FIRST-ORDER ORDINARY DIFFERENTIAL EQUATIONS III: Numerical and More Analytic Methods

David Levermore
Department of Mathematics
University of Maryland

30 September 2012

Because the presentation of this material in lecture will differ from that in the book, I felt that notes that closely follow the lecture presentation might be appreciated.

Contents

8. First-Order Equations: Numerical Methods

8.1. **Numerical Approximations.** Analytic methods are either difficult or impossible to apply to many first-order differential equations. In such cases direction fields might be the only graphical method that we have covered that can be applied. However, it can be very hard to understand how any particular solution behaves from the direction field of its governing equation. If one is interested in understanding how a particular solution behaves it is often easiest to use numerical methods to construct accurate approximations to the solution. These approximations can then be graphed much as we would an explicit solution.

Suppose we are interested in the solution $Y(t)$ of the initial-value problem

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t,y)\,, \qquad y(t_I) = y_I\,, \tag{8.1}$$

over the time interval $[t_I, t_F]$ — i.e. for $t_I \le t \le t_F$. Here t_I is called the *initial time* while t_F is called the *final time*. A numerical method selects times $\{t_n\}_{n=0}^N$ such that

$$t_I = t_0 < t_1 < t_2 < \cdots < t_{N-1} < t_N = t_F\,,$$

and computes values $\{y_n\}_{n=0}^N$ such that

$$y_0 = Y(t_0) = y_I\,,$$
$$y_n \text{ approximates } Y(t_n) \text{ for } n = 1, 2, \cdots, N\,.$$

For good numerical methods, these approximations will improve as N increases. So for sufficiently large N we can plot the points $\{(t_n, y_n)\}_{n=0}^N$ in the (t,y)-plane and "connect the dots" to get an accurate picture of how $Y(t)$ behaves over the time interval $[t_I, t_F]$.

Here we will introduce a few basic numerical methods in simple settings. The numerical methods used in software packages such as MATLAB are generally far more sophisticated than those we will study here. They are however built upon the same fundamental ideas as the simpler methods we will study. Throughout this section we will make the following two basic simplifications.

- We will employ *uniform time steps*. This means that given N we set

$$h = \frac{t_F - t_I}{N}\,, \qquad \text{and} \qquad t_n = t_I + nh \quad \text{for } n = 0, 1, \cdots, N\,, \tag{8.2}$$

  where h is called the *time step*.
- We will employ *one-step methods*. This means that given $f(t,y)$ and h the value of y_{n+1} for $n = 0, 1, \cdots, N-1$ will depend only on y_n.

Sophisticated software packages use methods in which the time step is chosen adaptively. In other words, the choice of t_{n+1} will depend on the behavior of recent approximations — for example, on (t_n, y_n) and (t_{n-1}, y_{n-1}). Employing uniform time steps will greatly simplify the algorithms, and thereby simplify the programming we you have to do. If you do not like the way a run looks, you will simply try again with a larger N.

Similarly, sophisticated software packages will sometimes use so-called *multi-step methods* for which the value of y_{n+1} for $n = m, m+1, \cdots, N-1$ will depend on y_n, y_{n-1}, $\cdots$, and y_{n-m} for some positive integer m. Employing one-step methods will again simplify the algorithms, and thereby simplify the programming you will have to do.

8.2. **Explicit and Implicit Euler Methods.** The simplest (and least accurate) numerical methods are the Euler methods. These can be derived many ways. Here we give a simple approach based on the definition of the derivative through difference quotients.

If we start with the fact that

$$\lim_{h\to 0} \frac{Y(t+h)-Y(t)}{h} = Y'(t) = f(t,Y(t))\,,$$

then for small positive h we have

$$\frac{Y(t+h)-Y(t)}{h} \approx f(t,Y(t))\,.$$

Upon solving this for $Y(t+h)$ we find that

$$Y(t+h) \approx Y(t) + hf(t,Y(t))\,.$$

If we let $t = t_n$ above (so that $t+h = t_{n+1}$) this is equivalent to

$$Y(t_{n+1}) \approx Y(t_n) + hf(t_n,Y(t_n))\,.$$

Because y_n and y_{n+1} approximate $Y(t_n)$ and $Y(t_{n+1})$ respectively, this suggests setting

$$y_{n+1} = y_n + hf(t_n,y_n) \quad \text{for } n = 0,1,\cdots,N-1\,. \tag{8.3}$$

This so-called Euler method was introduced by Leonhard Euler in 1768.

Alternatively, we could have started with the fact that

$$\lim_{h\to 0} \frac{Y(t)-Y(t-h)}{h} = Y'(t) = f(t,Y(t))\,.$$

Then for small positive h we have

$$\frac{Y(t)-Y(t-h)}{h} \approx f(t,Y(t))\,.$$

Upon solving this for $Y(t-h)$ we find that

$$Y(t-h) \approx Y(t) - hf(t,Y(t))\,.$$

If we let $t = t_{n+1}$ above (so that $t-h = t_n$) this is equivalent to

$$Y(t_{n+1}) - hf(t_{n+1},Y(t_{n+1})) \approx Y(t_n)\,.$$

Because y_n and y_{n+1} approximate $Y(t_n)$ and $Y(t_{n+1})$ respectively, this suggests setting

$$y_{n+1} - hf(t_{n+1},y_{n+1}) = y_n \quad \text{for } n = 0,1,\cdots,N-1\,. \tag{8.4}$$

This method is called the *implicit Euler* or *backward Euler* method. It is called the implicit Euler method because equation (8.4) implicitly relates y_{n+1} to y_n. It is called the backward Euler method because the difference quotient upon which it is based steps backward in time (from t to $t-h$). In contrast, the Euler method (8.3) sometimes called the *explicit Euler* or *forward Euler* method because it gives y_{n+1} explicitly and because the difference quotient upon which it is based steps forward in time (from t to $t+h$).

The implicit Euler method can be very inefficient unless equation (8.4) can be explicitly solved for y_{n+1}. This can be done when $f(t,y)$ is a fairly simple function if y. For example, it can be done when $f(t,y)$ is linear or quadratic in either y or $\sqrt{y}$. However, there are equations for which the implicit Euler method will outperform the explicit Euler method.

8.3. **Explicit One-Step Methods Based on Taylor Approximation.** The explicit (or forward) Euler method can be understood as the first in a sequence of explicit methods that can be derived from the Taylor approximation formula.

8.3.1. *Explicit Euler Method Revisited.* The explicit Euler method can be derived from the first-order Taylor approximation, which is also known as the tangent line approximation. This approximation states that if $Y(t)$ is twice continuously differentiable then

$$Y(t+h) = Y(t) + hY'(t) + O(h^2)\,. \tag{8.5}$$

Here the $O(h^2)$ means that the remainder vanishes at least as fast as h^2 as h tends to zero. It is clear from (8.5) that for small positive h we have

$$Y(t+h) \approx Y(t) + hY'(t)\,.$$

Because $Y(t)$ satisfies (8.1), this is the same as

$$Y(t+h) \approx Y(t) + hf(t, Y(t))\,.$$

If we let $t = t_n$ above (so that $t + h = t_{n+1}$) this is equivalent to

$$Y(t_{n+1}) \approx Y(t_n) + hf(t_n, Y(t_n))\,.$$

Because y_n and y_{n+1} approximate $Y(t_n)$ and $Y(t_{n+1})$ respectively, this suggests setting

$$y_{n+1} = y_n + hf(t_n, y_n) \qquad \text{for } n = 0, 1, \cdots, N-1\,, \tag{8.6}$$

which is exactly the Euler method (8.3).

8.3.2. *Local and Global Errors.* One advantage of viewing the Euler method through the tangent line approximation (8.5) is that we gain some understanding of how its error behaves as we increase N, the number of time steps — or what is equivalent by (8.2), as we decrease h, the time step. The $O(h^2)$ term in (8.5) represents the *local error*, which is error the approximation makes at each step.

Roughly speaking, if we halve the time step h then by (8.5) the local error will reduce by a factor of one quarter, while by (8.2) the number of steps N we must take to get to a prescribed time (say t_F) will double. If we assume that errors add (which is often the case) then the error at t_F will reduce by a factor of one half. In other words, doubling the number of time steps will reduce the error by about a factor of one half. Similarly, tripling the number of time steps will reduce the error by about a factor of one third. Indeed, it can be shown (but we will not do so) that the error of the explicit Euler method is $O(h)$ over the interval $[t_I, t_F]$. The best way to think about this is that if we take N steps and the error made at each step is $O(h^2)$ then we can expect that the acummulation of the local errors will lead to a *global error* of $O(h^2)N$. Because (8.2) states that $hN = t_F - t_I$, which is a number that is independent of h and N, we see that global error of the explicit Euler method is $O(h)$. This was shown by Cauchy in 1824. Moreover, it can be shown that the error of the implicit Euler method behaves the same way.

Global error is a more meaningful concept than local error because it tells us how fast a method converges over the entire interval $[t_I, t_F]$. *Therefore we identify the order of a method by the order of its global error.* In particular, methods like the Euler methods with global errors of $O(h)$ are *first-order methods*. By reasoning similar to that given in the previous paragraph, methods whose local error is $O(h^{m+1})$ will have a global error of $O(h^{m+1})N = O(h^m)$ and thereby are m^{th}*-order methods*.

Higher-order methods are more complicated than the explicit Euler method. The hope is that this cost is overcome by the fact that its error improves faster as you increase N — or what is equivalent by (8.2), as you decrease h. For example, if we halve the time step h of a fourth-order method then the global error will reduce by a factor of $1/16$. Similarly, tripling the number of time steps will reduce the error by about a factor of $1/81$.

8.3.3. *Higher-Order Taylor-Based Methods.* The second-order Taylor approximation states that if $Y(t)$ is thrice continuously differentiable then

$$Y(t+h) = Y(t) + hY'(t) + \tfrac{1}{2}h^2Y''(t) + O(h^3)\,. \tag{8.7}$$

Here the $O(h^3)$ means that the remainder vanishes at least as fast as h^3 as h tends to zero. It is clear from (8.7) that for small positive h one has

$$Y(t+h) \approx Y(t) + hY'(t) + \tfrac{1}{2}h^2Y''(t)\,. \tag{8.8}$$

Because $Y(t)$ satisfies (8.1), we see by the chain rule from multivariable calculus that

$$\begin{aligned} Y''(t) = \frac{\mathrm{d}}{\mathrm{d}t}\big(Y'(t)\big) = \frac{\mathrm{d}}{\mathrm{d}t}f(t,Y(t)) &= \partial_t f(t,Y(t)) + Y'(t)\,\partial_y f(t,Y(t)) \\ &= \partial_t f(t,Y(t)) + f(t,Y(t))\,\partial_y f(t,Y(t))\,. \end{aligned}$$

Hence, equation (8.8) is the same as

$$Y(t+h) \approx Y(t) + hf(t,Y(t)) + \tfrac{1}{2}h^2\Big(\partial_t f(t,Y(t)) + f(t,Y(t))\,\partial_y f(t,Y(t))\Big)\,.$$

If we let $t = t_n$ above (so that $t + h = t_{n+1}$) this is equivalent to

$$Y(t_{n+1}) \approx Y(t_n) + hf(t_n,Y(t_n)) + \tfrac{1}{2}h^2\Big(\partial_t f(t_n,Y(t_n)) + f(t_n,Y(t_n))\,\partial_y f(t_n,Y(t_n))\Big)\,.$$

Because y_n and y_{n+1} approximate $Y(t_n)$ and $Y(t_{n+1})$ respectively, this suggests setting

$$\begin{aligned} y_{n+1} &= y_n + hf(t_n,y_n) + \tfrac{1}{2}h^2\Big(\partial_t f(t_n,y_n) + f(t_n,y_n)\,\partial_y f(t_n,y_n)\Big) \\ &\qquad \text{for } n = 0, 1, \cdots, N-1\,. \end{aligned} \tag{8.9}$$

We call this the second-order Taylor-based method.

Remark. We can generalize our derivation of the second-order Taylor-based method by using the m^{th}-order Taylor approximation to derive an explicit numerical method whose error is $O(h^m)$ over the interval $[t_I, t_F]$ — a so-called *m^{th}-order method*. However, the formulas for these methods grow in complexity. For example, the third-order method is

$$\begin{aligned} y_{n+1} = y_n &+ hf(t_n,y_n) + \tfrac{1}{2}h^2\Big(\partial_t f(t_n,y_n) + f(t_n,y_n)\,\partial_y f(t_n,y_n)\Big) \\ &+ \tfrac{1}{6}h^3\Big[\partial_{tt} f(t_n,y_n) + 2f(t_n,y_n)\,\partial_{yt} f(t_n,y_n) + f(t_n,y_n)^2\partial_{yy} f(t_n,y_n) \\ &\qquad + \Big(\partial_t f(t_n,y_n) + f(t_n,y_n)\,\partial_y f(t_n,y_n)\Big)\partial_y f(t_n,y_n)\Big] \\ &\text{for } n = 0, 1, \cdots, N-1\,. \end{aligned} \tag{8.10}$$

This complexity of these methods makes them far less practical for general algorithms than the next class of methods we will study.

8.4. **Explicit One-Step Methods Based on Quadrature.** The starting point for our next class of methods will be the Fundamental Theorem of Calculus — specifically, the fact

$$Y(t+h) - Y(t) = \int_t^{t+h} Y'(s)\,\mathrm{d}s\,.$$

Because $Y(t)$ satisfies (8.1), this becomes

$$Y(t+h) = Y(t) + \int_t^{t+h} f(s, Y(s))\,\mathrm{d}s\,. \tag{8.11}$$

In 1895 Carl Runge proposed using quadrature rules (numerical integration) to construct approximations to the definite integral above in the form

$$\int_t^{t+h} f(s, Y(s))\,\mathrm{d}s = K(h, t, Y(t)) + O(h^{m+1})\,, \tag{8.12}$$

where m is some positive integer. The key point here is that $K(h, t, Y(t))$ depends on $Y(t)$, but does not depend on $Y(s)$ for any $s \neq t$. When approximation (8.12) is placed into (8.11) we obtain

$$Y(t+h) = Y(t) + K(h, t, Y(t)) + O(h^{m+1})\,.$$

If we let $t = t_n$ above (so that $t + h = t_{n+1}$) this is equivalent to

$$Y(t_{n+1}) = Y(t_n) + K(h, t_n, Y(t_n)) + O(h^{m+1})\,.$$

Because y_n and y_{n+1} approximate $Y(t_n)$ and $Y(t_{n+1})$ respectively, this suggests setting

$$y_{n+1} = y_n + K(h, t_n, y_n) \quad \text{for } n = 0, 1, \cdots, N-1\,, \tag{8.13}$$

Hence, every approximation of the form (8.12) yields the m^{th}-order explicit one-step method (8.13) for approximating solutions of (8.1). Here we will present methods associated with four basic quadrature rules that are covered in most calculus courses: the left-hand rule, the trapezoidal rule, the midpoint rule, and the Simpson rule.

8.4.1. *Explicit Euler Method Revisited Again.* The left-hand rule approximates the definite integral on the left-hand side of (8.12) as

$$\int_t^{t+h} f(s, Y(s))\,\mathrm{d}s = hf(t, Y(t)) + O(h^2)\,.$$

This approximation is already in the form (8.12) with $K(h, t, y) = hf(t, y)$. Method (8.13) thereby becomes

$$y_{n+1} = y_n + hf(t_n, y_n) \quad \text{for } n = 0, 1, \cdots, N-1\,,$$

which is exactly the explicit Euler method (8.3).

In practice, the explicit Euler method is implemented by initializing $y_0 = y_I$ and then for $n = 0, \cdots, N-1$ cycling through the instructions

$$f_n = f(t_n, y_n)\,, \qquad y_{n+1} = y_n + hf_n\,,$$

where $t_n = t_I + nh$.

Example. Let $Y(t)$ be the solution of the initial-value problem

$$\frac{\mathrm{d}y}{\mathrm{d}t} = t^2 + y^2\,, \qquad y(0) = 1\,.$$

Use the explicit Euler method with $h = .1$ to approximate $Y(.2)$.

Solution. We initialize $t_0 = 0$ and $y_0 = 1$. The explicit Euler method then gives

$$\begin{aligned} f_0 &= f(t_0, y_0) = 0^2 + 1^2 = 1 \\ y_1 &= y_0 + hf_0 = 1 + .1 \cdot 1 = 1.1 \\ f_1 &= f(t_1, y_1) = (.1)^2 + (1.1)^2 = .01 + 1.21 = 1.22 \\ y_2 &= y_1 + hf_1 = 1.1 + .1 \cdot 1.22 = 1.1 + .122 = 1.222 \end{aligned}$$

Therefore $Y(.2) \approx y_2 = 1.222$. □

The explicit Euler method is implemented by the following MATLAB function M-file.

```
function [t,y] = EulerExplicit(f, tI, yI, tF, N)

t = zeros(N + 1, 1); y = zeros(N + 1, 1);
t(1) = tI; y(1) = yI; h = (tF - tI)/N;
for j = 1:N
t(j + 1) = t(j) + h; y(j + 1) = y(j) + h*f(t(j), y(j));
end
```

Remark. There are some things you should notice. First, t(j) is t_{j-1} and y(j) is y_{j-1}, the approximation of $y(t_{j-1})$. In particular, y(j) is *not* the same as $Y(j)$, which denotes the solution $Y(t)$ evaluated at $t = j$. (You must pay attention to the font in which a letter is written!) The shift of the indices by one is needed because indexed variables in MATLAB begin with the index 1. In particular, t(1) and y(1) denote the initial time t_0 and value y_0. Consequently, all subsequent indices are shifted too, so that t(2) and y(2) denote t_1 and y_1, t(3) and y(3) denote t_2 and y_2, etc.

8.4.2. *Runge-Trapezoidal Method.* The trapezoidal rule approximates the definite integral on the left-hand side of (8.12) as

$$\int_t^{t+h} f(s, Y(s))\,\mathrm{d}s = \frac{h}{2}\big[f(t, Y(t)) + f(t+h, Y(t+h))\big] + O(h^3)\,.$$

This approximation is not in the form (8.12) because of the $Y(t+h)$ on the right-hand side. If we approximate this $Y(t+h)$ by the explicit Euler method then we obtain

$$\int_t^{t+h} f(s, Y(s))\,\mathrm{d}s = \frac{h}{2}\big[f(t, Y(t)) + f\big(t+h, Y(t) + hf(t, Y(t))\big)\big] + O(h^3)\,.$$

This approximation is in the form (8.12) with

$$K(h, t, y) = \frac{h}{2}\big[f(t, y) + f\big(t+h, y + hf(t, y)\big)\big]\,.$$

Method (8.13) thereby becomes

$$y_{n+1} = y_n + \frac{h}{2}\big[f(t_n, y_n) + f\big(t_{n+1}, y_n + hf(t_n, y_n)\big)\big] \quad \text{for } n = 0, 1, \cdots, N-1\,.$$

The book calls this the *improved Euler* method. However, that name is sometimes used for other methods by other books and is not very descriptive. Rather, we will call this the *Runge-trapezoidal* method because it was proposed by Runge based on the trapeziodal rule.

In practice, the Runge-trapezoidal method is implemented by initializing $y_0 = y_I$ and then for $n = 0, \cdots, N-1$ cycling through the instructions

$$f_n = f(t_n, y_n)\,, \qquad \tilde{y}_{n+1} = y_n + hf_n\,,$$
$$\tilde{f}_{n+1} = f(t_{n+1}, \tilde{y}_{n+1})\,, \qquad y_{n+1} = y_n + \tfrac{1}{2}h[f_n + \tilde{f}_{n+1}]\,,$$

where $t_n = t_I + nh$.

Example. Let $y(t)$ be the solution of the initial-value problem

$$\frac{\mathrm{d}y}{\mathrm{d}t} = t^2 + y^2\,, \qquad y(0) = 1\,.$$

Use the Runge-trapezoidal method with $h = .2$ to approximate $y(.2)$.

Solution. We initialize $t_0 = 0$ and $y_0 = 1$. The Runge-trapezoidal method then gives

$$f_0 = f(t_0, y_0) = 0^2 + 1^2 = 1$$
$$\tilde{y}_1 = y_0 + hf_0 = 1 + .2 \cdot 1 = 1.2$$
$$\tilde{f}_1 = f(t_1, \tilde{y}_1) = (.2)^2 + (1.2)^2 = .04 + 1.44 = 1.48$$
$$y_1 = y_0 + \tfrac{1}{2}h\big[f_0 + \tilde{f}_1\big] = 1 + .1 \cdot (1 + 1.48) = 1 + .1 \cdot 2.48 = 1.248$$

We then have $y(.2) \approx y_1 = 1.248$. □

Remark. Notice that two steps of the explicit Euler method with $h = .1$ gave $y(.2) \approx 1.222$, while one step of the Runge-trapezoidal method with $h = .2$ gave $y(.2) \approx 1.248$, which is much closer to the exact value. As these two calculations required roughly the same computational effort, this shows the advantange of using the second-order method.

The Runge-trapezoidal method is implemented by the following MATLAB function M-file.

```
function [t,y] = RungeTrap(f, tI, yI, tF, N)

t = zeros(N + 1, 1); y = zeros(N + 1, 1);
t(1) = tI; y(1) = yI; h = (tF - tI)/N; hhalf = h/2;
for j = 1:N
t(j + 1) = t(j) + h;
fnow = f(t(j), y(j));
yplus = y(j) + h*fnow;
fplus = f(t(j + 1), yplus);
y(j + 1) = y(j) + hhalf*(fnow + fplus);
end
```

Remark. Here t(j) and y(j) have the same meaning as they did in the M-file for the explicit Euler method. In particular, we have the same shift of the indices by one. Here we have introduced the so-called *working variables* fnow, yplus, and fplus to temporarily hold the values of f_{j-1}, $\tilde{y}_j$, and $\tilde{f}_j$ during each cycle of the loop. These values do not have to be saved, and so are overwritten with each new cycle. Here we have isolated the function evaluations for fnow and fplus into two separate lines. This is good coding practice that makes adaptations easier. For example, you can replace the function calls to f(t,y) by explicit formulas in those two lines without touching the rest of the coding.

8.4.3. *Runge-Midpoint Method.* The midpoint rule approximates the definite integral on the left-hand side of (8.12) as

$$\int_t^{t+h} f(s,Y(s))\,\mathrm{d}s = hf\big(t+\tfrac{1}{2}h, Y(t+\tfrac{1}{2}h)\big) + O(h^3)\,.$$

This approximation is not in the form (8.12) because of the $Y(t+\frac{1}{2}h)$ on the right-hand side. If we approximate this $Y(t+\frac{1}{2}h)$ by the explicit Euler method then we obtain

$$\int_t^{t+h} f(s,Y(s))\,\mathrm{d}s = hf\big(t+\tfrac{1}{2}h, Y(t)+\tfrac{1}{2}hf(t,Y(t))\big) + O(h^3)\,.$$

This approximation is in the form (8.12) with

$$K(h,t,y) = hf\big(t+\tfrac{1}{2}h, y+\tfrac{1}{2}hf(t,y)\big)\,.$$

Method (8.13) thereby becomes

$$y_{n+1} = y_n + hf\big(t_{n+\frac{1}{2}}, y_n+\tfrac{1}{2}hf(t_n,y_n)\big) \quad \text{for } n=0,1,\cdots,N-1\,.$$

The text calls this the *modified Euler* method. However, that name is sometimes used for other methods by other books and is not very descriptive. Rather, we will call this the *Runge-midpoint* method because it was proposed by Runge based on the midpoint rule.

In practice, the Runge-midpoint method is implemented by initializing $y_0 = y_I$ and then for $n = 0,\cdots,N-1$ cycling through the instructions

$$\begin{aligned} f_n &= f(t_n,y_n)\,, & y_{n+\frac{1}{2}} &= y_n + \tfrac{1}{2}hf_n\,,\\ f_{n+\frac{1}{2}} &= f(t_{n+\frac{1}{2}}, y_{n+\frac{1}{2}})\,, & y_{n+1} &= y_n + hf_{n+\frac{1}{2}}\,, \end{aligned}$$

where $t_n = t_I + nh$ and $t_{n+\frac{1}{2}} = t_I + (n+\frac{1}{2})h$.

Remark. The half-integer subscripts on $t_{n+\frac{1}{2}}$, $y_{n+\frac{1}{2}}$, and $f_{n+\frac{1}{2}}$ indicate that those variables are associated with the time $t = t_n + \frac{1}{2}h$, which is halfway between the times t_n and t_{n+1}. While it may seem strange at first, this notational device is a handy way to help keep track of the meanings of different variables.

Example. Let $y(t)$ be the solution of the initial-value problem

$$\frac{\mathrm{d}y}{\mathrm{d}t} = t^2 + y^2\,, \qquad y(0) = 1\,.$$

Use the Runge-midpoint method with $h = .2$ to approximate $y(.2)$.

Solution. We initialize $t_0 = 0$ and $y_0 = 1$. Then the Runge-midpoint method gives

$$\begin{aligned} f_0 &= f(t_0,y_0) = 0^2 + 1^2 = 1\\ y_{\frac{1}{2}} &= y_0 + \tfrac{1}{2}hf_0 = 1 + .1\cdot 1 = 1.1\\ f_{\frac{1}{2}} &= f(t_{\frac{1}{2}}, y_{\frac{1}{2}}) = (.1)^2 + (1.1)^2 = .01 + 1.21 = 1.22\,,\\ y_1 &= y_0 + hf_{\frac{1}{2}} = 1 + .2\cdot(1.22) = 1 + .244 = 1.244\,. \end{aligned}$$

We then have $y(.2) \approx y_1 = 1.244$. □

Remark. Notice that the Runge-trapezoidal method gave $y(.2) \approx 1.248$ while the Runge-midpoint method gave $y(.2) \approx 1.244$. The results are about the same because both methods are second-order. Here the Runge-trapezoidal method gave a better approximation. For other problems the Runge-midpoint method might give a better approximation.

The Runge-midpoint method is implemented by the following MATLAB function M-file.

```
function [t,y] = RungeMid(f, tI, yI, tF, N)

t = zeros(N + 1, 1); y = zeros(N + 1, 1);
t(1) = tI; y(1) = yI; h = (tF - tI)/N; hhalf = h/2;
for j = 1:N
thalf = t(j) + hhalf;
t(j + 1) = t(j) + h;
fnow = f(t(j), y(j));
yhalf = y(j) + hhalf*fnow;
fhalf = f(thalf, yhalf);
y(j + 1) = y(j) + h*fhalf;
end
```

Remark. Here t(j) and y(j) have the same meaning as they did in the M-file for the explicit Euler method. In particular, we have the same shift of the indices by one. Here we have introduced the working variables fnow, thalf, yhalf, and fhalf to temporarily hold the values of f_{j-1}, $t_{j-\frac{1}{2}}$, $y_{j-\frac{1}{2}}$, and $f_{j-\frac{1}{2}}$ during each cycle of the loop. These values do not have to be saved, and so are overwritten with each new cycle.

8.4.4. *Runge-Kutta Method.* The Simpson rule approximates the definite integral on the left-hand side of (8.12) as

$$\int_t^{t+h} f(s, Y(s))\,\mathrm{d}s = \frac{h}{6}\big[f(t, Y(t)) + 4f\big(t + \tfrac{1}{2}h, Y(t + \tfrac{1}{2}h)\big) + f\big(t + h, Y(t + h)\big)\big] + O(h^5)\,.$$

This approximation is not in the form (8.12) because of the $Y(t + \frac{1}{2}h)$ and $Y(t + h)$ on the right-hand side. If we approximate these with the explicit Euler method as we did before then we will degrade the local error to $O(h^3)$. We would like to find an approximation that is consistent with the $O(h^5)$ local error of the Simpson rule. In 1901 Wilhelm Kutta found such an approximation, which led to the so-called *Runge-Kutta* method. We will not give a derivation of this method here. Such derivations can be found in numerical analysis books.

In practice the Runge-Kutta method is implemented by initializing $y_0 = y_I$ and then for $n = 0, \cdots, N - 1$ cycling through the instructions

$$\begin{aligned}
f_n &= f(t_n, y_n)\,, & \tilde{y}_{n+\frac{1}{2}} &= y_n + \tfrac{1}{2}hf_n\,,\\
\tilde{f}_{n+\frac{1}{2}} &= f(t_{n+\frac{1}{2}}, \tilde{y}_{n+\frac{1}{2}})\,, & y_{n+\frac{1}{2}} &= y_n + \tfrac{1}{2}h\tilde{f}_{n+\frac{1}{2}}\,,\\
f_{n+\frac{1}{2}} &= f(t_{n+\frac{1}{2}}, y_{n+\frac{1}{2}})\,, & \tilde{y}_{n+1} &= y_n + hf_{n+\frac{1}{2}}\,,\\
\tilde{f}_{n+1} &= f(t_{n+1}, \tilde{y}_{n+1})\,, & y_{n+1} &= y_n + \tfrac{1}{6}h\big[f_n + 2\tilde{f}_{n+\frac{1}{2}} + 2f_{n+\frac{1}{2}} + \tilde{f}_{n+1}\big]\,,
\end{aligned}$$

where $t_n = t_I + nh$ and $t_{n+\frac{1}{2}} = t_I + (n + \frac{1}{2})h$.

Remark. The Runge-Kutta method requires four evaluations of $f(t, y)$ to advance each time step, whereas the second-order methods each required only two. Therefore it requires roughly twice as much computational work per time step as those methods.

Remark. Notice that because

$$\begin{aligned} y_n &\approx Y(t_n)\,, & f_n &\approx f\big(t_n, Y(t_n)\big) \\ \tilde{y}_{n+\frac{1}{2}} &\approx Y(t_n + \tfrac{1}{2}h)\,, & \tilde{f}_{n+\frac{1}{2}} &\approx f\big(t_n + \tfrac{1}{2}h, Y(t_n + \tfrac{1}{2}h)\big)\,, \\ y_{n+\frac{1}{2}} &\approx Y(t_n + \tfrac{1}{2}h)\,, & f_{n+\frac{1}{2}} &\approx f\big(t_n + \tfrac{1}{2}h, Y(t_n + \tfrac{1}{2}h)\big)\,, \\ \tilde{y}_{n+1} &\approx Y(t_n + h)\,, & \tilde{f}_{n+1} &\approx f\big(t_n + h, Y(t_n + h)\big)\,, \end{aligned}$$

we see that

$$y_{n+1} \approx Y(t_n) + \frac{h}{6}\big[f(t_n, Y(t_n)) + 4f\big(t_n + \tfrac{1}{2}h, Y(t_n + \tfrac{1}{2}h)\big) + f\big(t_n + h, Y(t_n + h)\big)\big]\,.$$

The Runge-Kutta method thereby looks consistant with the Simpson rule approximation. This argument does not show that the Runge-Kutta method is fourth order, but it is.

Example. Let $y(t)$ be the solution of the initial-value problem

$$\frac{\mathrm{d}y}{\mathrm{d}t} = t^2 + y^2\,, \qquad y(0) = 1\,.$$

Use the Runge-Kutta method with $h = .2$ to approximate $y(.2)$.

Solution. We initialize $t_0 = 0$ and $y_0 = 1$. The Runge-Kutta method then gives

$$\begin{aligned} f_0 &= f(t_0, y_0) = 0^2 + 1^2 = 1 \\ \tilde{y}_{\frac{1}{2}} &= y_0 + \tfrac{1}{2}hf_0 = 1 + .1 \cdot 1 = 1.1 \\ \tilde{f}_{\frac{1}{2}} &= f(t_{\frac{1}{2}}, \tilde{y}_{\frac{1}{2}}) = (.1)^2 + (1.1)^2 = .01 + 1.21 = 1.22 \\ y_{\frac{1}{2}} &= y_0 + \tfrac{1}{2}h\tilde{f}_{\frac{1}{2}} = 1 + .1 \cdot 1.22 = 1.122 \\ f_{\frac{1}{2}} &= f(t_{\frac{1}{2}}, y_{\frac{1}{2}}) = (.1)^2 + (1.122)^2 = .01 + 1.258884 = 1.268884 \\ \tilde{y}_1 &= y_0 + hf_{\frac{1}{2}} = 1 + .2 \cdot 1.268884 = 1 + .2517768 = 1.2517768 \\ \tilde{f}_1 &= f(t_1, \tilde{y}_1) = (.2)^2 + (1.2517768)^2 \approx .04 + 1.566945157 = 1.606945157 \\ y_1 &= y_0 + \tfrac{1}{6}h\big[f_0 + 2\tilde{f}_{\frac{1}{2}} + 2f_{\frac{1}{2}} + \tilde{f}_1\big] \\ &\approx 1 + .033333333\big[1 + 2 \cdot 1.22 + 2 \cdot 1.268884 + 1.606945157\big]\,. \end{aligned}$$

We then have $y(.2) \approx y_1 \approx 1.252823772$. Of course, you would not be expected to carry out such arithmetic calculations to nine decimal places on an exam. □

Remark. One step of the Runge-Kutta method with $h = .2$ yielded the approximation $y(.2) \approx 1.252823772$. This is more accurate than the approximations we had obtained with either second-order method. However, that is not a fair comparison because the Runge-Kutta method required roughly twice the computational work. A better comparison would be with the approximation produced by two steps of either second-order method with $h = .1$.

Remark. You will not be required to memorize the Runge-Kutta method. You also will not be required to carry out one step of it on an exam or quiz because, as the above example illustrates, the arithmetic gets messy even for fairly simple differential equations. However, you should understand the implications of it being a fourth-order method — namely, the relationship between its error and time step h. You also should be able to recognize the Runge-Kutta method if it is presented to you in MATLAB code.

The Runge-Kutta method is implemented by the following MATLAB function M-file.

```
function [t,y] = RungeKutta(f, tI, yI, tF, N)

t = zeros(N + 1, 1); y = zeros(N + 1, 1);
t(1) = tI; y(1) = yI; h = (tF - tI)/N; hhalf = h/2; hsixth = h/6;
for j = 1:N
thalf = t(j) + hhalf;
t(j + 1) = t(j) + h;
fnow = f(t(j), y(j));
yhalfone = y(j) + hhalf*fnow;
fhalfone = f(thalf, yhalfone);
yhalftwo = y(j) + hhalf*fhalfone;
fhalftwo = f(thalf, yhalftwo);
yplus = y(j) + h*fhalftwo;
fplus = f(t(j + 1), yplus);
y(j + 1) = y(j) + hsixth*(fnow + 2*fhalfone + 2*fhalftwo + fplus);
end
```

Remark. Here t(j) and y(j) have the same meaning as they did in the M-file for the explicit Euler method. In particular, we have the same shift of the indices by one. Here we have introduced the working variables fnow, thalf, yhalfone, fhalfone, yhalftwo, fhalftwo, yplus, and fplus to temporarily hold the values of f_{j-1}, $t_{j-\frac{1}{2}}$, $\tilde{y}_{j-\frac{1}{2}}$, $\tilde{f}_{j-\frac{1}{2}}$, $y_{j-\frac{1}{2}}$, $f_{j-\frac{1}{2}}$, $\tilde{y}_j$, and $\tilde{f}_j$.

8.4.5. *General Runge-Kutta Methods.* All the methods presented in this section are members of the family of general Runge-Kutta methods. The MATLAB command "ode45" uses the Dormand-Prince method, which is another member of this Runge-Kutta family that was discovered in 1980! The Runge-Kutta family continues to be enlarged by new methods, some of which might replace the Dormand-Prince method in future versions of MATLAB. An introduction to these modern methods requires a graduate course in numerical analysis. Here we have the more modest goal of introducing those family members presented by Wilhelm Kutta in his 1901 paper.

Carl Runge had described just a few methods in his 1895 paper, including the Runge trapezoid and midpoint methods. In 1900 Karl Heun presented a family of methods that included all those studied by Runge as special cases. Heun characterized the computational effort of these methods by how many evaluations of $f(t, y)$ are needed to compute $K(h, t, y)$. We say a method that requires s evaluations of $f(t, y)$ is an s-stage method. The explicit Euler method, for which $K(h, t, y) = hf(t, y)$, is the only one-stage method.

Heun considered the family of two-stage methods in the form

$$K(h,t,y) = \alpha_0 k_0 + \alpha_1 k_1\,, \qquad \text{with} \quad \alpha_0 + \alpha_1 = 1\,, \tag{8.14a}$$

where k_0 and k_1 are given by two evaluations of $f(t, y)$ as

$$k_0 = hf(t,y)\,, \qquad k_1 = hf(t+\beta h, y+\beta k_0)\,, \qquad \text{for some } \beta > 0\,. \tag{8.14b}$$

Heun showed that a two-stage method (8.14) is second-order for any $f(t, y)$ if and only if

$$\alpha_0 = 1 - \frac{1}{2\beta}\,, \qquad \alpha_1 = \frac{1}{2\beta}\,.$$

These include the Runge trapeziodal method, which is given by $\alpha_0 = \alpha_1 = \frac{1}{2}$ and $\beta = 1$, and the Runge midpoint method, which is given by $\alpha_0 = 0$, $\alpha_1 = 1$, and $\beta = \frac{1}{2}$. Heun also showed that no two-stage method (8.14) is third-order for any $f(t, y)$.

Remark. Second-order, two-stage methods are often called Heun methods in recognition of his work. Of these Heun favored the method given by $\alpha_0 = \frac{1}{4}$, $\alpha_1 = \frac{3}{4}$, and $\beta = \frac{2}{3}$, which is third order in the special case when $\partial_y f = 0$.

Heun also considered families of three- and four-stage methods in his 1900 paper. However in 1901 Kutta introduced families of s-stage methods that where more general when $s \geq 3$. For example, Kutta considered the family of three-stage methods in the form

$$K(h,t,y) = \alpha_0 k_0 + \alpha_1 k_1 + \alpha_2 k_2\,, \qquad \text{with} \quad \alpha_0 + \alpha_1 + \alpha_2 = 1\,, \tag{8.15a}$$

where k_0, k_1, and k_2 are given by three evaluations of $f(t, y)$ as

$$\begin{aligned} k_0 &= hf(t,y)\,, \\ k_1 &= hf(t+\beta_1 h, y+\gamma_{10}k_0)\,, && \text{with} \quad \beta_1 = \gamma_{10}\,, \\ k_2 &= hf(t+\beta_2 h, y+\gamma_{20}k_0+\gamma_{21}k_1)\,, && \text{with} \quad \beta_2 = \gamma_{20}+\gamma_{21}\,. \end{aligned} \tag{8.15b}$$

Kutta showed that a three-stage method (8.15) is second-order for any $f(t, y)$ if and only if

$$\alpha_1\beta_1 + \alpha_2\beta_2 = \tfrac{1}{2}\,;$$

and is third-order for any $f(t, y)$ if and only if in addition

$$\alpha_1\beta_1^2 + \alpha_2\beta_2^2 = \tfrac{1}{3}\,, \qquad \alpha_2\gamma_{21}\beta_1 = \tfrac{1}{6}\,.$$

Kutta also showed that no three-stage method (8.15) is fourth-order for any $f(t, y)$. Heun had shown the analogus results restricted to the case $\gamma_{20} = 0$. He favored the third-order method given by

$$\alpha_0 = \tfrac{1}{4}\,, \quad \alpha_1 = 0\,, \quad \alpha_2 = \tfrac{3}{4}\,, \quad \beta_1 = \gamma_{10} = \tfrac{1}{3}\,, \quad \beta_2 = \gamma_{21} = \tfrac{2}{3}\,, \quad \gamma_{20} = 0\,,$$

which is the third-order method requiring the fewest arithmetic operations. Kutta favored the third-order method given by

$$\alpha_0 = \tfrac{1}{6}\,, \quad \alpha_1 = \tfrac{2}{3}\,, \quad \alpha_2 = \tfrac{1}{6}\,, \quad \beta_1 = \gamma_{10} = \tfrac{1}{2}\,, \quad \beta_2 = 1\,, \quad \gamma_{20} = -1\,, \quad \gamma_{21} = 2\,,$$

which agrees with the Simpson rule in the special case when $\partial_y f = 0$.

Similarly, Kutta considered the family of four-stage methods in the form

$$K(h,t,y) = \alpha_0 k_0 + \alpha_1 k_1 + \alpha_2 k_2 + \alpha_3 k_3\,, \qquad \text{with} \quad \alpha_0 + \alpha_1 + \alpha_2 + \alpha_3 = 1\,, \tag{8.16a}$$

where k_0, k_1, k_2, and k_3 are given by four evaluations of $f(t, y)$ as

$$\begin{aligned} k_0 &= hf(t,y)\,, \\ k_1 &= hf(t+\beta_1 h, y+\gamma_{10}k_0)\,, && \text{with} \quad \beta_1 = \gamma_{10}\,, \\ k_2 &= hf(t+\beta_2 h, y+\gamma_{20}k_0+\gamma_{21}k_1)\,, && \text{with} \quad \beta_2 = \gamma_{20}+\gamma_{21}\,, \\ k_3 &= hf(t+\beta_3 h, y+\gamma_{30}k_0+\gamma_{31}k_1+\gamma_{32}k_2)\,, && \text{with} \quad \beta_3 = \gamma_{30}+\gamma_{31}+\gamma_{32}\,. \end{aligned} \tag{8.16b}$$

Kutta showed that a four-stage method (8.16) is second-order for any $f(t, y)$ if and only if

$$\alpha_1\beta_1 + \alpha_2\beta_2 + \alpha_3\beta_3 = \tfrac{1}{2}\,;$$

is third-order for any $f(t, y)$ if and only if in addition

$$\alpha_1\beta_1^2 + \alpha_2\beta_2^2 + \alpha_3\beta_3^2 = \tfrac{1}{3}\,, \qquad \alpha_2\gamma_{21}\beta_1 + \alpha_3\big(\gamma_{31}\beta_1 + \gamma_{32}\beta_2\big) = \tfrac{1}{6}\,;$$

and is fourth-order for any $f(t,y)$ if and only if in addition

$$\alpha_1\beta_1^3 + \alpha_2\beta_2^3 + \alpha_3\beta_3^3 = \tfrac{1}{4}\,, \qquad \alpha_2\gamma_{21}\beta_1^2 + \alpha_3\big(\gamma_{31}\beta_1^2 + \gamma_{32}\beta_2^2\big) = \tfrac{1}{12}\,,$$
$$\alpha_2\beta_2\gamma_{21}\beta_1 + \alpha_3\beta_3\big(\gamma_{31}\beta_1 + \gamma_{32}\beta_2\big) = \tfrac{1}{8}\,, \qquad \alpha_3\gamma_{32}\gamma_{21}\beta_1 = \tfrac{1}{24}\,, \qquad \beta_3 = 1\,.$$

Kutta also showed that no four-stage method (8.16) is fifth-order for any $f(t,y)$. Heun had shown the analogus results restricted to the case $\gamma_{20} = \gamma_{30} = \gamma_{31} = 0$. Kutta favored the classical Runge-Kutta method presented in the previous subsection, which is given by

$$\alpha_0 = \tfrac{1}{6}\,, \quad \alpha_1 = \tfrac{1}{3}\,, \quad \alpha_2 = \tfrac{1}{3}\,, \quad \alpha_3 = \tfrac{1}{6}\,,$$
$$\beta_1 = \gamma_{10} = \tfrac{1}{2}\,, \quad \beta_2 = \gamma_{21} = \tfrac{1}{2}\,, \quad \gamma_{20} = 0\,, \quad \beta_3 = \gamma_{32} = 1\,, \quad \gamma_{30} = \gamma_{31} = 0\,.$$

This is the fourth-order method that both requires the fewest arithmetic operations and is consistant with the Simpson rule.

More generally, Kutta considered the family of s-stage methods in the form

$$K(h,t,y) = \sum_{j=0}^{s-1} \alpha_j k_j\,, \qquad \text{with} \quad \sum_{j=0}^{s-1} \alpha_j = 1\,, \tag{8.17a}$$

where k_j for $j = 0, \cdots, s-1$ are given by s evaluations of $f(t,y)$ as

$$k_0 = hf(t,y)\,,$$
$$k_j = hf\left(t + \beta_j h\,,\; y + \sum_{i=0}^{j-1} \gamma_{ji} k_i\right)\,, \qquad \text{with} \quad \beta_j = \sum_{i=0}^{j-1} \gamma_{ji}\,, \quad \text{for } j = 1, \cdots, s-1\,. \tag{8.17b}$$

Kutta showed that no five-stage method (8.17) is fifth-order for any $f(t,y)$. This result was surprising because for $s = 1$, 2, 3, and 4 there were s-stage methods that were s^{th}-order. Kutta then characterized those six-stage methods (8.17) that are fifth-order for any $f(t,y)$. We will not give the conditions he found here.

Remark. Programmable electrionic computers were invented over fifty years after Runge, Heun, and Kutta carried out their work. Early numerical computations had less precision than they do today. Higher-order methods suffer from round-off error more than lower-order methods. Because round-off error is larger on machines with less precision, there was little advantage to using higher-order methods on early machines. As machines became more precise, the classical Runge-Kutta method became widely used to solve differential equations because it offers a nice balance between order and round-off error.

Remark. Perhaps the most important development in Runge-Kutta methods since their invention is *embedded methods*, which first appeared in 1957. These methods maintain a prescribed error tolerance for an m^{th}-order Runge-Kutta method by selecting a different h for each time step based on an error estimate made with a related $(m+1)^{th}$-order Runge-Kutta method. By "related" we mean that the two methods are built from the same evaluations of $f(t,y)$, so that they can be computed almost simultaniously. The MATLAB command "ode45" uses an embedded fourth-order/fifth-order method. Originally it used the Fehlberg embedded method, which was invented in 1969. Currently it uses the Dormand-Prince embedded method, which was invented in 1980. This method might be replaced by a higher-order embedded method as machines become faster and have smaller round-off error.

9. Exact Differential Forms and Integrating Factors

9.1. **Implicit General Solutions.** Consider a first-order ordinary equation in the form

$$\frac{\mathrm{d}y}{\mathrm{d}x} = f(x,y)\,, \tag{9.1}$$

where $f(x,y)$ is continuously differentiable over a region R of the xy-plane. Our basic existence and uniqueness theorem then insures that for every point (x_I, y_I) within the interior of R there exists a unique solution $y = Y(x)$ of the differential equation (9.1) that satisfies the initial condition $Y(x_I) = y_I$ for so long as $(x, Y(x))$ remains within the interior of R.

Let us ask the following question. When are the solutions of the differential equation (9.1) determined by an equation of the form

$$H(x,y) = c \qquad \text{where } c \text{ is some constant ?} \tag{9.2}$$

This means that we seek a function $H(x,y)$ defined over the interior of R such that the unique solution $y = Y(x)$ of the differential equation (9.1) that satisfies the initial condition $Y(x_I) = y_I$ is also the unique solution $y = Y(x)$ that satisfies

$$H(x,y) = H(x_I, y_I)\,, \qquad Y(x_I) = y_I\,. \tag{9.3}$$

Such an $H(x,y)$ is called an *integral* of (9.1). Because every solution of (9.1) that lies within the interior of R can be obtained in this way, we call relation (9.2) an *implicit general solution* of (9.1) over R.

The question can now be recast as "When does (9.1) have an integral?" This question is easily answered if we assume that all functions involved are as differentiable as we need. Suppose that an integral $H(x,y)$ exists, and that $y = Y(x)$ is a solution of differential equation (9.1). Then

$$H(x, Y(x)) = H(x_I, Y(x_I))\,,$$

where x_I is any point in the interval of definition of Y. By differentiating this equation with respect to x we find that

$$\partial_x H(x, Y(x)) + Y'(x)\, \partial_y H(x, Y(x)) = 0\,.$$

Therefore, wherever $\partial_y H(x, Y(x)) \neq 0$ we see that

$$Y'(x) = -\frac{\partial_x H(x, Y(x))}{\partial_y H(x, Y(x))}\,.$$

For this to hold for every solution of (9.1), we must have

$$\frac{\mathrm{d}y}{\mathrm{d}x} = -\frac{\partial_x H(x,y)}{\partial_y H(x,y)}\,,$$

or equivalently

$$f(x,y) = -\frac{\partial_x H(x,y)}{\partial_y H(x,y)}\,, \tag{9.4}$$

wherever $\partial_y H(x,y) \neq 0$. The question then arises as to whether we can find an $H(x,y)$ such that (9.4) holds for any given $f(x,y)$? It turns out that this cannot always be done. In this section we explore how to seek such an $H(x,y)$.

9.2. **Exact Differential Forms.** The starting point is to recast equation (9.1) in a so-called *differential form*

$$M(x,y)\,\mathrm{d}x + N(x,y)\,\mathrm{d}y = 0\,, \tag{9.5}$$

where $M(x,y)$ and $N(x,y)$ are continuously differentiable over a region R in the xy-plane and

$$f(x,y) = -\frac{M(x,y)}{N(x,y)}\,.$$

There is not a unique way to do this. Just pick one that looks natural. If you are lucky then there will exist a function $H(x,y)$ such that

$$\partial_x H(x,y) = M(x,y)\,, \qquad \partial_y H(x,y) = N(x,y)\,. \tag{9.6}$$

When this is the case the differential form (9.5) is said to be *exact* over the region R.

Leonhard Euler showed that there is a simple test you can apply to find out if you are lucky. It derives from the fact that "mixed partials commute" — namely, the fact that for any $H(x,y)$ that is twice continously differentiable over R we have

$$\partial_y\big(\partial_x H(x,y)\big) = \partial_x\big(\partial_y H(x,y)\big)\,.$$

This fact implies that if (9.6) holds for such an $H(x,y)$ then $M(x,y)$ and $N(x,y)$ satisfy

$$\partial_y M(x,y) = \partial_y\big(\partial_x H(x,y)\big) = \partial_x\big(\partial_y H(x,y)\big) = \partial_x N(x,y)\,.$$

In other words, if the differential form (9.5) is exact then $M(x,y)$ and $N(x,y)$ satisfy

$$\partial_y M(x,y) = \partial_x N(x,y)\,. \tag{9.7}$$

Euler showed that whenever R has no holes the converse holds too. Namely, if the differential form (9.5) satisfies (9.7) for every (x,y) in R then it is exact — i.e. there exists an $H(x,y)$ such that (9.6) holds. Moreover, the problem of finding $H(x,y)$ is reduced to finding two primitives. We illustrate this fact with examples.

Example. Solve the initial-value problem

$$\frac{\mathrm{d}y}{\mathrm{d}x} + \frac{e^x y + 2x}{2y + e^x} = 0\,, \quad y(0) = 0\,.$$

Solution. Express this equation in the differential form

$$(e^x y + 2x)\,\mathrm{d}x + (2y + e^x)\,\mathrm{d}y = 0\,.$$

Because

$$\partial_y(e^x y + 2x) = e^x \qquad = \qquad \partial_x(2y + e^x) = e^x\,,$$

this differential form satisfies (9.7) everywhere in the xy-plane and thereby is *exact*. Therefore we can find $H(x,y)$ such that

$$\partial_x H(x,y) = e^x y + 2x\,, \qquad \partial_y H(x,y) = 2y + e^x\,. \tag{9.8}$$

You can now integrate either equation, and plug the result into the other equation to obtain a second equation to integrate.

If we first integrate the first equation in (9.8) then we find that

$$H(x,y) = \int (e^x y + 2x)\,\mathrm{d}x = e^x y + x^2 + h(y)\,.$$

Here we are integrating with respect to x while treating y as a constant. The function $h(y)$ is the "constant of integration". We plug this expression for $H(x,y)$ into the second equation in (9.8) to obtain

$$e^x + h'(y) = \partial_y H(x,y) = 2y + e^x \,.$$

This reduces to $h'(y) = 2y$. Notice that this equation for $h'(y)$ only depends on y. Take $h(y) = y^2$, so that $H(x,y) = e^x y + x^2 + y^2$ is an integral of the differential equation. Therefore an implicit general solution is

$$H(x,y) = e^x y + x^2 + y^2 = c \,.$$

The initial condition $y(0) = 0$ implies that

$$c = e^0 \cdot 0 + 0^2 + 0^2 = 0 \,.$$

Therefore

$$y^2 + e^x y + x^2 = 0 \,.$$

The quadratic formula then yields

$$y = \frac{-e^x + \sqrt{e^{2x} - 4x^2}}{2} \,,$$

where the positive square root is taken so that solution satisfies the initial condition. This is a solution wherever $e^{2x} > 4x^2$. □

Alternative Solution. If we first integrate the second equation in (9.8) then we find that

$$H(x,y) = \int (2y + e^x)\,\mathrm{d}y = y^2 + e^x y + h(x) \,.$$

Here we are integrating with respect to y while treating x as a constant. The function $h(x)$ is the "constant of integration". We plug this expression for $H(x,y)$ into the first equation in (9.8) to obtain

$$e^x y + h'(x) = \partial_x H(x,y) = e^x y + 2x \,.$$

This reduces to $h'(x) = 2x$. Notice that this equation for $h'(x)$ only depends on x. Taking $h(x) = x^2$, so $H(x,y) = e^x y + x^2 + y^2$, we see that a general solution satisfies

$$e^x y + x^2 + y^2 = c \,.$$

Because this is the same relation for a general solution that we had found previously, the evaluation of c is done as before. □

The points to be made here are the following:

- In principle you can integrate either equation in (9.7) first.
- If you integrate with respect to x first then the "constant of integration" $h(y)$ will depend on y and the equation for $h'(y)$ should only depend on y.
- If you integrate with respect to y first then the "constant of integration" $h(x)$ will depend on x and the equation for $h'(x)$ should only depend on x.
- In either case, if your equation for h' involves both x and y you have made a mistake!

Sometimes the differential equation will be given to you already in differential form. In that case, use that form as the starting point.

Example. Give an implicit general solution to the differential equation

$$(xy^2 + y + e^x)\,\mathrm{d}x + (x^2 y + x)\,\mathrm{d}y = 0 \,.$$

Solution. Because

$$\partial_y(xy^2 + y + e^x) = 2xy + 1 \qquad = \qquad \partial_x(x^2y + x) = 2xy + 1\,.$$

this differential form satisfies (9.7) everywhere in the xy-plane and thereby is exact. Therefore we can find $H(x, y)$ such that

$$\partial_x H(x, y) = xy^2 + y + e^x\,, \qquad \partial_y H(x, y) = x^2y + x\,.$$

By integrating the second equation you obtain

$$H(x, y) = \int (x^2y + x)\,\mathrm{d}y = \tfrac{1}{2}x^2y^2 + xy + h(x)\,.$$

When we plug this expression for $H(x, y)$ into the first equation we obtain

$$xy^2 + y + h'(x) = \partial_x H(x, y) = xy^2 + y + e^x\,,$$

which yields $h'(x) = e^x$. (Notice that this only depends on x!) Take $h(x) = e^x$, so that $H(x, y) = \frac{1}{2}x^2y^2 + xye^x$. Therefore an implicit general solution is

$$\tfrac{1}{2}x^2y^2 + xy + e^x = c\,.$$

□

In the last example we could just as easily have integrated the equation for $\partial_x H(x, y)$ first and plugged the resulting expression into the equation for $\partial_y H(x, y)$. The next example shows that it can be helpful to first integrate whichever equation for $H(x, y)$ is easier to integrate.

Example. Give an implicit general solution to the differential equation

$$\big(x^2\cos(x)e^y + 2x\sin(x)e^y + e^x\cos(y)\big)\,\mathrm{d}x + \big(x^2\sin(x)e^y - e^x\sin(y)\big)\,\mathrm{d}y = 0\,.$$

Solution. Because

$$\begin{aligned}\partial_y\big(x^2\cos(x)e^y + 2x\sin(x)e^y + e^x\cos(y)\big) &= x^2\cos(x)e^y + 2x\sin(x)e^y - e^x\sin(y)\,,\\ \partial_x\big(x^2\sin(x)e^y - e^x\sin(y)\big) &= x^2\cos(x)e^y + 2x\sin(x)e^y - e^x\sin(y)\,,\end{aligned}$$

this differential form satisfies (9.7) and thereby is exact. Therefore we can find $H(x, y)$ such that

$$\begin{aligned}\partial_x H(x, y) &= x^2\cos(x)e^y + 2x\sin(x)e^y + e^x\cos(y)\,,\\ \partial_y H(x, y) &= x^2\sin(x)e^y - e^x\sin(y)\,.\end{aligned}$$

Now notice that it is more apparent how to integrate the bottom equation in y than how to integrate the top equation in x. (Recall that integrating terms like $x^2\cos(x)$ requires two integration-by-parts.) So integrating the bottom equation we obtain

$$H(x, y) = \int \big(x^2\sin(x)e^y - e^x\sin(y)\big)\,\mathrm{d}y = x^2\sin(x)e^y + e^x\cos(y) + h(x)\,.$$

When we plug this expression for $H(x, y)$ into the top equation we obtain

$$x^2\cos(x)e^y + 2x\sin(x)e^y + e^x\cos(y) + h'(x) = \partial_x H(x, y) = x^2\cos(x)e^y + 2x\sin(x)e^y + e^x\cos(y)\,,$$

which yields $h'(x) = 0$. Taking $h(x) = 0$, so that $H(x, y) = x^2\sin(x)e^y + e^x\cos(y)$, we see that a general solution is given by

$$x^2\sin(x)e^y + e^x\cos(y) = c\,.$$

□

Remark. Of course, had you seen that $x^2\cos(x) + 2x\sin(x)$ is the derivative of $x^2\sin(x)$ then you could have just as easily started by integrating the $\partial_x H(x,y)$ equation with respect to x in the previous example. But such insights do not always arrive when you need them.

We will now derive formulas for $H(x,y)$ in terms of definite integrals that apply whenever R is a rectangle in the xy-plane and (x_I, y_I) is any point that lies within the interior of R. These formulas will encode the two steps given above. They thereby show that those steps can always be carried out in this setting. We consider a differential form

$$M(x,y)\,\mathrm{d}x + N(x,y)\,\mathrm{d}y = 0\,, \tag{9.9}$$

where $M(x,y)$ and $N(x,y)$ are continuously differentiable over R and satisfy

$$\partial_y M(x,y) = \partial_x N(x,y)\,. \tag{9.10}$$

Now seek $H(x,y)$ such that

$$\partial_x H(x,y) = M(x,y)\,, \qquad \partial_y H(x,y) = N(x,y)\,. \tag{9.11}$$

By integrating the first equation with respect to x we obtain

$$H(x,y) = \int_{x_I}^{x} M(r,y)\,\mathrm{d}r + h(y)\,.$$

When we plug this expression for $H(x,y)$ into the second equation, use (9.10) to assert that $\partial_y M(r,y) = \partial_r N(r,y)$, and apply the First Fundamental Theorem of Calculus, we obtain

$$\begin{aligned} N(x,y) = \partial_y H(x,y) &= \int_{x_I}^{x} \partial_y M(r,y)\,\mathrm{d}r + h'(y) \\ &= \int_{x_I}^{x} \partial_r N(r,y)\,\mathrm{d}r + h'(y) = N(x,y) - N(x_I,y) + h'(y)\,. \end{aligned}$$

This yields $h'(y) = N(x_I, y)$, which only depends on y because x_I is a number. Let

$$h(y) = \int_{y_I}^{y} N(x_I,s)\,\mathrm{d}s\,.$$

An implicit general solution of (9.9) thereby is $H(x,y) = c$, where $H(x,y)$ is given by

$$H(x,y) = \int_{x_I}^{x} M(r,y)\,\mathrm{d}r + \int_{y_I}^{y} N(x_I,s)\,\mathrm{d}s\,.$$

Notice that $H(x_I, y_I) = 0$. If the second equation in (9.11) had been integrated first then we would have found that an implicit general solution of (9.9) is $H(x,y) = c$, where $H(x,y)$ is given by

$$H(x,y) = \int_{x_I}^{x} M(r,y_I)\,\mathrm{d}r + \int_{y_I}^{y} N(x,s)\,\mathrm{d}s\,.$$

The above formulas give two expressions for the same function $H(x,y)$. Rather than memorize these formulas, I strongly recommend that you simply learn the steps underlying them.

Remark. In the two examples given previously the rectange R was the entire xy-plane. This will be the case whenever $M(x,y)$ and $N(x,y)$ appearing in the differential form (9.9) are continuously differentiable over the entire xy-plane and satisfy (9.10).

Remark. Our recipe for separable equations can be viewed as a special case of our recipe for exact differential forms. Consider the separable first-order ordinary differential equation

$$\frac{\mathrm{d}y}{\mathrm{d}x} = f(x)g(y)\,.$$

It can be put into the differential form

$$f(x)\,\mathrm{d}x - \frac{1}{g(y)}\,\mathrm{d}y = 0\,.$$

This differential form is exact because

$$\partial_y f(x) = 0 \qquad = \qquad \partial_x\left(\frac{1}{g(y)}\right) = 0\,.$$

Therefore we can find $H(x,y)$ such that

$$\partial_x H(x,y) = f(x)\,, \qquad \partial_y H(x,y) = \frac{1}{g(y)}\,.$$

Indeed, we find that

$$H(x,y) = F(x) - G(y)\,, \qquad \text{where} \quad F'(x) = f(x) \quad \text{and} \quad G'(y) = \frac{1}{g(y)}\,.$$

Therefore an implicit general solution is $F(x) - G(y) = c$. This is precisely the recipe for solving separable equations that we derived earlier.

9.3. **Integrating Factors.** Suppose you had considered the differential form

$$M(x,y)\,\mathrm{d}x + N(x,y)\,\mathrm{d}y = 0\,, \tag{9.12}$$

and found that is not exact. Just because you were unlucky the first time, do not give up! Recall that this differential form has the same solutions as any differential form in the form

$$M(x,y)\mu(x,y)\,\mathrm{d}x + N(x,y)\mu(x,y)\,\mathrm{d}y = 0\,, \tag{9.13}$$

where $\mu(x,y)$ any *nonzero* function. Indeed, both (9.12) and (9.13) are differential forms associated with the first-order differential equation

$$\frac{\mathrm{d}y}{\mathrm{d}x} = f(x,y)\,, \qquad \text{where} \quad f(x,y) = -\frac{M(x,y)}{N(x,y)} = -\frac{M(x,y)\mu(x,y)}{N(x,y)\mu(x,y)}\,.$$

Therefore we can seek a nonzero function $\mu(x,y)$ that makes the differential form (9.13) exact! This means that $\mu(x,y)$ must satisfy

$$\partial_y\big[M(x,y)\mu(x,y)\big] = \partial_x\big[N(x,y)\mu(x,y)\big]\,.$$

Expanding the above partial derivatives using the product rule, we see that μ must satisfy

$$M(x,y)\partial_y\mu + [\partial_y M(x,y)]\mu = N(x,y)\partial_x\mu + [\partial_x N(x,y)]\mu\,. \tag{9.14}$$

This is a first-order linear partial differential equation for μ. Finding its general solution is equivalent to finding the general solution of the original ordinary differential equation. Fortunately, we do not need this general solution. All we need is one nonzero solution. Such a μ is called an *integrating factor* for the differential form (9.12).

A trick that sometimes yields a solution of (9.14) is to assume either that μ is only a function of x, or that μ is only a function of y. When μ is only a function of x then $\partial_y \mu = 0$ and (9.14) reduces to the first-order linear ordinary differential equation

$$\frac{\mathrm{d}\mu}{\mathrm{d}x} = \frac{\partial_y M(x,y) - \partial_x N(x,y)}{N(x,y)}\,\mu\,.$$

This equation will be consistent with our assumption that μ is only a function of x when the fraction on its right-hand side is independent of y. In that case we can integrate the equation to find the integrating factor

$$\mu(x) = e^{A(x)}\,, \qquad \text{where} \quad A'(x) = \frac{\partial_y M(x,y) - \partial_x N(x,y)}{N(x,y)}\,.$$

Similarly, when μ is only a function of y then $\partial_x \mu = 0$ and (9.14) reduces to the first-order linear ordinary differential equation

$$\frac{\mathrm{d}\mu}{\mathrm{d}y} = \frac{\partial_x N(x,y) - \partial_y M(x,y)}{M(x,y)}\,\mu\,.$$

This equation will be consistent with our assumption that μ is only a function of y when the fraction on its right-hand side is independent of x. In that case we can integrate the equation to find the integrating factor

$$\mu(y) = e^{B(y)}\,, \qquad \text{where} \quad B'(y) = \frac{\partial_x N(x,y) - \partial_y M(x,y)}{M(x,y)}\,.$$

This will be the only method for finding integrating factors that we will use in this course.

Remark. Rather than memorize the above formulas for $\mu(x)$ and $\mu(y)$ in terms of primitives, I strongly recommend that you simply follow the steps by which they were derived. Namely, you seek an integrating factor μ that satisfies

$$\partial_y[M(x,y)\,\mu] = \partial_x[N(x,y)\,\mu]\,.$$

You then expand the partial derivatives using the product rule as

$$M(x,y)\partial_y\mu + [\partial_y M(x,y)]\,\mu = N(x,y)\partial_x\mu + [\partial_x N(x,y)]\,\mu\,,$$

and combine the μ terms. If the resulting equation reduces to an equation that only depends on x when you set $\partial_y\mu = 0$ then there is an integrating factor $\mu(x)$. On the other hand, if the equation reduces to an equation that only depends on y when you set $\partial_x\mu = 0$ then there is an integrating factor $\mu(y)$. We will illustrate this approach with the following examples.

Example. Give an implicit general solution to the differential equation

$$(2e^x + y^3)\,\mathrm{d}x + 3y^2\,\mathrm{d}y = 0\,.$$

Solution. This differential form is not exact because

$$\partial_y(2e^x + y^3) = 3y^2 \qquad \neq \qquad \partial_x(3y^2) = 0\,.$$

Therefore we seek an integrating factor μ such that

$$\partial_y[(2e^x + y^3)\mu] = \partial_x[(3y^2)\mu]\,.$$

Expanding the partial derivatives using the product rule gives

$$(2e^x + y^3)\partial_y\mu + 3y^2\mu = 3y^2\partial_x\mu\,.$$

Notice that if $\partial_y \mu = 0$ then this equation reduces to $\mu = \partial_x \mu$, whereby $\mu(x) = e^x$ is an integrating factor. (See how easy that was!)

Because e^x is an integrating factor, we *know* that

$$(2e^x + y^3)e^x \, \mathrm{d}x + 3y^2 e^x \, \mathrm{d}y = 0 \quad \text{is exact}\,.$$

(Of course, you should check that this is exact. If it is not then you made a mistake in finding μ!) Therefore we can find $H(x, y)$ such that

$$\partial_x H(x, y) = 2e^{2x} + y^3 e^x\,, \qquad \partial_y H(x, y) = 3y^2 e^x\,.$$

By intgrating the second equation we see that $H(x, y) = y^3 e^x + h(x)$. When this expression for $H(x, y)$ is plugged into the first equation we obtain

$$y^3 e^x + h'(x) = \partial_x H(x, y) = 2e^{2x} + y^3 e^x\,,$$

which yields $h'(x) = 2e^{2x}$. Upon taking $h(x) = e^{2x}$, so that $H(x, y) = y^3 e^x + e^{2x}$, a general solution satisfies

$$y^3 e^x + e^{2x} = c\,.$$

In this case the general solution can be given explicitly as

$$y = (ce^{-x} - e^x)^{\frac{1}{3}}\,,$$

where c is an arbitrary constant. □

Example. Give an implicit general solution to the differential equation

$$2xy \, \mathrm{d}x + (2x^2 - e^y) \, \mathrm{d}y = 0\,.$$

Solution. This differential form is not exact because

$$\partial_y(2xy) = 2x \qquad \neq \qquad \partial_x(2x^2 - e^y) = 4x\,.$$

Therefore we seek an integrating factor μ such that

$$\partial_y[(2xy)\mu] = \partial_x[(2x^2 - e^y)\mu]\,.$$

Expanding the partial derivatives using the product rule gives

$$2xy\partial_y \mu + 2x\mu = (2x^2 - e^y)\partial_x \mu + 4x\mu\,.$$

Combining the μ terms then yields

$$2xy\partial_y \mu = (2x^2 - e^y)\partial_x \mu + 2x\mu\,.$$

Notice that if $\partial_x \mu = 0$ then this equation reduces to $y\partial_y \mu = \mu$, whereby $\mu(y) = y$ is an integrating factor. (See how easy that was!)

Because y is an integrating factor, we *know* that

$$2xy^2 \, \mathrm{d}x + (2x^2 - e^y)y \, \mathrm{d}y = 0 \quad \text{is exact}\,.$$

Therefore we can find $H(x, y)$ such that

$$\partial_x H(x, y) = 2xy^2\,, \qquad \partial_y H(x, y) = 2x^2 y - e^y y\,.$$

By intgrating the first equation we see that $H(x, y) = x^2 y^2 + h(y)$. When this expression for $H(x, y)$ is plugged into the second equation we obtain

$$2x^2 y + h'(y) = \partial_y H(x, y) = 2x^2 y - e^y y\,,$$

which yields $h'(y) = -e^y y$. Upon taking $h(y) = e^y(1-y)$, so that $H(x,y) = x^2y^2 + e^y(1-y)$, a general solution satisfies

$$x^2y^2 + e^y(1-y) = c\,.$$

In this case we cannot solve for y explicitly. □

Remark. Sometimes it might not be evident whether we should set $\partial_y\mu = 0$ or $\partial_x\mu = 0$ when searching for the integrating factor μ. The next example illustrates such a case.

Example. Give an implicit general solution to the differential equation

$$(4xy + 3y^3)\,\mathrm{d}x + (x^2 + 3xy^2)\,\mathrm{d}y = 0\,.$$

Solution. This differential form is *not exact* because

$$\partial_y(4xy + 3y^3) = 4x + 9y^2 \qquad \neq \qquad \partial_x(x^2 + 3xy^2) = 2x + 3y^2\,.$$

Therefore we seek an *integrating factor* μ such that

$$\partial_y[(4xy + 3y^3)\mu] = \partial_x[(x^2 + 3xy^2)\mu]\,.$$

Expanding the partial derivatives gives

$$(4xy + 3y^3)\partial_y\mu + (4x + 9y^2)\mu = (x^2 + 3xy^2)\partial_x\mu + (2x + 3y^2)\mu\,.$$

Combining the μ terms yields

$$(4xy + 3y^3)\partial_y\mu + (2x + 6y^2)\mu = (x^2 + 3xy^2)\partial_x\mu\,.$$

At this point it might not be evident whether we should set $\partial_y\mu = 0$ or $\partial_x\mu = 0$. However, the picture becomes clear once we notice that $(x + 3y^2)$ is a common factor of the last two terms. Hence, if we set $\partial_y\mu = 0$ then this becomes

$$2(x + 3y^2)\mu = (x + 3y^2)x\partial_x\mu\,,$$

which reduces to $2\mu = x\partial_x\mu$. This yields the integrating factor $\mu = x^2$.

Because x^2 is an integrating factor, the differential form

$$(4xy + 3y^3)x^2\,\mathrm{d}x + (x^2 + 3xy^2)x^2\,\mathrm{d}y = 0 \quad \text{is exact}\,.$$

Therefore we can find $H(x,y)$ such that

$$\partial_x H(x,y) = 4x^3y + 3x^2y^3\,, \qquad \partial_y H(x,y) = x^4 + 3x^3y^2\,.$$

Integrating the first equation with respect to x yields

$$H(x,y) = x^4y + x^3y^3 + h(y)\,,$$

whereby

$$\partial_y H(x,y) = x^4 + 3x^3y^2 + h'(y)\,.$$

Plugging this expression for $\partial_y H(x,y)$ into the second equation gives

$$x^4 + 3x^3y^2 + h'(y) = x^4 + 3x^3y^2\,,$$

which yields $h'(y) = 0$. Taking $h(y) = 0$, an implicit general solution is therefore given by

$$x^4y + x^3y^3 = c\,.$$

Solving for y explicitly requires the cubic formula, which you are not expected to know. □

Remark. Integrating factors for the linear equations can be viewed as a special case of the foregoing method. Consider the linear first-order ordinary differential equation

$$\frac{\mathrm{d}y}{\mathrm{d}x} + a(x)y = f(x)\,.$$

It can be put into the differential form

$$\big(a(x)y - f(x)\big)\,\mathrm{d}x + \mathrm{d}y = 0\,.$$

This differential form is generally not exact because when $a(x) \neq 0$ we have

$$\partial_y\big(a(x)y - f(x)\big) = a(x) \qquad \neq \qquad \partial_x 1 = 0\,.$$

Therefore we seek an integrating factor μ such that

$$\partial_y\big[\big(a(x)y - f(x)\big)\,\mu\big] = \partial_x\mu\,.$$

Expanding the partial derivatives by the product rule gives

$$\big(a(x)y - f(x)\big)\,\partial_y\mu + a(x)\mu = \partial_x\mu\,.$$

Notice that if $\partial_y\mu = 0$ then this equation reduces to $a(x)\mu = \partial_x\mu$, whereby an integrating factor is $\mu(x) = e^{A(x)}$ where $A'(x) = a(x)$.

Because $e^{A(x)}$ is an integrating factor, we *know* that

$$e^{A(x)}\big(a(x)y - f(x)\big)\,\mathrm{d}x + e^{A(x)}\,\mathrm{d}y = 0 \quad \text{is exact}\,.$$

Therefore we can find $H(x,y)$ such that

$$\partial_x H(x,y) = e^{A(x)}\big(a(x)y - f(x)\big)\,, \qquad \partial_y H(x,y) = e^{A(x)}\,.$$

By intgrating the second equation we see that $H(x,y) = e^{A(x)}y + h(x)$. When this expression for $H(x,y)$ is plugged into the first equation we obtain

$$e^{A(x)}a(x)y + h'(x) = \partial_x H(x,y) = e^{A(x)}\big(a(x)y - f(x)\big)\,,$$

which yields $h'(x) = -e^{A(x)}f(x)$. Therefore a general solution is $H(x,y) = c$ with $H(x,y)$ given by

$$H(x,y) = e^{A(x)}y - B(x)\,, \qquad \text{where} \quad A'(x) = a(x) \quad \text{and} \quad B'(x) = e^{A(x)}f(x)\,.$$

This can be solved to obtain the explicit general solution

$$y = e^{-A(x)}c + e^{-A(x)}B(x)\,.$$

This is equivalent to the recipe for solving linear equations that we derived previously.

10. Special First-Order Equations and Substitution

So far we have developed analytical methods for linear equations, separable equations, and equations that can be expressed as an exact differential form. There are other first-order equations that can be solved by analytical methods. Here we present a few of them. In each case a substitution will transform the problem into a form that we know how to solve.

10.1. **Linear Argument Equations.** These equations can be put into the form

$$\frac{\mathrm{d}y}{\mathrm{d}x} = k(ax+by)\,, \tag{10.1}$$

where $k(z)$ is a differentiable function over an interval (z_L, z_R) while a and b are constants with $b \neq 0$. Upon setting $z = ax + by$ we see that

$$\frac{\mathrm{d}z}{\mathrm{d}x} = \frac{\mathrm{d}}{\mathrm{d}x}(ax+by) = a + b\,\frac{\mathrm{d}y}{\mathrm{d}x} = a + b\,k(ax+by) = a + b\,k(z)\,.$$

Therefore z satisfies the autonomous equation

$$\frac{\mathrm{d}z}{\mathrm{d}x} = a + b\,k(z)\,. \tag{10.2}$$

If we can solve this equation for z then the solution of (10.1) is obtained by setting

$$y = \frac{z-ax}{b}\,.$$

Solutions of (10.2) are given implicitly by

$$G(z) = x + c\,, \qquad \text{where} \quad G'(z) = \frac{1}{a+b\,k(z)}\,. \tag{10.3}$$

Of course, we will not be able to find an explicit primitive $G(z)$ for every $k(z)$. And when we can find $G(z)$, often we will not be able to solve (10.3) for z as an explicit function of x.

Example. Solve the equation

$$\frac{\mathrm{d}y}{\mathrm{d}x} = (x+y)^2\,.$$

Solution. This has the form (10.1) with $k(z) = z^2$ and $a = b = 1$. Rather than remember the form (10.2), it is easier to remember the substitution $z = x + y$ and rederive (10.2). Indeed, we see that

$$\frac{\mathrm{d}z}{\mathrm{d}x} = \frac{\mathrm{d}}{\mathrm{d}x}(x+y) = 1 + \frac{\mathrm{d}y}{\mathrm{d}x} = 1 + (x+y)^2 = 1 + z^2\,.$$

Solutions of this autonomous equation satisfy

$$x = \int \frac{1}{1+z^2}\,\mathrm{d}z = \tan^{-1}(z) + c\,.$$

This equation can be solved explicitly to obtain $z = \tan(x-c)$. Because $y = z - x$, a family of solutions to the original equation is

$$y = \tan(x-c) - x\,.$$

□

10.2. **Dilation Invariant Equations.** These equations can be put into the form

$$\frac{\mathrm{d}y}{\mathrm{d}x} = x^{p-1}k\Big(\frac{y}{x^p}\Big)\,, \qquad \text{for some } p \neq 0\,, \tag{10.4}$$

where $k(z)$ is a differentiable function over $(-\infty, \infty)$. A first-order equation in the form

$$\frac{\mathrm{d}y}{\mathrm{d}x} = f(x,y)\,,$$

can be put into the form (10.4) if and only if $f(x,y)$ satisfies the dilation symmetry

$$f(\lambda x, \lambda^p y) = \lambda^{p-1} f(x,y) \qquad \text{for every } \lambda \neq 0\,. \tag{10.5}$$

Indeed, if $f(x,y)$ satisfies has this symmetry then by choosing $\lambda = 1/x$ we see that

$$f(x,y) = \lambda^{1-p}\, f(\lambda x, \lambda^p y) = x^{p-1} f\left(\frac{1}{x}\,x\,,\, \frac{1}{x^p}\,y\right) = x^{p-1} f\Big(1, \frac{y}{x^p}\Big)\,,$$

whereby $k(z) = f(1,z)$. When $f(x,y)$ satisfies (10.5) with $p = 1$ then equation (10.4) is said to be *homogeneous*. This notion of homogeneous should not be confused with the notion of homogeneous that arises in the context of linear equations.

We can transform (10.4) into a separable equation by setting $z = y/x^p$. By using (10.4) we see that

$$\begin{aligned}\frac{\mathrm{d}z}{\mathrm{d}x} = \frac{1}{x^p}\,\frac{\mathrm{d}y}{\mathrm{d}x} - p\,\frac{y}{x^{p+1}} &= \frac{1}{x^p}\,x^{p-1}k\Big(\frac{y}{x^p}\Big) - p\,\frac{y}{x^{p+1}} \\ &= \frac{1}{x}\left(k\Big(\frac{y}{x^p}\Big) - \frac{y}{x^p}\right) = \frac{1}{x}\big(k(z) - pz\big)\,.\end{aligned}$$

Therefore z satisfies the separable equation

$$\frac{\mathrm{d}z}{\mathrm{d}x} = \frac{k(z) - pz}{x}\,. \tag{10.6}$$

If we can solve this equation for z then the solution of (10.4) is obtained by setting $y = zx^p$.

Solutions of (10.6) are given implicitly by

$$\log(|x|) = G(z) + c\,, \qquad \text{where} \quad G'(z) = \frac{1}{k(z) - pz}\,. \tag{10.7}$$

Of course, we will not be able to find an explicit primitive $G(z)$ for every $k(z)$. And when we can find $G(z)$, often we will not be able to solve (10.7) for z as an explicit function of x.

Example. Solve the equation

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{y + \sqrt{x^2 + y^2}}{x} \qquad \text{for } x > 0\,.$$

Solution. This equation can be expressed as

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{y}{x} + \sqrt{1 + \frac{y^2}{x^2}}\,.$$

It thereby has the dilation invariant form (10.4) with $p = 1$ and $k(z) = z + \sqrt{1 + z^2}$. Rather than remember the form (10.6), it is easier to remember the substitution $z = y/x$ and rederive (10.6). Indeed, we see that

$$\frac{\mathrm{d}z}{\mathrm{d}x} = \frac{\mathrm{d}}{\mathrm{d}x}\frac{y}{x} = \frac{1}{x}\,\frac{\mathrm{d}y}{\mathrm{d}x} - \frac{y}{x^2} = \frac{1}{x}\left(\frac{y}{x} + \sqrt{1 + \frac{y^2}{x^2}}\right) - \frac{y}{x^2} = \frac{1}{x}\sqrt{1 + \frac{y^2}{x^2}} = \frac{\sqrt{1 + z^2}}{x}\,.$$

Solutions of this separable equation satisfy

$$\log(x) = \int \frac{1}{\sqrt{1+z^2}}\,\mathrm{d}z = \sinh^{-1}(z) + c\,.$$

This equation can be solved explicitly to obtain

$$z = \sinh\big(\log(x) - c\big) = \frac{e^{\log(x)-c} - e^{-\log(x)+c}}{2} = \frac{1}{2}\left(\frac{x}{e^c} - \frac{e^c}{x}\right).$$

Because $y = xz$, a family of solutions to the original equation is

$$y = \frac{x^2 - e^{2c}}{2e^c}\,.$$

□

10.3. **Bernoulli Equations.** These equations can be put into the form

$$\frac{\mathrm{d}y}{\mathrm{d}t} = a(t)y - b(t)y^{1+m}\,, \tag{10.8}$$

where $a(t)$ and $b(t)$ are continous over an interval (t_L, t_R) while m is a constant. If $m = 0$ this reduces to a homogeneous linear equation, which can be solved the method of Section 3.2. So here we will treat only the case $m \neq 0$. If $m = -1$ then equation (10.8) is a nonhomogeneous linear equation, which can be solved the method of Section 3.3.

Remark. Jacob Bernoulli wrote down such equations in a 1695 letter to Gottfried Leibniz. The next year Leibniz showed that they can be transformed into a nonhomogeneous linear equation for every $m \neq 0$ by a simple substitution. Bernoulli then showed that they can be transformed into a separable equation by another simple substitution.

Leibniz transformed (10.8) into a linear equation by setting $z = y^{-m}$. We see that

$$\begin{aligned}\frac{\mathrm{d}z}{\mathrm{d}t} = -my^{-m-1}\frac{\mathrm{d}y}{\mathrm{d}t} &= -my^{-m-1}\left(a(t)y - b(t)y^{1+m}\right)\\ &= -m\,a(t)y^{-m} + m\,b(t)\\ &= -m\,a(t)z + m\,b(t)\,.\end{aligned}$$

Therefore z satisfies the nonhomogeneous linear equation

$$\frac{\mathrm{d}z}{\mathrm{d}t} + m\,a(t)z = m\,b(t)\,. \tag{10.9}$$

If we can solve this equation for z then a solution of (10.8) is obtained by setting $y = z^{-\frac{1}{m}}$.

Equation (10.9) can be solved by the recipe of Section 3.3. Let $A(t)$ and $B(t)$ satisfy

$$A'(t) = m\,a(t)\,, \qquad\qquad B'(t) = m\,e^{A(t)}b(t)\,.$$

Then the general solution of (10.9) is given by

$$z = e^{-A(t)}B(t) + e^{-A(t)}c\,, \qquad \text{where } c \text{ is an arbitrary constant}\,. \tag{10.10}$$

Therefore a solution of (10.8) is given by

$$y = \left(e^{-A(t)}B(t) + e^{-A(t)}c\right)^{-\frac{1}{m}}\,, \qquad \text{where } c \text{ is an arbitrary constant}\,. \tag{10.11}$$

Remark. Because equation (10.9) is linear, if $a(t)$ and $b(t)$ are continuous over a time interval (t_L, t_R) then its solution z will exist over (t_L, t_R) and be given by (10.10). However, formula (10.11) for the solution y of (10.8) can break down for several reasons.

Example. Solve the logistic model for populations

$$\frac{\mathrm{d}p}{\mathrm{d}t} = (r - ap)p\,.$$

Remark. Earlier we solved this using our autonomous equation recipe. Here we treat it as a Bernoulli equation.

Solution. The equation has the form

$$\frac{\mathrm{d}p}{\mathrm{d}t} = rp - ap^2\,,$$

which is the Bernoulli form (10.8) with $a(t) = r$, $b(t) = a$, and $m = 1$. If we apply formula (10.10) with $A(t) = rt$ and

$$B(t) = \int e^{rt} a \,\mathrm{d}t = \frac{a}{r} e^{rt} + c\,,$$

then we obtain

$$p = \frac{1}{\dfrac{a}{r} + e^{-rt}c}\,, \qquad \text{where } c \text{ is an arbitrary constant}\,.$$

This solution breaks down where the denominator vanishes. □

Remark. Bernoulli transformed (10.8) into a separable equation by setting

$$z = e^{-A(t)}y\,, \qquad \text{where} \quad A'(t) = a(t)\,.$$

We see that

$$\begin{aligned}\frac{\mathrm{d}z}{\mathrm{d}t} &= \frac{\mathrm{d}}{\mathrm{d}t}\left(e^{-A(t)}y\right) = e^{-A(t)}\frac{\mathrm{d}y}{\mathrm{d}t} - e^{-A(t)}a(t)y \\ &= e^{-A(t)}\left(a(t)y - b(t)y^{1+m}\right) - e^{-A(t)}a(t)y = -e^{-A(t)}b(t)y^{1+m} \\ &= -b(t)e^{mA(t)}\left(e^{-A(t)}y\right)^{1+m} = -b(t)e^{mA(t)}z^{1+m}\,.\end{aligned}$$

Therefore z satisfies the separable equation

$$\frac{\mathrm{d}z}{\mathrm{d}t} = -b(t)e^{mA(t)}z^{1+m}\,. \tag{10.12}$$

If we can solve this equation for z then a solution of (10.8) is obtained by setting $y = e^{A(t)}z$.

Equation (10.12) is separable and can be solved by the recipe of Section 4.2. Because $m \neq 0$, it has the separated form

$$-\frac{m}{z^{1+m}}\,\mathrm{d}z = mb(t)e^{mA(t)}\,\mathrm{d}t\,.$$

Therefore an implicit general solution is

$$\frac{1}{z^m} = F(t) + c\,, \qquad \text{where} \quad F'(t) = mb(t)e^{mA(t)}\,.$$

An explicit general solution is $z = (F(t) + c)^{-\frac{1}{m}}$ whenever this expression makes sense.

10.4. **Substitution.** The idea behind each of the examples above is to transform the original differential equation into an equation with a form that we know how to solve. In general, let the original equation be

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t,y)\,. \tag{10.13}$$

Upon setting $z = Z(t,y)$ and using (10.13) we see that

$$\frac{\mathrm{d}z}{\mathrm{d}t} = \partial_t Z(t,y) + \partial_y Z(t,y)\,\frac{\mathrm{d}y}{\mathrm{d}t} = \partial_t Z(t,y) + \partial_y Z(t,y)\,f(t,y)\,.$$

We then assume the relation $z = Z(t,y)$ can be inverted to obtain $y = Y(t,z)$, and substitute this result into the above to find

$$\frac{\mathrm{d}z}{\mathrm{d}t} = \partial_t Z\big(t,Y(t,z)\big) + \partial_y Z\big(t,Y(t,z)\big)\,f\big(t,Y(t,z)\big)\,.$$

We thereby obtain the transformed equation

$$\frac{\mathrm{d}z}{\mathrm{d}t} = g(t,z)\,, \tag{10.14}$$

where $g(t,z)$ is given in terms of $f(t,y)$, $Z(t,y)$, and $Y(t,z)$ by

$$g(t,z) = \partial_t Z\big(t,Y(t,z)\big) + \partial_y Z\big(t,Y(t,z)\big)\,f\big(t,Y(t,z)\big)\,.$$

This relation can be inverted to give $f(t,y)$ in terms of $g(t,z)$, $Y(t,z)$, and $Z(t,y)$ as

$$f(t,y) = \partial_t Y\big(t,Z(t,y)\big) + \partial_z Y\big(t,Z(t,y)\big)\,g\big(t,Z(t,y)\big)\,.$$

Therefore if you can solve equation (10.14) then you can solve equation (10.13), and vice versa.