

Numerical Solution to Ordinary Differential Equations

By Gilberto E. Urroz, September 2004

In this document I present some notes related to finite difference approximations and the numerical solution of single and systems of ordinary differential equations (ODEs).

Finite differences and numerical solutions

To solve differential equations numerically we can replace the derivatives in the equation with finite difference approximations on a discretized domain. This results in a number of algebraic equations that can be solved one at a time (explicit methods) or simultaneously (implicit methods) to obtain values of the dependent function y_i corresponding to values of the independent function x_i in the discretized domain.

Finite differences

A finite difference is a technique by which derivatives of functions are approximated by differences in the values of the function between a given value of the independent variable, say x_0 , and a small increment (x_0+h) . For example, from the definition of derivative,

$$df/dx = \lim_{h \rightarrow 0} (f(x+h)-f(x))/h,$$

we can approximate the value of df/dx by using the finite difference approximation

$$(f(x+h)-f(x))/h$$

with a small value of h .

The following table shows approximations to the derivative of the function

$$f(x) = \exp(-x) \sin(x^2/2),$$

at $x = 2$, using finite differences. The actual value of the derivative is -0.23569874791. The third column in the table shows the error in evaluating the derivative, i.e., the difference between the numerical derivative $\Delta f/\Delta x$ and the actual value.

h	$\Delta f/\Delta x$	error
0.1	-0.244160077	0.00846132909
0.01	-0.236684829	0.00098608109
0.001	-0.235798686	0.00009993809
0.0001	-0.235708734	0.00000998609
0.00001	-0.235699726	0.00000097809
0.000001	-0.235698825	0.00000007709
0.0000001	-0.235698734	0.00000001391
0.00000001	-0.235698724	0.00000002391
0.000000001	-0.235698752	0.00000000409

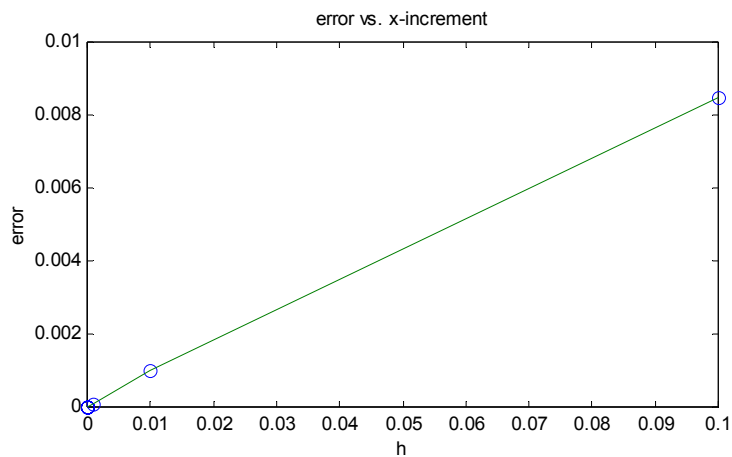
This exercise illustrates the fact that, as $h \rightarrow 0$, the value of the finite difference approximation, $(f(x+h)-f(x))/h$, approaches that of the derivative, df/dx , at the point of interest.

A plot of the error as a function of h also reveals the fact that the error is proportional to the value of the x -increment h . The following plots, using different ranges of h and the error, are produced with MATLAB out of the data in the table.

```

» h = [1e-1,1e-2,1e-3,1e-4,1e-5,1e-6,1e-7,1e-8,1e-9];
» er = [0.00846132909,0.00098608109,0.00009993809,0.00000998609,...
        0.00000097809,0.00000007709,0.00000001391,0.00000002391,...
        0.00000000409];
» plot(h,er,'o',h,er,'-');axis([0 0.1 0 0.01]);
» title('error vs. x-increment');xlabel('h');ylabel('error');

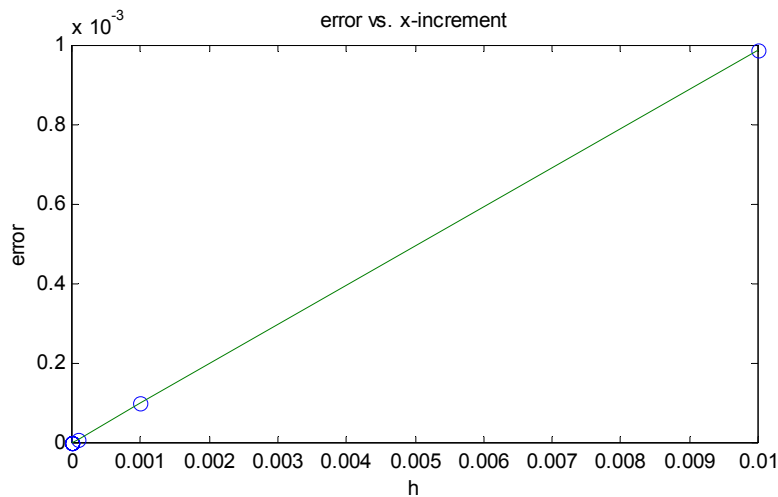
```



```

» axis([0 0.01 0 0.01]);

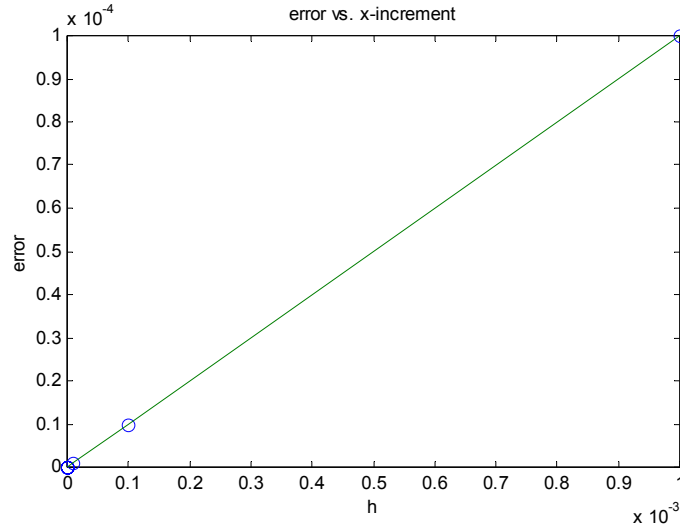
```



```

» axis([0 0.001 0 0.0001]);

```



The graphs seem indicate that the error varies linearly with the increment h in the independent variable. It is very common to indicate this dependency by saying that "the error is of order h ", or $error = O(h)$. The magnitude of the error can be estimated by using Taylor series expansions of the function $f(x+h)$.

Finite difference formulas based on Taylor series expansions

The Taylor series expansion of the function $f(x)$ about the point $x = x_0$ is given by the formula

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!} \cdot (x - x_0)^n.$$

Where $f^{(n)}(x_0) = (d^n f/dx^n)|_{x=x_0}$, and $f^{(0)}(x_0) = f(x_0)$.

If we let $x = x_0 + h$, then $x - x_0 = h$, and the series can be written as

$$f(x_0 + h) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!} \cdot h^n = f(x_0) + \frac{f'(x_0)}{1!} \cdot h + \frac{f''(x_0)}{2!} \cdot h^2 + O(h^3),$$

Where the expression $O(h^3)$ represents the remaining terms of the series and indicates that the leading term is of order h^3 . Because h is a small quantity, we can write $1 > h$, and $h > h^2 > h^3 > h^4 > \dots$. Therefore, the remaining of the series represented by $O(h^3)$ provides the order of the error incurred in neglecting this part of the series expansion when calculating $f(x_0+h)$.

From the Taylor series expansion shown above we can obtain an expression for the derivative $f'(x_0)$ as

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} + \frac{f''(x_0)}{2!} \cdot h + O(h^2) = \frac{f(x_0 + h) - f(x_0)}{h} + O(h).$$

In practical applications of finite differences, we will replace the first-order derivative df/dx at $x = x_0$, with the expression $(f(x_0+h)-f(x_0))/h$, selecting an appropriate value for h , and indicating that the error introduced in the calculation is of order h , i.e., $error = O(h)$.

Forward, backward and centered finite difference approximations to the first derivative

The approximation

$$df/dx = (f(x_0+h)-f(x_0))/h$$

is called a forward difference formula because the derivative is based on the value $x = x_0$ and it involves the function $f(x)$ evaluated at $x = x_0+h$, i.e., at a point located forward from x_0 by an increment h .

If we include the values of $f(x)$ at $x = x_0 - h$, and $x = x_0$, the approximation is written as

$$df/dx = (f(x_0)-f(x_0-h))/h$$

and is called a backward difference formula. The order of the error is still $O(h)$.

A centered difference formula for df/dx will include the points $(x_0-h, f(x_0-h))$ and $(x_0+h, f(x_0+h))$. To find the expression for the formula as well as the order of the error we use the Taylor series expansion of $f(x)$ once more. First we write the equation corresponding to a forward expansion:

$$f(x_0+h) = f(x_0) + f'(x_0) \cdot h + 1/2 \cdot f''(x_0) \cdot h^2 + 1/6 \cdot f^{(3)}(x_0) \cdot h^3 + O(h^4).$$

Next, we write the equation for a backward expansion:

$$f(x_0-h) = f(x_0) - f'(x_0) \cdot h + 1/2 \cdot f''(x_0) \cdot h^2 - 1/6 \cdot f^{(3)}(x_0) \cdot h^3 + O(h^4).$$

Subtracting these two equations results in

$$f(x_0+h) - f(x_0-h) = 2f'(x_0) \cdot h + 1/3 \cdot f^{(3)}(x_0) \cdot h^3 + O(h^5).$$

Notice that the even terms in h , i.e., h^2, h^4, \dots , vanish. Therefore, the order of the remaining terms in this last expression is $O(h^5)$. Solving for $f'(x_0)$ from the last result produces the following centered difference formula for the first derivative:

$$\left. \frac{df}{dx} \right|_{x=x_0} = \frac{f(x_0+h) - f(x_0-h)}{2 \cdot h} + \frac{1}{3} \cdot f^{(3)}(x) \cdot h^2 + O(h^4),$$

or,

$$\frac{df}{dx} = \frac{f(x_0 + h) - f(x_0 - h)}{2 \cdot h} + O(h^2).$$

This result indicates that the centered difference formula has an error of the order $O(h^2)$, while the forward and backward difference formulas had an error of the order $O(h)$. Since $h^2 < h$, the error introduced in using the centered difference formula to approximate a first derivative will be smaller than if the forward or backward difference formulas are used.

Forward, backward and centered finite difference approximations to the second derivative

To obtain a centered finite difference formula for the second derivative, we'll start by using the equations for the forward and backward Taylor series expansions from the previous section but including terms up to $O(h^5)$, i.e.,

$$f(x_0+h) = f(x_0) + f'(x_0) \cdot h + 1/2 f''(x_0) \cdot h^2 + 1/6 f^{(3)}(x_0) \cdot h^3 + 1/24 f^{(4)}(x_0) \cdot h^4 + O(h^5).$$

and

$$f(x_0-h) = f(x_0) - f'(x_0) \cdot h + 1/2 f''(x_0) \cdot h^2 - 1/6 f^{(3)}(x_0) \cdot h^3 + 1/24 f^{(4)}(x_0) \cdot h^4 - O(h^4).$$

Next, add the two equations and solve for $f''(x_0)$:

$$d^2f/dx^2 = [f(x_0+h) - 2f(x_0) + f(x_0-h)]/h^2 + O(h^2).$$

Forward and backward finite difference formulas for the second derivatives are given, respectively, by

$$d^2f/dx^2 = [f(x_0+2 \cdot h) - 2f(x_0+h) + f(x_0)]/h^2 + O(h),$$

and

$$d^2f/dx^2 = [f(x_0) - 2f(x_0-h) + f(x_0-2 \cdot h)]/h^2 + O(h).$$

Solution of a first-order ODE using finite differences - Euler forward method

Consider the ordinary differential equation,

$$dy/dx = g(x,y),$$

subject to the boundary condition,

$$y(x_1) = y_1.$$

To solve this differential equation numerically, we need to use one of the formulas for finite differences presented earlier. Suppose that we use the forward difference approximation for dy/dx ; i.e.,

$$dy/dx = (y(x+h)-y(x))/h.$$

Then, the differential equation is transformed into the following difference equation:

$$(y(x+h)-y(x))/h = g(x,y),$$

from which,

$$y(x+h) = y(x)+h \cdot g(x,y).$$

This result is known as Euler's forward method for numerical solution of first-order ODEs.

Since we know the boundary condition (x_1, y_1) we can start by solving for y at $x_2 = x_1+h$, then we solve for y at $x_3 = x_2+h$, and so on. In this way, we generate a series of points $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, which will represent the numerical solution to the original ODE. The upper limit of the independent variable x_n is either given or selected arbitrarily during the solution.

The term "*discretizing the domain of the independent variable*" refers to obtaining a series of values of the independent variable, namely, $x_i, i = 1, 2, \dots, n$, that will be used in the solution. Suppose that the range of the independent variable (a, b) is known, and that we use a constant value $h = \Delta x$ to divide the range into n equal intervals. By making $x_1 = a$, and $x_n = b$, then we find that the values of $x_i, i = 2, 3, \dots, n$, are given by

$$x_i = x_1 + (i-1) \cdot \Delta x = a + (i-1) \cdot \Delta x,$$

and that for $i = n, x_n = x_1 + (n-1) \cdot \Delta x$. This latter result can be used to find n given Δx ,

$$n = (x_n - x_1) / \Delta x + 1 = (b-a) / \Delta x + 1,$$

or, to find Δx given n ,

$$\Delta x = (x_n - x_1) / (n-1) = (b-a) / (n-1).$$

The recurrent equation for solving for y is given by

$$y_{i+1} = y_i + \Delta x \cdot g(x_i, y_i),$$

for $i = 1, 2, \dots, n-1$. Because the method solves $y_{i+1} = f(x_i, y_i, \Delta x)$, i.e., one value of the dependent variable at a time, the method is said to be an explicit method.

The following example illustrates the application of the Euler first-order method to the solution of the differential equation $dy/dx = g(x,y) = x + y$ using MATLAB. First, we define function $g(x,y)$:

```

>> g = inline('x+y','x','y')

g =

    Inline function:
    g(x,y) = x+y

```

We solve the equation in the range of values of x from $x_0 = 0$ to $x_n = 2.0$ with an increment $Dx = 0.1$. The initial condition is $y_0 = 1.0$ for $x_0 = 0$:

```

>>x0 = 0; y0 = 1; Dx = 0.1; xn = 2.0;

```

The following commands generate a vector of values of x , a vector y of the same length of x , initialized with zeros, and determines the value of n as the length of vector y (or x):

```

>> x = [x0:Dx:xn]; n = length(x); y = zeros(1,n);

```

The following *for...end* loop takes care of calculating the values of y_i for $i = 2, 3, \dots, n$:

```

>> for j = 1:n-1
    y(j+1) = y(j) + Dx*g(x(j),y(j));
end;

```

To produce a plot of the results we determine the minimum and maximum values of y :

```

>> ymin = min(min(y)), ymax = max(max(y))

```

```

ymin = 0

```

```

ymax = 3.7275

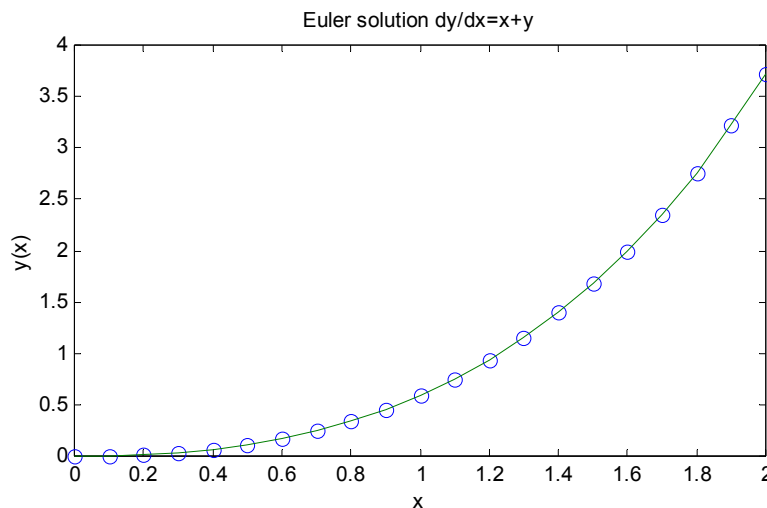
```

The plot is generated by using:

```

>> plot(x,y,'o',x,y,'-');title('Euler solution dy/dx=x+y');
>> xlabel('x');ylabel('y(x)');

```



A function to implement Euler's first-order method

The following function, *Euler1*, implements the calculation steps outlined in the previous example. The function detects if there is overflowing introduced in the solution and stops the calculation at that point providing the current results.

```
function [x,y] = Euler1(x0,y0,xn,Dx,g)
% Euler 1st order method solving ODE
% dy/dx = g(x,y), with initial
% conditions y=y0 at x = x0. The
% solution is obtained for x = [x0:Dx:xn]
% and returned in y
ymaxAllowed = 1e+100;
x = [x0:Dx:xn]; y = zeros(x); n = length(y); y(1) = y0;
for j = 1:n-1
    y(j+1) = y(j) + Dx*g(x(j),y(j));
    if y(j+1) > ymaxAllowed
        disp('Euler 1 - WARNING: underflow or overflow');
        disp('Solution sought in the following range:');
        disp([x0 Dx xn]);
        disp('Solution evaluated in the following range:');
        disp([x0 Dx x(j)]);
        n = j; x = x(1,1:n); y = y(1,1:n);
        break;
    end;
end;
end;
```

Next, we use function *Euler1* to solve the differential equation from the previous example, namely, $dy/dx = g(x,y) = x+y$, for different values of the x increment, $\Delta x = 0.5, 0.2, 0.1,$ and 0.05 , with the same initial conditions and range of values of x as before:

```
» [x1,y1] = Euler1(0,1,2,0.5,g);
» [x2,y2] = Euler1(0,1,2,0.2,g);
» [x3,y3] = Euler1(0,1,2,0.1,g);
» [x4,y4] = Euler1(0,1,2,0.05,g);
```

The exact solution for this equation is $y(x) = -x - 1 + 2e^x$. Set of values of the exact solution are calculated as follows:

```
» xx = [0:0.1:2]; yy = -xx-1+2.*exp(xx);
```

To plot the exact and numerical solutions we first determine the minimum and maximum values of y:

```
-->ymax = max([y1 y2 y3 y4 yy])
ymax = 11.778112

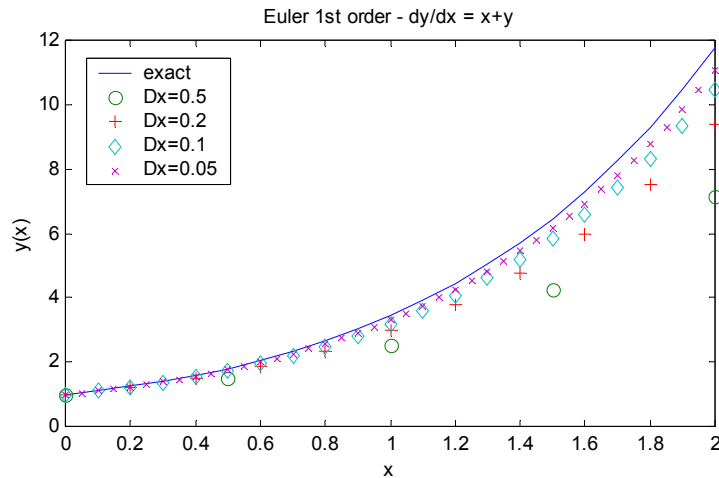
-->ymin = min([y1 y2 y3 y4 yy])
ymin = 1.
```

The plot of the solutions is produced through the use of the following calls to function *plot*:

```

» plot(xx,yy,'-',x1,y1,'o',x2,y2,'+',x3,y3,'d',x4,y4,'x');
» title('Euler 1st order - dy/dx = x+y');xlabel('x');ylabel('y(x)');
» legend('exact','Dx=0.5','Dx=0.2','Dx=0.1','Dx=0.05');

```



A second example of application of function *Euler1* is shown next for the differential equation $dy/dx = xy + 1$, with initial condition $x_0 = 0, y_0 = 1$, in the range $0 < x < 2$, with $\Delta x = 0.5, 0.2, 0.1, 0.05$, and 0.01 . The MATLAB commands used are exactly the same as before except for the definition of function $g(x,y)$ and the title of the plot. The function $g(x,y) = xy+1$ is defined as:

```

» g = inline('x*y+1','x','y')

```

g =

```

Inline function:
g(x,y) = x*y+1

```

Numerical solutions to the differential equation for the different values of Δx are obtained from:

```

» [x1,y1] = Euler1(0,1,2,0.5,g);
» [x2,y2] = Euler1(0,1,2,0.2,g);
» [x3,y3] = Euler1(0,1,2,0.1,g);
» [x4,y4] = Euler1(0,1,2,0.05,g);

```

Next, we determine the minimum and maximum values of y :

```

» ymin = min(min([y1 y2 y3 y4 yy]))

```

ymin = 1

```

EDU» ymax = max(max([y1 y2 y3 y4 yy]))

```

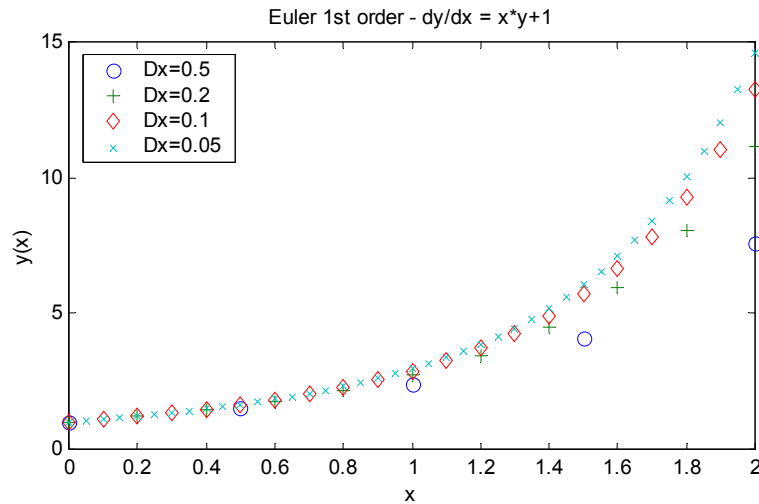
ymax = 14.5874

The plot of the numerical solution is accomplished through:

```

» plot(x1,y1,'o',x2,y2,'+',x3,y3,'d',x4,y4,'x');
» legend('Dx=0.5','Dx=0.2','Dx=0.1','Dx=0.05');
» title('Euler 1st order - dy/dx = x*y+1');xlabel('x');ylabel('y(x)');

```



The following example solves the differential equation $dy/dx = g(x,y) = x + \sin(xy)$ in the interval $0 < x < 6.5$, with initial conditions $x_0 = 0, y_0 = 1$, for $\Delta x = 0.5, 0.2, 0.1, 0.05$, and 0.01 . The steps are the same as in the two previous example:

```

» g = inline('x+sin(x*y)', 'x', 'y')
g =
    Inline function:
    g(x,y) = x+sin(x*y)

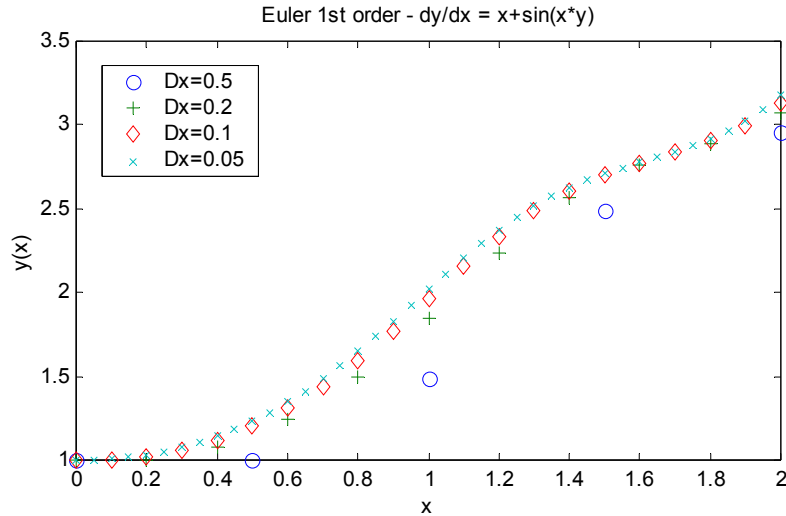
» [x1,y1] = Euler1(0,1,2,0.5,g);
» [x2,y2] = Euler1(0,1,2,0.2,g);
» [x3,y3] = Euler1(0,1,2,0.1,g);
» [x4,y4] = Euler1(0,1,2,0.05,g);

» ymin = min(min([y1 y2 y3 y4 yy]));
ymin = 1

» ymax = max(max([y1 y2 y3 y4 yy]));
ymax = 11.7781

» plot(xx,yy,'-',x1,y1,'o',x2,y2,'+',x3,y3,'d',x4,y4,'x');
» legend('exact','Dx=0.5','Dx=0.2','Dx=0.1','Dx=0.05');
» title('Euler 1st order - dy/dx = x*y+1');xlabel('x');ylabel('y(x)');

```



Finite difference formulas using indexed variables

In the presentation of the Euler forward method, above, we demonstrated how you can get, from the general formula for the first derivative,

$$dy/dx = [y(x+h)-y(x)]/h,$$

the recurrence formula for the explicit solution, namely,

$$y_{i+1} = y_i + \Delta x \cdot g(x_i, y_i),$$

for $i = 1, 2, \dots, n-1$. This suggests re-writing the formula for the derivative as,

$$dy/dx = (y_{i+1}-y_i)/\Delta x + O(\Delta x).$$

Using this sub-index notation, we can summarize the forward, centered, and backward approximations for the first and second derivatives as shown below:

First Derivative

FORWARD: $dy/dx = (y_{i+1}-y_i)/\Delta x + O(\Delta x).$

CENTERED: $dy/dx = (y_{i+1}-y_{i-1})/(2 \cdot \Delta x) + O(\Delta x^2).$

BACKWARD: $dy/dx = (y_i - y_{i-1})/\Delta x + O(\Delta x).$

Second Derivative

FORWARD:
$$d^2y/dx^2 = (y_{i+2} - 2y_{i+1} + y_i)/(\Delta x^2) + O(\Delta x).$$

CENTERED:
$$d^2y/dx^2 = (y_{i+1} - 2y_i + y_{i-1})/(\Delta x^2) + O(\Delta x^2).$$

BACKWARD:
$$d^2y/dx^2 = (y_i - 2y_{i-1} + y_{i-2})/(\Delta x^2) + O(\Delta x).$$

Solution of a first-order ODE using finite differences - an implicit method

Consider again the ordinary differential equation, $dy/dx = g(x,y)$, subject to the boundary condition, $y(x_1) = y_1$. This time, however, we use the centered difference approximation for dy/dx , i.e.

$$dy/dx = (y(x+h) - y(x-h))/(2 \cdot h).$$

With this approximation the ODE becomes,

$$(y(x+h) - y(x-h))/(2 \cdot h) = g(x,y).$$

In terms of sub-indexed variables, this latter equation can be written as:

$$y_{i-1} + 2 \cdot \Delta x \cdot g(x_i, y_i) - y_{i+1} = 0, \quad (i = 2, 3, \dots, n-1)$$

where the substitutions $y(x) = y_i$, $y(x+h) = y_{i+1}$, $y(x-h) = y_{i-1}$, and $h = \Delta x$, have been used.

If the function $g(x,y)$ is linear in y , then the equations described above consist of a set of $(n-2)$ equations. For example, if $n = 5$, we have 3 equations:

$$\begin{aligned} y_1 + 2 \cdot \Delta x \cdot g(x_2, y_2) - y_3 &= 0 \\ y_2 + 2 \cdot \Delta x \cdot g(x_3, y_3) - y_4 &= 0 \\ y_3 + 2 \cdot \Delta x \cdot g(x_4, y_4) - y_5 &= 0 \end{aligned}$$

Since y_1 is known (it is the initial condition), there are still 4 unknowns, y_2 , y_3 , y_4 , and y_5 . We need to find a fourth equation to obtain a solution. We could use, for example, the forward difference equation applied to $i = 1$, i.e.,

$$(y_2 - y_1)/\Delta x = g(x_1, y_1),$$

or

$$y_2 - \Delta x \cdot g(x_1, y_1) - y_1 = 0.$$

The values of x_i , and n (or Δx), can be obtained as in the Euler forward (explicit) solution.

Example 1 -- Solve the ODE

$$dy/dx = y \sin(x),$$

with initial conditions $y(0) = 1$, in the interval $0 < x < 5$. Use $\Delta x = 0.5$, or $n = (5-0)/0.5 + 1 = 11$.

Exact solution: the exact is $y(x) = \exp(-\cos(x))/(\cosh(1)-\sinh(1))$.

Numerical solution: Using a centered difference formula for dy/dx , i.e.,

$$dy/dx = (y_{i+1}-y_{i-1})/(2 \cdot \Delta x),$$

into the ODE, we get $(y_{i+1}-y_{i-1})/(2 \cdot \Delta x) = y_i \sin(x_i)$, which results in the $(n-2)$ implicit equations:

$$y_{i-1} + 2 \cdot \Delta x \cdot \sin(x_i) y_i - y_{i+1} = 0, \quad (i = 2, 3, \dots, n-1).$$

We already know that

$$y_1 = 1$$

(initial condition), thus we have $(n-1)$ unknowns left. We still need to come up with an additional equation, which could be obtained by using a forward difference formula for $i = 1$, i.e.,

$$dy/dx|_{x=1} = (y_2 - y_1)/\Delta x = -y_1 \sin(x_1),$$

or

$$(1 + \Delta x \sin(x_1)) y_1 - y_2 = 0.$$

These equations can be written in the form of a matrix equation, for example, for $n = 5$:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 + \Delta x \cdot \sin(x_1) & -1 & 0 & 0 & 0 \\ 1 & 2 \cdot \Delta x \cdot \sin(x_2) & -1 & 0 & 0 \\ 0 & 1 & 2 \cdot \Delta x \cdot \sin(x_3) & -1 & 0 \\ 0 & 0 & 1 & 2 \cdot \Delta x \cdot \sin(x_4) & -1 \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} y_0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where y_0 represents the initial condition for y . [**Note:** The data requires $n = 11$. The example for $n = 5$ is presented above to provide a sense of the algorithm to fill out the matrix of data]. The matrixial equation can be written as $\mathbf{A} \cdot \mathbf{y} = \mathbf{b}$. Matrix \mathbf{A} and column vector \mathbf{b} can be defined using MATLAB, as indicated below, and the solution found by using left-division. First, we enter the basic data for the problem:

```
» x0 = 0; xn = 5; Dx = 0.5; y0 = 1; x = [x0:Dx:xn]; n = (xn-x0)/Dx+1
```

```
    n = 11.
```

Next, we fill the main diagonal, and the two diagonals below the main diagonal in matrix A using:

```
» A = zeros(n,n); A(1,1) = 1; for j = 2:n, A(j,j) = -1; end; % main diagonal
» A(2,1) = 1+Dx*sin(x(1)); for j = 3:n, A(j,j-1) = 2*Dx*sin(x(j-1)); end;
» for j = 3:n, A(j,j-2) = 1; end; % third diagonal
```

The right-hand side vector is defined as:

```
>> b = zeros(n,1); b(1) = 1; % Right-hand side vector
```

The implicit solution is obtained from:

```
>> y = A\b; % Solving for y
```

To compare the implicit solution we calculate also the explicit solution obtained through the Euler first-order solution:

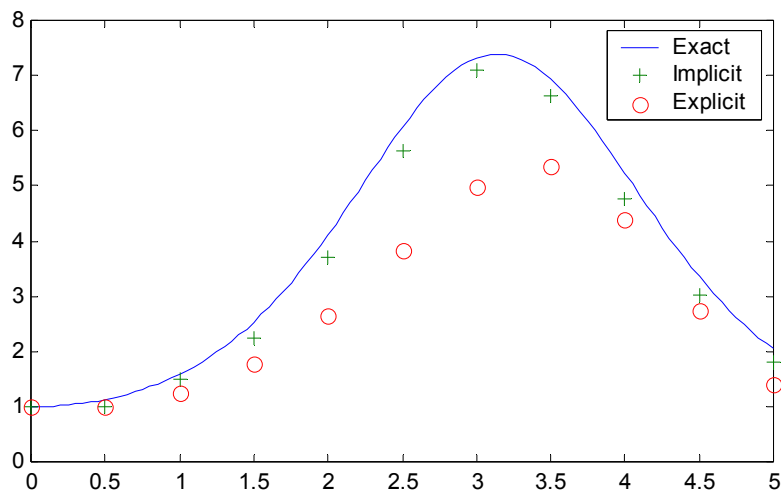
```
» ff = inline('y*sin(x)', 'x', 'y');
» [xx,yy] = Euler1(x0,y0,xn,Dx,ff);
```

To produce data reproducing the exact solution we use:

```
» fE = inline('exp(-cos(x))/(cosh(1)-sinh(1))');
» xE = [0:0.05:5]; yE = fE(xE);
```

The following commands will generate the plot showing the exact, implicit, and explicit solution in the same set of axes:

```
» plot(xE,yE,'-',x',y,'+',xx',yy','o');
» legend('Exact','Implicit','Explicit');
```



Explicit versus implicit methods

The idea behind the *explicit method* is to be able to obtain values such as

$$y_{i+1} = f(x_i, y_i), y_{i+2} = f(x_i, x_{i+1}, y_i, y_{i+1}), \text{ etc.}$$

In other words, your solution proceeds by solving explicitly for a new unknown value in the solution array, given all previous values in the array. On the other hand, *implicit methods* imply the simultaneous solution of n linear algebraic equations that provide, at once, the elements of the solution array. With this distinction in mind between explicit and implicit methods, we outline explicit and implicit solutions for second-order, linear ODEs.

Outline of explicit solution for a second-order ODE

For example, to solve the ODE

$$d^2y/dx^2 + y = 0,$$

in the x -interval $(0, 20)$ subject to $y(0) = 1$, $dy/dx = 1$ at $y = 0$. Use $\Delta x = 0.1$.

First, we discretize the differential equation using the finite difference approximation

$$d^2y/dx^2 = (y_{i+2} - 2y_{i+1} + y_i) / (\Delta x^2),$$

which results in

$$(y_{i+2} - 2y_{i+1} + y_i) / (\Delta x^2) + y_i = 0.$$

An explicit solution can be obtained from the recurrence equation:

$$y_{i+2} = 2y_{i+1} - (1 + \Delta x^2)y_i, \quad i = 1, 2, \dots, n-2;$$

This equation is based on the two previous values of y_i , therefore, to get started we need the values $y = y_1$, and $y = y_2$. The value y_1 is provided in the initial condition, $y(0) = 1$, i.e.,

$$y_1 = 1.$$

The value of y_2 can be obtained from the second initial condition, $dy/dx = 1$, by replacing the derivative with the finite difference approximation:

$$dy/dx = (y_2 - y_1) / \Delta x,$$

which results in

$$(y_2 - y_1) / \Delta x = 1,$$

or

$$y_2 = y_1 + \Delta x.$$

The x-domain is discretized in a similar fashion as in the previous examples for first derivatives, i.e., by making $x_1 = a$, and $x_n = b$, and computing the values of x_i , $i = 2, 3, \dots, n$, with

$$x_i = x_1 + (i-1) \cdot \Delta x = a + (i-1) \cdot \Delta x,$$

where,

$$n = (x_n - x_1) / \Delta x + 1 = (b - a) / \Delta x + 1.$$

The implementation of the solution for this example is left as an exercise for the reader.

Outline of the implicit solution for a second-order ODE

We use the same problem from the previous section: solve the ODE

$$d^2y/dx^2 + y = 0,$$

in the x-interval (0,20) subject to $y(0) = 1$, $dy/dx = 1$ at $x = 0$. Use $\Delta x = 0.1$.

We discretize the differential equation using the finite difference approximation

$$d^2y/dx^2 = (y_{i+2} - 2y_{i+1} + y_i) / (\Delta x^2),$$

which results in

$$(y_{i+2} - 2y_{i+1} + y_i) / (\Delta x^2) + y_i = 0.$$

From this result we get the following implicit equations:

$$y_{i-1} - (2 - \Delta x^2) \cdot y_i + y_{i+1} = 0,$$

for $i = 2, 3, \dots, n-1$. There are a total of $(n-2)$ equations. Since we have n unknowns, i.e., y_1, y_2, \dots, y_n , we need two more equations to solve a system of linear equations. The remaining equations are provided by the two initial conditions:

From the initial condition, $y(0) = 1$, we can write $y_1 = 1$. For the second initial condition, $dy/dx = 1$, at $x = 0$, we will use a forward difference, i.e.,

$$dy/dx = (y_2 - y_1) / \Delta x,$$

or

$$y_2 - y_1 = \Delta x.$$

The x-domain is discretized in a similar fashion as in the previous examples. The n equations resulting from discretizing the domain can be written as a matrix equation similar to that of Example 1. Solution to the matrix equation can be accomplished, for example, through the use of left-division for matrices. The implementation of the solution for this example is left as an exercise for the reader.

MATLAB provides a number of functions for the numerical solution of differential equations. These functions are designed to operate on single differential equations (i.e., similar to the examples presented so far), as well as on systems of differential equations. Therefore, before presenting the MATLAB functions for solving ordinary differential equations, we present some concepts related to systems of such equations.

Systems of ordinary differential equations

To introduce the idea of systems of differential equations we will limit the coverage of the subject to first-order, linear equations with constant coefficients. A system of ordinary differential equations consists of a set of two or more equations with an equal number of unknown functions, $y_1(x)$, $y_2(x)$, etc. As an example consider the following *homogeneous* system:

$$\frac{dy_1}{dx} + 3y_1 - 2y_2 = 0, \quad \frac{dy_2}{dx} - y_1 + y_2 = 0.$$

In a homogeneous system the right-hand sides of the equations are zero. The following example represents a *non-homogeneous* system of ordinary differential equations:

$$\frac{dy_1}{dx} + 2y_1 - 5y_2 = \sin(x), \quad \frac{dy_2}{dx} - 4y_1 + 3y_2 = e^x.$$

Systems of ordinary differential equations using matrices

A homogeneous system of ODEs can be written as a single matrix differential equation by using vector functions and a matrix of coefficients as illustrated in the following example. First, we re-write the homogeneous system presented above to read:

$$\frac{dy_1}{dx} = -3y_1 + 2y_2,$$

$$\frac{dy_2}{dx} = y_1 - y_2.$$

Then, we define the vector function $\mathbf{f}(x) = [y_1(x) \ y_2(x)]^T$, and the matrix $\mathbf{A} = [-3 \ 2; 1 \ -1]$, and write the differential equation:

$$\frac{d}{dx} \mathbf{f}(x) = \mathbf{A} \mathbf{f}(x).$$

This result is equivalent to writing:

$$\frac{d}{dx} \begin{bmatrix} y_1(x) \\ y_2(x) \end{bmatrix} = \begin{bmatrix} -3 & 2 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} y_1(x) \\ y_2(x) \end{bmatrix}.$$

The non-homogeneous system presented earlier can be re-written as

$$\frac{dy_1}{dx} = -2y_1 + 5y_2 - \sin(x),$$

$$\frac{dy_2}{dx} = 4y_1 - 3y_2 + e^x.$$

For this system we will use the same vector function $\mathbf{f}(x)$ defined earlier, but change the matrix \mathbf{A} to $\mathbf{A} = \begin{bmatrix} -2 & 5 \\ 4 & -3 \end{bmatrix}$. We also need to define a new vector function, $\mathbf{g}(x) = \begin{bmatrix} -\sin(x) \\ \exp(x) \end{bmatrix}^T$. With these definitions, we can re-write the non-homogeneous system as:

$$\frac{d}{dx} \mathbf{f}(x) = \mathbf{A} \mathbf{f}(x) + \mathbf{g}(x),$$

or

$$\frac{d}{dx} \begin{bmatrix} y_1(x) \\ y_2(x) \end{bmatrix} = \begin{bmatrix} -2 & 5 \\ 4 & -3 \end{bmatrix} \cdot \begin{bmatrix} y_1(x) \\ y_2(x) \end{bmatrix} + \begin{bmatrix} -\sin(x) \\ \exp(x) \end{bmatrix}.$$

Converting second-order linear equations to a system of equations

A second-order linear ODE of the form $\frac{d^2y}{dx^2} + \frac{b}{dx} \frac{dy}{dx} + cy = r(x)$, can be transformed into a linear system of equations by introducing the relationship, $u(x) = \frac{dy}{dx}$, so that $\frac{d^2y}{dx^2} = \frac{du}{dx}$, thus, the equation reduces to $\frac{du}{dx} + bu + cy = r(x)$, or $\frac{du}{dx} = -bu - cy + r(x)$. The resulting system of equations is:

$$\begin{aligned} \frac{du}{dx} &= -bu - cy + r(x), \\ \frac{dy}{dx} &= u. \end{aligned}$$

Which can be written in matricial form as $d\mathbf{f}/dx = \mathbf{A} \mathbf{f}(x) + \mathbf{g}(x)$, with

$$f(x) = \begin{bmatrix} u \\ y \end{bmatrix}, A = \begin{bmatrix} -b & -c \\ 1 & 0 \end{bmatrix}, g(x) = \begin{bmatrix} r(x) \\ 0 \end{bmatrix}.$$

For example, the solution to the second order differential equation

$$\frac{d^2 y}{dx^2} + \frac{5 dy}{dx} - 3 y = x,$$

can be obtained by solving the equivalent first-order linear system:

$$\frac{du}{dx} = -5 u + 3 y + x,$$

$$\frac{dy}{dx} = u.$$

The procedure outlined above to transform a second order linear equation can be used to convert a linear equation of order n into a system of first-order linear equations. For example, if the original ODE is written as:

$$\frac{d^n y}{dx^n} + a_{n-1} \frac{d^{(n-1)} y}{dx^{(n-1)}} + \dots + a_2 \frac{d^2 y}{dx^2} + \frac{a_1 dy}{dx} + a_0 y = r(x),$$

we can re-write it as

$$\frac{d^n y}{dx^n} = -a_{n-1} \frac{d^{(n-1)} y}{dx^{(n-1)}} - \dots - a_2 \frac{d^2 y}{dx^2} - \frac{a_1 dy}{dx} - a_0 y + r(x),$$

and transform it into a system of n first-order linear equations given by:

$$\frac{du_{n-1}}{dx} = -a_{n-1} u_{n-1} - a_{n-2} u_{n-2} - \dots - a_2 u_2 - a_1 u_1 - a_0 y + r(x),$$

$$\frac{du_{n-2}}{dx} = u_{n-1}, \quad \frac{du_{n-3}}{dx} = u_{n-2}, \quad \dots, \quad \frac{du_1}{dx} = u_2, \quad \frac{dy}{dx} = u_1,$$

or, in matricial form,

$$\mathbf{f}(x) = \begin{bmatrix} u_{n-1} \\ u_{n-2} \\ \vdots \\ u_2 \\ u_1 \\ y \end{bmatrix}, \quad A = \begin{bmatrix} -a_{n-1} & -a_{n-2} & -a_{n-3} & \dots & -a_1 & -a_0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix}, \quad \mathbf{g}(x) = \begin{bmatrix} r(x) \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

For example, to transform the following fourth-order ($n=4$) linear ODE

$$\frac{d^4 y}{dx^4} + \frac{3 d^3 y}{dx^3} - \frac{2 d^2 y}{dx^2} + \frac{5 dy}{dx} + y = 0,$$

subjected to $y = 1$, $\frac{dy}{dx} = -1$, $\frac{d^2 y}{dx^2} = 0$, $\frac{d^3 y}{dx^3} = -1$, at $x = 0$, into a first-order linear system, we would write:

$$du_3/dx = -3u_3(x) + 2u_2(x) - 5u_1(x) - y(x) + x^2/2, \quad du_2/dx = u_3(x), \quad du_1/dx = u_2(x), \quad \text{and } dy/dx = u_1(x),$$

or

$$\frac{d}{dx} \begin{bmatrix} u_3(x) \\ u_2(x) \\ u_1(x) \\ y(x) \end{bmatrix} = \begin{bmatrix} -3 & 2 & -5 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_3(x) \\ u_2(x) \\ u_1(x) \\ y(x) \end{bmatrix} + \begin{bmatrix} x^2/2 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

with $v(x) = [u_3(x); u_2(x); u_1(x); y(x)]^T$, $A = [-3, 2, -5, -1; 1, 0, 0, 0; 0, 1, 0, 0; 0, 0, 1, 0]$; and $g(x) = [x^2/2; 0; 0; 0]^T$, the system of differential equations is written as $dv/dx = Av + g(x)$. The initial conditions are $y(0) = 1$, $u_1(0) = dy/dx = -1$, $u_2(0) = du_1/dx = d^2 y/dx^2 = 0$, $u_3(0) = du_2/dx = d^2 u_1/dx^2 = d^3 y/dx^3 = -1$, or $u_0 = [-1; 0; -1; 1]$.

Initial-value problems

Any differential equation of the form $df/dx = \mathbf{A} f(x) + \mathbf{g}(x)$, subject to initial conditions $\mathbf{f}(x_0) = \mathbf{f}_0$, is referred to as an *initial-value problem* (IVP). For a constant matrix \mathbf{A} , the solution can be found using one of several functions available in Matlab. Some of these functions are:

- ode23: IVP solver of order 2 or 3
- ode45: IVP solver of order 4 or 5

Use the following commands to get additional information on these functions:

```
» help ode23
```

```
» help ode45
```

Also, read Chapter 8 (*Ordinary Differential Equations*) in the *Using Matlab* guide for additional applications of these functions.