Algoritmos Numéricos por Computadora

COM - 14105

"Actually, a person does not really understand something until he can teach it to a computer"
Donald Knuth, 1974

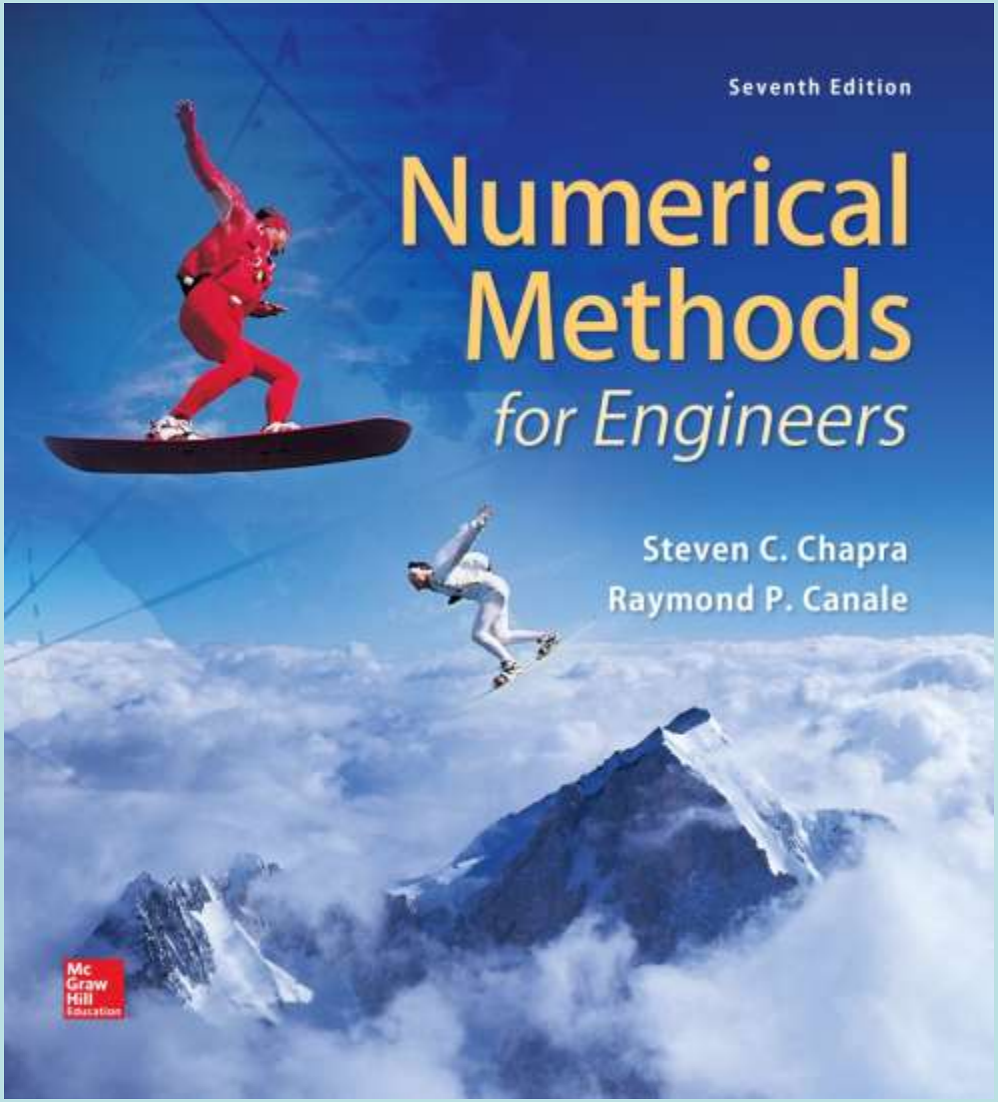
Objetivos

- Solucionar sistemas de ecuaciones lineales y ecuaciones diferenciales de forma numérica
- Entender el funcionamiento de diversos métodos numéricos
- Identificar los errores numéricos de las soluciones computacionales
- Familiarizarse con el modelado matemático de sistemas físicos
- Utilizar un lenguaje de programación matricial de manera eficiente

Temario

1. Introducción
    1. Modelado de sistemas dinámicos
    2. Truncamiento y redondeo
    3. Raíces de funciones y optimización
2. Sistemas de ecuaciones
    1. Valores y vectores propios
    2. Eliminación de Gauss
    3. Factorizaciones
    4. Métodos iterativos
    5. Sistemas no lineales
3. Ecuaciones diferenciales ordinarias
    1. Soluciones analíticas sencillas
    2. Problemas con valor inicial
    3. Sistemas de ecuaciones lineales de primer orden
    4. ODE de orden superior
    5. Métodos de paso variable, multipasos e implícitos
    6. Problemas con valores en la frontera

Ecuaciones Diferenciales



Ordinary Differential Equations

Equations which are composed of an unknown function and its derivatives are called *differential equations*.

Differential equations play a fundamental role in engineering because many physical phenomena are best formulated mathematically in terms of their rate of change.

$$\frac{dv}{dt} = g - \frac{c_d}{m} v^2$$

v - dependent variable

t - independent variable

When a function involves one dependent variable, the equation is called an *ordinary differential equation (or ODE).* A *partial differential equation (or PDE)* involves two or more independent variables.

Differential equations are also classified as to their order.

- *A first order equation* includes a first derivative as its highest derivative.
- *A second order equation* includes a second derivative.

Higher order equations can be reduced to a system of first order equations, by redefining a variable.

Physical law $\longrightarrow$ $F = ma$

ODE $\longrightarrow$ $$\frac{dv}{dt} = g - \frac{c_d}{m}v^2 \qquad \frac{dx}{dt} = v$$

Analytical (calculus)

Numerical (computer)

Solution $\longrightarrow$ $$v = \sqrt{\frac{gm}{c_d}}\tanh\left(\sqrt{\frac{gc_d}{m}}t\right) \qquad v_{i+1} = v_i + \left(g - \frac{c_d}{m}v^2\right)\Delta t$$

Initial-Value Problems

Solving ordinary differential equations of the form

$$\frac{dy}{dx} = f(x, y) \qquad y(x_0) = y_0$$

$$y_{i+1} = y_i + \phi h$$

One-step methods
Constant step size

Numerical Solution

A numerical solution of an initial value problem

$$y' = f(t, y) \qquad\qquad y(t_0) = y_0$$

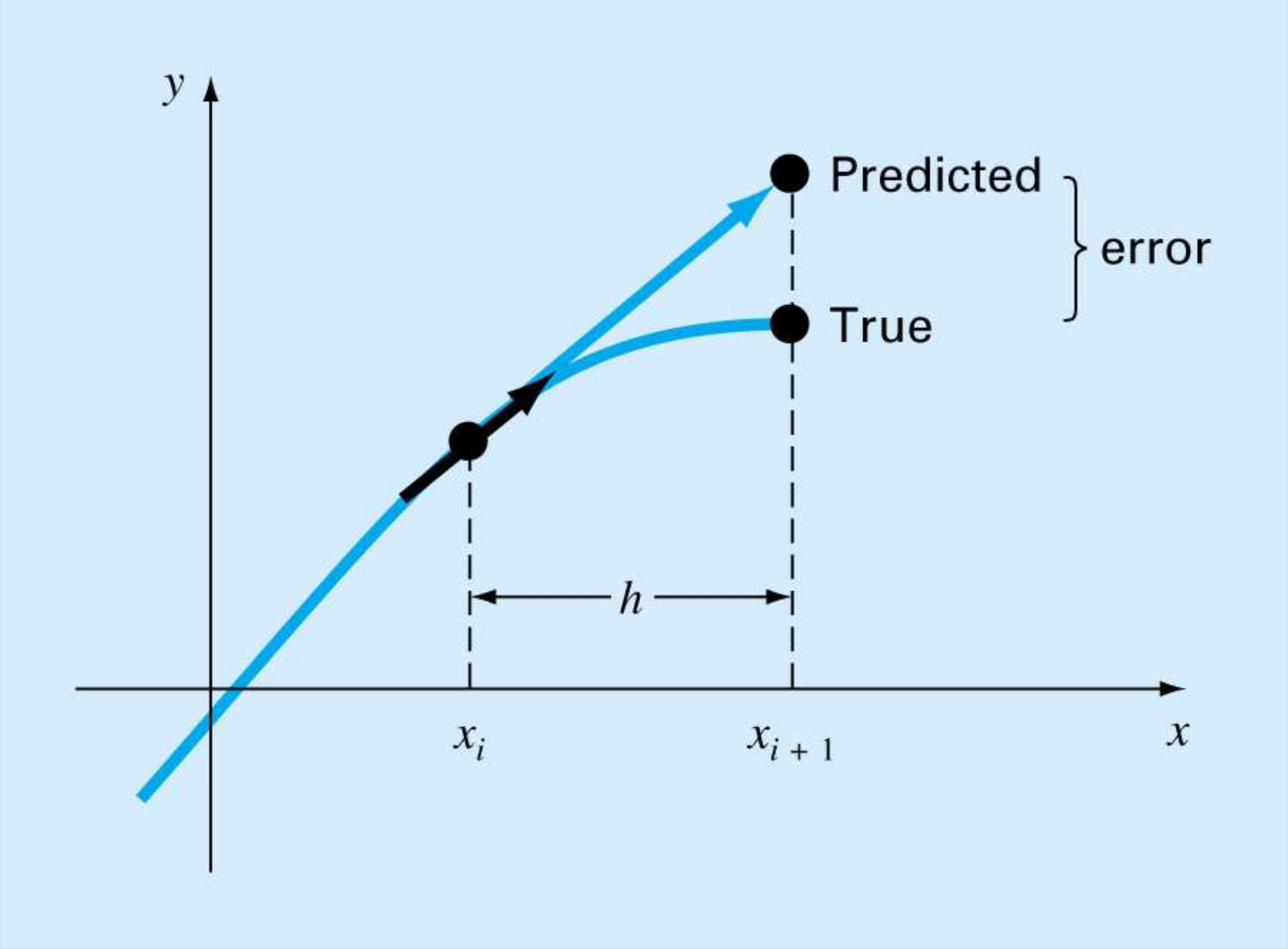
is essentially a table of *t* and *y* values listed at discrete intervals of *t*.

$t(1) = t_0 \qquad t(i+1) = t(i) + h$

$y(1) = y_0 \qquad y(i+1) = y(i) + phi*h$

t						
y						

Euler’s Method



First-order RK method

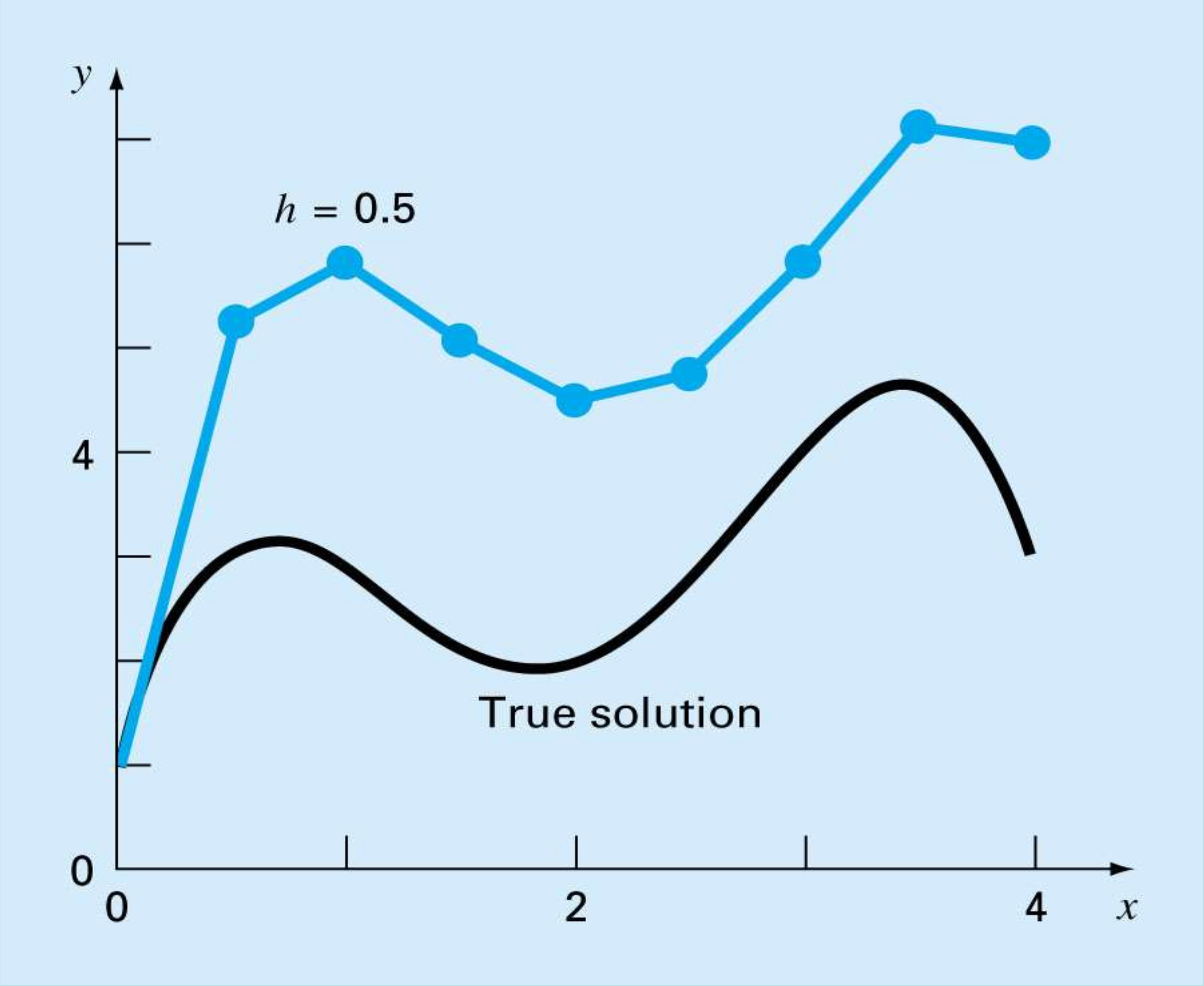
The first derivative (*dy/dx*) provides a direct estimate of the slope of y at x_i

$$\phi = f(x_i, y_i)$$

where $f(x_i,y_i)$ is the differential equation evaluated at x_i and y_i. This estimate can be substituted into the equation:

$$y_{i+1} = y_i + f(x_i, y_i)h$$

A new value of y is predicted using the slope to extrapolate linearly over the step size h.


y
h = 0.5
4
True solution
0
0
2
4
x

We can derive Euler's method directly from the Taylor series expansion, using the notation:

$h = t_{i+1} - t_i$

$y_i = y(t_i)$

$y_{i+1} = y(t_{i+i}) = y(t_i + h)$

$$y_{i+1} = y_i + y_i' h + \frac{y_i''}{2!} h^2 + \cdots + \frac{y_i^{(n)}}{n!} h^n + R_n$$

$$y_{i+1} = y_i + f(t_i, y_i) h + \frac{f'(t_i, y_i)}{2!} h^2 + \cdots + \frac{f^{(n-1)}(t_i, y_i)}{n!} h^n + O(h^{n+1})$$

Error Analysis for Euler's Method

Numerical solutions of ODEs involves two types of error:

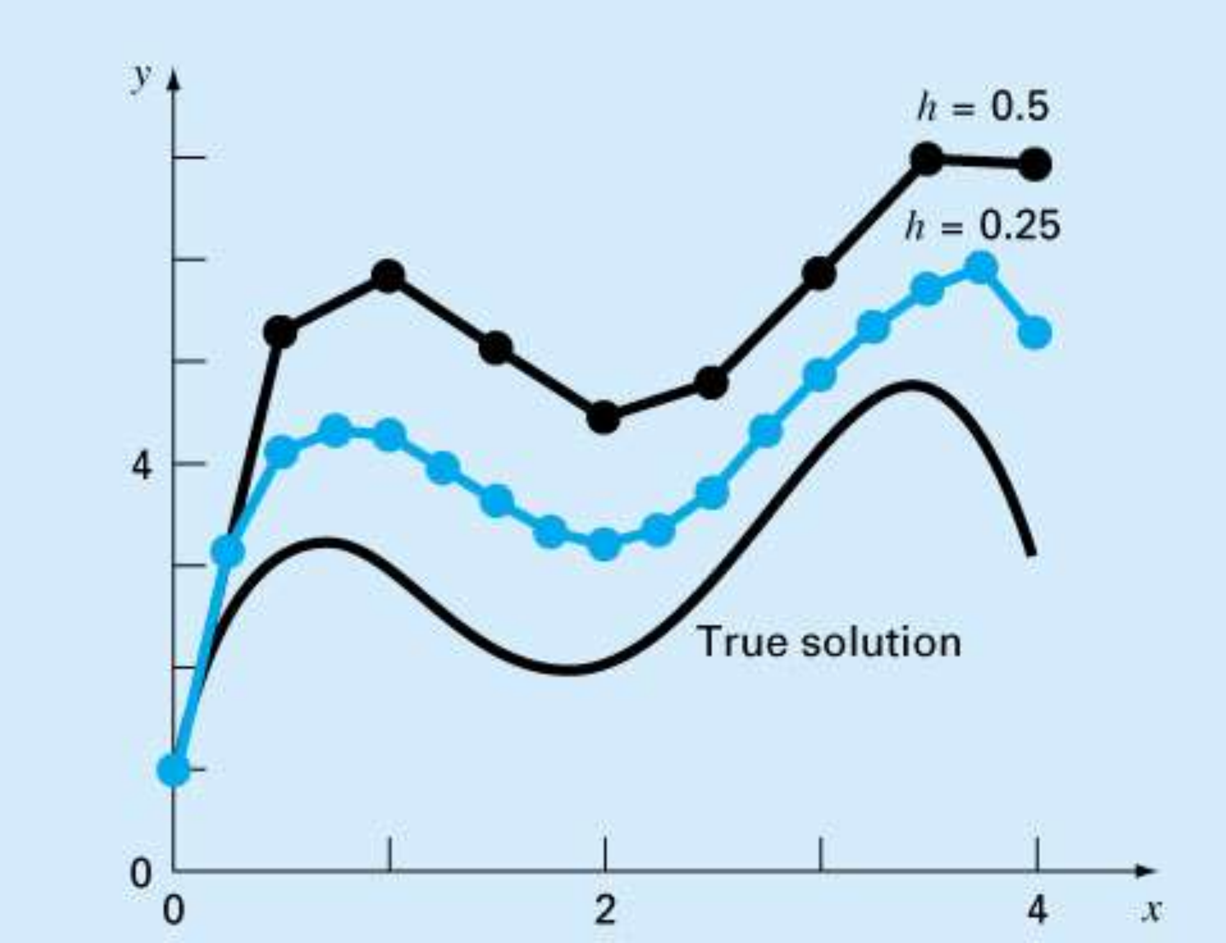
- *Truncation* error
  - *Local truncation error* $O(h^2)$
  - *Global truncation error* $O(h)$ (first order method)
- *Round-off* errors

The Taylor series provides a means of quantifying the error in Euler’s method, however

- The Taylor series provides only an estimate of the local truncation error-that is, the error created during a single step of the method.
- In actual problems, the functions are more complicated than simple polynomials. Consequently, the derivatives needed to evaluate the Taylor series expansion would not always be easy to obtain.

In conclusion

- The error can be reduced by reducing the step size
- If the solution to the differential equation is linear, the method will provide error free predictions as for a straight line the second derivative would be zero.


y
h = 0.5
h = 0.25
4
True solution
0
0
2
4
x

Improvements of Euler's method

A fundamental source of error in Euler's method is that the derivative at the beginning of the interval is assumed to apply across the entire interval.

Two simple modifications are available to circumvent this shortcoming:

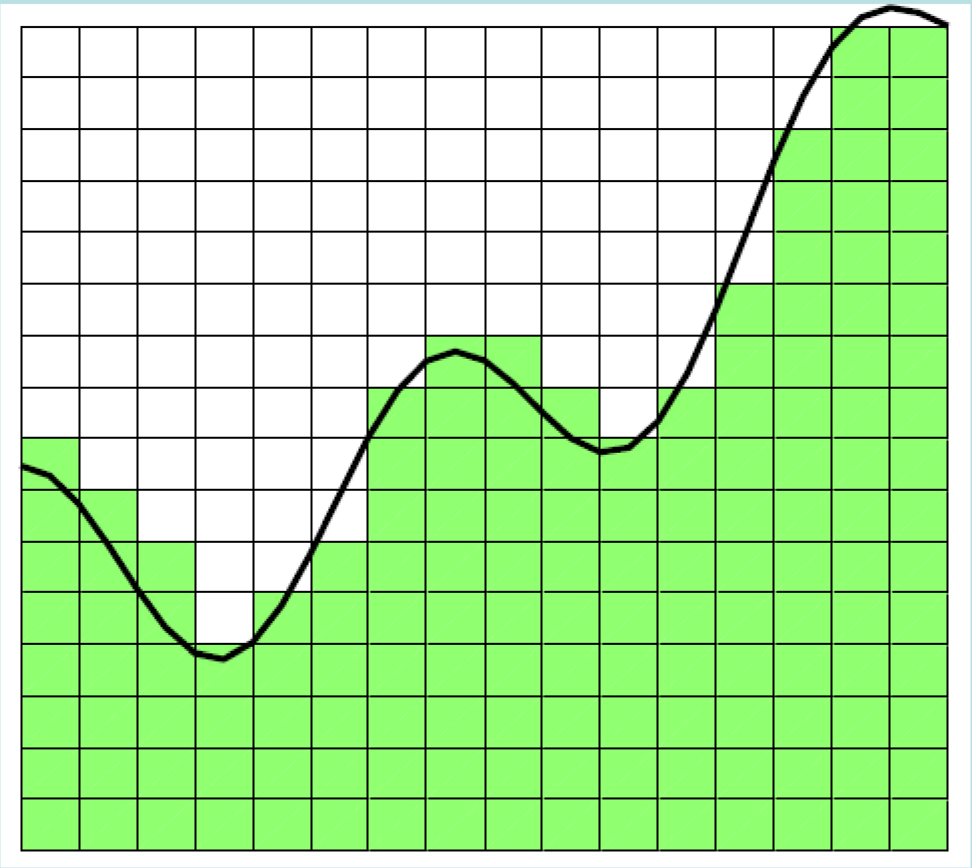
- Midpoint Method
- Heun's Method

Second-order RK methods

Numerical Integration

$$\frac{dy}{dx} = f(x, y)$$

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x, y)dx$$



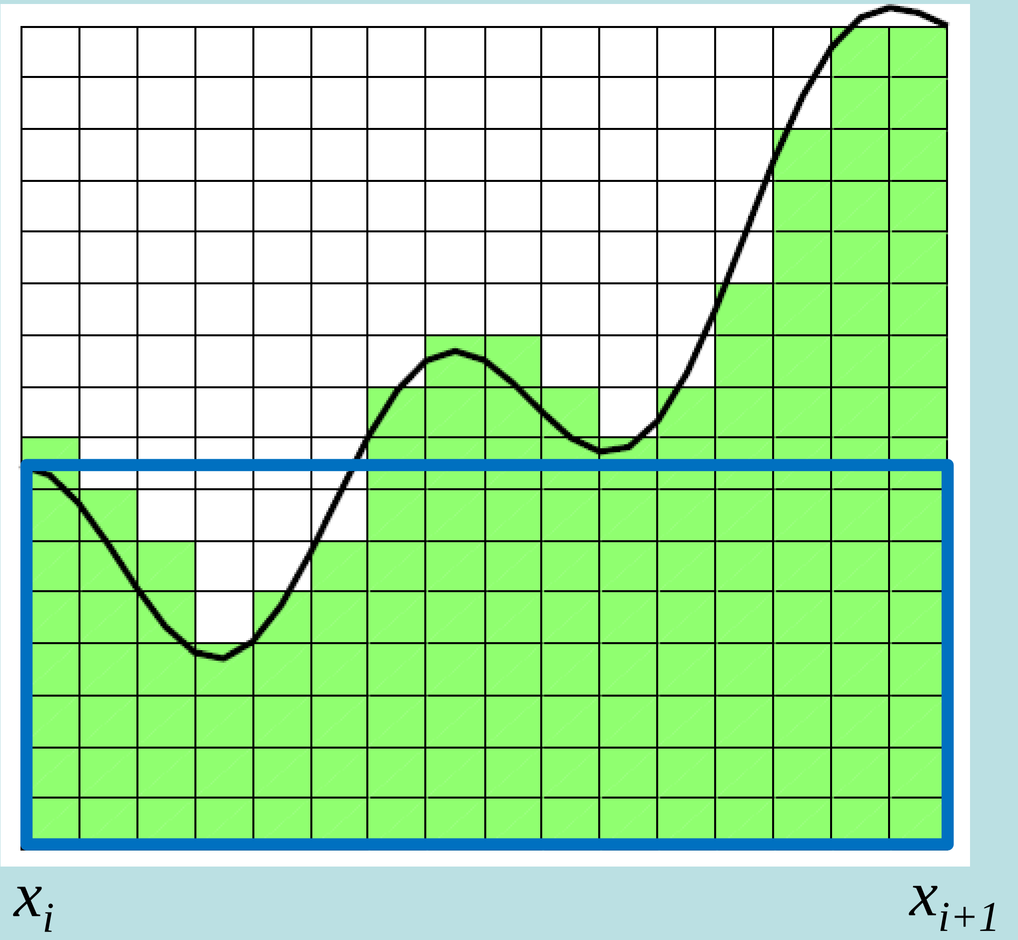
Numerical Integration

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x,y)dx$$

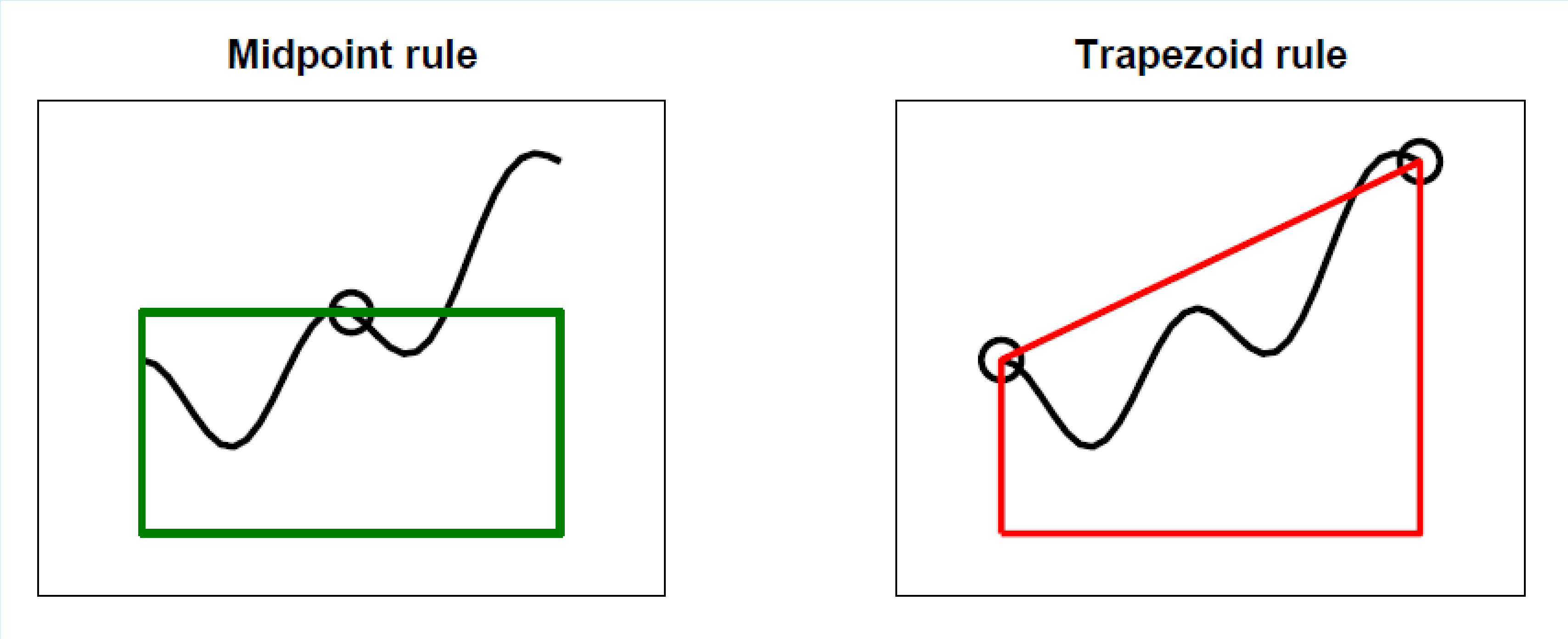
Euler

$$y_{i+1} = y_i + f(x_i, y_i) \int_{x_i}^{x_{i+1}} dx$$

$$y_{i+1} = y_i + f(x_i, y_i)h$$



Numerical Integration



$f(x) \approx b$

$f(x) \approx ax + b$

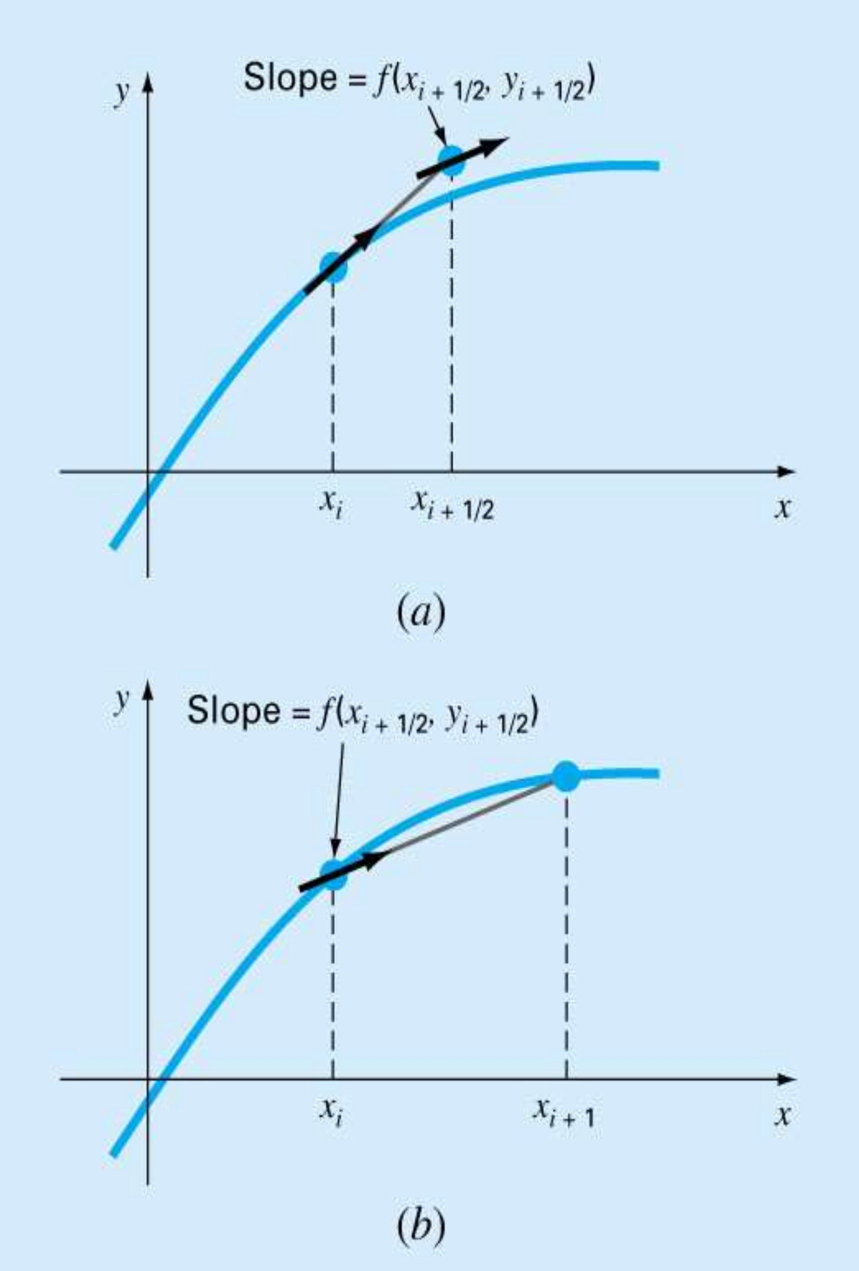
Midpoint Method

Uses Euler's method to predict a value $y_p = y_{i+1/2}$ at the midpoint $x_{i+1/2} = x_i + h/2$ of the interval

$$y_{i+1/2} = y_i + f(x_i, y_i)(h/2)$$

This derivative is then used as an improved estimate of the slope for the entire interval.

$$\boxed{y_{i+1} = y_i + f(x_{i+1/2}, y_{i+1/2})h}$$



(a)

(b)

Heun's Method

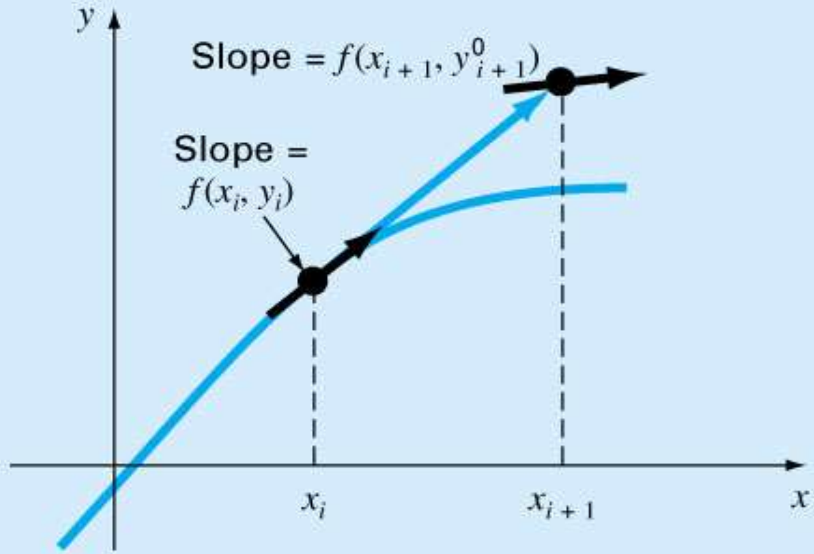
Use two derivatives to improve the estimate of the slope for the entire interval:

- At the initial point
- At the end point

The two derivatives are averaged.

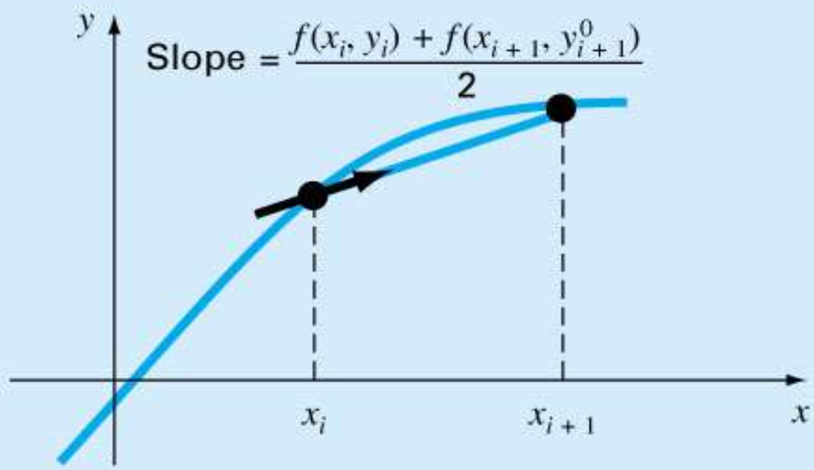
$$\text{Predictor:} \quad y_{i+1}^{0} = y_i + f(x_i, y_i)h$$

$$\text{Corrector:} \quad y_{i+1} = y_i + \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{0})}{2} h$$



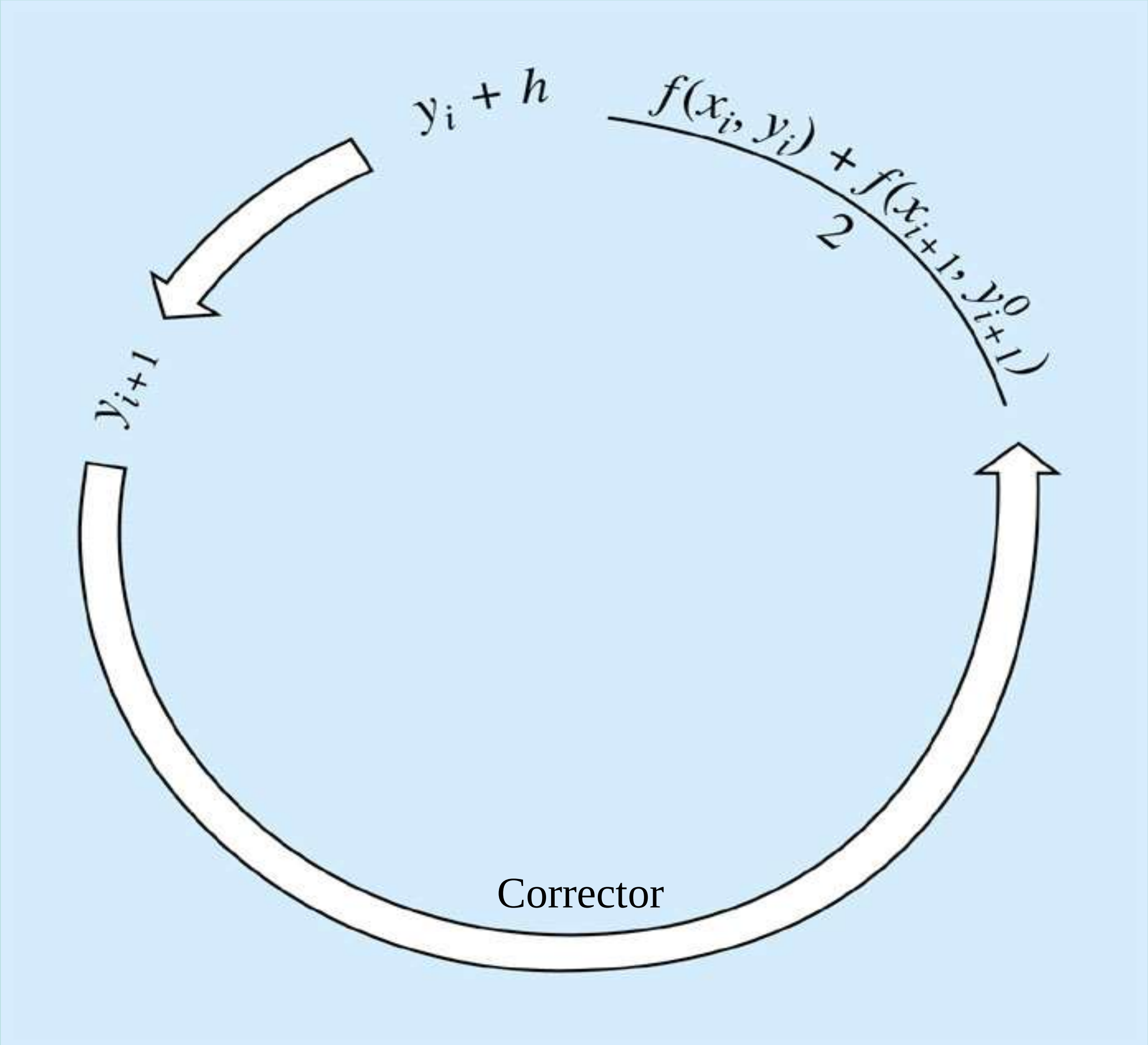
Slope = $f(x_{i+1}, y^0_{i+1})$
Slope = $f(x_i, y_i)$
y
x
x_i
x_{i+1}

(a)



Slope = $\dfrac{f(x_i, y_i) + f(x_{i+1}, y^0_{i+1})}{2}$
y
x
x_i
x_{i+1}

(b)



$y_i + h \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^0)}{2}$
y_{i+1}
Corrector

Runge-Kutta Methods (RK)

Runge-Kutta methods achieve the accuracy of a higher-order Taylor series approach without requiring the calculation of higher derivatives.

$$y_{i+1} = y_i + \phi(x_i, y_i, h)h$$

$$\phi = a_1 k_1 + a_2 k_2 + \cdots + a_n k_n$$

Increment function

$$a's = \text{constants}$$

$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + p_1 h, y_i + q_{11} k_1 h)$$

p's and q's are constants

$$k_3 = f(x_i + p_3 h, y_i + q_{21} k_1 h + q_{22} k_2 h)$$

$$\vdots$$

$$k_n = f(x_i + p_{n-1} h, y_i + q_{n-1} k_1 h + q_{n-1,2} k_2 h + \cdots + q_{n-1,n-1} k_{n-1} h)$$

k's are recurrence functions. Because each k is a function evaluation, this recurrence makes RK methods efficient for computer calculations.

Various types of RK methods can be devised by employing different number of terms in the increment function as specified by n.

Once n is chosen, values of a's, p's, and q's must be evaluated.

First order RK method with $n=1$ is in fact Euler's method ($a_1=1$).

Second order RK methods ($n=2$)

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2)h$$
$$k_1 = f(x_i, y_i)$$
$$k_2 = f(x_i + p_1 h, y_i + q_{11} k_1 h)$$

Two function evaluations

Values of a_1, a_2, p_1, and q_{11} are evaluated by using a second order Taylor series of a function of two variables $f(x, y)$ to calculate k_2

$$f(x_i + p_1h, y_i + q_{11}k_1h) = f(x_i, y_i) + p_1h\frac{\partial f}{\partial x} + q_{11}k_1h\frac{\partial f}{\partial y}$$

and comparing the second-order general formula to

$$y_{i+1} = y_i + f(x_i, y_i)h + \frac{f'(x_i, y_i)}{2!}h^2$$

$$y_{i+1} = y_i + f(x_i, y_i)h + \left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\frac{dy}{dx}\right)\frac{h^2}{2!}$$

Three equations to evaluate four unknown constants are derived.

$$a_1 + a_2 = 1$$

$$a_2 p_1 = \frac{1}{2}$$

$$a_2 q_{11} = \frac{1}{2}$$

$$a_1 = 1 - a_2$$

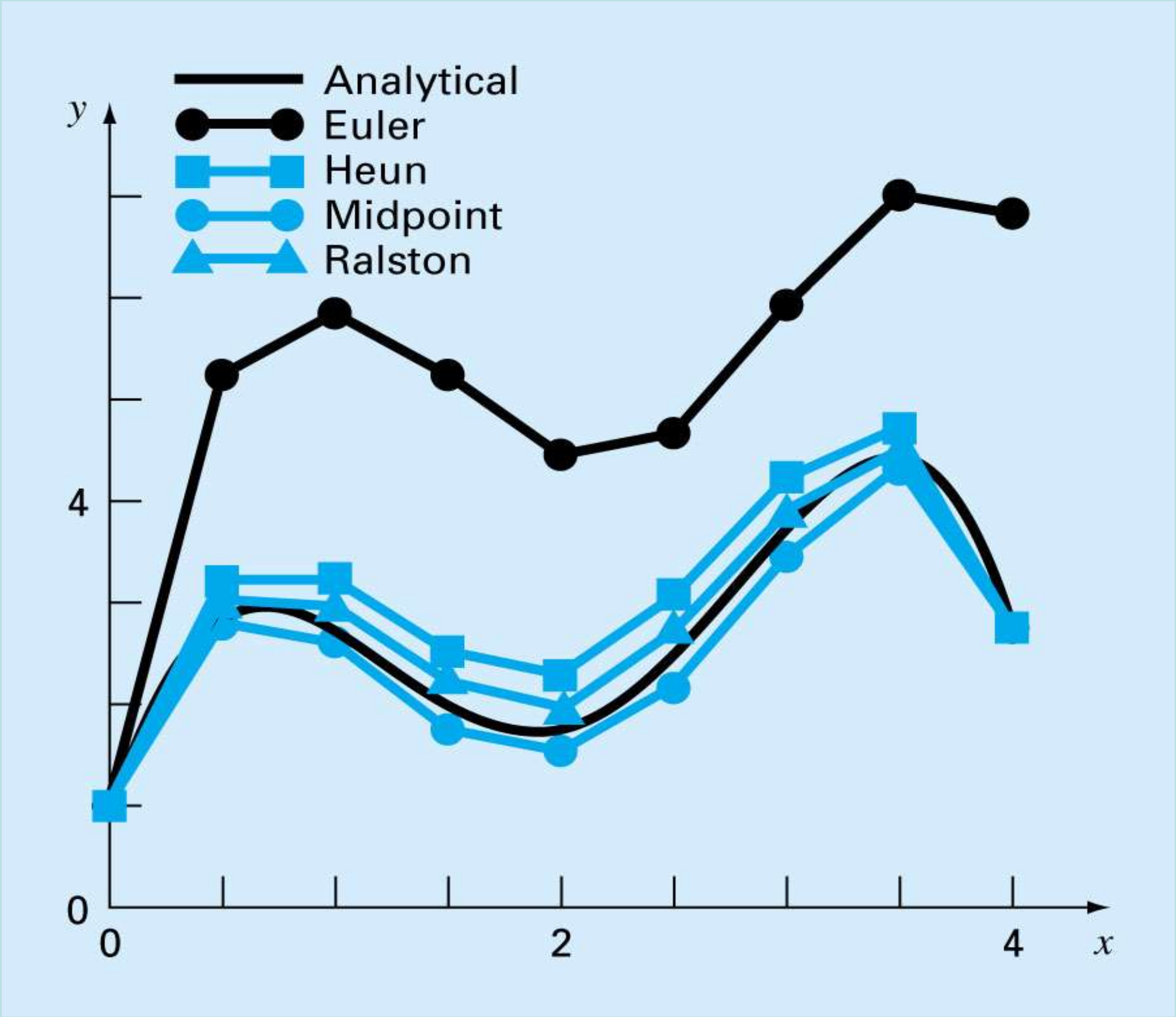
$$p_1 = q_{11} = \frac{1}{2a_2}$$

Because we can choose an infinite number of values for a_2, there are an infinite number of second-order RK methods.

Every version would yield exactly the same results if the solution to ODE were quadratic, linear, or a constant. However, they yield different results if the solution is more complicated.

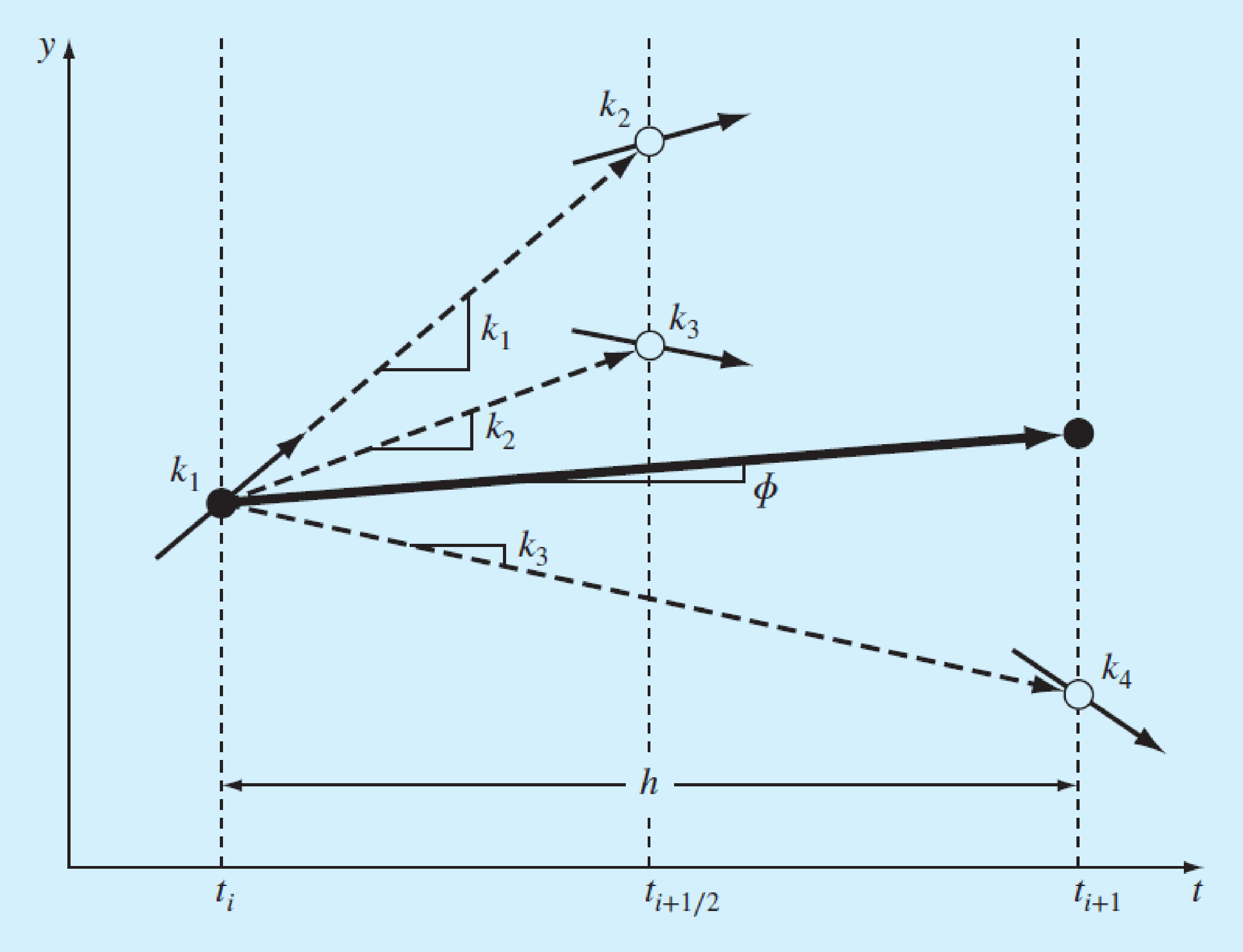
Three of the most commonly used methods are:

- Midpoint Method ($a_2=1$)
- Huen's Method with a Single Corrector ($a_2=1/2$)
- Raltson's Method ($a_2=2/3$)



Analytical
Euler
Heun
Midpoint
Ralston
y
4
0
0
2
4
x

Classical RK-4



Classical RK-4

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h$$

$$k_1 = f(t_i, y_i)$$

$$k_2 = f\left(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1 h\right)$$

$$k_3 = f\left(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2 h\right)$$

$$k_4 = f(t_i + h, y_i + k_3 h)$$

Four function evaluations

Systems of Equations

Many practical problems in engineering and science require the solution of a system of simultaneous first order ODEs rather than a single equation:

$$
\begin{aligned}
\frac{dy_1}{dx} &= f_1(x, y_1, y_2, \ldots, y_n) \\
\frac{dy_2}{dx} &= f_2(x, y_1, y_2, \ldots, y_n) \\
&\vdots \\
\frac{dy_n}{dx} &= f_n(x, y_1, y_2, \ldots, y_n)
\end{aligned}
$$

$$y = [y_1; y_2; \ldots; y_n]$$

$$y' = [f_1; f_2; \ldots; f_n]$$

Solution requires that *n* initial conditions be known at the starting value of *x*.

Numerical Solution

Now, each value of y is a column vector

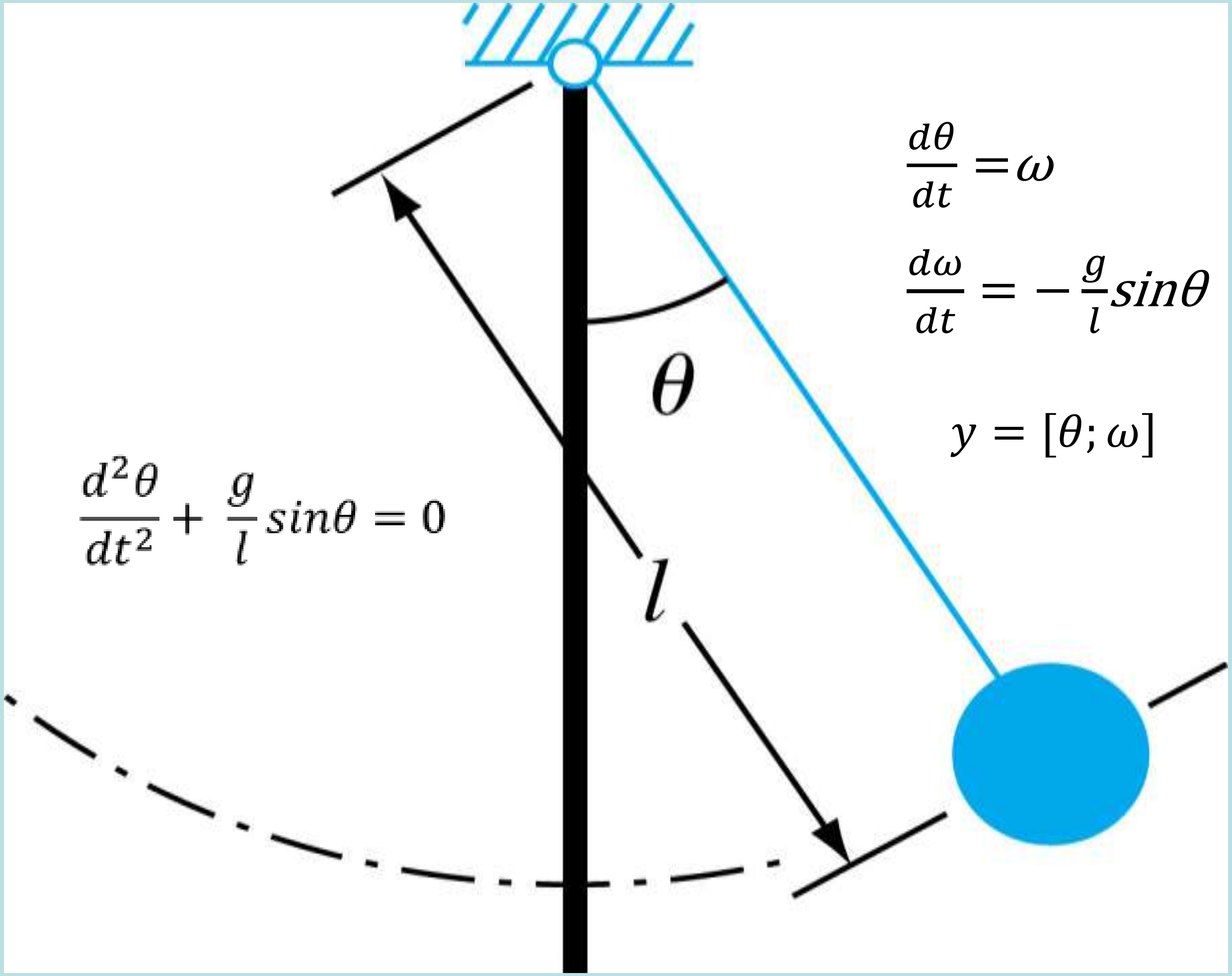
$$y = [y_1; y_2; \dots; y_n]$$

t						

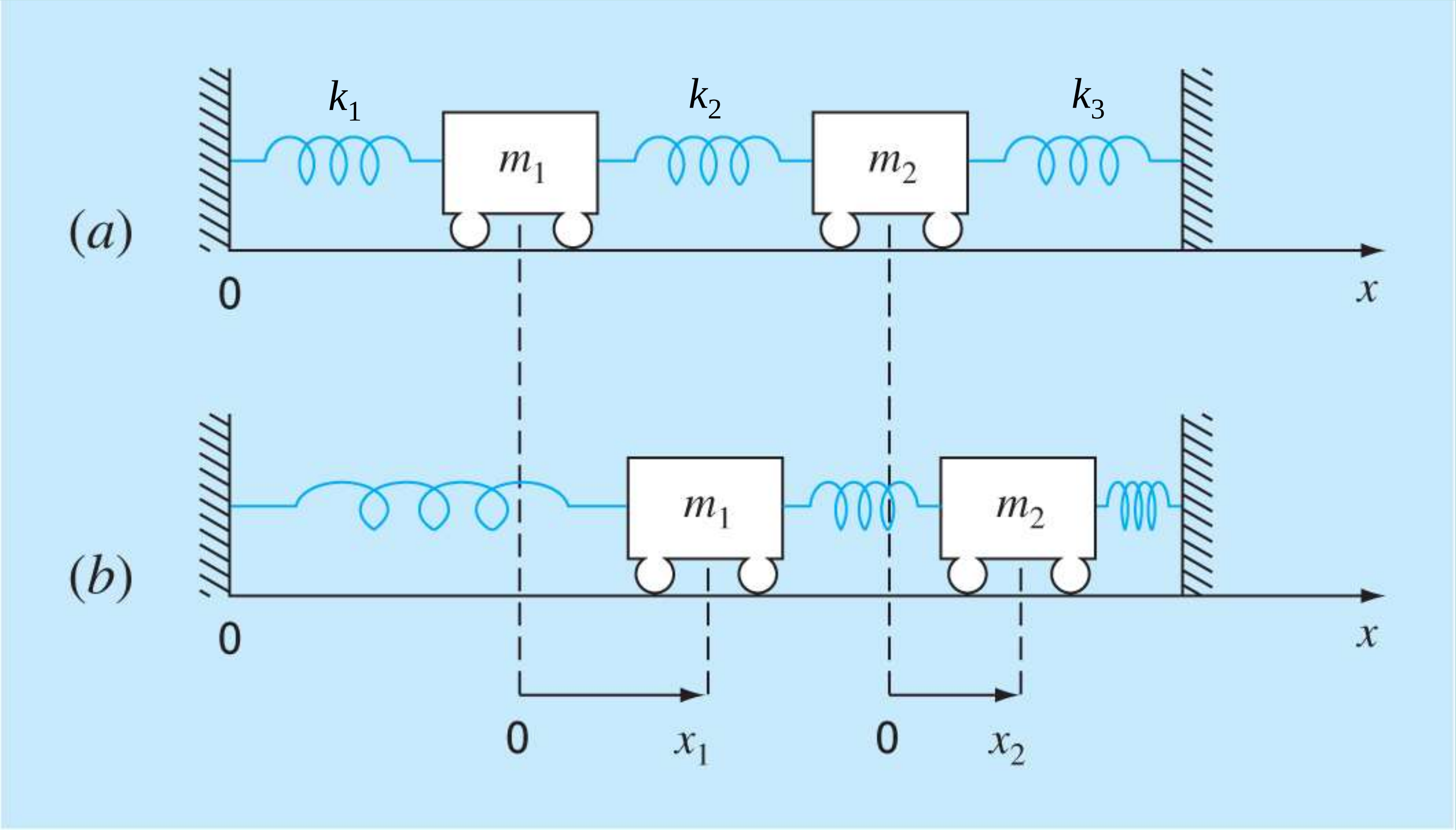
y_1						
y_2						
…						
y_n						

Vector differential equations

Second Order ODEs



Second Order ODEs



Sources of error when we solve a problem

- Error due to the simplifying assumptions made in the development of a mathematical model for the physical problem.
- Uncertainty in physical data: error in collecting and measuring data.
- Mathematical truncation error: error that results from the use of numerical methods in solving a problem, such as solving a differential equation by a numerical method.
- Machine errors: rounding, underflow, overflow.
- Programming errors.

Advanced Methods

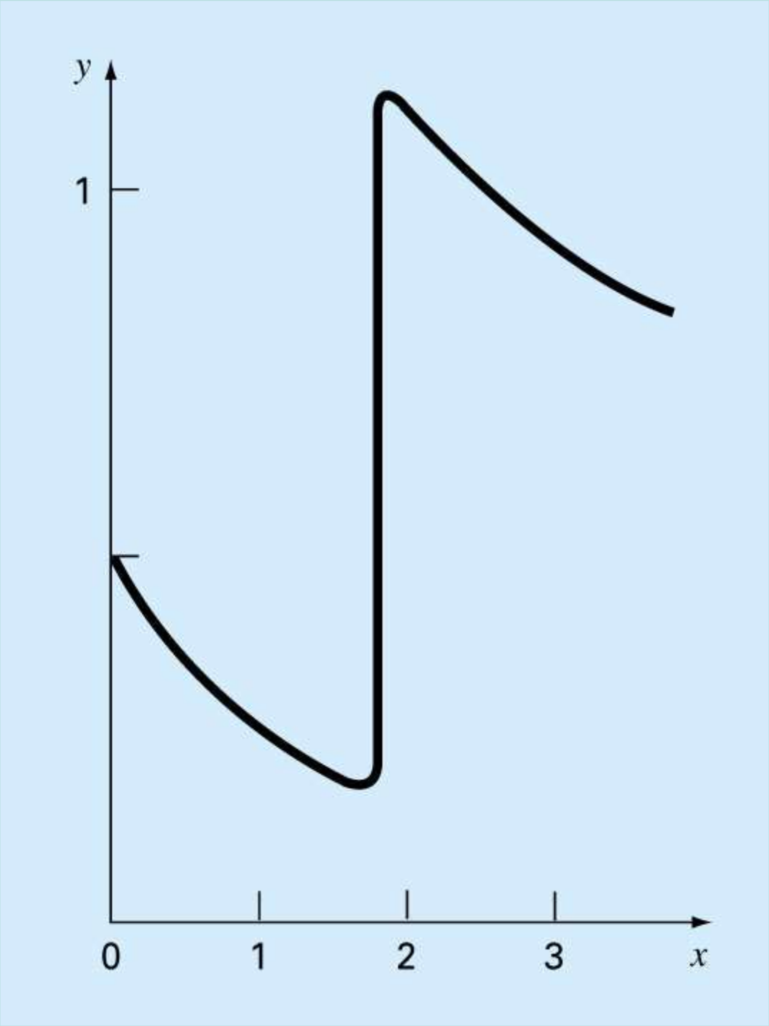
Three methods for solving initial-value problems are covered:

- *Variable step*
- *Multistep*
- *Implicit*

Variable Step Methods

For an ODE with an abrupt changing solution, a constant step size can represent a serious limitation.

Adaptive step-size control



Implementation of adaptive methods requires an estimate of the local truncation error at each step.

The error estimate can then serve as a basis for either lengthening or decreasing step size.

In the embedded pair RK methods, the local truncation error is estimated as the difference between two predictions using different order RK methods.

RK-Fehlberg order4/order5 embedded pair

Fourth order

$$w_{i+1} = w_i + h\left(\frac{25}{216}s_1 + \frac{1408}{2565}s_3 + \frac{2197}{4104}s_4 - \frac{1}{5}s_5\right)$$

Fifth order

$$z_{i+1} = w_i + h\left(\frac{16}{135}s_1 + \frac{6656}{12825}s_3 + \frac{28561}{56430}s_4 - \frac{9}{50}s_5 + \frac{2}{55}s_6\right)$$

re = abs($(z_{i+1} - w_{i+1}) / z_{i+1}$)

RK-Fehlberg order4/order5 embedded pair

$$s_1 = f(t_i, w_i)$$

$$s_2 = f\left(t_i + \frac{1}{4}h, w_i + \frac{1}{4}hs_1\right)$$

$$s_3 = f\left(t_i + \frac{3}{8}h, w_i + \frac{3}{32}hs_1 + \frac{9}{32}hs_2\right)$$

$$s_4 = f\left(t_i + \frac{12}{13}h, w_i + \frac{1932}{2197}hs_1 - \frac{7200}{2197}hs_2 + \frac{7296}{2197}hs_3\right)$$

$$s_5 = f\left(t_i + h, w_i + \frac{439}{216}hs_1 - 8hs_2 + \frac{3680}{513}hs_3 - \frac{845}{4104}hs_4\right)$$

$$s_6 = f\left(t_i + \frac{1}{2}h, w_i - \frac{8}{27}hs_1 + 2hs_2 - \frac{3544}{2565}hs_3 + \frac{1859}{4104}hs_4 - \frac{11}{40}hs_5\right)$$

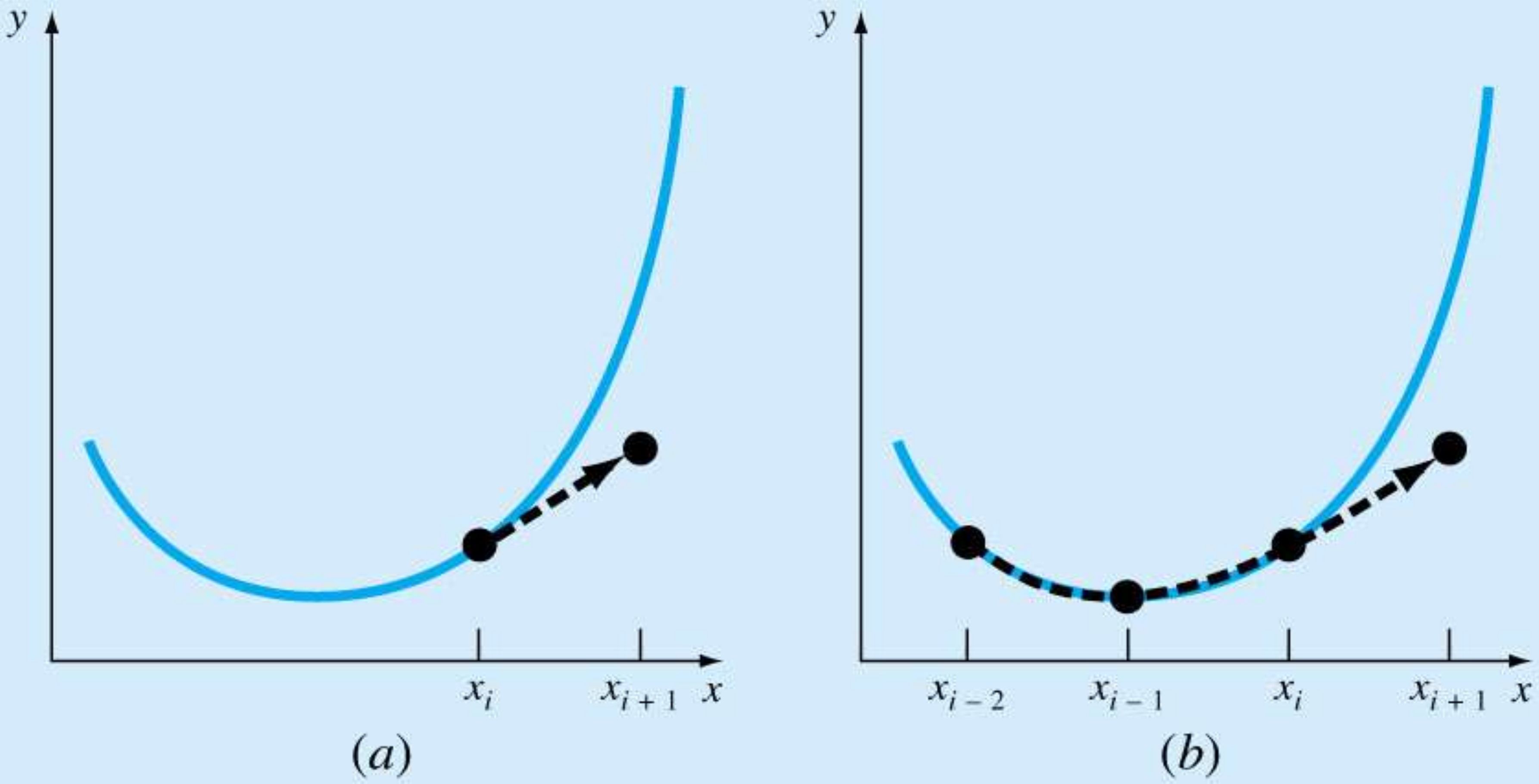
Six function evaluations

Multistep Methods

The one-step methods utilize information at a single point t_i to predict a value of the dependent variable y_{i+1} at a future point t_{i+1}.

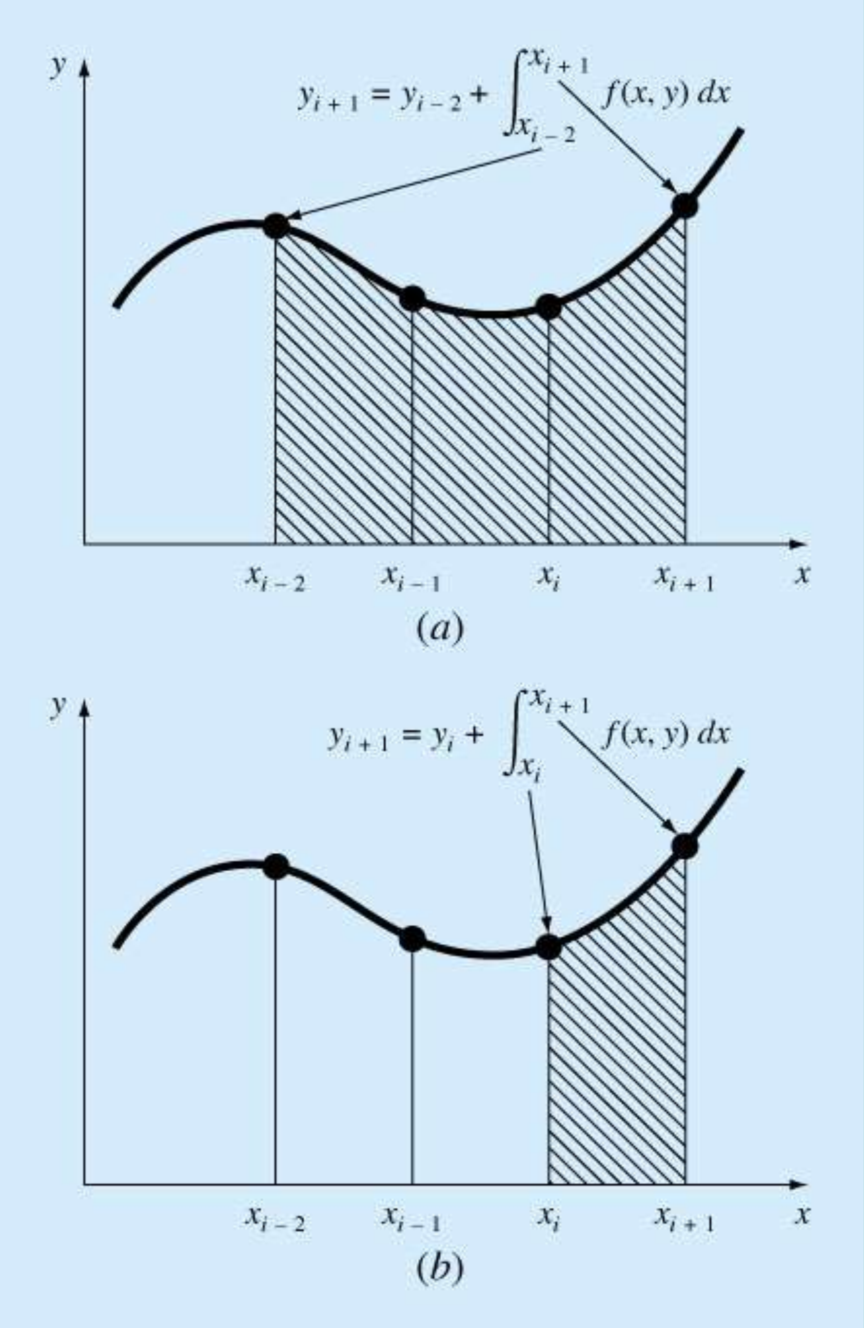
Multistep methods, are based on the insight that, once the computation has begun, valuable information from previous points is at our command.

Start-up phase, using one-step method.



One-step method

Multistep method



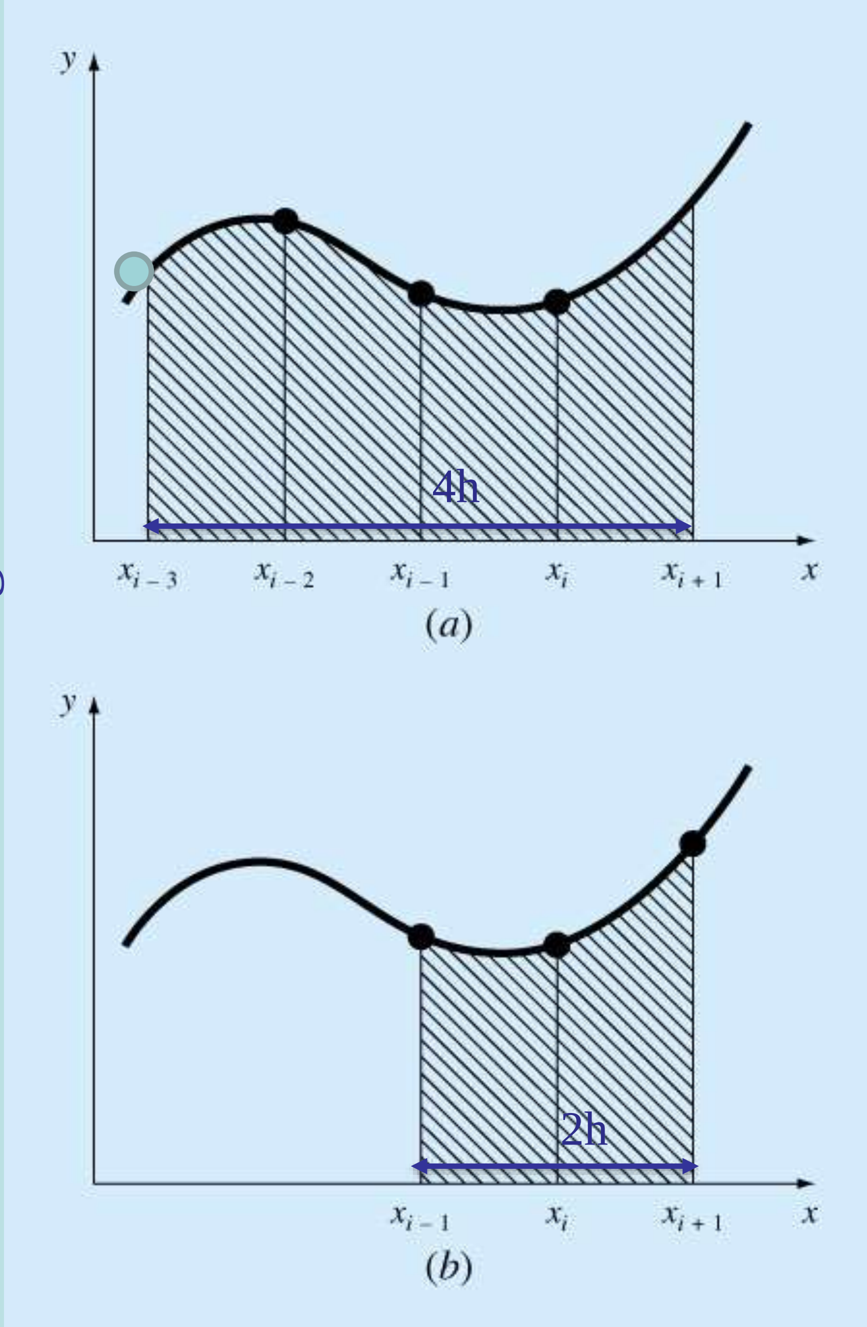
Integration formulas

Newton-Cotes

Adams

2nd degree interpolating polynomial

Exact integration



Newton-Cotes open

$$y_{i+1} = y_{i-3} + \frac{4h}{3}(2f_i - f_{i-1} + 2f_{i-2})$$

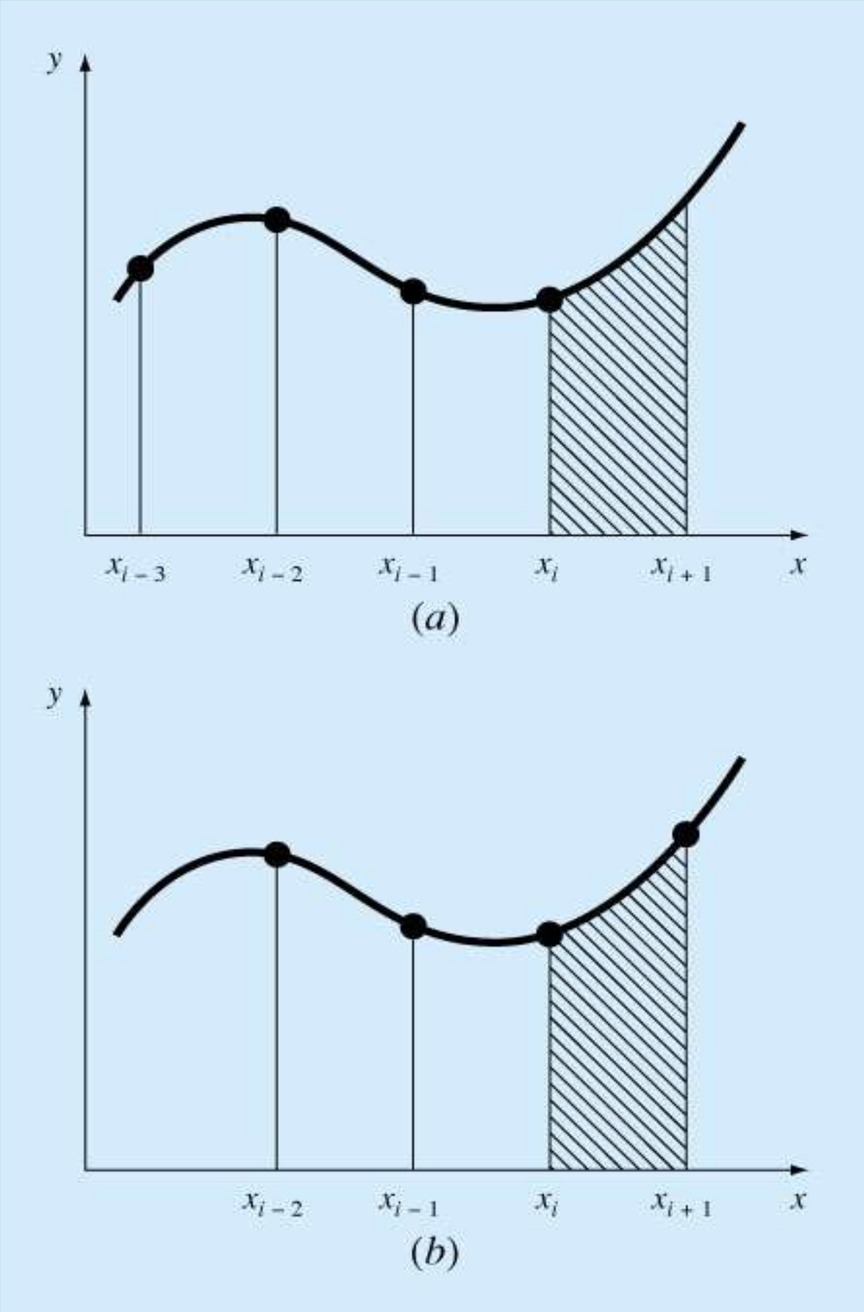
Newton-Cotes closed

$$y_{i+1} = y_{i-1} + \frac{h}{3}(f_{i-1} + 4f_i + f_{i+1})$$

One new function evaluation

Forward Taylor series expansion

Backward finite-differences for derivatives



Adams-Bashforth
open

Adams-Moulton
closed

Predictor-Corrector Multistep Methods

Milne's Method.

Uses the three point Newton-Cotes open formula as a predictor and the three point Newton-Cotes closed formula as a corrector.

Fourth-Order ABM Method.

Predictor: (fourth Adams-Bashforth)

$$y_{i+1}^{0} = y_i^m + h\left(\frac{55}{24}f_i^m - \frac{59}{24}f_{i-1}^m + \frac{37}{24}f_{i-2}^m - \frac{9}{24}f_{i-3}^m\right)$$

Corrector: (fourth Adams-Moulton)

$$y_{i+1}^{j} = y_i^m + h\left(\frac{9}{24}f_{i+1}^{j-1} + \frac{19}{24}f_i^m - \frac{5}{24}f_{i-1}^m + \frac{1}{24}f_{i-2}^m\right)$$

Stiffness

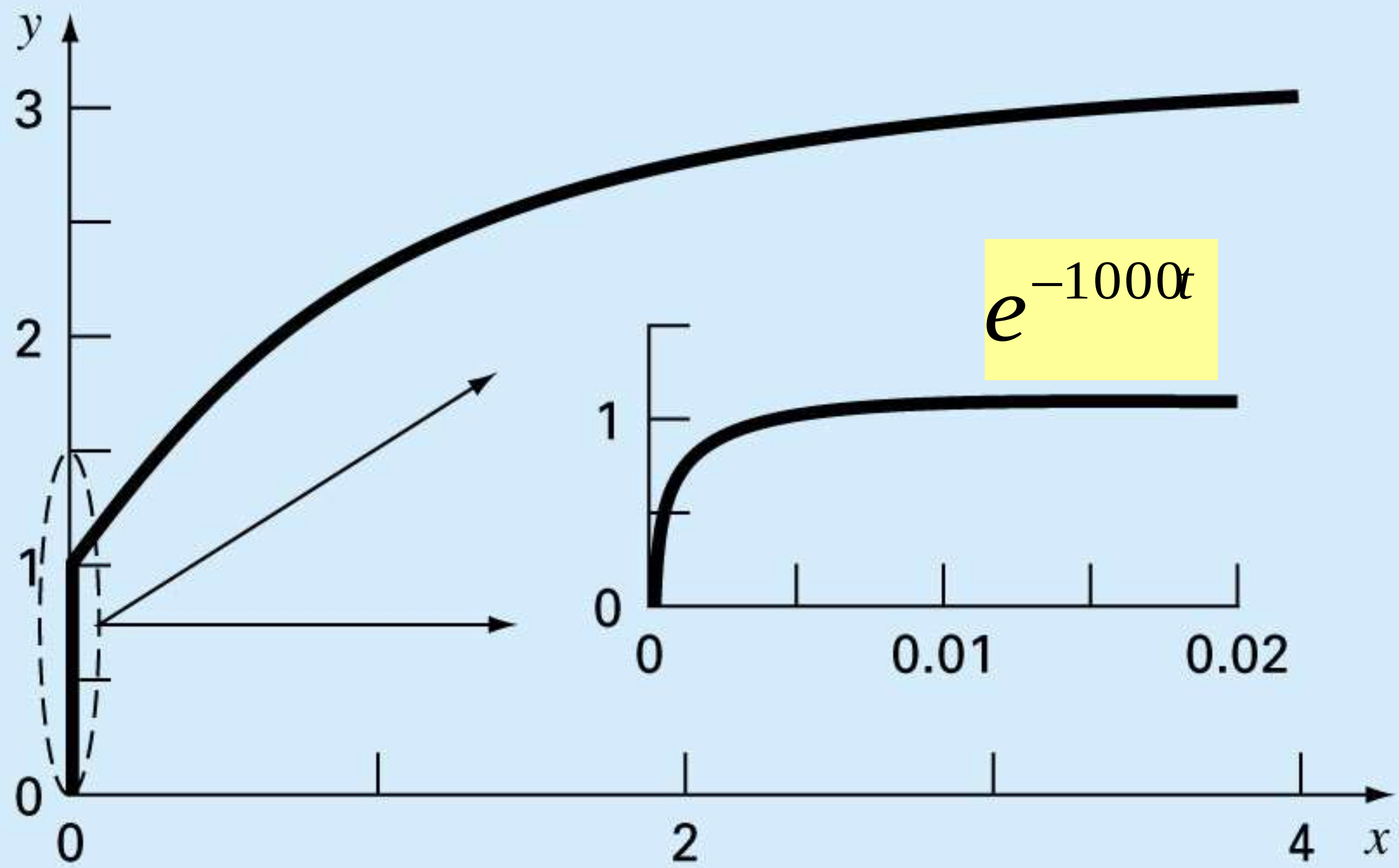
A *stiff system* is one involving rapidly changing components together with slowly changing ones.

Both individual and systems of ODEs can be stiff:

$$\frac{dy}{dt} = -1000y + 3000 - 2000e^{-t}$$

If y(0)=0, the analytical solution is developed as:

$$y = 3 - 0.998e^{-1000t} - 2.002e^{-t}$$


y
3
2
1
0
0
2
4
x
1
0
0
0.01
0.02
e^{-1000t}

Insight into the step size required for stability of such a solution can be gained by examining the homogeneous part of the ODE:

$$\frac{dy}{dt} = -ay$$

$$y = y_0 e^{-at}$$

The solution starts at $y(0)=y_0$ and asymptotically approaches zero.

If Euler's method is used to solve the problem numerically:

$$y_{i+1} = y_i + \frac{dy_i}{dt} h$$

$$y_{i+1} = y_i - a y_i h \qquad \text{or} \qquad y_{i+1} = y_i (1 - ah)$$

The stability of this formula depends on the step size *h*:

$$|1 - ah| \prec 1$$

$$h \succ 2/a \quad \Rightarrow \quad |y_i| \to \infty \quad \text{as} \quad i \to \infty$$

Thus, the step size h must be <2/1000=0.002 to maintain stability.

While this criterion maintains stability, an even smaller step size would be required to obtain an accurate solution.

Rather than using explicit approaches, *implicit* methods offer an alternative remedy.

An implicit form of Euler's method can be developed by evaluating the derivative at a future time.

Implicit Euler Method

$$y_{i+1} = y_i + \frac{dy_{i+1}}{dt} h$$

$$y_{i+1} = y_i - a y_{i+1} h$$

$$y_{i+1} = \frac{y_i}{1 + ah}$$

Backward or implicit method

$f(x_{i+1}, y_{i+1})$

The approach is called *unconditionally stable*. Regardless of the step size:

$$|yi| \to 0 \quad as \quad i \to \infty$$

Implicit Euler Method

The method does not directly give a formula for the new y_{i+1}. Instead, we must work a little to get it.

For the example $y' = 10(1 - y)$, the method gives

$$y_{i+1} = y_i + 10(1 - y_{i+1})h,$$

which, after a little algebra, can be expressed as

$$y_{i+1} = (y_i + 10h)/(1 + 10h)$$

Implicit Euler Method

In the general case, we calculate the value

$$y_{i+1} = y_i + f(x_{i+1}, y_{i+1})*h$$

solving the roots problem

$$y_{i+1} - y_i - f(x_{i+1}, y_{i+1})*h = 0$$

$$g(y_{i+1}) = 0$$

$$g(z) = @(z)\ z - y_i - f(x_{i+1}, z)*h$$

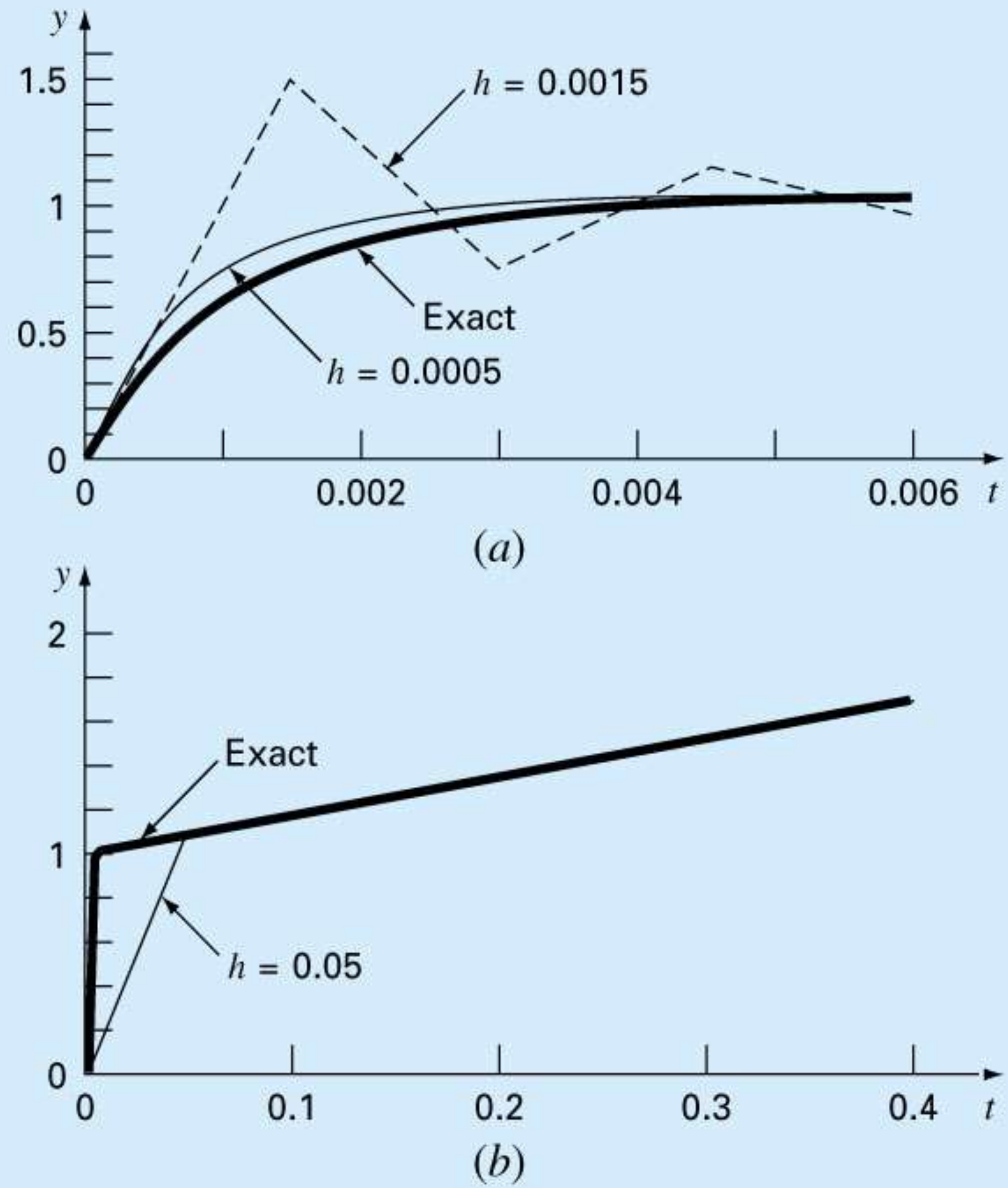
using Newton-Raphson with an initial guess equal to the value of the previous point y_i.

Implicit Euler Method

Use both the explicit and implicit Euler methods to solve

$$\frac{dy}{dt} = -1000\,y + 3000 - 2000e^{-t} \qquad y(0)=0$$

- Use the explicit Euler with step sizes of 0.0005 and 0.0015 to solve for *y* between *t* = 0 and 0.006.
- Use the implicit Euler with a step size of 0.05 to solve for *y* between 0 and 0.4.



Euler explícito

Euler implícito

Initial-Value Problems

Solving ordinary differential equations of the form

$$\frac{dy}{dx} = f(x, y) \qquad y(x_0) = y_0$$

$$y_{i+1} = y_i + \phi h$$

Constant step size
One-step methods

Variable step size
Multistep methods

Boundary-Value Problems

An ODE is accompanied by auxiliary conditions. These conditions are used to evaluate the integral that result during the solution of the equation. An n^{th} order equation requires n conditions.

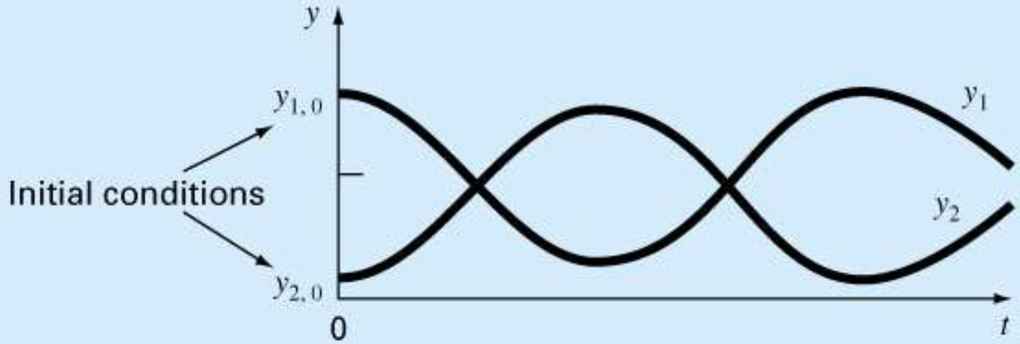
If all conditions are specified at the same value of the independent variable, then we have an *initial-value problem*.

If the conditions are specified at different values of the independent variable, usually at extreme points or boundaries of a system, then we have a *boundary-value problem*.

$$\frac{dy_1}{dt} = f_1(t, y_1, y_2)$$

$$\frac{dy_2}{dt} = f_2(t, y_1, y_2)$$

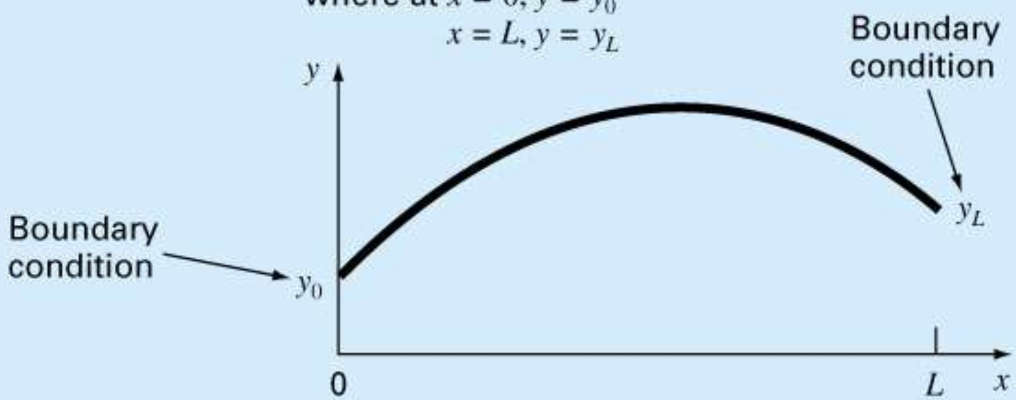
where at $t = 0$, $y_1 = y_{1,0}$ and $y_2 = y_{2,0}$



(a)

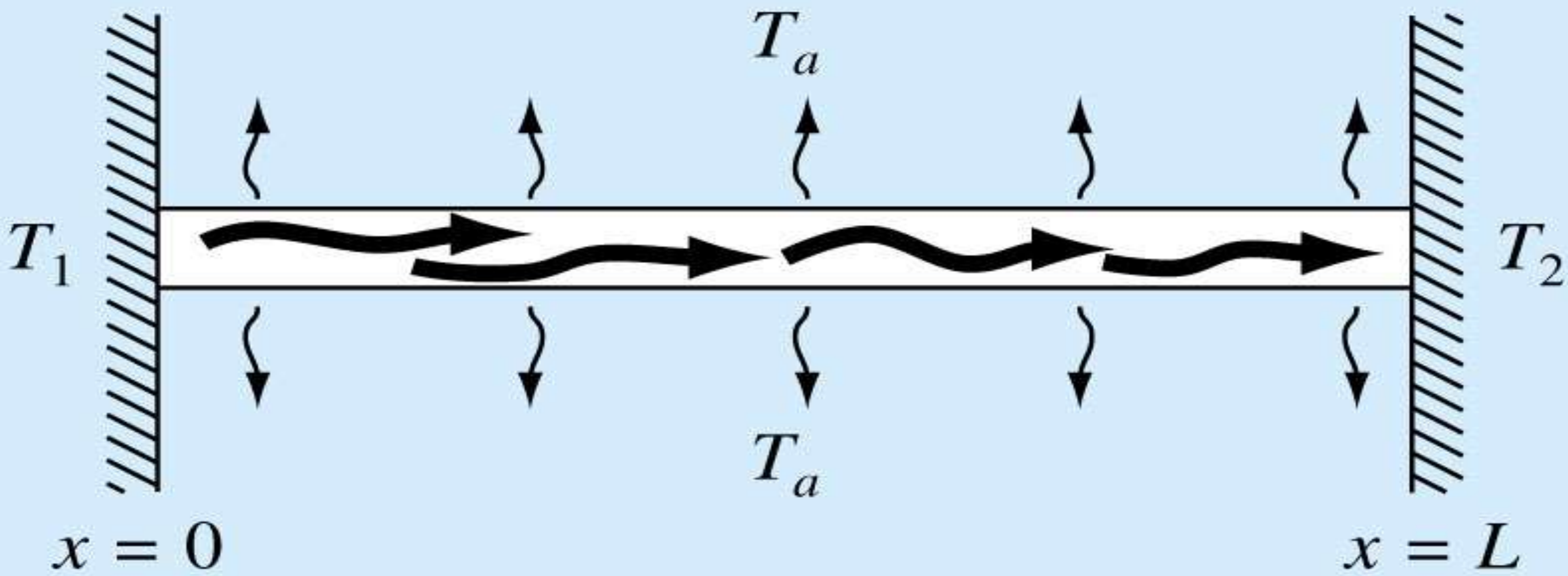
$$\frac{d^2y}{dx^2} = f(x, y)$$

where at $x = 0$, $y = y_0$
$x = L$, $y = y_L$



(b)

Heated Rod



T(x) in [0,L]

$$\frac{d^2T}{dx^2} + h'(T_a - T) = 0$$

$$T_a = 20$$

$$L = 10\,m$$

$$h' = 0.01\,m^{-2}$$

$$\left.\begin{aligned} T(0) &= T_1 = 40 \\ T(L) &= T_2 = 200 \end{aligned}\right\}$$

T' = z

z' = h´(T-T_a)

z(0) = ¿?

Dirichlet
Boundary Conditions

Using the RK4 method with $z(0)=10$ and h=2 we obtain

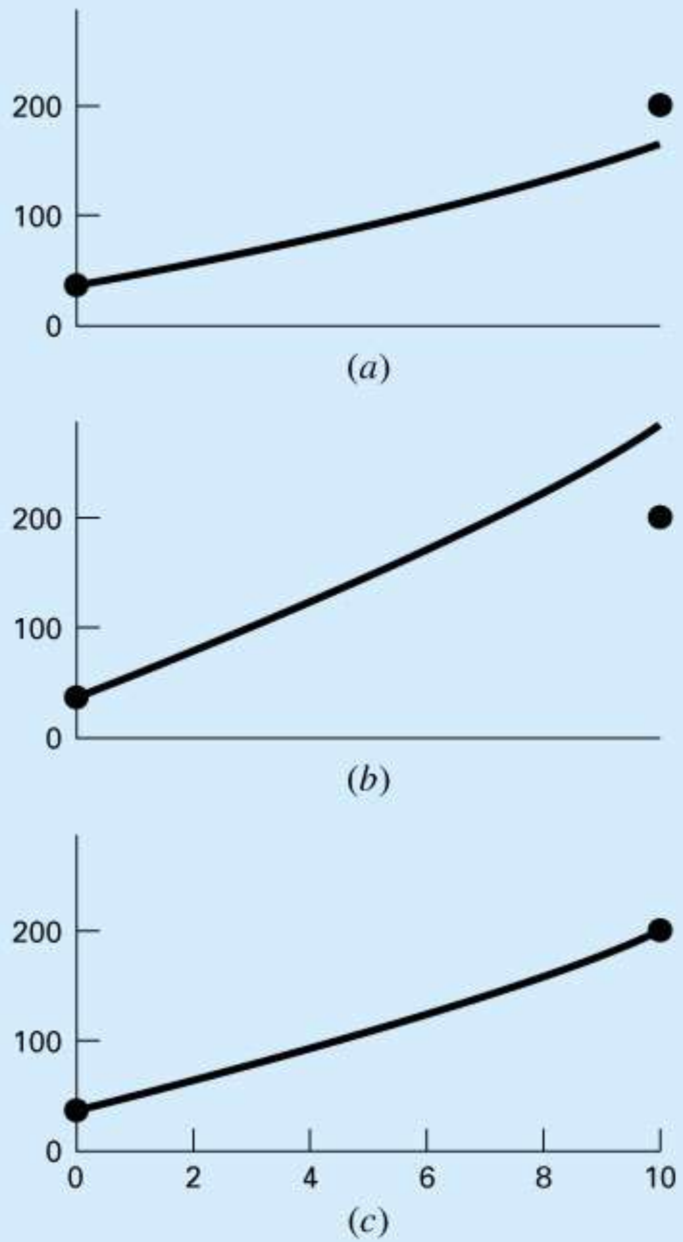
$T(10)=168.3797$.

This differs from $T(10)=200$.

Therefore a new guess is made, $z(0)=20$, and the shooting is performed again.

$T(10)=285.8980$

Since the two sets of points, $(z, T)_1$ and $(z, T)_2$, are linearly related, a linear interpolation formula can be used to compute the value of $z(0)$ as 12.6907 to determine the correct solution.



(z_{01}, T_{L1})

$T_L = \text{shooting}(f, T_0, z_0)$

(z_{02}, T_{L2})

(z_0^*, T_L)

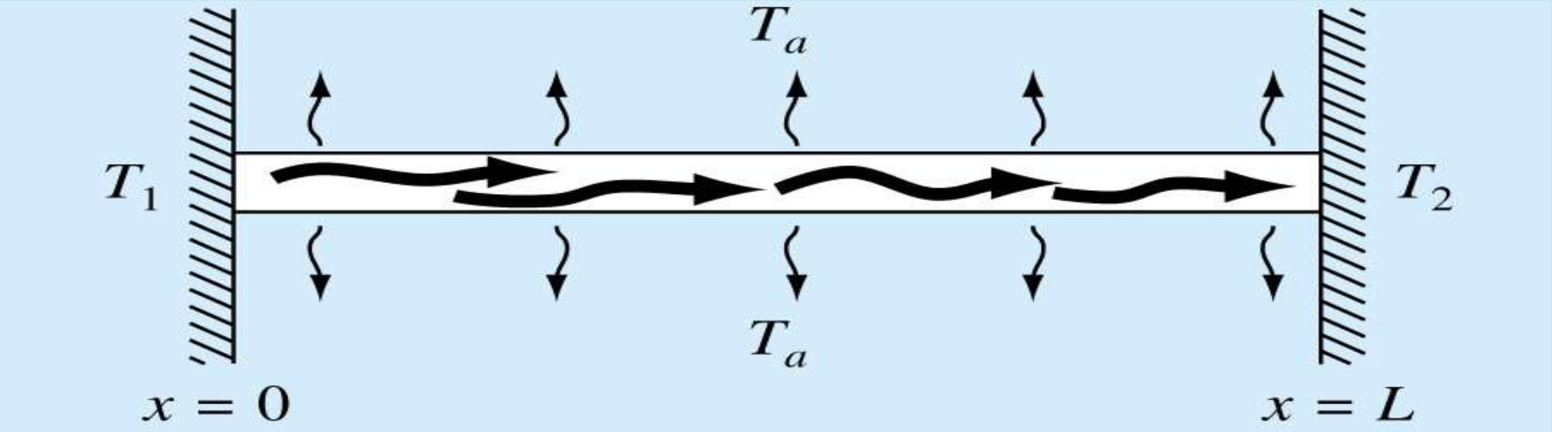
For a nonlinear problem a better approach involves recasting it as a roots problem.

```
g = @(z) shooting(f, h, 0, L, T0, z) - TL
```

Driving this new function $g(z_0)$ to zero, using bisection, provides the solution z_0^*.

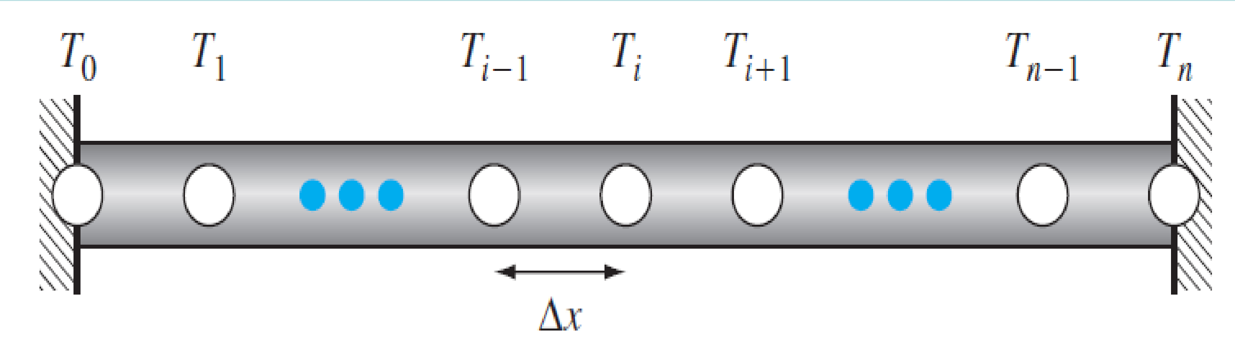
And using the value z_0^* we can solve for $T(x)$ in $[0,L]$.

The most common alternative to the shooting method is the finite differences method.



$$\frac{d^2T}{dx^2} + h'(T_a - T) = 0$$

We divide the range of integration (0, L) into n equal subintervals of length h each.



h

Finite differences are substituted for the derivatives in the original equation.

$$y_i' = \frac{y_{i+1} - y_{i-1}}{2h} \qquad y_i'' = \frac{y_{i-1} - 2y_i + y_{i+1}}{h^2}$$

$$\frac{d^2T}{dx^2} = \frac{T_{i-1} - 2T_i + T_{i+1}}{\Delta x^2}$$

$$\frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} - h'(T_i - T_a) = 0$$

$$-T_{i-1} + (2 + h'\Delta x^2)T_i - T_{i+1} = h'\Delta x^2 T_a$$

The last equation can be written for each of the n-1 interior nodes. The first and last nodes are specified by the boundary conditions T_0 and T_L.

Using four interior nodes with a segment length of $\Delta x=2$ and $h'=0.01$, we have $h'*(\Delta x)^2=0.04$ and

$$\begin{bmatrix} 2.04 & -1 & 0 & 0 \\ -1 & 2.04 & -1 & 0 \\ 0 & -1 & 2.04 & -1 \\ 0 & 0 & -1 & 2.04 \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{Bmatrix} = \begin{Bmatrix} 40.8 \\ 0.8 \\ 0.8 \\ 200.8 \end{Bmatrix}$$

The linear system is tridiagonal and diagonally dominant.