Differential equations

Differential equation is an equation containing an unknown function $y = y(x)$ and its derivatives. The problem is to find y.

Many physical theories can be expressed in terms of differential equations: what happens here or right now depends only on what happened in the nearest neighborhood or a short time before.

If the unknown function depends only on one variable, the equation is an *ordinary differential equation* (ODE):

An object falling in a constant gravitational field:

$$\frac{\mathrm{d}}{\mathrm{d}t}h(t) = -gt.$$

Here the height h is known at some initial time, e.g. $h(0) = H$, where H is a constant. Such a problem is called an *initial value problem*.

Usually a unique solution exists.

The general solution of the previous equation is obtained by integration:

$$h(t) = \int_0^t -gt\mathrm{d}t = -\frac{gt^2}{2} + C.$$

This gives a family of solutions. The initial value is needed to select a specific solution.

Uniformly loaded beam (like a bridge):

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2} w(x) = Aw(x) + Bx(x - L).$$

The displacement w must vanish at both ends of the bridge: $w(0) = w(L) = 0$. This is a *boundary value problem*.

Usually more difficult to solve than the initial value problems.

Depending on the boundary values there may be a solution or no solution at all.

If the unknown function depends on several variables and the equation contains its partial derivatives, we have a *partial differential equation* (PDE).

Hamiltonian equations of motion

$$\frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial \mathcal{H}}{\partial q_i}$$
$$\frac{\mathrm{d}q_i}{\mathrm{d}t} = \frac{\partial \mathcal{H}}{\partial p_i}$$

are initial value partial equations.

If the edges of a plate are kept in a certain temperature the heat flow in the plate is a boundary value problem.

The highest derivative in the equation determines the *order* of the equation. For example,

$$y' - xy = y''$$

is of the second order.

The general solution of an equation of order n contains n integration constants that are used to select a specific solution.

If the equation contains only linear combinations of the unknown function y and its derivatives, the equation is *linear*:

$$y - 2y' + y'' = 0,$$
$$y + xy' = e^x.$$

Otherwise the equation is nonlinear:

$$yy' = y'',$$
$$y^2 = y'.$$

If the coefficients of the unknown function and its derivatives are constants, the equation is a *constant coefficient differential equation*.

There are analytical solution methods for first order equations and higher order linear constant coefficient equations.

A solution of other kinds of equations may in some special cases be found e.g. by applying some integral transform (Laplace, Fourier, Hankel etc.)

Even if an analytical solution exists, it may be very complicated or difficult to find.

A linear constant coefficient equation

$$y'' + ay' + by = 0$$

can be solved using its *characteristic equation*

$$\lambda^2 + a\lambda + b = 0.$$

If the roots λ_1 and λ_2 are real and distinct, the solution is

$$y = C_1 e^{\lambda_1 x} + C_2 e^{\lambda_2 x},$$

where the constant C_1 and C_2 are chosen so that the solution satisfies the initial conditions.

The solution is a linear combination of two different linearly independent solutions.

If the roots are identical the solution is

$$y = C_1 e^{\lambda_1 x} + C_2 x e^{\lambda_1 x},$$

Finally, if the roots are complex conjugates, the solution can be written in the form

$$y = C_1 e^{ax} \cos bx + C_2 e^{bx} \sin bx,$$

(1) Numerical methods can usually be easily generalized to a set of several functions.

(2) Any equation of order n can always be replaced with a set of n first order equations by taking the derivatives as new unknown functions.

For example, the equation

$$y'' + xy' - y = 0$$

is equivalent to the pair of equations

$$\begin{aligned} u &= y', \\ u' + xu - y &= 0. \end{aligned}$$

From (1) and (2) it follows that it is sufficient to study solutions of first order equations only.

Such an equation can always be expressed in the form

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}(t)).$$

If, in addition, the value of the unknown function is known at one point, $\mathbf{y}(t_0) = \mathbf{y}_0$, the problem is an initial value problem.

Lipschitz condition: an initial value problem has a unique continuously differentiable solution in $A \subset \mathbf{R} \times \mathbf{R}^n$, if there is a constant L such that for all $(t, \mathbf{z}_1) \in A$, $(t, \mathbf{z}_2) \in A$

$$\| \mathbf{f}(t, \mathbf{z}_1) - \mathbf{f}(t, \mathbf{z}_2) \| < L \| \mathbf{z}_1 - \mathbf{z}_2 \| .$$

Stability

The second order initial value problem

$$\begin{aligned} &y'' - 10y' - 11y = 0, \\ &y(0) = 1, \\ &y'(0) = -1 \end{aligned}$$

has a solution $y(x) = e^{-x}$. The solution will approach zero when $x \to \infty$.

If the initial value is $y(0) = 1 + a$, the solution is

$$y(x) = \left(\frac{11}{12}a + 1\right) e^{-x} + \frac{a}{12} e^{11x}.$$

When $x \to \infty$, the solution will increase beyond all limits however small a is.

The solution is unstable. Whatever method is used to solve the equations, even a small rounding error may lead to a totally erroneous solution.

Such equations are called *stiff*.

Even a stable problem can be *ill-conditioned* (e.g. discretization may lead to a nearly singular set of linear equations.)

Let $\mathbf{y_1}$ and $\mathbf{y_2}$ be two solutions corresponding to two different initial values. The solution is *stable*, if for all $\epsilon > 0$ there is $\delta > 0$ such that

$$|\mathbf{y_1}(t) - \mathbf{y_2}(t)| \leq \epsilon$$

for all $t \geq 0$ always when

$$|\mathbf{y_1}(0) - \mathbf{y_2}(0)| \leq \delta.$$

Thus a small disturbance in the initial value has only a small effect on the solution.

A solution is *asymptotically stable* if

$$|\mathbf{y_1}(t) - \mathbf{y_2}(t)| \to 0,$$

when $t \to \infty$. Small disturbances in the initial values may first affect the solutions, but further away the deviation will diminish and the solutions approach each others.

Especially nonlinear equations can be very sensitive to initial values and therefore be unstable (chaos).

The Duffing equation is an example of an equation that has badly behaving solutions:

$$y'' + y + ry^3 = a \sin \Omega t,$$

where r, a and Ω are constants.

When $r = 0$, $\Omega = 1$ and $y(0) = y'(0) = 0$ the solution is

$$y = \frac{1}{2} a \sin t - \frac{1}{2} at \cos t.$$

This is unstable and the amplitude approaches infinity (resonance).

When $r \neq 0$ the behaviour is difficult to predict. For suitable values of the constants the solutions seems to show relatively stable oscillation but then suddenly jumps far away.

Initial value problems, single step methods

The value of the function (and its derivative) is known at one point. This value is used to evaluate the function at the next point. Earlier values of the function are not used.

Let's begin by trying to solve a differential equation using a Taylor series.

Assume that the equation is written as

$$y'(x) = f(x, y(x)),$$

where y is the function to be determined and $f(x, y(x))$ is some expression of x and $y(x)$.

The function y can be expressed as a Taylor series

$$y(x) = y(x_0) + y'(x_0)(x - x_0) + \frac{1}{2}y''(x_0)(x - x_0)^2 + \dots.$$

As an example, take the equation

$$y'(x) = -2xy(x), \quad y(0) = 1.$$

(The exact solution is $y(x) = e^{-x^2}$.)

Since the function is known at $x = 0$, expand y as a series around zero. At $x = h$ the function has the value

$$y(h) = y(0) + y'(0)h + \frac{1}{2}y''(0)h^2 + \frac{1}{6}y'''(0)h^3 + +\frac{1}{2}y^{(4)}(0)h^4 + \dots.$$

Using the initial value $y(0) = 1$ and the original equation we get $y'(0) = -2 \times 0 \times y(0) = 0$. The higher derivatives in the expansion can be evaluated by differentiating the original equations as many times as needed:

$$\begin{aligned} y'' &= -2y - 2xy' = -2y + 4x^2y, \\ y''' &= -4y' - 2xy'' = 12xy - 8x^3y, \\ y^{(4)} &= -6y'' - 2xy''' = 12y - 48x^2y + 16x^3y. \end{aligned}$$

At $x = 0$ we get the values

$$y''(0) = -2$$
$$y'''(0) = 0$$
$$y^{(4)}(0) = +12$$

Substituting these into the series expansion we get

$$y(h) = 1 - h^2 + 2h^4 + \dots.$$

When a finite number of terms is included we get a polynomial that gives an approximate solution around zero. Using the polynomial we can find the value of the unknown function at $x = h$.

The accuracy of a truncated series expansion will become worse with increasing h. Hence it is not usually sufficinet to give the whole solution. However, applying the same method, we can find a new expansion at $x = h$; using the new expansion we can evaluate the function at $x = 2h$.

Idea: find some approximation for the function to be determined, and use it to extrapolate the function a small step further. By repeating this sufficiently many times the values of the function can be evaluated for an arbitrary range,

h is the *step length* of the method. The step has to be short enough to avoid large errors in the extrapolation.

Taylor series is practical only if analytical expressions of the derivatives can be easily obtained.

Solution of the example problem with the Taylor series method and two different steps, $h = 0.2$ and $h = 0.1$.

x	y	y	e^{-x^2}
0.00	1.00000	1.00000	1.00000
0.10		0.99050	0.99005
0.20	0.96400	0.96166	0.96079
0.30		0.91512	0.91393
0.40	0.85765	0.85352	0.85214
0.50		0.78022	0.77880
0.60	0.70336	0.69898	0.69768
0.70		0.61369	0.61263
0.80	0.53105	0.52803	0.52729
0.90		0.44524	0.44486
1.00	0.36881	0.36792	0.36788

The function was approximated with a Taylor series omitting all terms proportional to h^5 and higher powers. Thus the error in one step is of the order of h^5. This is the *local error* of the method.

The order of the method is 4, which is often denoted as $\mathcal{O}(h^5)$.

In each step the error can increase by an amount that is of the order h^5. At the end of the whole interval the accumulated error is $1/h \times h^5 = h^4$. This is the *global error* of the solution.

Euler's method

A very simple method is obtained when the step is chosen so short that already the two first terms of the Taylor series give a satisfactory approximation.

The method is not very accurate, so it is not recommended. However, it illustrates basic ideas that are used in more advanced methods.

When the function and its derivative at x are known, their values at $x + h$ are obtained from

$$y(x + h) = y(x) + hy'(x),$$
$$y'(x + h) = f(x + h, y(x + h)).$$

```
program euler
! Solve the differential equation y'=f(x,y)
! with initial value y(a)=y0
! in the range a <= x <= b
! test with y'=-xy, the solution of which
! is y=exp(-x**2/2)
  implicit none
```

```
  call euler1(0.0, 1.0, 1.0, 0.1)
contains


real function f(x, y)
 real, intent(in) :: x, y
 f = -x*y
end function

subroutine euler1 (a, b, y0, h)
  real, intent(in) :: a, b, y0, h
  real :: x, y, dy, e
  x=a
  y=y0
dy=f(x,y)
e=exp(-x**2/2); e=abs((e-y)/e)
write(*,'(4F10.6)') x,y,dy,e

do while (x < b)
 x=x+h
 y=y+h*dy
 dy=f(x,y)
 e=exp(-x**2/2);  e=abs((e-y)/e)
```

```
      write(*,'(4F10.6)') x,y,dy,e
    end do
  end subroutine
  end program
```

```
 x        y        y'       rel.error

0.000000  1.000000  0.000000  0.000000
0.100000  1.000000 -0.100000  0.005013
0.200000  0.990000 -0.198000  0.009999
0.300000  0.970200 -0.291060  0.014856
0.400000  0.941094 -0.376438  0.019475
0.500000  0.903450 -0.451725  0.023743
0.600000  0.858278 -0.514967  0.027545
0.700000  0.806781 -0.564747  0.030761
0.800000  0.750306 -0.600245  0.033268
0.900000  0.690282 -0.621254  0.034941
1.000000  0.628156 -0.628156  0.035655
```

Usually the step has to be very short to get satisfactory accuracy.

Let Y be the exact solution. The error of Euler's method at $x+h$ is

$$\begin{aligned} d &= Y(x+h) - y(x+h) \\ &= Y(x) - y(x) + h[f(x, Y(x)) - f(x, y(x))] + h^2 Y''(z)/2, \end{aligned}$$

where $x < z < x+h$.

If $Y(x) = y(x)$ (i.e the solution is exact at x), after one step the error is $h^2 Y''(z)/2$. The local error of the method is of the second order as a function of the step size ($\mathcal{O}(h^2)$); thus it is a first order method.

In general, if the local error is $\mathcal{O}(h^{n+1})$ the method is of the order n.

The total error after each integration step depends on the error accumulated this far + the local error in the next step.

The total error of Euler's method is

$$\begin{aligned} d &= Y(x) - y(x) + hf_y(x,\xi)(Y(x) - y(x)) + h^2Y'' \\ &= (1 + hf_y)(Y(x) - y(x)) + h^2Y'' \\ &= (1 + hf_y)[\text{total error at } x]+ \\ &\quad [\text{local error at } x + h]. \end{aligned}$$

Euler's method is stable, if $|1 + hf_y| < 1$. The method is stable only if the step is short enough.

In the case of a set of equations the stability criterion of Euler's method is $|1 + h\lambda_1| < 1$, where λ_1 is the largest eigenvalue of the coefficient matrix (Jacobian matrix). Even a stable set of equations may be difficult to solve. If the real parts of the eigenvalues of the coefficient matrix have very different magnitudes, the set of equations is stiff.

Example:

$$\begin{aligned} y' &= -2000y + 999.75u + 1000.25, \\ u' &= y - u, \\ y(0) &= 0, u(0) = -2. \end{aligned}$$

$$A = \begin{pmatrix} -2000 & 999.75 \\ 1 & -1 \end{pmatrix}.$$

The eigenvalues are -2000.5 and -0.5 and the exact solution

$$y = -1.499875e^{-0.5x} + 0.499875e^{-2000.5x} + 1,$$
$$u = -2.99975e^{-0.5x} - 0.00025e^{-2000.5x} + 1.$$

In each of these the second term has noticeable effect only at very small values of x. Near the origin, an extremely short step length is needed, although the first term varies only very slowly.

Leapfrog

Particularly dynamical systems are described by equations like

$$\frac{\mathrm{d}^2 x}{\mathrm{d}t^2} = F(x).$$

In the leapfrog method the positions x and velocities v are updated alternatingly:

$$\begin{aligned} x_i &= x_{i-1} + v_{i-1/2}\Delta t \\ a_i &= F(x_i) \\ v_{i+1/2} &= v_{i-1/2} + a_i \Delta t \end{aligned}$$

Another version is:

$$\begin{aligned} a_i &= F(x_i) \\ x_{i+1} &= x_i + v_i \Delta t + \frac{1}{2} a_i (\Delta t)^2 \\ v_{i+1} &= v_i + \frac{1}{2}(a_i + a_{i+1})\Delta t \end{aligned}$$

The method is stable also for periodic motion if the time step is constant and $\Delta t \leq 2/\omega$, where ω is the frequency.

The method is simple (and fast) and the energy of the system is conserved, which makes it accurate.

Implicit Euler's method

Euler's method is an *explicit method* : all values needed to calculate the next point are known in advance.

In Euler's method the derivative is evaluated at the beginning of each subinterval and used to estimate the slope of the function in the whole subinterval. The result is correct only if the function is a straight line. If the graph of the function is curved, it would be better to use the mean value of the derivative. It could be estimated by e.g.

$$\bar{y}' = \frac{y'(x) + y'(x+h)}{2}.$$

Here the value $y'(x+h) = f(x+h, y(x+h))$ is not known in advance, Thus this is a *implicit method* .

The unknown derivative can be estimeted using the original Euler's method:

$$\begin{aligned} y(x+h) &= y(x) + hy'(x), \\ y'(x+h) &= f(x+h, y(x+h)), \\ y(x+h) &= y(x) + \frac{h}{2}(y'(x) + y'(x+h)). \end{aligned}$$

This iteration could even be repeated by using the corrected y to calculate a more accurate value for the derivative, and then again a new y.

Implicit methods are often more stable than explicit methods.

Idea of a *predictor-corrector method*: Using the values known at the current point find an approximate value (predictor) of the function at the next point. Using the prediction find an improved value (corrector).

Runge–Kutta method

As an example, we'll derive the second order Runge–Kutta method. Let the equation be

$$y' = f(x, y),$$

whence

$$y'' = f' = f_x + f_y \frac{dy}{dx} = f_x + f_y f.$$

The Taylor series gives the estimate

$$\begin{aligned} y_{n+1} &= y_n + hy' + \frac{h^2}{2} y'' = y_n + hf + \frac{h^2}{2} f' \\ &= y_n + hf + h^2 (\frac{1}{2} f_x + \frac{1}{2} f_y f). \end{aligned}$$

Denote

$$\begin{aligned} k_1 &= hf(x_n, y_n), \\ k_2 &= hf(x_n + \alpha h, y_n + \beta k_1). \end{aligned}$$

Because $f = y'$, k_1 and k_2 are estimates of the change of y when x increases by h.

The actual change is expressed as a linear combination of these:

$$\begin{aligned} y_{n+1} &= y_n + ak_1 + bk_2 \\ &= y_n + ahf + bhf(x_n + \alpha h, y_n + \beta hf) \\ &\approx y_n + ahf + bh\left[f + \alpha hf_x + \beta hf_y f\right] \\ &= y_n + (a+b)hf + h^2(\alpha bf_x + \beta bf_y f). \end{aligned}$$

Comparing this with the Taylor series we get a set of equations

$$\begin{aligned} a + b &= 1, \\ \alpha b &= \frac{1}{2}, \\ \beta b &= \frac{1}{2}. \end{aligned}$$

This is satisfied, if we choose, for example, $a = b = 1/2$, $\alpha = \beta = 1$, which gives

$$\begin{aligned} k_1 &= hf(x_n, y_n), \\ k_2 &= hf(x_n + h, y_n + k_1) = hf(x_n + h, y_n + hf(x_n, y_n)), \\ y_{n+1} &= y_n + k_1/2 + k_2/2 = y_n + \frac{h}{2}(f(x_n, y_n) + f(x_n + h, y_n + hf(x_n, y_n)) \end{aligned}$$

i.e. we end up with the implict Euler's method.

The fourth order Runge–Kutta is a frequently used method. Its derivation is rather laborious. It involves a set of 11 equations with 13 unknowns.

The most common method is:

$$
\begin{aligned}
k_1 &= hf(x_n, y_n), \\
k_2 &= hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1), \\
k_3 &= hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2), \\
k_4 &= hf(x_n + h, y_n + k_3), \\
y_{n+1} &= y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4).
\end{aligned}
$$

The global error of this is of the order h^4.

Example: The second order equation

$$y'' + y = 0$$

can be written as a pair of equations

$$\begin{aligned} y' &= u \\ u' &= -y. \end{aligned}$$

The analytic solution with the initial values $y(0) = 0$, $y'(0) = 1$ is $y = \sin x$.

The solution using the Runge–Kutta method is essentially as

```
h=0.1
x=0.0; y= 0.0 ;  u= 1.0
do while (x <= 1.0)
 ky1 = h * u
 ku1 = h * (-y)
 ky2 = h * (u+ku1/2)
 ku2 = h * (-(y+ky1/2))
 ky3 = h * (u+ku2/2)
 ku3 = h * (-(y+ky2/2))
 ky4 = h * (u+ku3)
 ku4 = h * (-(y+ky3))
 y = y + (ky1 + 2*ky2 + 2*ky3 + ky4)/6.0
 u = u + (ku1 + 2*ku2 + 2*ku3 + ku4)/6.0
 x = x+h
 write(*,*) x,y
end do
```

The output of the program is (the rightmost column gives the true value):

```
 0.1000000    9.9833339E-02 9.9833421E-02
 0.2000000    0.1986692    0.1986693
 0.3000000    0.2955200    0.2955202
 0.4000000    0.3894180    0.3894183
 0.5000000    0.4794252    0.4794255
 0.6000000    0.5646421    0.5646425
 0.7000000    0.6442173    0.6442177
 0.8000001    0.7173556    0.7173561
 0.9000001    0.7833264    0.7833270
  1.000000    0.8414705    0.8414711
```

The accuracy can be tested by a method called step doubling. Proceed from x to $x + 2h$ by taking one step of $2h$ to get y_1 and two steps of h to get y_2. If the exact solution is $y(x + 2h)$, we get

$$y(x + 2h) = y_1 + (2h)^5 C...$$
$$y(x + 2h) = y_2 + 2(h)^5 C...,$$

where C is a constant within the step. The difference

$$\Delta = y_2 - y_1$$

is an indicator of the truncation error. It is proportional to h^5.

If Δ_0 is the desired accuracy, the step size h_0 should have been

$$h_0 = h \left| \frac{\Delta_0}{\Delta} \right|^{1/5}.$$

If $\Delta > \Delta_0$, the step size must be reduced to obtain the required accuracy.

If $\Delta < \Delta_0$, the step size can be lengthened to fasten calculations.

The fourth order RK requires four evaluations of the derivative function f. In higher order methods the number of derivative evaluations is higher than the order, and they become progressively slower.

Bulirsch–Stoer method

See Stoer, Bulirsch: *Introductio to Numerical Analysis*, Springer 1980.

A brief summary in *Numerical Recipes*.

(1) Use a modified midpoint method to integrate from x to $x + H$ using n substeps of length $h = H/n$:

$$
\begin{aligned}
z_0 &= y(x) \\
z_1 &= z_0 + hf'(x, z_0), \\
&\dots \\
z_{m+1} &= z_{m-1} + 2hf'(x + mh, z_m), \qquad m = 1, 2, ..., n-1 \\
y(x + H) \approx y_n &= \frac{1}{2}[z_n + z_{n-1} + hf'(x + H, z_n)]
\end{aligned}
$$

It can be shown that the errot expressed as a power series of h contains only even powers of h. Combining results of different order to remove error terms gains two orders at a time.

(2) An idea similar to the Romberg integration: Integrate over a relatively long interval using different step sizes h.

(3) Express the result as a rational function of the step size, $R(h)$, and extrapolate to $h = 0$.

A suitable sequence of subdivisions is e.g.

$$n = 2, 4, 6, 8, 12, 16, 24, 32, ..., n_i = 2n_{i-2}$$

Multistep methods

Single step methods do not utilize values computed earlier.

In multistep methods several previous points are used to extrapolate the solution further.

Step size can be longer; thus the methods are faster.

Initial values are usually known at one point only. Therefore, a few points must be first calculated using some single step method, like Runge-Kutta. Another possibility is to iterate the solution at the initial polints until it satisfies the equation.

In single step methods each step is independent from the previous ones. Hence the step size can be easily changed, as necessary. In multistep methods changing the step size is more complicated; a new set of initial points must be calculated.

Milne's method

Assume we know the function at four points, x_{n-3}, x_{n-2}, x_{n-1} and x_n (calculated e.g. by RK or Taylor series).

Write the equation as

$$dy = f(x, y)\, dx$$

and integrate one step further using a second degree polynomial to describe the function in the range x_{n-2} to x_n:

$$\int_{x_{n-3}}^{x_{n+1}} f(x, y)\mathrm{d}x = \int_{x_{n-3}}^{x_{n+1}} P_2(x)\mathrm{d}x,$$

which gives

$$y_{n+1} - y_{n-3} = \frac{4h}{3}(2f_n - f_{n-1} + 2f_{n-2}).$$

This is used to predict f_{n+1}.

An improved value is obtained from

$$\int_{x_{n-1}}^{x_{n+1}} f(x,y)\mathrm{d}x = \int_{x_{n-1}}^{x_{n+1}} P_2(x)\mathrm{d}x,$$

where P_2 is now a polynomial determined at the points x_{n-1}, x_n and x_{n+1}. Thus we get

$$y_{n+1,c} - y_{n-3} = \frac{h}{3}(2f_{n+1} - 4f_n + f_{n-1}).$$

The method is simple and the local error is $\mathcal{O}(h^5)$.

However, the method sometimes suffers from instability, and then reducing the step size does not help.

In general, the method cannot be recommended, but it can be used with appropriate caution.

Adams' method

Write the equation again as

$$dy = f(x, y)\, dx$$

and integrate over one step

$$\int_{x_n}^{x_{n+1}} dy = y_{n+1} - y_n = \int_{x_n}^{x_{n+1}} f(x, y)\, dx.$$

Replace $f(x, y)$ by an interpolation polynomial (now we have to use a polynomial based on backward differences):

$$\begin{aligned}
y_{n+1} - y_n &\approx \int_{x_n}^{x_{n+1}} \left(f_n + s\Delta f_{n-1} + \frac{(s+1)s}{2}\Delta^2 f_{n-2} \right) dx \\
&= \int_{s=0}^{s=1} \left(f_n + s\Delta f_{n-1} + \frac{(s+1)s}{2}\Delta^2 f_{n-2} \right) h\, ds \\
&= h(f_n + \frac{1}{2}\Delta f_{n-1} + \frac{5}{12}\Delta^2 f_{n-2}) \\
&= h\left(f_n + \frac{f_n - f_{n-1}}{2} + \frac{5(f_n - 2f_{n-1} + f_{n-2})}{12} \right) \\
&= \frac{h}{12}(23f_n - 16f_{n-1} + 5f_{n-2}).
\end{aligned}$$

Thus the integration step is

$$y_{n+1} = y_n + \frac{h}{12}(23f_n - 16f_{n-1} + 5f_{n-2}).$$

The local error is $\mathcal{O}h^4$ and global error $\mathcal{O}h^3$.

Let's solve our example equation $y'' + y = 0$ using Adams method:

```
! initial values; we'll cheat and assume
! they are known
! these should be determined using
! some single step method
  g2 = cos(0)
  g1 = cos(0.1)
  g0 = cos(0.2)
  f2 = -sin(0)
  f1 = -sin(0.1)
  f0 = -sin(0.2)
  h = 0.1
  x = 0.2
  y = sin(0.2)
  u = cos(0.2)
```

```
 ! the actual integration loop
 do while (x <=2.0)
   y = y + h/12*(23*g0-16*g1+5*g2)
   u = u + h/12*(23*f0-16*f1+5*f2)

   g2=g1; g1=g0; g0=u
   f2=f1; f1=f0; f0=-y
   x = x+h
   write(*,*) x,y,u
end do
```

```
x           y        y (exact)          u        y'(exact)
0.3000000   0.2955149   0.2955202   0.9552994   0.9553365
0.4000000   0.3893969   0.3894183   0.9209886   0.9210610
0.5000000   0.4793826   0.4794255   0.8774783   0.8775826
0.6000000   0.5645716   0.5646425   0.8252031   0.8253356
0.7000000   0.6441129   0.6442177   0.7646864   0.7648422
0.8000001   0.7172123   0.7173561   0.6965334   0.6967067
0.9000001   0.7831398   0.7833270   0.6214256   0.6216099
1.000000    0.8412372   0.8414711   0.5401140   0.5403022
```

The previous method is one of the more general Adams-Bashforth methods. Some other methods of this type are e.g.

$$
\begin{aligned}
y_{n+1} &= y_n + hf_n, \\
y_{n+1} &= y_n + \frac{h}{2}(3f_n - f_{n-1}), \\
y_{n+1} &= y_n + \frac{h}{12}(23f_n - 16f_{n-1} + 5f_{n-2}), \\
y_{n+1} &= y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}),
\end{aligned}
$$

These are explicit methods.

Implicit methods contain an equation

$$y_{n+1} = y_n + g + \beta h f(x_{n+1}, y_{n+1}),$$

where g depends only on the previous values of x and y. The function f appearing here may be of such a form that an analytical expression for y_{n+1} cannot be obtained, but it has to be solved by some iterative method.

Adams-Moulton methods are implicit methods:

$$
\begin{aligned}
y_{n+1} &= y_n + hf_{n+1}, \\
y_{n+1} &= y_n + \frac{h}{2}(f_{n+1} + f_n), \\
y_{n+1} &= y_n + \frac{h}{12}(5f_{n+1} + 8f_n - f_{n-1}), \\
y_{n+1} &= y_n + \frac{h}{24}(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}),
\end{aligned}
$$

Implicit methods are more stable than explicit methods.

Explicit and implicit methods can be combined to avoid solving an equation.

Use a Adams-Bashforth method as a predictor and a Adams-Moulton method as a corrector. For example

$$y_{n+1} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3})$$
$$y_{n+1} = y_n + \frac{h}{24}(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}).$$

Solve the example equation using an Adams predictor-corrector method

```
do while (x <=1.0)
 ! predictor
 y1 = y + h/12*(23*g0-16*g1+5*g2)
 u1 = u + h/12*(23*f0-16*f1+5*f2)

 g2=g1; g1=g0; g0=u1
 f2=f1; f1=f0; f0=-y1

 ! corrector
 y2 = y +h/12*(5*g0+8*g1-g2)
 u2 = u +h/12*(5*f0+8*f1-f2)

 u = u2 ; y = y2 ;
 x = x+h
 write(*,*) x,y,u
end do
```

x	y	y (exact)	u	y'(exact)
0.3000000	0.2955195	0.2955202	0.9553408	0.9553365
0.4000000	0.3894152	0.3894183	0.9210703	0.9210610
0.5000000	0.4794213	0.4794255	0.8775975	0.8775826
0.6000000	0.5646383	0.5646425	0.8253561	0.8253356
0.7000000	0.6442145	0.6442177	0.7648684	0.7648422
0.8000001	0.7173551	0.7173561	0.6967387	0.6967067
0.9000001	0.7833292	0.7833270	0.6216474	0.6216099
1.000000	0.8414777	0.8414711	0.5403448	0.5403022

Backward multistep methods

In these methods the derivative is evaluated at one point only. This will reduce the increasing effect of f on errors. The methods are very stable, and good for stiff problems.

$$
\begin{aligned}
y_{n+1} &= y_n + hf_{n+1}, \\
y_{n+1} &= \frac{4}{3}y_n - \frac{1}{3}y_{n-1} + \frac{2h}{3}f_{n+1}, \\
y_{n+1} &= \frac{18}{11}y_n - \frac{9}{11}y_{n-1} + \frac{2}{11}y_{n-2} + \frac{6h}{11}f_{n+1}, \\
y_{n+1} &= \frac{48}{25}y_n - \frac{36}{25}y_{n-1} + \frac{16}{25}y_{n-2} - \frac{3}{25}y_{n-3} + \frac{12h}{25}f_{n+1},
\end{aligned}
$$

The first one is just the implicit Euler's method.

Potential problems

Some equations are inherently unstable and difficult to solve, independently of the method.

A method can be stable or unstable depending on the equation and step size.

The step size must be short enough. Too long a step may make the method unstable.

But: a very short step slows down the computation and may lead to accumulation of round-off errors.

A good check is to recalculate the solution using a step that is half of the original. If the values are almost the same, the solution is probably close to the correct one. If the values differ considerably, either the step is too long, or there are some other problems that must be found out.

Potential singularities: For example, the gravitational force increases with diminishing distance and becomes infinite. When objects approach each others the time step must be decreased. (E.g. the orbit of a comet far away from other objects is almost a straight line and motion very slow allowing a long time step, but close to the Sun the time step must be very short.)

Easy to correct in single-step methods. In multi-step methods new initial values must be calculated when the step size is changed.

The problem can be removed (at least partly) by regularization, i.e. suitable transformations of time and geometry.

Differential equations — boundary value problems

The values of the unknown function are known at both ends of the interval. For example, in a stellar model the mass distribution has the boundary values $M(0) = 0$, $M(R) = m$, where $M(r)$ is the mass inside radius r and m is the total mass of the star.

An initial value problem has usually a unique solution, if the functions involved are smooth enough.

A boundary value problem may not have a solution at all, or there may be an infinite number of solutions.

Shooting method

Assume the values of the function to be determined are known at both ends of the interval, but its derivatives are unknown. To use the methods for solving an initial value problem the derivative should be known at the starting point. Try different values of the derivative until the solution passes through the endpoint.

Consider the equation

$$y'' = f(x, y, y'), \quad y(a) = \alpha, \quad y(b) = \beta.$$

Convert this to an initial value problem

$$y'' = f(x, y, y'), \quad y(a) = \alpha, \quad y'(a) = k.$$

The solution depends on the shooting angle k:

$$y = y(x; k).$$

Thus the value of k must be solved from

$$y(b; k) = \beta.$$

In principle this is an equation that can be solved using any numerical method. Evaluation of the function requires now finding one particular solution of the differential equation.

Begin by finding two solutions corresponding to $k = k_0$ and k_1. Then improve the solution e.g. by the secant method:

$$k_2 = k_1 - (y(b; k_1) - \beta)\frac{k_1 - k_0}{y(b; k_1) - y(b; k_0)},$$

Example: Solve the equation $y'' + y = 0$ with the boundary conditions $y(0) = 1$, $y(1) = 1$.
(The exact solution is $0.5463 \sin x + \cos x$.)

We can use e.g. the 4. order Runge-Kutta.

Try first $y'(0) = k_0 = 0$:

```
0.100000 0.995004
0.200000 0.980067
   ...
0.800000 0.696707
0.900000 0.621611
1.000000 0.540303
```

Another trial $y'(0) = k_1 = 1$:

```
0.100000 1.094837
0.200000 1.178736
   ...
0.800000 1.414063
0.900000 1.404937
1.000000 1.381773
```

An improved value of the slope:

$$y'(0) = k_2 = 1 - (1.381773 - 1)\frac{1 - 0}{1.381773 - 0.540303} = 0.5463.$$

Solution with the new initial value $y'(0) = k_2 = 0.5463$:

```
0.100000 1.049543
0.200000 1.088600
   ...
0.800000 1.088599
0.900000 1.049542
1.000000 0.999998
```

The accuracy is no coincidence. The solution of a second order linear boundary value problem is always obtained in this way from two different solutions.

In linear problems a linear combination of solutions is also a solution satisfying the equation. Thus it is possible the find the actual solution by combining several trial solutions.

Not true for nonlinear problems, which usually require several iterations.

Try to solve the nonlinear equation

$$yy' = -e^{-2x}$$

with the boundary conditions $y(0) = 1$, $y(1) = 1/e$. The actual solution is $y(x) = e^{-x}$ and thus we should have $y'(0) = -1$

In this case $h = 0.1$ and the implicit Euler method gave

```
 h      x      y
-1.500  1.000  0.353
-1.400  1.000  0.365
-1.300  1.000  0.377
-1.200  1.000  0.388
-1.100  1.000  0.399
-1.000  1.000  0.410
-0.900  1.000  0.421
-0.800  1.000  0.431
-0.700  1.000  0.442
-0.600  1.000  0.452
-0.500  1.000  0.462
-0.400  1.000  0.472
```

Only when the step was reduced to $h = 0.001$ we get close to the real solution. However, now the final value is not very sensitive to the derivative:

```
-1.500  1.000  0.368
-1.400  1.000  0.368
-1.300  1.000  0.368
-1.200  1.000  0.368
-1.100  1.000  0.368
-1.000  1.000  0.368
-0.900  1.000  0.369
-0.800  1.000  0.369
-0.700  1.000  0.369
-0.600  1.000  0.369
-0.500  1.000  0.369
-0.400  1.000  0.369
-0.300  1.000  0.369
-0.200  1.000  0.370
-0.100  1.000  0.370
 0.000  1.000  0.370
```

Difference method

Replace derivatives by finite differences:

$$y'(x_i) \approx \frac{y_{i+1} - y_{i-1}}{2h},$$
$$y''(x_i) \approx \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}.$$

Example: equation $y'' + y = 0$:

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + y_i = 0$$

or

$$y_{i-1} + (h^2 - 2)y_i + y_{i+1} = 0.$$

In addition, boundary values give

$$y_0 = 1, \qquad y_n = 1.$$

If the step is $h = 0.25$, we obtain equations

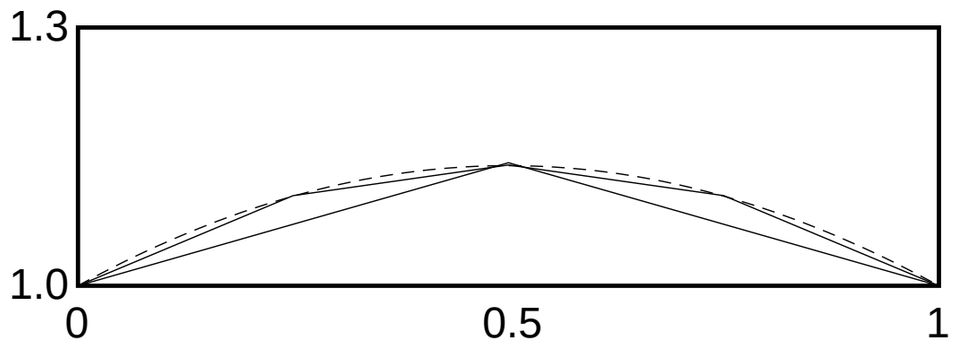
$$\begin{aligned} y_0 &= 1, \\ y_0 - 1.9375y_1 + y_2 &= 0, \\ y_1 - 1.9375y_2 + y_3 &= 0, \\ y_2 - 1.9375y_3 + y_4 &= 0, \\ y_4 &= 1 \end{aligned}$$

or

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & -1.9375 & 1 & 0 & 0 \\ 0 & 1 & -1.9375 & 1 & 0 \\ 0 & 0 & 1 & -1.9375 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

The solution of the set of equations is

```
 x      y      y(exact)
0.0000 1.0000   1.0000
0.2500 1.1047   1.1041
0.5000 1.1403   1.1395
0.7500 1.1047   1.1041
1.0000 1.0000   1.0000
```



A linear differential equation is replaced by a set of linear equations; the coefficient matris is a band matrix, like the tridiagonal matrix above. Space can be saved by storing the nonzero elements only.

Usually a normal Gaussian elimination can be used. In a tridiagonal system each variable is eliminated from two equations; thus the time is $\propto n$.

If higher order differences are used the matrix will still be a band matrix.

If the step is short, the number of equations can be very big, possibly resulting in accumulation of rounding errors.

If the matrix is diagonally dominant, iterative methods (Gauss–Seidel or Gauss–Jordan) can be faster and more stable than the elimination method.

Accuracy can be improved by

- Using shorter steps

- Using higher order approximations for derivatives; this means bigger bandwidth of the matrix. Also, the equations will contain points outside the original interval.

If the original equation is nonlinear, also the resulting set of equations will be nonlinear.
It may be possible to solve them by iteration.

Finding a good initial value leading to a convergent solution may be difficult.

If a converging solution is difficult to fkind, the shooting method should be used.

Boundary values may also include derivatives.

Example: equation $y'' + y = 0$, boundary values $y'(0) = 0$, $y'(1) = 0.5$. (The exact solution is $y = -0.5942 \cos x$, and $y(1) = -0.3210$.)

If the step is $h = 0.25$, we get the equations

$$\begin{aligned} y_{-1} - 1.9375y_0 + y_1 &= 0, \\ y_0 - 1.9375y_1 + y_2 &= 0, \\ y_1 - 1.9375y_2 + y_3 &= 0, \\ y_2 - 1.9375y_3 + y_4 &= 0, \\ y_3 - 1.9375y_4 + y_5 &= 0, \end{aligned}$$

y_{-1} and y_5 are outside the interval.

Boundary values give also the equations

$$\begin{aligned} y'(0) &= \frac{y_1 - y_{-1}}{0.5} = 0, \\ y'(1) &= \frac{y_5 - y_3}{0.5} = 0.5, \end{aligned}$$

$$\begin{pmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1.9375 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1.9375 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1.9375 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1.9375 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1.9375 & 1 \\ 0 & 0 & 0 & 0 & -2 & 0 & 2 \end{pmatrix} \begin{pmatrix} y_{-1} \\ y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.5 \end{pmatrix}.$$

The solution is

$$y = \begin{pmatrix} -0.5792 \\ -0.5979 \\ -0.5792 \\ -0.5243 \\ -0.4367 \\ -0.3217 \\ -0.1867 \end{pmatrix}.$$

Galerkin's method

The solution can also be sought in terms of some basis functions ϕ_i, $i = 1, ..., N$:

$$y(x) = \sum_{i=1}^{N} c_i \phi_i(x).$$

If the equation is of the form

$$y'' = f(x, y, y')$$

it will become

$$\sum_{i=1}^{N} c_i \phi_i''(x_j) = f\left(x_j, \sum_{i=1}^{N} c_i \phi_i(x_j), \sum_{i=1}^{N} c_i \phi_i'(x_j)\right), j = 1, ..., N$$

and the boundary values give

$$y(a) = \sum_{i=1}^{N} c_i \phi_i(a) = \alpha$$
$$y(b) = \sum_{i=1}^{N} c_i \phi_i(b) = \beta$$

These equations are used to determine the coefficients c_i.

The basis functions can be anything, but they should be chosen in a suitable way for the problem.

Differential equations — characteristic value problems

Problems concerning vibrations of strings or membranes as well as solving the Schrödinger equation are a special form of boundary value problems.

Consider the equation

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

where k is a constant.

The general solution is of the form

$$y = a \sin kx + b \cos kx,$$

where a and b are the two constants needed for the solution of the second order equation.

Since $y(0) = 0$, the initial value requires that $b = 0$.

At $x = 1$ we must have $y = a \sin kx = 0$.

This is satisfied if $a = 0$. A homogeneous equation like this always has the *trivial solution* $y = 0$, which is not very interesting. A nontrivial solution requires that $\sin k = 0$, Thus the solution is

$$y = a \sin n\pi x, \quad n = 1, 2, ...$$

The constant a can have any value whatsoever without violating the boundary conditions.

The equation has a solution only for certain characteristic values or eigenvalues of the parameter k. An eigenfunction of the equation is a solution corresponding to a certain eigenvalue.

We can discretize the equation as

$$\frac{1}{h^2}(y_{i+} - 2y_i + y_{i-1}) + k^2 y_i = 0.$$

If we take $h = 0.3$, we have for the interior points

$$\begin{aligned}
y_1 - (2 - 0.04k^2)y_2 + y3 &= 0 \\
y_2 - (2 - 0.04k^2)y_3 + y4 &= 0 \\
y_3 - (2 - 0.04k^2)y_4 + y5 &= 0 \\
y_4 - (2 - 0.04k^2)y_5 + y6 &= 0
\end{aligned}$$

The boundary conditions require $y_1 = y_6 = 0$. Thus we get

$$\begin{pmatrix}
2 - 0.04k^2 & -1 & 0 & 0 \\
-1 & 2 - 0.04k^2 & -1 & 0 \\
0 & -1 & 2 - 0.04k^2 & -1 \\
0 & 0 & -1 & 2 - 0.04k^2
\end{pmatrix}
\begin{pmatrix} y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix} = 0.$$

This set of equations has a nontrivial solution only if the determinant of the coefficient matrix is zero.

This means that we have to find the eigenvalues of the matrix

$$\begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}$$

If you have a QR program of linear algebra you can use it to find the eigenvalues.

Also, Matlab, Octave etc. have functions that evaluate the eigenvalues.

In the example we get

```
octave:4> l=eig(A)
l =

  0.38197
  1.38197
  2.61803
  3.61803

k=sqrt(l/0.04)
k =

  3.09
  5.88
  8.09
  9.51
```

This is not very accurate, since the values should be multiples of π. To improve the accuracy, a shorter step length h is needed