MACSI One Day Graduate Course:
Numerical Solution to Differential Equations using Matlab

Part 1: Numerical Differentiation and Finite Difference Methods

Niall Madden and Meghan Stephens
Department of Mathematics

National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh

3rd of April 2007.

Numerical Differentiation

The problem is: *Given values for a function* f *at points* $x = x_0, x_1, \ldots, x_n$, *find approximations for* $f'(x_i)$ *and* $f''(x_i)$.

Why Bother? After all, unlike, say, numerical integration, differentiation is easy.

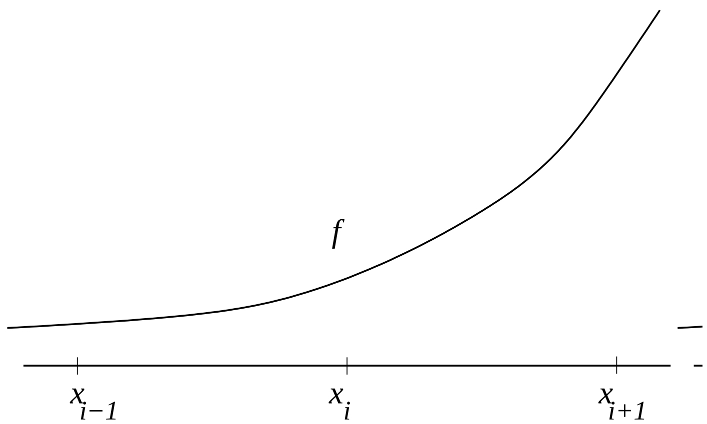
1. Often we don't know f for any x other than the x_i.
2. We might want a simple rule that can be easily programmed.
3. It is a gentle introduction of finite difference methods for boundary value problems.

Numerical Differentiation

Constructing the rules can be done in several ways:

1. Geometrically. Recall, all we are really looking for is the slope of the tangent to $f(x)$ at $x = x_i$.
2. By finding a polynomial that interpolates f and differentiating that.
3. Undetermined coefficients.
4. **Taylor's Theorem**.

Numerical Differentiation Geometry



As suggested by the diagram, one could take

Backward Differencing

$$\frac{df}{dx}(x_i) = f'(x_i) \approx \frac{f_i - f_{i-1}}{x_i - x_{i-1}} =: D^-(f)_i.$$

Forward Differencing

$$f'(x_i) \approx \frac{f_{i+1} - f_i}{x_{i+1} - x_i} =: D^+(f)_i.$$

Since one approach seems to over-estimate $f'(x_i)$, and the other seems to under-estimate it, we could take the average:

Central Differencing

$$f'(x_i) \approx \frac{-f_{i-1} + f_{i+1}}{x_{i+1} - x_{i-1}} =: D^c(f)_i.$$

We are also/primarily interested in approximating

$$\frac{d^2}{dx^2}f(x_i) = f''(x_i).$$

But of course,

$$\frac{d^2}{dx^2}f(x_i) = \frac{d}{dx}\left(\frac{d}{dx}f(x_i)\right) = f''(x_i).$$

so we can use the forward and backward difference operators:

$$f''(x_i) \approx D^+\big(D^-(f)\big)_i.$$

The result is:

2nd Order Central Differencing

$$f''(x_i) \approx \frac{1}{h^2}(f_{i-1} - 2f_i + f_{i+1}) =: \delta^2(f)_i.$$

One may also deduce the rules for D^c and δ^2 above using *undetermined coefficients*: Assume that the rule is of the form

$$a_o f_{i-1} + a_1 f_i + a_{i+1} f_{i+1}$$

and that it will give exactly the right answer for $f(x) \equiv 1$, $f(x) = x$ and $f(x) = x^2$. Then solve for a_0, a_1 and a_2.

This is quite a handy approach if you want to quickly construct a method for estimating derivatives.

The best way to construct differentiation methods — and get error estimates — is to use the celebrated:

Theorem (Taylor's Theorem)

$$f(x) = f(a) + (x-a)f'(a) + \frac{(x-a)^2}{2}f''(a) + \frac{(x-a)^3}{3!}f'''(a) + \cdots + \frac{(x-a)^n}{n!}f^{(n)}(a) + R_n.$$

where the remainder R_n *is be given by*

$$R_n = \frac{(x-a)^{n+1}}{(n+1)!}f^{(n+1)}(\eta), \qquad \text{for some } \eta \in (a, b).$$

To simplify things, take the points to be equally spaced: $h = x_i - x_{i-1}$.
Suppose we take $a = x_i$ and $x = x_{i-1}$ or $x = x_{i+1}$. Now we can write the truncated Taylor Series:

$$f_{i-1} = f_i - hf'(x_i) + \frac{h^2}{2}f''(\tau),$$

for $\tau \in (x_{i-1}, x_i)$. Rearranging we get

$$f'(x_i) = \frac{1}{h}(f_i - f_{i-1}) + \frac{h}{2}f''(\tau),$$

the backward difference scheme, with an error term.

For the central difference scheme, write:

$$f_{i+1} = f_i + hf'(x_i) + \frac{h^2}{2}f''(x_i) + \frac{h^3}{6}f'''(\tau_1), \quad \text{for } \tau_1 \in (x_i, x_{i+1}),$$
$$f_{i-1} = f_i - hf'(x_i) + \frac{h^2}{2}f''(x_i) - \frac{h^3}{6}f'''(\tau_2), \quad \text{for } \tau_2 \in (x_{i-1}, x_i).$$

Now subtract the 2nd equation from the 1st and divide by $2h$ to get the central difference operator $D^c(f)_i$,
To complete the error terms, just observe that, if $f'''(x)$ is continuous on $[x_{i-1}, x_{i+1}]$, then there must be a point $\tau \in [x_{i-1}, x_{i+1}]$ such that $f'''(\tau) = (f'''(\tau_1) + f'''(\tau_2))/2$.

To get the second-order difference operator $\delta^2(f)_i$, just extend the formulae above by one more term

$$f_{i+1} = f_i + hf'(x_i) + \frac{h^2}{2}f''(x_i) + \frac{h^3}{6}f'''(x_i) + \frac{h^4}{24}f^{(iv)}(\tau_1),$$
$$f_{i-1} = f_i - hf'(x_i) + \frac{h^2}{2}f''(x_i) - \frac{h^3}{6}f'''(x_i) + \frac{h^4}{24}f^{(iv)}(\tau_2),$$

add them and divide by h^2:

$$\begin{aligned} f''(x_i) &= \delta^2 f_i + \frac{h^2}{12}f^{(iv)}(\tau) \\ &= \frac{1}{h^2}(f_{i-1} - 2f_i + f_{i+1}) + \frac{h^2}{12}f^{(iv)}(\tau) \end{aligned}$$

for some $\tau \in (x_{i-1}, x_{i+1}$.

Our differential equation is

$$-u''(x) + r(x)u(x) = f(x) \quad \text{for } 0 < x < 1.$$
$$u(0) = \alpha, u(1) = \beta.$$

Idea:

- Choose a number $N \geqslant 2$ of points in your mesh:

  $0 = x_0 < x_1 < \cdots < x_{n-1} < x_n = 1$, where $x_k - x_{k-1} = h = \dfrac{1}{N}$.
- Replace the 2nd derivative in the equation with the 2nd order difference formula that we found earlier.

$$u''(x_i) \approx \delta^2 u_i = \frac{1}{h^2}(u_{i-1} - 2u_i + u_{i+1}).$$

- Solve the resulting system of linear equations to get an approximation for the solution to the DE *at the mesh points*.

Finite Difference Method

Let u_k be the approximation for $u(x_k)$. Then:

$$u_0 = A$$

$$-\frac{1}{h^2}(u_{k-1} - 2u_k + u_{k+1}) + r_k u_k = f_k$$

$$u_N = B$$

where r_k and f_k are short-hand for $r(x_k)$ and $f(x_k)$.

This is just the linear system of equations:

$$\mathbf{A}\mathbf{u} = \mathbf{f}$$

where $\mathbf{u}$ and $\mathbf{v}$ are vectors:

$$\mathbf{u} = \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_n \end{pmatrix} \quad \mathbf{f} = \begin{pmatrix} \alpha \\ f_1 \\ f_2 \\ f_3 \\ \vdots \\ \beta \end{pmatrix}.$$

And A is the matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ -1/h^2 & 2/h^2 + r_1 & -1/h^2 & 0 & 0 & \dots & 0 \\ 0 & -1/h^2 & 2/h^2 + r_2 & -1/h^2 & 0 & \dots & 0 \\ 0 & 0 & -1/h^2 & 2/h^2 + r_2 & -1/h^2 & \dots & 0 \\ \vdots & & & & & \ddots & \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix}$$

This is a *tridiagonal* linear system of equations: it is easily solved.