Chapter 6

Finite difference models: one dimension

6.1 overview

Our goal in building numerical models is to represent differential equations in a computationally manageable way. A large class of numerical schemes, including our initial value models of chapter 3, do so using finite difference representations of the derivative terms. The model domain is divided into a set of discrete, typically uniform, intervals of the independent variable (or variables). Arithmetic differences of function evaluations at the *nodes* between those intervals are used to approximate the derivatives of the function over those intervals.

6.2 the difference approximation

Finite difference techniques rely on the approximation of a derivative as the change (or *difference*) in the dependent variable over a small interval of the independent variable,

$$\frac{\mathrm{d}y}{\mathrm{d}x} \approx \frac{\Delta y}{\Delta x}$$

Those approximations are written using a small set of difference operators. These operators are handy, they can be used to write formulae to interpolate, differentiate, and integrate "data." The data might in fact be a set of measurements or, as in the case of a numerical integration scheme, might be the values of the dependent variable in a differential equation. The operators may be used singly to approximate first derivatives or in combination to approximate derivatives of any order.

The following definitions for finite difference operators use a particular set of Greek letters to identify the operators. Don't get sidetracked by that. The symbols are just a convenient shorthand (and you may find different symbols used in different texts). The names for the operations are universal and you should always be able to find formulas or strategies for their implementation in an appropriate reference book. This presentation uses the function name $f(x)$.

6.2.1 Taylor Series

Our approximation of the derivatives begins with their representation using the Taylor series. We have experience with the forward looking step:

$$f\left(x_{j+1}\right) \;=\; f\left(x_j\right) \;+\; hf^{(1)}\left(x_j\right) \;+\; \frac{h^2}{2}f^{(2)}\left(x_j\right) \;+\; \frac{h^3}{6}f^{(3)}\left(x_j\right) \;+\; \ldots \tag{6.1}$$

where $h = x_{j+1} - x_j$ is the interval over which we wish to approximate the derivative and the superscripts indicate the order of the derivative of $f(x)$. Truncating after the first derivative and rearranging yields an approximation of the first derivative

$$f^{(1)}\left(x_j\right) \;\approx\; \frac{f\left(x_{j+1}\right) \;-\; f\left(x_j\right)}{h} \tag{6.2}$$

The numerator $\left(f\left(x_{j+1}\right) - f\left(x_j\right)\right)$ is called the first forward difference of the function at x_j.

A backward looking series is written

$$f\left(x_{j-1}\right) \;=\; f\left(x_j\right) \;-\; hf^{(1)}\left(x_j\right) \;+\; \frac{h^2}{2}f^{(2)}\left(x_j\right) \;-\; \frac{h^3}{6}f^{(3)}\left(x_j\right) \;-\; \ldots \tag{6.3}$$

6.2.2 differences

Although we didn't call it such at the time, the first forward difference was used in the Euler single-step method for solving an initial value problem. The basic difference operations may be used in combination to create other differences, as in the double-interval central difference.

forward difference

$$\Delta\; f(x_j) \;=\; f(x_{j+1}) \;-\; f(x_j) \tag{6.4}$$

backward difference

$$\nabla f(x_j) \;= f(x_j) \;-\; f(x_{j-1}) \tag{6.5}$$

single-interval central difference

$$\delta\; f(x_j) \;=\; f(x_{j+\frac{1}{2}}) \;-\; f(x_{j-\frac{1}{2}}) \tag{6.6}$$

double-interval central difference

$$\frac{1}{2}\left\{\Delta\ f(x_j) - \nabla f(x_j)\right\} \ = \ \frac{1}{2}\ \left\{f(x_{j+1})\ -\ f(x_{j-1})\right\} \tag{6.7}$$

The double interval, created by differencing the forward and backward differences at x_j is attractive because nodes $j+\frac{1}{2}$ and $j-\frac{1}{2}$ in the single-interval central difference are not in the set of x_j where $j = \{1 : N+1\}$ are whole numbers.

6.2.3 higher-order differences

Higher-order differences are created by repeated differences. For example, applying the first forward difference to a first forward difference of a function gives us the

second forward difference:

$$\Delta^2 f(x_j) \ = \ \Delta\ \left\{\Delta\ f(x_j)\right\}$$

$$\Delta^2\ f(x_j) \ = \ \Delta\ \left\{f(x_{j+1})\ -\ f(x_j)\right\}$$

$$\Delta^2\ f(x_j)\ = f(x_{j+2})\ -\ 2\ f(x_{j+1})\ +\ f(x_j) \tag{6.8}$$

The superscript on Δ indicates that the operation is to be performed twice.

Similarly, we can derive the expression for the second central difference:

$$\delta^2 f(x_j) \ = \ \delta\ \left\{\delta\ f(x_j)\right\}$$

$$\delta^2\ f(x_j) \ = \ \delta\ \left\{\ f(x_{j+\frac{1}{2}})\ -\ f(x_{j-\frac{1}{2}})\right\}$$

$$\delta^2\ f(x_j)\ = f(x_{j+1})\ -\ 2\ f(x_j)\ +\ f(x_{j-1}) \tag{6.9}$$

There are many ways in which the differences may be combined, including inverses and fractional values.

6.2.4 using difference operators to compute derivatives

Derivatives are produced by dividing a finite difference by the interval over which the difference is computed. .

The first derivative of $f(x)$ can be represented using a double-interval central difference:

$$f^{(1)}(x_j) \;=\; \frac{f(x_{j+1}) \;-\; f(x_{j-1})}{2\;h} \;+\; error \tag{6.10}$$

The magnitude of the error is discussed in section (6.2.5).

The second derivative of $f(x)$ can be represented using the second central difference operator:

$$f^{(2)}(x_j) \;=\frac{f(x_{j+1}) \;-\; 2\;f(x_j) \;+\; f(x_{j-1})}{h^2} \;+\; error \tag{6.11}$$

6.2.5 numerical error

Truncation of the Taylor series introduces error into the numerical scheme. The magnitude of that error can be determined according to the order of the first truncated term in the series. For example, the first order Taylor expansion used in the first forward difference of the continuous function $f(x)$ is

$$f(x+h) \;=\; f(x) \;+\; hf^{(1)}(x) \;+\; \frac{h^2}{2}f^{(2)}(\xi)$$

in which ξ is an unknown number between x and $x+h$. The corresponding finite difference expression is

$$\frac{f(x+h) \;-\; f(x)}{h} \;=\; f^{(1)}(x) \;+\; \frac{h}{2}f^{(2)}(\xi)$$

in which we can see that the error in the approximation is proportional to h. Expressed more generally, this is

$$f^{(1)}(x) \;=\frac{f(x+h) \;-\; f(x)}{h} \;+\; \mathrm{O}(h) \tag{6.12}$$

in which the term O is understood to indicate the order of the error in the approximation. The exact value of the error cannot be determined because the value of ξ is unknown.

The accuracy of the finite difference approximation may be improved by using more function evaluations (more "samples" of the function) over the interval. The centered difference is made by subtracting a backward from a forward difference. Taylor expansions for these two differences are:

$$\begin{aligned} f(x+h) &= f(x) + \; hf^{(1)}(x) \;+\; \frac{h^2}{2}f^{(2)}(x) \;+\; \frac{h^3}{6}f^{(3)}(\xi_+) \\ f(x-h) &= f(x) - \; hf^{(1)}(x) \;+\; \frac{h^2}{2}f^{(2)}(x) \;-\; \frac{h^3}{6}f^{(3)}(\xi_-) \end{aligned}$$

Subtracting and rearranging yields

$$f^{(1)}(x) = \frac{f(x+h) - f(x-h)}{2h} \;-\; \frac{f^{(3)}(\xi_+) + f^{(3)}(\xi_-)}{12}h^2$$

	$f^{(1)}(x)$	$f^{(2)}(x)$
numerical	12.25	30.00
exact	12	30

Table 6.1: finite difference approximate and exact derivatives of x^3; $x = 2$, h=0.5

which may be simplified using the intermediate value theorem from Calculus

$$f^{(1)}(x) = \frac{f(x+h) - f(x-h)}{2h} \quad - \quad \frac{f^{(3)}(\xi)}{6}(h^2) \tag{6.13}$$

and we see that the centered difference approximation is second-order accurate (due to the cancellation of the terms involving $f^{(2)}(x)$). A happy turn of events.

The exact error due to the finite difference approximation may be evaluated for a function for which the exact solution is available. Consider the function

$$f(x) = x^3$$

The exact and numerical approximations of the first and second derivatives are

$$\begin{aligned} f^{(1)}(x) &= 3x^2 \approx \frac{(x+h)^3 - (x-h)^3}{2h} \\ f^{(2)}(x) &= 6x \approx \frac{(x+h)^3 - 2\,(x)^3 + (x-h)^3}{h^2} \end{aligned}$$

Example calculations of the approximate and exact derivatives are given in table (6.1).

6.3 using finite differences to solve differential equations: a generic example

6.3.1 discretization: writing the governing equation in a finite difference form

Consider the the general linear second-order differential equation:

$$Q(x)\,\frac{\mathrm{d}^2 y}{\mathrm{d}x^2} \;+\; R(x)\,\frac{\mathrm{d}y}{\mathrm{d}x} \;+\; S(x)\,y \;+\; T(x) \;=\; 0 \tag{6.14}$$

Our goal is to write this equation in a form that can be solved numerically for the unknown values of the dependent variable, y, at specified values of the independent variable $x_j \quad j = \{1, 2, \ldots, N, N+1\}$, where x_1 and x_{N+1} are the boundaries of the model domain. The dependent variable (or derivatives thereof) is known at x_1 and x_{N+1}.

N is the number of intervals between the $N+1$ nodes in the model domain.

The finite difference form of equation (6.14) is produced by replacing the first and second derivatives with first and second central differences:

$$Q\left(x_j\right)\frac{y(x_{j+1})-2y(x_j)+y(x_{j-1})}{h^2}+R\left(x_j\right)\frac{y(x_{j+1})-y(x_{j-1})}{2h}+S\left(x_j\right)y_j+T\left(x_j\right)=0 \tag{6.15}$$

Simplifying the subscripts for clarity, $Q(x_j) \equiv Q_j$, equation (6.15) is written:

$$Q_j\frac{y_{j+1}-2y_j+y_{j-1}}{h^2}+R_j\frac{y_{j+1}-y_{j-1}}{2h}+S_jy_j+T_j=0 \tag{6.16}$$

Rearranging equation (6.16) to group terms by node number for the dependent variable,

$$\left(\frac{Q_j}{h^2}-\frac{R_j}{2h}\right)y_{j-1}+\left(S_j-\frac{2Q_j}{h^2}\right)y_j+\left(\frac{Q_j}{h^2}+\frac{R_j}{2h}\right)y_{j+1}=-T_j \tag{6.17}$$

Terms involving the dependent variable are grouped on the left-hand side of (6.17) and terms involving constants are grouped on the right-hand side of the equation.

It is worth noting that at the first and last unknown nodes, the derivative terms include known values of the dependent variable. These are the boundary conditions. The first and last equations in our system are:

$$\left(\frac{Q_2}{h^2}-\frac{R_2}{2h}\right)y_1+\left(S_2-\frac{2Q_2}{h^2}\right)y_2+\left(\frac{Q_2}{h^2}+\frac{R_2}{2h}\right)y_3=-T_2 \tag{6.18}$$

$$\left(\frac{Q_N}{h^2}+\frac{R_N}{2h}\right)y_{N+1}+\left(\frac{Q_N}{h^2}-\frac{R_N}{2h}\right)y_{N-1}+\left(S_N-\frac{2Q_N}{h^2}\right)y_N=-T_N \tag{6.19}$$

Equation (6.18) is the finite difference approximation of the governing differential equation at the first unknown node in the domain, $j = 2$ and so it includes the boundary value y_1. Equation (6.19) is the approximation at the last unknown node, $j = N$ and so it includes the boundary value y_{N+1}.

6.3.2 linear alegebra representation of the system

The system of equations (6.17) that form the numerical solution to (6.14) over the range of x is tridiagonal. That is, three nodes in the model domain, and thus three values of y appear in each discretized equation and the y are sequential: $(j, j-1), (j, j), (j, j+1)$. The system can be written

$$\boldsymbol{A}y \;=\; b$$

in which the matrix $\boldsymbol{A}$ contains the coefficients of the dependent variables, the vector y represents those variables, and the vector b contains coefficients that do not involve unknown values of the dependent variable. The matrix equation visualized:

$$\begin{bmatrix} 1 & & & & & & \\ a_{21} & a_{22} & a_{23} & & & & \\ & a_{32} & a_{33} & a_{34} & & & \\ & & a_{i,i-1} & a_{i,i} & a_{i,i+1} & & \\ & & & \vdots & & & \\ & & & & & \vdots & \\ & & & a_{N-1,N-2} & a_{N-1,N-1} & a_{N-1,N} & \\ & & & & a_{N,N-1} & a_{N,N} & a_{N,N+1} \\ & & & & & & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_i \\ \vdots \\ \vdots \\ y_{N-1} \\ y_N \\ y_{N+1} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_i \\ \vdots \\ \vdots \\ b_{N-1} \\ b_N \\ b_{N+1} \end{bmatrix}$$

where the j'th row represents the equation for the j'th value of the dependent variable (equation 6.17). The coefficients a are:

$$\begin{aligned} a_{j,j-1} &= \frac{Q_j}{h^2} - \frac{R_j}{2h} \\ a_{j,j} &= S_j - \frac{2Q_j}{h^2} \\ a_{j,j+1} &= \frac{Q_j}{h^2} + \frac{R_j}{2h} \end{aligned}$$

The b are:

$$\begin{aligned} b_1 &= y_1 \\ b_j &= -T_j \\ b_{N+1} &= y_{N+1} \end{aligned}$$

where y_1 and y_{N+1} are boundary values.

The equations described by rows $j = \{2, N\}$ contain boundary values. The resulting known products, $a_{2,1}y_1$ and $a_{N,N+1}y_{N+1}$ may be moved to the right-hand side of the equation (as in equations 6.18 and 6.19) and stored in the vector b, in which case $a(2,1)$ and $a(N, N+1)$ are zero and the b are:

$$\begin{aligned} b_1 &= y_1 \\ b_2 &= -T_2 - \left(\frac{Q_2}{h^2} - \frac{R_2}{2h}\right) y_1 \\ b_j &= -T_j \\ b_N &= -T_N - \left(\frac{Q_N}{h^2} + \frac{R_N}{2h}\right) y_{N+1} \\ b_{N+1} &= y_{N+1} \end{aligned}$$

The matrix $\boldsymbol{A}$ and vector b are easily constructed using **for** loops. The only tricky part is treating the second and Nth rows differently than the other interior rows.

Once $\boldsymbol{A}$ and b are created, the y are found by solving $y = \boldsymbol{A}^{-1}b$. We will use MATLAB's backslash operator to perform the calculation.

6.4 an example problem

6.4.1 a second-order linear differential equation with two boundary values

$$x\frac{\mathrm{d}^2y}{\mathrm{d}x^2} - 2\frac{\mathrm{d}y}{\mathrm{d}x} + 2 = 0 \tag{6.20}$$

$$y\,(x=0) = 0 \qquad \text{and} \qquad y\,(x=1) = 0$$

The equation may be discretized using central differences:

$$x_j\left(\frac{y_{j+1} - 2y_j + y_{j-1}}{\Delta x^2}\right) - 2\left(\frac{y_{j+1} - y_{j-1}}{2\Delta x}\right) + 2 = 0 \tag{6.21}$$

$$\left(\frac{x_j}{\Delta x^2} + \frac{1}{\Delta x}\right)y_{j-1} - \frac{2x_j}{\Delta x^2}y_j + \left(\frac{x_j}{\Delta x^2} - \frac{1}{\Delta x}\right)y_{j+1} = -2 \tag{6.22}$$

where Δx represents the interval (or step) size.

The coefficients of equation (6.22) are used to build the known parts of the matrix equation representing the system of equations that together represent the solution to equation (6.20) over the model domain.

As before, we define N as the number of intervals between $x_1 = 0$ and $x_{N+1} = 1$. The interval (step) size $\Delta x = (x_{N+1} - x_1)/N$.

The following formulae are used to generate the interior-row elements of $\boldsymbol{A}$:

$$\left.\begin{array}{rcl} a(j,j-1) & = & \frac{x_j}{\Delta x^2} + \frac{1}{\Delta x} \\ a(j,j) & = & \frac{2x_j}{\Delta x^2} \\ a(j,j+1) & = & \frac{x_j}{\Delta x^2} - \frac{1}{\Delta x} \end{array}\right\} j = 2\ldots N$$

At this point, we have a decision to make. We can choose to keep the coefficients of the boundary values in the matrix $\boldsymbol{A}$ (rows $j = 2$ and $j = N$) or we can choose to move the products of the coefficients $a(2,1)$ and $a(N,N+1)$ and the corresponding boundary values into the vector b. For our current purpose, the decision is trivial but for large problems with constant boundary conditions, it would make sense to store them at the start of the calculation and not create them repeatedly within a for loop.

In this example we will adopt the simpler approach and the vector b is constructed:

$$\begin{aligned} b_1 &= y_1 \\ b_j &= -2 \quad \} j = 2 \ldots N \\ b_{N+1} &= y_{N+1} \end{aligned}$$

Had we moved the boundary value products, the equations for b would have been:

$$\begin{aligned} b_1 &= y_1 \\ b_2 &= -2 - \left(\frac{x_2}{\Delta x^2} + \frac{1}{\Delta x} \right) y_1 \\ b_j &= -2 \quad \} i = 3 \ldots N-1 \\ b_N &= -2 - \left(\frac{x_N}{\Delta x^2} - \frac{1}{\Delta x} \right) y_{N+1} \\ b_{N+1} &= y_{N+1} \end{aligned}$$

and the coefficients $a(2,1)$ and $a(N, N+1)$ would be zero.

6.4.2 Matlab script

As usual, the first step is to set up the model domain.

```
%* set up model domain

xi=0;              % one end point of domain
xf=1;              % the other end point
N=20;              % number of intervals
dx=(xf-xi)/N;      % interval size ("step" size)
x=[xi:dx:xf]';     % vector holding model node values of independent variable
                   % transpose to make a column vector

%* boundary conditions on dependent variable
yi=0;              % y(xi), node 1
yf=0;              % y(xf), node N+1

%* dependent variable
y=zeros(N+1,1);

% boundary conditions into y
y(1)=yi;
y(N+1)=yf;
```

Next, the matrix containing coefficients of the unknown dependent variables from the finite difference form of the model equation is created. For a second order differential equation discretized using central differences, each row of $\boldsymbol{A}$ contains at most three non-zero elements. We have two special statements, for $j = 1$ and $j = N + 1$ and a **for** loop for the interior rows $j = \{2 : N\}$.

```
%** left-hand side of discretized equation
%   A matrix in Ay=b

A=zeros(N+1,N+1);

% boundary values
A(1,1)= 1;
A(N+1,N+1)= 1;

% fill j=2...N
for j=2:N
        A(j,j-1) = x(j)/dx^2 + 1/dx;
        A(j,j) = -2*x(j)/dx^2;
        A(j,j+1) = x(j)/dx^2 - 1/dx;
end
```

The vector containing constants and known values of the dependent variable is constructed in a similar manner. The first and last elements of b contain boundary conditions and the interior elements are the coefficients that do not involve the dependent variable.

```
%** right-hand side of Ay=b

b=zeros(N+1,1); % column vector

% boundary values
b(1)= y(1);
b(N+1)= y(N+1);

b(2:N)= -2;
```

The solution to our system of equations is the product of the inverse of $\boldsymbol{A}$ and the vector $\boldsymbol{b}$.

```
%** solve the equation

y=A\b;  % unknown y's into y

%** plot the result
figure(1)
clf
plot(x, y, 'b-'), hold on
title('finite difference solution to x d^2y/dx^2-2dy/dx+2=0 y(0)=0 y(1)=0')
ylabel('y')xlabel('x')
```

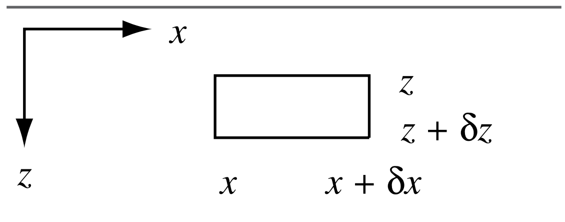

Figure 6.1: geometry for force balance in a fluid flow between two parallel plates

6.5 laminar flow in a channel (derivation for exercise 2)

A variety of fluid flow problems can be simplified by assuming that the flow is between parallel plates. For example, in exercise 6 you will consider flow in the asthenosphere due to an uneven surface load. Over sufficiently long distances, the asthenosphere looks like a parallel-sided slab between the lithosphere and its lower boundary. Such flows are called *channel flows*.

We will simplify the problem by assuming that our channel is infinite in two horizontal directions and that the Reynolds number is less than 1, that is, that the flow is laminar. Fluid flow within our channel may be driven by a pressure gradient, relative motion of the channel walls, or both. Taking x to represent one horizontal direction and z to represent the vertical direction, our goal is to develop an equation for the fluid velocity in the horizontal direction $u(z)$.

Consider an little box with sides aligned parallel and perpendicular to the z direction (Figure 7.1). The z faces of the box are at some locations z and $z + \delta z$, where the Greek letter δ ("del") indicates a small change. Similarly, the x faces (z-parallel) of the box are at x and $x + \delta x$. If $u(z)$ is not constant, there will be tractions (shear stresses) on the z faces of the little box, $\tau(z)$ and $\tau(z + \delta z)$ such that

$$\tau\left(z + \delta z\right) = \tau\left(z\right) + \frac{\mathrm{d}\tau}{\mathrm{d}z}\delta z \tag{6.23}$$

Pressure p at either end of the little pox can be similarly described:

$$p\left(x + \delta x\right) = p\left(x\right) + \frac{\mathrm{d}p}{\mathrm{d}z}\delta x \tag{6.24}$$

The shear stress τ is proportional to the gradient in u across the channel. If we assume a *linear-viscous* behavior for our fluid, the relationship between shear stress and fluid velocity is:

$$\tau = \mu\frac{\mathrm{d}u}{\mathrm{d}z} \tag{6.25}$$

where μ represents the viscosity of the fluid. This expression can be used, along with Newton's Second Law of motion, to derive an equation for u.

Newton's Second Law of motion states that the sum of forces acting on an object must equal the product of the object's mass and acceleration. In our laminar flow case, the acceleration must be zero, and we can use Newton's Law to write a force balance equation for our little box within the fluid flow. The z faces experience shear forces

$$\left(\tau\left(z\right)+\frac{\mathrm{d}\tau}{\mathrm{d}z}\delta z\right)\delta x$$

and

$$-\tau\left(z\right)\delta x$$

The x faces experience normal forces

$$p\left(x\right)\delta z$$

and

$$-\left(p\left(x\right)+\frac{\mathrm{d}p}{\mathrm{d}x}\delta x\right)\delta z$$

The force balance is thus:

$$\begin{aligned}\sum \boldsymbol{F} &= 0\\ \frac{\mathrm{d}\tau}{\mathrm{d}z}\delta z\delta x-\frac{\mathrm{d}p}{\mathrm{d}x}\delta x\delta z &= 0\\ \frac{\mathrm{d}\tau}{\mathrm{d}z} &= \frac{\mathrm{d}p}{\mathrm{d}x} \qquad (6.26)\end{aligned}$$

Substituting (6.25) into (6.26),

$$\frac{\mathrm{d}}{\mathrm{d}z}\left(\mu\frac{\mathrm{d}u}{\mathrm{d}z}\right)=\frac{\mathrm{d}p}{\mathrm{d}x} \qquad (6.27)$$

and if μ is uniform,

$$\mu\frac{\mathrm{d}^2u}{\mathrm{d}z^2}=\frac{\mathrm{d}p}{\mathrm{d}x} \qquad (6.28)$$

Equation (6.28) can be solved numericaly or exactly with appropriate boundary conditions on u.

6.6 exercises

1. Derive the expression for the second backward difference. Write out all your steps.

2. The error associated with various approximations can be investigated by comparing finite difference solutions with exact solutions for simple functions. Consider the function

$$f(x) = \sin(x)$$

   Using $x = 1$ and $h = 0.1$, show that a centered difference yields a better approximation of the first derivative $f^{(1)}(x)$ than does a forward difference. To do this, you will need to compute the local error at $x = 1$.

3. Consider a vector x that represents an independent variable and a second vector y that represents a dependent variable that we believe to be a function of x. Here's a bit of code that could be used to calculate the first forward difference of y, assuming you are given some set of x and y:

```
% first forward difference of y with respect to x

N=length(y)-1;
h=x(2)-x(1);

% first forward difference
ffd=zeros(1, N);

for n=1:N
        ffd(n)=(y(n+1) - y(n)) / h;
end

figure(1)
clf
plot((x(1:N)+x(2:N+1))/2 , ffd , 'r-')
```

   (a) Why is the dimension of `ffd` in the example code `length(y)-1`?

   (b) Write a script or collection of functions that will accept any size x and y vectors (you will need to measure the length of x and then use that dimension to create whatever other variables are necessary), and will calculate derivatives using the:

      i. first backward difference of y with respect to x

      ii. centered difference of y with respect to x

      iii. second centered difference of y with respect to x

4. Two data files, `testdat.mat` and `testdat2.mat` are available via ftp from the class website. Download the files. Each file contains two arrays, `x` and `y`, and `x` and `y2`. Use your collection of scripts to compute the first derivative using forward, backward, and centered differences. Plot your results.

(a) Make a plot that compares the first forward difference and centered difference for the data in `testdat.mat`.

(b) Make the same comparison for the data in `testdat2.mat`. Why are the two derivatives different for this data set but not for the data set in exercise 4a?

5. Implement the MATLAB script in section 6.4. Try several domain resolutions. Turn in a figure showing solutions to the equation for three different resolutions.

6. The one-dimensional, non-turbulent flow of a viscous fluid in a channel is described by equation (6.28). The fluid flow may be driven by a pressure gradient dp/dx, by motion of one of the channel boundaries with magnitude u, or both.

   Suppose you want to use this equation to investigate the response of the asthenosphere to a change in the growth rate of a mid-ocean ridge. If the ridge growth rate has been steady for some time, then it is likely to be in isostatic equilibrium, and it does not drive a flow in the asthenosphere. However, if the ridge growth rate changes, the extra (or missing), uncompensated, thickness of ocean crust increases (or decreases) the pressure under the ridge relative to the pressure at an equal depth at some distance from the ridge and the asthenosphere must adjust in order to return the system to isostatic equilibrium. That is, according to the hypothesis, an asthenosphere flow is driven by a gradient in lithostatic pressure (the weight of the uncompensated overlying rock) associated with the growing ridge. The gradient is

$$\frac{\mathrm{d}p}{\mathrm{d}x} = \rho g \frac{\mathrm{d}h}{\mathrm{d}x} \tag{6.29}$$

   in which ρ represents the density of the asthenosphere, g represents the acceleration due to gravity, and dh/dx represents the gradient in uncompensated oceanic crust thickness across the ridge.

   If we assume that the speed of the asthenosphere is zero at its upper and lower boundaries, the boundary conditions for (6.28) are $u\,(z = 0) = u\,(z = h) = 0$.

   This model is simple but it gives us a chance to practice using an equation with some physical meaning and it is a base from which a more complete model could be built.

   (a) The first step in building the numerical model for this exercise is to approximate the second derivative of u and use the result to discretize the governing equation. Re-write equation (6.28) using central differences.

   (b) Using the program written in section 6.4 as a guide, write a MATLAB script to model $u(z)$ in a 200 km thick asthenosphere at a point along the flank of a growing mid-ocean ridge where $dh/dx = -5 \times 10^{-4}$. Use the coefficient values supplied in table (6.2).

   Write out, by hand, the equations you will use to construct the matrix of coefficients of the unknown u (the matrix $\boldsymbol{A}$) and the vector of known values(b) and use them to write a MATLAB script that solves (6.28) for the boundary conditions and constants given here. Plot the simulated $u(z)$, orienting the axes in a way that makes sense in the context of the process under investigation. Turn in your script and the figure.

coefficient	*value*	*units*
μ	4 x 10^{19}	Pa s
ρ	3000	kg m^{-3}
g	9.81	m s^{-2}

Table 6.2: coefficients for asthenosphere flow problem

7. Suppose you wish to include motion of the oceanic crust in your model of asthenosphere flow. Explain how that would be accomplished in this model, implement the change, and produce a figure using a crust displacement rate of 50 mm a^{-1}.