

Numerical Differentiation

We have already introduced the notion of numerical differentiation in Chap. 4. Recall that we employed Taylor series expansions to derive finite-divided-difference approximations of derivatives. In Chap. 4, we developed forward, backward, and centered difference approximations of first and higher derivatives. Recall that, at best, these estimates had errors that were $O(h^2)$ —that is, their errors were proportional to the square of the step size. This level of accuracy is due to the number of terms of the Taylor series that were retained during the derivation of these formulas. We will now illustrate how to develop more accurate formulas by retaining more terms.

23.1 HIGH-ACCURACY DIFFERENTIATION FORMULAS

As noted above, high-accuracy divided-difference formulas can be generated by including additional terms from the Taylor series expansion. For example, the forward Taylor series expansion can be written as [Eq. (4.21)]

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2}h^2 + \dots \quad (23.1)$$

which can be solved for

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} - \frac{f''(x_i)}{2}h + O(h^2) \quad (23.2)$$

In Chap. 4, we truncated this result by excluding the second- and higher-derivative terms and were thus left with a final result of

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} + O(h) \quad (23.3)$$

In contrast to this approach, we now retain the second-derivative term by substituting the following approximation of the second derivative [recall Eq. (4.24)]

$$f''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i))}{h^2} + O(h) \quad (23.4)$$

into Eq. (23.2) to yield

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} - \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{2h^2}h + O(h^2)$$

or, by collecting terms,

$$f'(x_i) = \frac{-f(x_{i+2}) + 4f(x_{i+1}) - 3f(x_i)}{2h} + O(h^2) \quad (23.5)$$

Notice that inclusion of the second-derivative term has improved the accuracy to $O(h^2)$. Similar improved versions can be developed for the backward and centered formulas as well as for the approximations of the higher derivatives. The formulas are summarized in Figs. 23.1 through 23.3 along with all the results from Chap. 4. The following example illustrates the utility of these formulas for estimating derivatives.

FIGURE 23.1

Forward finite-divided-difference formulas: two versions are presented for each derivative. The latter version incorporates more terms of the Taylor series expansion and is, consequently, more accurate.

First Derivative	Error
$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h}$	$O(h)$
$f'(x_i) = \frac{-f(x_{i+2}) + 4f(x_{i+1}) - 3f(x_i)}{2h}$	$O(h^2)$
Second Derivative	
$f''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{h^2}$	$O(h)$
$f''(x_i) = \frac{-f(x_{i+3}) + 4f(x_{i+2}) - 5f(x_{i+1}) + 2f(x_i)}{h^2}$	$O(h^2)$
Third Derivative	
$f'''(x_i) = \frac{f(x_{i+3}) - 3f(x_{i+2}) + 3f(x_{i+1}) - f(x_i)}{h^3}$	$O(h)$
$f'''(x_i) = \frac{-3f(x_{i+4}) + 14f(x_{i+3}) - 24f(x_{i+2}) + 18f(x_{i+1}) - 5f(x_i)}{2h^3}$	$O(h^2)$
Fourth Derivative	
$f^{(4)}(x_i) = \frac{f(x_{i+4}) - 4f(x_{i+3}) + 6f(x_{i+2}) - 4f(x_{i+1}) + f(x_i)}{h^4}$	$O(h)$
$f^{(4)}(x_i) = \frac{-2f(x_{i+5}) + 11f(x_{i+4}) - 24f(x_{i+3}) + 26f(x_{i+2}) - 14f(x_{i+1}) + 3f(x_i)}{h^4}$	$O(h^2)$

First Derivative	Error
$f'(x_i) = \frac{f(x_i) - f(x_{i-1}))}{h}$	$O(h)$
$f'(x_i) = \frac{3f(x_i) - 4f(x_{i-1}) + f(x_{i-2}))}{2h}$	$O(h^2)$
Second Derivative	
$f''(x_i) = \frac{f(x_i) - 2f(x_{i-1}) + f(x_{i-2}))}{h^2}$	$O(h)$
$f''(x_i) = \frac{2f(x_i) - 5f(x_{i-1}) + 4f(x_{i-2}) - f(x_{i-3}))}{h^2}$	$O(h^2)$
Third Derivative	
$f'''(x_i) = \frac{f(x_i) - 3f(x_{i-1}) + 3f(x_{i-2}) - f(x_{i-3}))}{h^3}$	$O(h)$
$f'''(x_i) = \frac{5f(x_i) - 18f(x_{i-1}) + 24f(x_{i-2}) - 14f(x_{i-3}) + 3f(x_{i-4}))}{2h^3}$	$O(h^2)$
Fourth Derivative	
$f^{(4)}(x_i) = \frac{f(x_i) - 4f(x_{i-1}) + 6f(x_{i-2}) - 4f(x_{i-3}) + f(x_{i-4}))}{h^4}$	$O(h)$
$f^{(4)}(x_i) = \frac{3f(x_i) - 14f(x_{i-1}) + 26f(x_{i-2}) - 24f(x_{i-3}) + 11f(x_{i-4}) - 2f(x_{i-5}))}{h^4}$	$O(h^2)$

FIGURE 23.2

Backward finite-divided-difference formulas: two versions are presented for each derivative. The latter version incorporates more terms of the Taylor series expansion and is, consequently, more accurate.

FIGURE 23.3

Centered finite-divided-difference formulas: two versions are presented for each derivative. The latter version incorporates more terms of the Taylor series expansion and is, consequently, more accurate.

First Derivative	Error
$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1}))}{2h}$	$O(h^2)$
$f'(x_i) = \frac{-f(x_{i+2}) + 8f(x_{i+1}) - 8f(x_{i-1}) + f(x_{i-2}))}{12h}$	$O(h^4)$
Second Derivative	
$f''(x_i) = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1}))}{h^2}$	$O(h^2)$
$f''(x_i) = \frac{-f(x_{i+2}) + 16f(x_{i+1}) - 30f(x_i) + 16f(x_{i-1}) - f(x_{i-2}))}{12h^2}$	$O(h^4)$
Third Derivative	
$f'''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + 2f(x_{i-1}) - f(x_{i-2}))}{2h^3}$	$O(h^2)$
$f'''(x_i) = \frac{-f(x_{i+3}) + 8f(x_{i+2}) - 13f(x_{i+1}) + 13f(x_{i-1}) - 8f(x_{i-2}) + f(x_{i-3}))}{8h^3}$	$O(h^4)$
Fourth Derivative	
$f^{(4)}(x_i) = \frac{f(x_{i+2}) - 4f(x_{i+1}) + 6f(x_i) - 4f(x_{i-1}) + f(x_{i-2}))}{h^4}$	$O(h^2)$
$f^{(4)}(x_i) = \frac{-f(x_{i+3}) + 12f(x_{i+2}) - 39f(x_{i+1}) + 56f(x_i) - 39f(x_{i-1}) + 12f(x_{i-2}) - f(x_{i-3}))}{6h^4}$	$O(h^4)$

EXAMPLE 23.1 High-Accuracy Differentiation Formulas

Problem Statement. Recall that in Example 4.4 we estimated the derivative of

$$f(x) = -0.1x^4 - 0.15x^3 - 0.5x^2 - 0.25x + 1.2$$

at $x = 0.5$ using finite divided differences and a step size of $h = 0.25$,

	Forward $O(h)$	Backward $O(h)$	Centered $O(h^2)$
Estimate	-1.155	-0.714	-0.934
ε_t (%)	-26.5	21.7	-2.4

where the errors were computed on the basis of the true value of -0.9125 . Repeat this computation, but employ the high-accuracy formulas from Figs. 23.1 through 23.3.

Solution. The data needed for this example are

$$\begin{aligned} x_{i-2} &= 0 & f(x_{i-2}) &= 1.2 \\ x_{i-1} &= 0.25 & f(x_{i-1}) &= 1.1035156 \\ x_i &= 0.5 & f(x_i) &= 0.925 \\ x_{i+1} &= 0.75 & f(x_{i+1}) &= 0.6363281 \\ x_{i+2} &= 1 & f(x_{i+2}) &= 0.2 \end{aligned}$$

The forward difference of accuracy $O(h^2)$ is computed as (Fig. 23.1)

$$f'(0.5) = \frac{-0.2 + 4(0.6363281) - 3(0.925)}{2(0.25)} = -0.859375 \quad \varepsilon_t = 5.82\%$$

The backward difference of accuracy $O(h^2)$ is computed as (Fig. 23.2)

$$f'(0.5) = \frac{3(0.925) - 4(1.1035156) + 1.2}{2(0.25)} = -0.878125 \quad \varepsilon_t = 3.77\%$$

The centered difference of accuracy $O(h^4)$ is computed as (Fig. 23.3)

$$f'(0.5) = \frac{-0.2 + 8(0.6363281) - 8(1.1035156) + 1.2}{12(0.25)} = -0.9125 \quad \varepsilon_t = 0\%$$

As expected, the errors for the forward and backward differences are considerably more accurate than the results from Example 4.4. However, surprisingly, the centered difference yields a perfect result. This is because the formulas based on the Taylor series are equivalent to passing polynomials through the data points.

23.2 RICHARDSON EXTRAPOLATION

To this point, we have seen that there are two ways to improve derivative estimates when employing finite divided differences: (1) decrease the step size or (2) use a higher-order formula that employs more points. A third approach, based on Richardson extrapolation, uses two derivative estimates to compute a third, more accurate approximation.

Recall from Sec. 22.2.1 that Richardson extrapolation provided a means to obtain an improved integral estimate I by the formula [Eq. (22.4)]

$$I \cong I(h_2) + \frac{1}{(h_1/h_2)^2 - 1} [I(h_2) - I(h_1)] \quad (23.6)$$

where $I(h_1)$ and $I(h_2)$ are integral estimates using two step sizes h_1 and h_2 . Because of its convenience when expressed as a computer algorithm, this formula is usually written for the case where $h_2 = h_1/2$, as in

$$I \cong \frac{4}{3}I(h_2) - \frac{1}{3}I(h_1) \quad (23.7)$$

In a similar fashion, Eq. (23.7) can be written for derivatives as

$$D \cong \frac{4}{3}D(h_2) - \frac{1}{3}D(h_1) \quad (23.8)$$

For centered difference approximations with $O(h^2)$, the application of this formula will yield a new derivative estimate of $O(h^4)$.

EXAMPLE 23.2

Richardson Extrapolation

Problem Statement. Using the same function as in Example 23.1, estimate the first derivative at $x = 0.5$ employing step sizes of $h_1 = 0.5$ and $h_2 = 0.25$. Then use Eq. (23.8) to compute an improved estimate with Richardson extrapolation. Recall that the true value is -0.9125 .

Solution. The first-derivative estimates can be computed with centered differences as

$$D(0.5) = \frac{0.2 - 1.2}{1} = -1.0 \quad \varepsilon_t = -9.6\%$$

and

$$D(0.25) = \frac{0.6363281 - 1.1035156}{0.5} = -0.934375 \quad \varepsilon_t = -2.4\%$$

The improved estimate can be determined by applying Eq. (23.8) to give

$$D = \frac{4}{3}(-0.934375) - \frac{1}{3}(-1) = -0.9125$$

which for the present case is a perfect result.

The previous example yielded a perfect result because the function being analyzed was a fourth-order polynomial. The perfect outcome was due to the fact that Richardson extrapolation is actually equivalent to fitting a higher-order polynomial through these data and then evaluating the derivatives by centered divided differences. Thus, the present case matched the derivative of the fourth-order polynomial precisely. For most other functions, of course, this would not occur and our derivative estimate would be improved

but not perfect. Consequently, as was the case for the application of Richardson extrapolation, the approach can be applied iteratively using a Romberg algorithm until the result falls below an acceptable error criterion.

23.3 DERIVATIVES OF UNEQUALLY SPACED DATA

The approaches discussed to this point are primarily designed to determine the derivative of a given function. For the finite-divided-difference approximations of Sec. 23.1, these data had to be evenly spaced. For the Richardson extrapolation technique of Sec. 23.2, these data had to be evenly spaced and generated for successively halved intervals. Such control of data spacing is usually available only in cases where we can use a function to generate a table of values.

In contrast, empirically derived information—that is, data from experiments or field studies—is often collected at unequal intervals. Such information cannot be analyzed with the techniques discussed to this point.

One way to handle nonequispaced data is to fit a second-order Lagrange interpolating polynomial [recall Eq. (18.23)] to each set of three adjacent points. Remember that this polynomial does not require that the points be equispaced. The second-order polynomial can be differentiated analytically to give

$$f'(x) = f(x_{i-1}) \frac{2x - x_i - x_{i+1}}{(x_{i-1} - x_i)(x_{i-1} - x_{i+1})} + f(x_i) \frac{2x - x_{i-1} - x_{i+1}}{(x_i - x_{i-1})(x_i - x_{i+1})} + f(x_{i+1}) \frac{2x - x_{i-1} - x_i}{(x_{i+1} - x_{i-1})(x_{i+1} - x_i)} \quad (23.9)$$

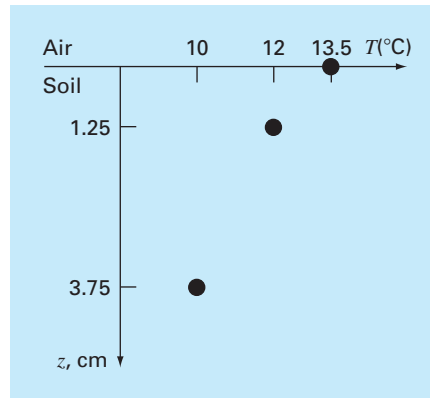
where x is the value at which you want to estimate the derivative. Although this equation is certainly more complicated than the first-derivative approximations from Figs. 23.1 through 23.3, it has some important advantages. First, it can be used to estimate the derivative anywhere within the range prescribed by the three points. Second, the points themselves do not have to be equally spaced. Third, the derivative estimate is of the same accuracy as the centered difference [Eq. (4.22)]. In fact, for equispaced points, Eq. (23.9) evaluated at $x = x_i$ reduces to Eq. (4.22).

EXAMPLE 23.3 Differentiating Unequally Spaced Data

Problem Statement. As in Fig. 23.4, a temperature gradient can be measured down into the soil. The heat flux at the soil-air interface can be computed with Fourier's law,

$$q(z = 0) = -k\rho C \left. \frac{dT}{dz} \right|_{z=0}$$

where q = heat flux (W/m^2), k = coefficient of thermal diffusivity in soil ($\cong 3.5 \times 10^{-7} \text{ m}^2/\text{s}$), ρ = soil density ($\cong 1800 \text{ kg}/\text{m}^3$), and C = soil specific heat ($\cong 840 \text{ J}/(\text{kg} \cdot ^\circ\text{C})$). Note that a positive value for flux means that heat is transferred from the air to the soil. Use numerical differentiation to evaluate the gradient at the soil-air interface and employ this estimate to determine the heat flux into the ground.

**FIGURE 23.4**

Temperature versus depth into the soil.

Solution. Equation (23.9) can be used to calculate the derivative as

$$\begin{aligned} f'(x) &= 13.5 \frac{2(0) - 1.25 - 3.75}{(0 - 1.25)(0 - 3.75)} + 12 \frac{2(0) - 0 - 3.75}{(1.25 - 0)(1.25 - 3.75)} \\ &\quad + 10 \frac{2(0) - 0 - 1.25}{(3.75 - 0)(3.75 - 1.25)} \\ &= -14.4 + 14.4 - 1.333333 = -1.333333^\circ\text{C}/\text{cm} \end{aligned}$$

which can be used to compute (note that $1 \text{ W} = 1 \text{ J/s}$),

$$\begin{aligned} q(z = 0) &= -3.5 \times 10^{-7} \frac{\text{m}^2}{\text{s}} \left(1800 \frac{\text{kg}}{\text{m}^3} \right) \left(840 \frac{\text{J}}{\text{kg} \cdot ^\circ\text{C}} \right) \left(-133.3333 \frac{^\circ\text{C}}{\text{m}} \right) \\ &= 70.56 \text{ W}/\text{m}^2 \end{aligned}$$

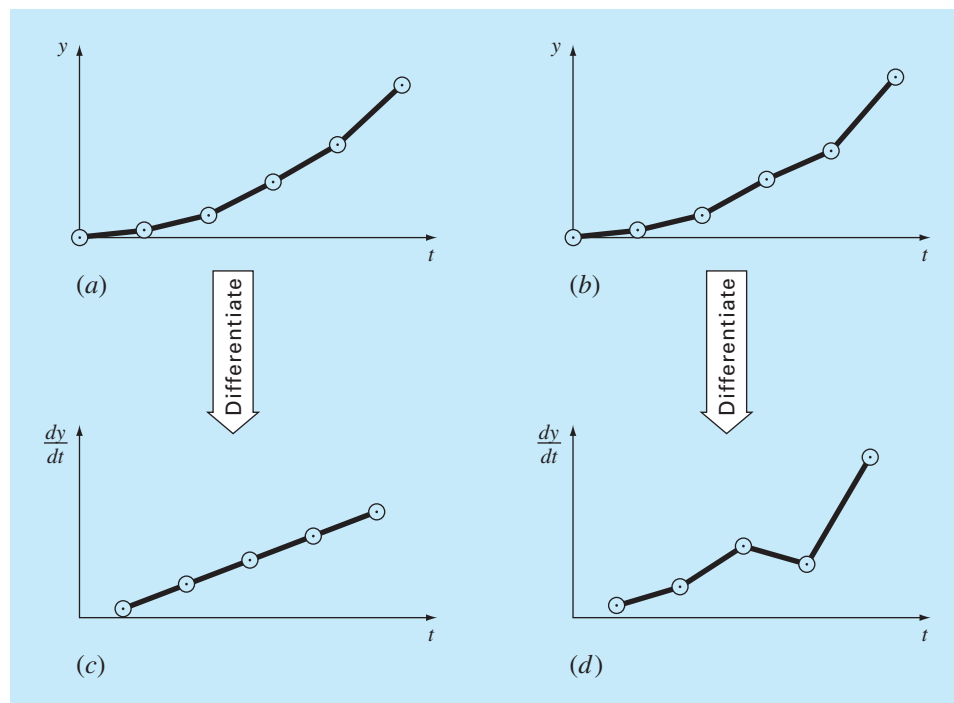
23.4 DERIVATIVES AND INTEGRALS FOR DATA WITH ERRORS

Aside from unequal spacing, another problem related to differentiating empirical data is that it usually includes measurement error. A shortcoming of numerical differentiation is that it tends to amplify errors in the data. Figure 23.5*a* shows smooth, error-free data that when numerically differentiated yield a smooth result (Fig. 23.5*c*). In contrast, Fig. 23.5*b* uses the same data, but with some points raised and some lowered slightly. This minor modification is barely apparent from Fig. 23.5*b*. However, the resulting effect in Fig. 23.5*d* is significant because the process of differentiation amplifies errors.

As might be expected, the primary approach for determining derivatives for imprecise data is to use least-squares regression to fit a smooth, differentiable function to these data. In the absence of any other information, a lower-order polynomial regression might be a good first choice. Obviously, if the true functional relationship between the dependent and independent variable is known, this relationship should form the basis for the least-squares fit.

FIGURE 23.5

Illustration of how small data errors are amplified by numerical differentiation: (a) data with no error, (b) data modified slightly, (c) the resulting numerical differentiation of curve (a), and (d) the resulting differentiation of curve (b) manifesting increased variability. In contrast, the reverse operation of integration [moving from (d) to (b) by taking the area under (d)] tends to attenuate or smooth data errors.



23.4.1 Differentiation versus Integration of Uncertain Data

Just as curve-fitting techniques like regression can be used to differentiate uncertain data, a similar process can be employed for integration. However, because of the difference in stability between differentiation and integration, this is rarely done.

As depicted in Fig. 23.5, differentiation tends to be unstable—that is, it amplifies errors. In contrast, the fact that integration is a summing process tends to make it very forgiving with regard to uncertain data. In essence, as points are summed to form an integral, random positive and negative errors tend to cancel out. In contrast, because differentiation is subtractive, random positive and negative errors tend to add.

23.5 PARTIAL DERIVATIVES

Partial derivatives along a single dimension are computed in the same fashion as ordinary derivatives. For example, suppose that we want to determine the partial derivatives for a two-dimensional function, $f(x, y)$. For equally-spaced data, the partial first derivatives can be approximated with centered differences,

$$\frac{\partial f}{\partial x} = \frac{f(x + \Delta x, y) - f(x - \Delta x, y)}{2\Delta x} \quad (23.10)$$

$$\frac{\partial f}{\partial y} = \frac{f(x, y + \Delta y) - f(x, y - \Delta y)}{2\Delta y} \quad (23.11)$$

All the other formulas and approaches discussed to this point can be applied to evaluate partial derivatives in a similar fashion.

For higher-order derivatives, we might want to differentiate a function with respect to two or more different variables. The result is called a *mixed partial derivative*. For example, we might want to take the partial derivative of $f(x, y)$ with respect to both independent variables

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial y} \right) \quad (23.12)$$

To develop a finite-difference approximation, we can first form a difference in x of the partial derivatives in y ,

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\frac{\partial f}{\partial y}(x + \Delta x, y) - \frac{\partial f}{\partial y}(x - \Delta x, y)}{2\Delta x} \quad (23.13)$$

Then, we can use finite differences to evaluate each of the partials in y ,

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\frac{f(x + \Delta x, y + \Delta y) - f(x + \Delta x, y - \Delta y)}{2\Delta y} - \frac{f(x - \Delta x, y + \Delta y) - f(x - \Delta x, y - \Delta y)}{2\Delta y}}{2\Delta x} \quad (23.14)$$

Collecting terms yields the final result

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{f(x + \Delta x, y + \Delta y) - f(x + \Delta x, y - \Delta y) - f(x - \Delta x, y + \Delta y) + f(x - \Delta x, y - \Delta y)}{4\Delta x \Delta y} \quad (23.15)$$

23.6 NUMERICAL INTEGRATION/DIFFERENTIATION WITH SOFTWARE PACKAGES

Software packages have great capabilities for numerical integration and differentiation. In this section, we will give you a taste of some of the more useful ones.

23.6.1 MATLAB

MATLAB software has a variety of built-in functions that allow functions and data to be integrated and differentiated (Table 23.1). In this section, we will illustrate some of these capabilities.

MATLAB can integrate both discrete data and functions. For example, `trapz` computes the integral of discrete values using the multiple-application trapezoidal rule. A simple representation of its syntax is

$$q = \text{trapz}(x, y)$$

where the two vectors, x and y , hold the independent and dependent variables, respectively, and q holds the resulting integral. It also has another function, `cumtrapz`, that computes the cumulative integral. For this case, the result is a vector whose elements $q(k)$ hold the integral from $x(1)$ to $x(k)$.

TABLE 23.1 MATLAB functions to implement (a) integration and (b) differentiation.

Function	Description
(a) Integration:	
cumtrapz	Cumulative trapezoidal numerical integration
dblquad	Numerically evaluate double integral
polyint	Integrate polynomial analytically
quad	Numerically evaluate integral, adaptive Simpson quadrature
quadgk	Numerically evaluate integral, adaptive Gauss-Kronrod quadrature
quadl	Numerically evaluate integral, adaptive Lobatto quadrature
quadv	Vectorized quadrature
trapz	Trapezoidal numerical integration
triplequad	Numerically evaluate triple integral
(b) Differentiation:	
del2	Discrete Laplacian
diff	Differences and approximate derivatives
gradient	Numerical gradient
polyder	Polynomial derivative

When the integrand is available in functional form, `quad` generates the definite integral using adaptive quadrature. A simple representation of its syntax is

$$q = \text{quad}(fun, a, b)$$

where `fun` is the function to be integrated, and `a` and `b` are the integration limits.

EXAMPLE 23.4**Using Numerical Integration to Compute Distance from Velocity**

Problem Statement. As described in Sec. PT6.1, integration can be used to compute the distance, $y(t)$, of an object based on its velocity, $v(t)$, as in,

$$y(t) = \int_0^t v(t) dt \quad (\text{E23.4.1})$$

Recall from Sec. 1.1, that the velocity of a free-falling parachutist, subject to linear drag and with zero initial velocity, can be computed with

$$v(t) = \frac{gm}{c} (1 - e^{-(c/m)t}) \quad (\text{E23.4.2})$$

If we substitute, Eq. (E23.4.2) into Eq. (E23.4.1), the result can be integrated analytically, with the initial condition, $y(0) = 0$, to yield

$$y(t) = \frac{gm}{c} t - \frac{gm^2}{c^2} (1 - e^{-(c/m)t})$$

This result can be used to compute that a 70-kg parachutist with a drag coefficient of 12.5 kg/s will fall 799.73 m over a 20-s period.

Use MATLAB functions to perform the same integration numerically. In addition, develop a plot of the analytical and computed distances along with velocity on the same graph.

Solution. We can first use Eq. (E23.4.2) to generate some unequally-spaced times and velocities. We can then round these velocities so that they are more like measured values; that is, they are not exact,

```
>> format short g
>> t=[0 1 2 3 4.3 7 12 16];
>> g=9.81;m=70;c=12.5;
>> v=round(g*m/c*(1-exp(-c/m*t)));
```

The total distance can then be computed as

```
>> y=trapz (t,v)

y =
    789.6
```

Thus, after 20 seconds, the jumper has fallen 789.6 m, which is reasonably close to the exact, analytical solution of 799.73 m.

If we desire the cumulative distance travelled at each time, `cumtrapz` can be employed to compute,

```
>> yc=cumtrapz (t,v)

yc =
    0    4.5    17   36.5   70.3  162.1  379.6  579.6  789.6
```

A graph of the numerical and analytical solutions along with both the exact and rounded velocities are generated with the following commands,

```
>> ta=linspace (t(1), t(length(t)));
>> ya=g*m/c*ta-g*m^2/c^2*(1-exp(-c/m*ta));
>> plot (ta, ya, t, yc, 'o')
>> title ('Distance versus time')
>> xlabel ('t (s)'), ylabel ('x (m)')
>> legend ('analytical', 'numerical')
```

As in Fig. 23.6, the numerical and analytical results match fairly well.

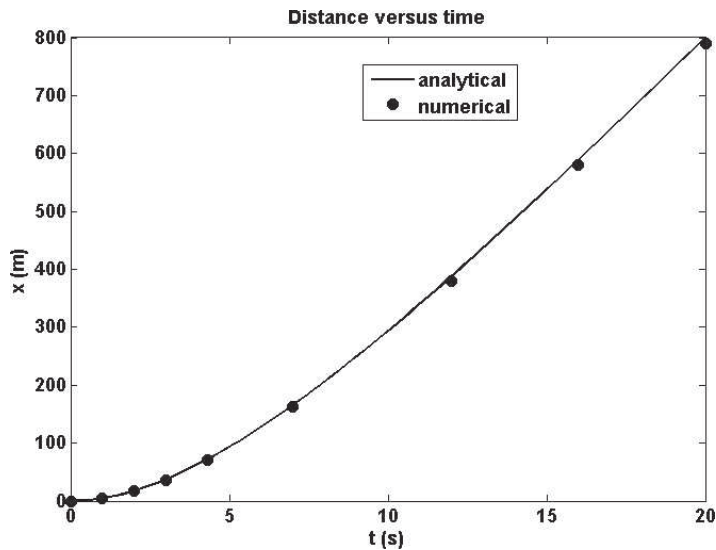
Finally, the `quad` function can be used to evaluate the integral with adaptive quadrature

```
>> va=@(t) g*m/c*(1-exp(-c/m*t));
>> yq=quad(va,t(1),t(length(t)))

yq =
    799.73
```

This result is identical to the analytical solution to within the 5 significant digits displayed.

As listed in Table 23.1b, MATLAB has a number of built-in functions for evaluating derivatives including the `diff` and `gradient` functions. When it is passed a one-dimensional vector of length n , the `diff` function returns a vector of length $n - 1$

**FIGURE 23.6**

Plot of distance versus time. The line was computed with the analytical solution, whereas the points were determined numerically with the `cumtrapz` function.

containing the differences between adjacent elements. These can then be employed to determine finite-difference approximations of first-derivatives.

The `gradient` function also returns differences. However, it does so in a manner that is more compatible with evaluating derivatives at the values themselves rather than in the intervals between values. A simple representation of its syntax is

$$fx = \text{gradient}(f)$$

where f = a one-dimensional vector of length n , and fx is a vector of length n containing differences based on f . Just as with the `diff` function, the first value returned is the difference between the first and second value. However, for the intermediate values, a centered difference based on the adjacent values is returned,

$$\text{diff}_i = \frac{f_{i+1} - f_{i-1}}{2}$$

The last value is then computed as the difference between the final two values. Hence, the results correspond to using centered differences for all the intermediate values, with forward and backward differences at the ends.

Note that the spacing between points is assumed to be one. If the vector represents equally-spaced data, the following version divides all the results by the interval and hence returns the actual values of the derivatives,

$$fx = \text{gradient}(f, h)$$

where h = the spacing between points.

EXAMPLE 23.5**Using `diff` and `gradient` for Differentiation**

Problem Statement. Explore how the MATLAB's `diff` and `gradient` functions can be employed to differentiate the function $f(x) = 0.2 + 25x - 200x^2 + 675x^3 - 900x^4 +$

$400x^5$ from $x = 0$ to 0.8 . Compare your results with the exact solution: $f'(x) = 25 - 400x^2 + 2025x^2 - 3600x^3 + 2000x^4$.

Solution. We can first express $f(x)$ as an anonymous function

```
>> f=@(x) 0.2+25*x-200*x.^2+675*x.^3-900*x.^4+400*x.^5;
```

We then generate a series of equally-spaced values of the independent and dependent variables,

```
>> x=0:0.1:0.8;
>> y=f(x);
```

The `diff` function is to determine the differences between adjacent elements of each vector. For example,

```
>> format short g
>> diff(x)

0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000
```

As expected, the result represents the differences between each pair of elements of x . To compute divided-difference approximations of the derivative, we merely perform a vector division of the y differences by the x differences by entering

```
>> d=diff(y)./diff(x)

10.89 -0.01 3.19 8.49 8.69 1.39 -11.01 -21.31
```

Note that because we are using equally-spaced values, after generating the x values, we could have simply performed the above computation concisely as

```
>> d=diff(f(x))/0.1;
```

The vector d now contains derivative estimates corresponding to the midpoint between adjacent elements. Therefore, in order to develop a plot of our results, we must first generate a vector holding the x values for the midpoint of each interval

```
>> n=length(x);
>> xm=(x(1:n-1)+x(2:n))./2;
```

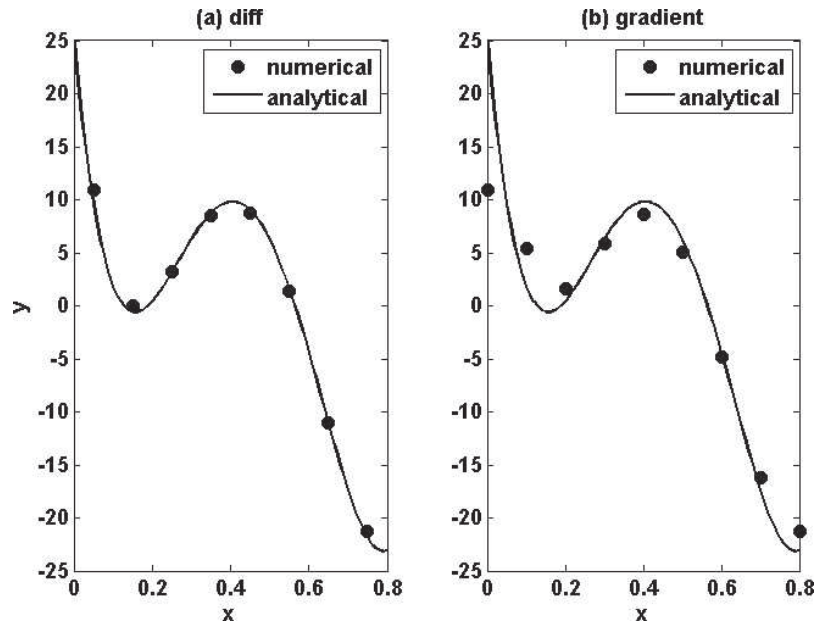
We can compute values for the analytical derivative at a finer level of resolution to include on the plot for comparison.

```
>> xa=0:.01:.8;
>> ya=25-400*xa+3*675*xa.^2-4*900*xa.^3+5*400*xa.^4;
```

A plot of the numerical and analytical estimates is then generated with

```
subplot(1,2,1), plot(xm,d,'o',xa,ya)
xlabel('x'), ylabel('y')
legend('numerical','analytical'),title('(a) diff')
```

As displayed in Fig. 23.7a, the results of the numerical approximation compare favorably with the exact, analytical solution for this case.

**FIGURE 23.7**

Comparison of the exact derivative (line) with numerical estimates (circles) computed with MATLAB's (a) `diff`, and (b) `gradient` functions.

We can also use the `gradient` function to determine the derivatives as

```
>> dy=gradient(y,0.1)
dy = 10.89  5.44  1.59  5.84  8.59  5.04 -4.81 -16.16 -21.31
```

As was done for the `diff` function, we can also display both the numerical and analytical estimates on a plot,

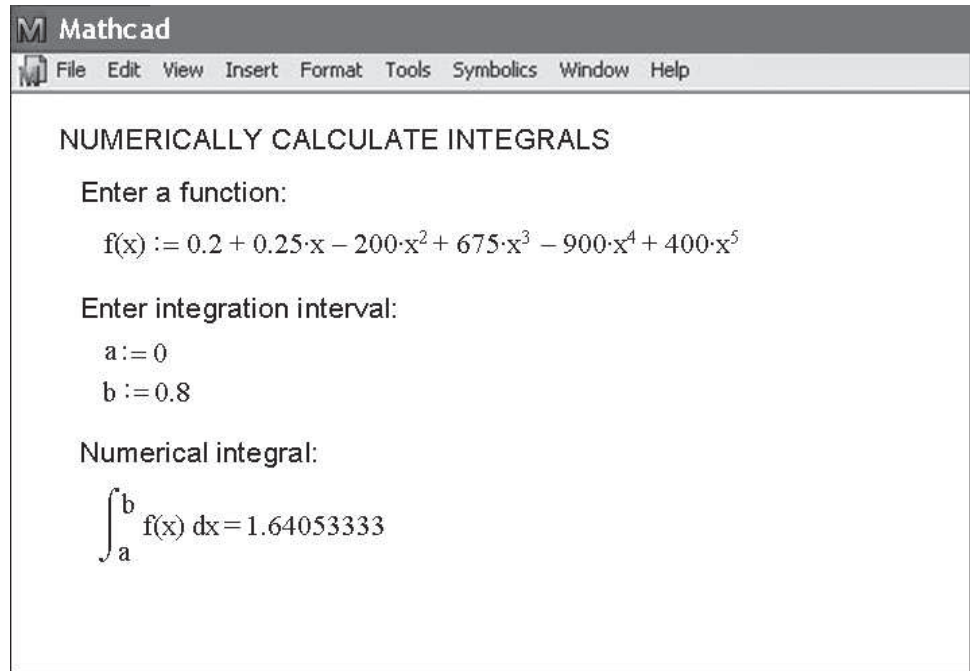
```
>> subplot(1,2,2), plot(x,dy,'o',xa,ya)
>> xlabel('x')
>> legend('numerical','analytical'),title('(b)gradient')
```

The results (Fig. 23.7b) are not as accurate as those obtained with the `diff` function (Fig. 23.7a). This is due to the fact that `gradient` employs intervals that are two times (0.2) as wide as for those used for `diff` (0.1).

Beyond one-dimensional vectors, the `gradient` function is particularly well-suited for determining the partial derivatives of matrices. For example, for a two-dimensional matrix, the function can be invoked as

```
[fx, fy] = gradient (f, h)
```

where f is a two-dimensional array, fx corresponds to the differences in the x (column) direction and fy corresponds to the differences in the y (row) direction, and h = the

**FIGURE 23.8**

Mathcad screen to determine the integral of a polynomial with Romberg integration.

spacing between points. If h is omitted, the spacing between points in both dimensions is assumed to be one. In Sec. 31.4.2, we will illustrate how this capability can be used to visualize vector fields.

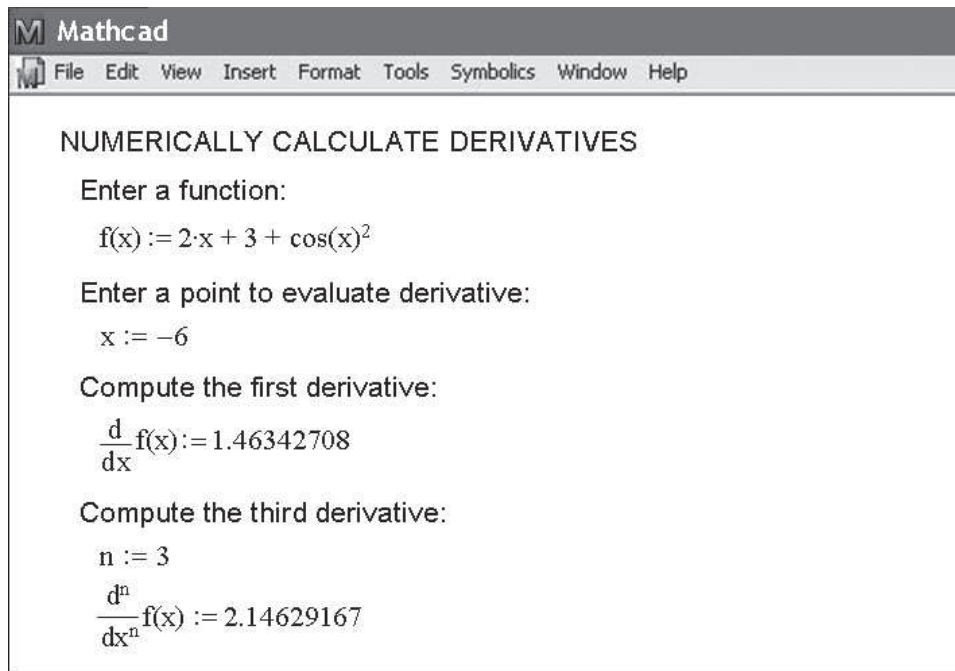
23.6.2 Mathcad

Mathcad has operators that perform numerical integration and differentiation. These operators employ and look like the same traditional mathematical symbols you have used since high school or your first semester of college.

The integration operator uses a sequence of trapezoidal rule evaluations of the integral and the Romberg algorithm. Iterations are performed until successive results vary by less than a tolerance. The derivative operator uses a similar method to compute derivatives between order 0 and 5. This operator creates a table of approximations based on divided-difference calculations of the derivative using various orders and step sizes. Extrapolation techniques are used to estimate values in a manner resembling Richardson's method.

Figure 23.8 shows a Mathcad example where $f(x)$ is created using the definition symbol ($:=$), and then the integral is calculated over a range from $x = 0$ to $x = 0.8$. In this case, we used the simple polynomial we evaluated throughout Chap. 21. Note that the range as defined by the variables a and b is input with the definition symbol.

Figure 23.9 shows a Mathcad example where a function $f(x)$ is created with the definition symbol ($:=$) and then first and third derivatives are calculated at a point where $x = -6$. Note that the location of the point and the order of the derivative are input with the definition symbol.

**FIGURE 23.9**

Mathcad screen to implement numerical differentiation.

PROBLEMS

23.1 Compute forward and backward difference approximations of $O(h)$ and $O(h^2)$, and central difference approximations of $O(h^2)$ and $O(h^4)$ for the first derivative of $y = \cos x$ at $x = \pi/4$ using a value of $h = \pi/12$. Estimate the true percent relative error ε_r for each approximation.

23.2 Repeat Prob. 23.1, but for $y = \log x$ evaluated at $x = 25$ with $h = 2$.

23.3 Use centered difference approximations to estimate the first and second derivatives of $y = e^x$ at $x = 2$ for $h = 0.1$. Employ both $O(h^2)$ and $O(h^4)$ formulas for your estimates.

23.4 Use Richardson extrapolation to estimate the first derivative of $y = \cos x$ at $x = \pi/4$ using step sizes of $h_1 = \pi/3$ and $h_2 = \pi/6$. Employ centered differences of $O(h^2)$ for the initial estimates.

23.5 Repeat Prob. 23.4, but for the first derivative of $\ln x$ at $x = 5$ using $h_1 = 2$ and $h_2 = 1$.

23.6 Employ Eq. (23.9) to determine the first derivative of $y = 2x^4 - 6x^3 - 12x - 8$ at $x = 0$ based on values at $x_0 = -0.5$, $x_1 = 1$, and $x_2 = 2$. Compare this result with the true value and with an estimate obtained using a centered difference approximation based on $h = 1$.

23.7 Prove that for equispaced data points, Eq. (23.9) reduces to Eq. (4.22) at $x = x_i$.

23.8 Compute the first-order central difference approximations of $O(h^4)$ for each of the following functions at the specified location and for the specified step size:

(a) $y = x^3 + 4x - 15$ at $x = 0$, $h = 0.25$

(b) $y = x^2 \cos x$ at $x = 0.4$, $h = 0.1$

(c) $y = \tan(x/3)$ at $x = 3$, $h = 0.5$

(d) $y = \sin(0.5\sqrt{x})/x$ at $x = 1$, $h = 0.2$

(e) $y = e^x + x$ at $x = 2$, $h = 0.2$

Compare your results with the analytical solutions.

23.9 The following data were collected for the distance traveled versus time for a rocket:

t, s	0	25	50	75	100	125
y, km	0	32	58	78	92	100

Use numerical differentiation to estimate the rocket's velocity and acceleration at each time.

23.10 Develop a user-friendly program to apply a Romberg algorithm to estimate the derivative of a given function.

23.11 Develop a user-friendly program to obtain first-derivative estimates for unequally spaced data. Test it with the following data:

x	1	1.5	1.6	2.5	3.5
$f(x)$	0.6767	0.3734	0.3261	0.08422	0.01596

where $f(x) = 5e^{-2x}$. Compare your results with the true derivatives.

23.12 The following data are provided for the velocity of an object as a function of time,

t, s	0	4	8	12	16	20	24	28	32	36
$v, m/s$	0	34.7	61.8	82.8	99.2	112.0	121.9	129.7	135.7	140.4

- (a) Using the best numerical method available, how far does the object travel from $t = 0$ to 28 s?
- (b) Using the best numerical method available, what is the object's acceleration at $t = 28$ s?
- (c) Using the best numerical method available, what is the object's acceleration at $t = 0$ s?

23.13 Recall that for the falling parachutist problem, the velocity is given by

$$v(t) = \frac{gm}{c} (1 - e^{-(c/m)t}) \tag{P23.13.1}$$

and the distance traveled can be obtained by

$$d(t) = \frac{gm}{c} \int_0^t (1 - e^{-(c/m)t}) dt \tag{P23.13.2}$$

Given $g = 9.81$, $m = 70$, and $c = 12$,

- (a) Use MATLAB or Mathcad to integrate Eq. (P23.13.1) from $t = 0$ to 10.
- (b) Analytically integrate Eq. (P23.13.2) with the initial condition that $d = 0$ at $t = 0$. Evaluate the result at $t = 10$ to confirm (a).
- (c) Use MATLAB or Mathcad to differentiate Eq. (P23.13.1) at $t = 10$.
- (d) Analytically differentiate Eq. (P23.13.1) at $t = 10$ to confirm (c).

23.14 The normal distribution is defined as

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

- (a) Use MATLAB or Mathcad to integrate this function from $x = -1$ to 1 and from -2 to 2.
- (b) Use MATLAB or Mathcad to determine the inflection points of this function.

23.15 The following data were generated from the normal distribution:

x	-2	-1.5	-1	-0.5	0	0.5	1	1.5	2
$f(x)$	0.05399	0.12952	0.24197	0.35207	0.39894	0.35207	0.24197	0.12952	0.05399

(a) Use MATLAB to integrate these data from $x = -1$ to 1 and -2 to 2 with the `trapz` function.

(b) Use MATLAB to estimate the inflection points of these data.

23.16 Evaluate $\partial f/\partial x$, $\partial f/\partial y$, and $\partial f/(\partial x \partial y)$ for the following function at $x = y = 1$ (a) analytically and (b) numerically $\Delta x = \Delta y = 0.0001$,

$$f(x, y) = 3xy + 3x - x^3 - 3y^3$$

23.17 Evaluate the following integral with MATLAB,

$$\int_0^{2\pi} \frac{\sin t}{t} dt$$

using both the `quad` and `quadl` functions. To learn more about `quadl`, type `help quadl` at the MATLAB prompt.

23.18 Use the `diff` command in MATLAB and compute the finite-difference approximation to the first and second derivative at each x -value in the table below, excluding the two end points. Use finite-difference approximations that are second-order correct, $O(\Delta x^2)$.

x	0	1	2	3	4	5	6	7	8	9	10
y	1.4	2.1	3.3	4.8	6.8	6.6	8.6	7.5	8.9	10.9	10

23.19 The objective of this problem is to compare second-order accurate forward, backward, and centered finite-difference approximations of the first derivative of a function to the actual value of the derivative. This will be done for

$$f(x) = e^{-2x} - x$$

- (a) Use calculus to determine the correct value of the derivative at $x = 2$.
- (b) To evaluate the centered finite-difference approximations, start with $x = 0.5$. Thus, for the first evaluation, the x values for the centered difference approximation will be $x = 2 \pm 0.5$ or $x = 1.5$ and 2.5 . Then, decrease in increments of 0.01 down to a minimum value of $\Delta x = 0.01$.
- (c) Repeat part (b) for the second-order forward and backward differences. (Note that these can be done at the same time that the centered difference is computed in the loop.)
- (d) Plot the results of (b) and (c) versus x . Include the exact result on the plot for comparison.

23.20 Use a Taylor series expansion to derive a centered finite-difference approximation to the third derivative that is second-order

accurate. To do this, you will have to use four different expansions for the points x_{i-2} , x_{i-1} , x_{i+1} , and x_{i+2} . In each case, the expansion will be around the point x_i . The interval Δx will be used in each case of $i-1$ and $i+1$, and $2\Delta x$ will be used in each case of $i-2$ and $i+2$. The four equations must then be combined in a way to eliminate the first and second derivatives. Carry enough terms along in each expansion to evaluate the first term that will be truncated to determine the order of the approximation.

23.21 Use the following data to find the velocity and acceleration at $t = 10$ seconds:

Time, t , s	0	2	4	6	8	10	12	14	16
Position, x , m	0	0.7	1.8	3.4	5.1	6.3	7.3	8.0	8.4

Use second-order correct (a) centered finite-difference, (b) forward finite-difference, and (c) backward finite-difference methods.

23.22 A plane is being tracked by radar, and data are taken every second in polar coordinates θ and r .

t , s	200	202	204	206	208	210
θ , rad	0.75	0.72	0.70	0.68	0.67	0.66
r , m	5120	5370	5560	5800	6030	6240

At 206 s, use the centered finite difference (second-order correct) to find the vector expressions for velocity \vec{v} , and acceleration \vec{a} . The velocity and acceleration given in polar coordinates are:

$$\vec{v} = \dot{r}\vec{e}_r + r\dot{\theta}\vec{e}_\theta \quad \text{and} \quad \vec{a} = (\ddot{r} - r\dot{\theta}^2)\vec{e}_r + (r\ddot{\theta} + 2\dot{r}\dot{\theta})\vec{e}_\theta$$

23.23 Develop an Excel VBA macro program to read in adjacent columns of x and y values from a worksheet. Evaluate the derivatives at each point using Eq. 23.9, and display the results in a third column adjacent to the x and y values back on the spreadsheet. Test your program by applying it to evaluate the velocities for the time-position values from Prob. 23.21.

23.24 Use regression to estimate the acceleration at each time for the following data with second-, third-, and fourth-order polynomials. Plot the results.

t	1	2	3.25	4.5	6	7	8	8.5	9.3	10
v	10	12	11	14	17	16	12	14	14	10

23.25 You have to measure the flow rate of water through a small pipe. In order to do it, you place a bucket at the pipe's outlet and measure the volume in the bucket as a function of time as tabulated below. Estimate the flow rate at $t = 7$ s.

Time, s	0	1	5	8
Volume, cm^3	0	1	8	16.4

23.26 The velocity v (m/s) of air flowing past a flat surface is measured at several distances y (m) away from the surface. Determine

the shear stress τ (N/m^2) at the surface ($y = 0$), using *Newton's viscosity law*

$$\tau = \mu \frac{dv}{dy}$$

Assume a value of dynamic viscosity $\mu = 1.8 \times 10^{-5} \text{ N} \cdot \text{s/m}^2$.

y , m	0	0.002	0.006	0.012	0.018	0.024
v , m/s	0	0.287	0.899	1.915	3.048	4.299

23.27 Chemical reactions often follow the model:

$$\frac{dc}{dt} = -kc^n$$

where c = concentration, t = time, k = reaction rate, and n = reaction order. Given values of c and dc/dt , k and n can be evaluated by a linear regression of the logarithm of this equation:

$$\log\left(-\frac{dc}{dt}\right) = \log k + n \log c$$

Use this approach along with the following data to estimate k and n :

t	10	20	30	40	50	60
c	3.52	2.48	1.75	1.23	0.87	0.61

23.28 The velocity profile of a fluid in a circular pipe can be represented as

$$v = 10 \left(1 - \frac{r}{r_0}\right)^{1/n}$$

where v = velocity, r = radial distance measured out from the pipe's centerline, r_0 = the pipe's radius, and n = a parameter. Determine the flow in the pipe if $r_0 = 0.75$ and $n = 7$ using (a) Romberg integration to a tolerance of 0.1%, (b) two-point Gauss-Legendre formula, and (c) the MATLAB `quad` function. Note that flow is equal to velocity times area.

23.29 The amount of mass transported via a pipe over a period of time can be computed as

$$M = \int_{t_1}^{t_2} Q(t)c(t) dt$$

where M = mass (mg), t_1 = the initial time (min), t_2 = the final time (min), $Q(t)$ = flow rate (m^3/min), and $c(t)$ = concentration (mg/m^3). The following functional representations define the temporal variations in flow and concentration,

$$Q(t) = 9 + 4\cos^2(0.4t)$$

$$c(t) = 5e^{-0.5t} + 2e^{0.15t}$$

Determine the mass transported between $t_1 = 2$ and $t_2 = 8$ min with (a) Romberg integration to a tolerance of 0.1%, and (b) the MATLAB `quad` function.