKE reduces to df/dα=0: $$\frac{\partial}{\partial\xi}\left[\left(1-\xi^2\right)\frac{\partial f}{\partial\xi}\right]=\frac{\partial^2 f}{\partial\alpha^2}+\underbrace{\frac{\cos\alpha}{\sin\alpha}}_{\pm\infty\text{ at }\alpha=0,\pi}\frac{\partial f}{\partial\alpha}$$
- Therefore we use a uniform grid in α with grid points at element centers instead of ends, so there is no singularity.
- Most of the structure in the solution is at small ξ (trapped particles or barely-passing particles), so extra resolution there would be valuable. It is straightforward to add resolution there using a coordinate transformation. Not considered here for simplicity.

Defect correction

Multigrid smoothing tends to only be stable for low-order upwinded discretizations, like D_{1a}. But we want the solution accurate to high order (>= 4). Therefore we will need to choose a *pair* of discretizations, and this choice strongly affects convergence.

Example: GMRES preconditioned with direct LU factorization of preconditioner (no multigrid yet), W7-X geometry, r/a=0.9, monoenergetic, E_r=0, collisionality $\nu qR/v_{\text{th}}$ = 2e-4, N_θ=47, N_ζ=139, N_ξ=85, tolerance = 1e-4.

Iterations required:

Discretization for preconditioner

Discretization for real matrix	D_{1a} DD_2	D_{1b} DD_2	D_{2a} DD_2	D_{2b} DD_2	D_{2c} DD_2	D_{3a} DD_2	D_{3b} DD_4	D_{4a} DD_4	D_{4b} DD_2	D_{5a} DD_4	D_{5b} DD_4
D_{1a} DD_2	1	12									
D_{1b} DD_2	12	1									
D_{2a} DD_2	628	272	1	1362	546						
D_{2b} DD_2	67	146	>10000	1	17						
D_{2c} DD_2	58	48	3800	18	1						
D_{3a} DD_2	65	42	2200	45	6	1	7				
D_{3b} DD_4	38	46	3160	19	5	7	1				
D_{4a} DD_4	1097	938	2523	1513	274	112	201	1	723		
D_{4b} DD_2	180	188	>10000	85	14	13	18	>10000	1		
D_{5a} DD_4	192	119	4762	133	24	13	26	1827	7	1	6
D_{5b} DD_4	154	124	4148	93	15	10	16	2121	4	7	1

Red entries are the most interesting: high-order (>3) discretizations in the real matrix that are effectively preconditioned by low-order (<4) upwinded discretizations.

y = 0.15 for D_{1b} to minimize # iterations when preconditioning D_{5a}.

y = 0.2 for D_{3b} and y = 0.04 for D_{5b}, to minimize # iterations when preconditioned by D_{1a}.

Algebraic Multigrid Results

Iterations for W7-X example without defect correction, 1 proc, using default GAMG/BoomerAMG/ML options, with GMRES acceleration: only D_{1a} works well

		Stencil										
		D_{1a} DD_2	D_{1b} DD_2	D_{2a} DD_2	D_{2b} DD_2	D_{2c} DD_2	D_{3a} DD_2	D_{3b} DD_4	D_{4a} DD_4	D_{4b} DD_2	D_{5a} DD_4	D_{5b} DD_4
Library	GAMG	51	NaN	NaN	>1000	>1000	NaN	804	NaN	>1000	NaN	>1000
	BoomerAMG	9	>1000	NaN	>1000	>1000	>1000	>1000	NaN	>1000	>1000	>1000
	ML	54	NaN	NaN	>1000	>1000	NaN	794	NaN	>1000	NaN	>1000

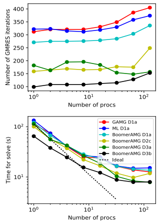
All 3 libraries have MANY tuning parameters/options. Best results so far, using 8 procs and D_{5a} for main matrix:

Preconditioner stencil	Library	Non-default options	# iterations	Time for solve (s)
D_{1a}	GAMG	Threshold=0.2	321	25
D_{1a}	ML	Threshold=0.5	312	27
D_{1a}	BoomerAMG	Threshold=0.15, modified Ruge-Stueben coarsening, <4 nonzeros per row of interpolation, relax in lexicographic order, extended+i interpolation	276	28
D_{2c}	BoomerAMG	Threshold=0.15, relax_weight=0.35, modified Ruge-Stueben coarsening, <4 nonzeros per row of interpolation, relax in lexicographic order, extended+i interpolation	165	22
D_{3a}	BoomerAMG	Threshold=0.15, relax_weight=0.16, modified Ruge-Stueben coarsening, <4 nonzeros per row of interpolation, relax in lexicographic order, extended+i interpolation	196	27
D_{3b} (y=0.2)	BoomerAMG	Threshold=0.15, relax_weight=0.9, modified Ruge-Stueben coarsening, <4 nonzeros per row of interpolation, relax in lexicographic order, extended+i interpolation	108	16

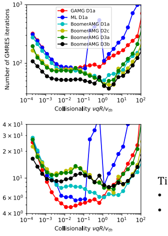
Memory usage for W7X example, N_θ=59, N_ζ=175, N_ξ=200, 32 procs, GMRES(200):

GAMG: **14 GB**, 4 min;   SFINCS (not multigrid): **291 GB**, 19 min

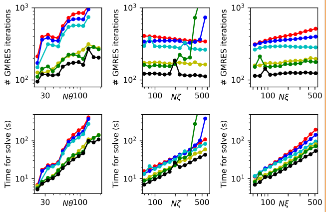
All methods show strong scaling a bit worse than ideal, saturating around ~ 64 procs.



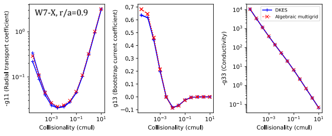
At fixed resolution, more iterations are needed at very low or high collisionality:



Scaling with resolution is different in the different coordinates, presumably due to anisotropy. D_{1a} methods perform poorly at high Nθ.



The new method has also been benchmarked against DKES [1]:



Timings on IPP computer Draco for lowest collisionality (cmul=3e-5):

- DKES gets radial transport to +/- 12% in **58,160 seconds** (1 proc).
- Algebraic multigrid code gets it within 4% in 168 seconds (1 proc) or **33 seconds** (32 procs)