Chapter 4: Numerical Methods for Common Mathematical Problems

Interpolation

Problem: Suppose we have data defined at a discrete set of points (x_i, y_i), $i = 0, 1, ..., N$. Often it is useful to have a smooth function $y(x)$ that passes through all these points so that we can calculate y at intermediate values of x.

Lagrange Polynomial Interpolation

There is a UNIQUE polynomial of degree N passing through the N+1 points (x_i, y_i), $i = 0, 1, ..., N$.

Formula

$$\begin{aligned} P(x) &= \frac{(x-x_1)(x-x_2)....(x-x_N)}{(x_0-x_2)(x_0-x_3)....(x_0-x_N)}\,y_0 + \frac{(x-x_0)(x-x_2)....(x-x_N)}{(x_1-x_0)(x_1-x_2)....(x_1-x_N)}\,y_1 \\ &\quad +.... + \frac{(x-x_0)(x-x_1)....(x-x_{N-1})}{(x_N-x_0)(x_N-x_1)....(x_N-x_{N-1})}\,y_N \end{aligned}$$

This is known as the LAGRANGE INTERPOLATING POLYNOMIAL. Note that the denominators are constants and the numerators are all polynomials of degree N.

Check

$$\begin{aligned} P(x_0) &= \frac{(x_0-x_1)(x_0-x_2)....(x_0-x_N)}{(x_0-x_2)(x_0-x_3)....(x_0-x_N)}\,y_0 + \frac{(x_0-x_0)(x_0-x_2)....(x_0-x_N)}{(x_1-x_0)(x_1-x_2)....(x_1-x_N)}\,y_1 \\ &\quad +.... + \frac{(x_0-x_0)(x_0-x_1)....(x_0-x_{N-1})}{(x_N-x_0)(x_N-x_1)....(x_N-x_{N-1})}\,y_N \\ &= 1.y_0 + 0.y_1 + ... + 0.y_N = y_0 \qquad \text{as claimed!} \end{aligned}$$

Similarly $P(x_1) = y_1$, $P(x_2) = y_2$, etc.

Linear and Cubic Interpolation

Sometimes N is so large that the Lagrange polynomials become impractical. In this case, alternatives include LINEAR and CUBIC interpolation.

Linear: For linear interpolation, draw a straight line between adjacent points (see diagram).

Cubic: For cubic interpolation, use Lagrange polynomials to interpolate between surrounding 4 points. i.e. if desired $x \in [x_j, x_{j+1}]$ find the Lagrange polynomial interpolating (x_{j-1}, y_{j-1}), (x_j, y_j), (x_{j+1}, y_{j+1}), (x_{j+2}, y_{j+2}). Note that if $x \in [x_0, x_1]$ then you need to use (x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3) and if $x \in [x_{N-1}, x_N]$ use (x_{N-3}, y_{N-3}), (x_{N-2}, y_{N-2}), (x_{N-1}, y_{N-1}), (x_N, y_N).

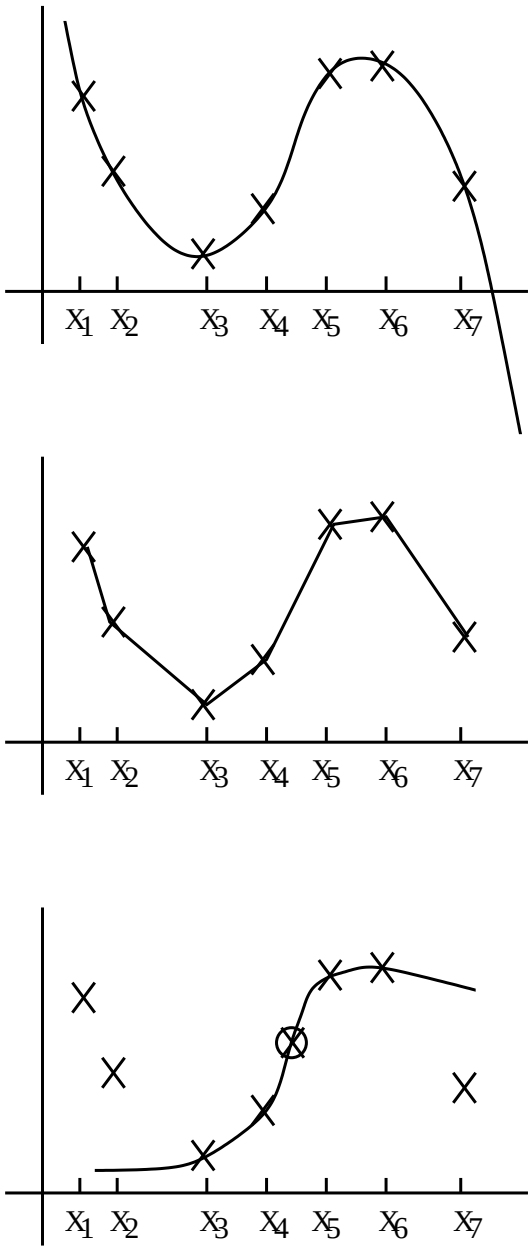

Figure: Illustrating different interpolation methods.

Root Finding

Problem: To solve a nonlinear equation $f(x) = 0$ (e.g. find the roots of a polynomial, or the solution of $\cosh x - x^3 = 0$).

Newton's Method

Advantages: Fast, Easy to Implement

Disadvantages: Need to be able to calculate $f'(x)$, sometimes goes wild and gives wrong answer

1. Start at a point x_1 hopefully near the root at $x = r$, $(f(r) = 0)$.
2. Calculate $f_1 = f(x_1)$ and $f_1' = f'(x_1)$
3. Draw tangent line between (x_1, f_1) and new point $(x_2, 0)$.
   Use equation of line in form
   $$\frac{y_2 - y_1}{x_2 - x_1} = m \qquad \text{(gradient)}$$
   in our case this gives
   $$\frac{0 - f_1}{x_2 - x_1} = f_1',$$
   or rearranging
   $$x_2 = x_1 - \frac{f_1'}{f_1}.$$
4. Repeat to get a sequence x_3, x_4,... using
   $$x_{n+1} = x_n - \frac{f_n}{f_n'}$$
5. Stop when $|x_{n+1} - x_n| < \delta$ for some tolerance δ, e.g. $\delta = 10^{-6}$.

Example: Find $\sqrt{3}$ without a calculation

$$a = \sqrt{3}, \quad a^2 - 3 = 0.$$

This suggests

$$f(x) = x^2 - 3, \qquad f'(x) = 2x$$

so Newton's method gives

$$x_{n+1} = x_n - \frac{x_n^2 - 3}{2x_n} \quad \text{or} \quad x_{n+1} = \frac{x_n^2 + 3}{2x_n}.$$

Start nearby, e.g. $x_1 = 2$.

$$x_2 = \frac{4+3}{4} = \frac{7}{4}, \quad x_3 = \frac{\left(\frac{7}{4}\right)^2 + 3}{\frac{7}{2}}, \quad x_4 = \text{etc.}$$

Difficulties:

Consider $p(x) = x^3 - x$, this has roots at $a = -1, 0, 1$.
Try a starting point $x_1 = 1/2$

$$x_2 = x_1 - \frac{f_1}{f_1'} = x_1 - \frac{x_1^3 - x_1}{3x_1^2 - 1} = \frac{1}{2} - \frac{\frac{1}{8} - \frac{1}{2}}{\frac{3}{4} - 1} = -1$$

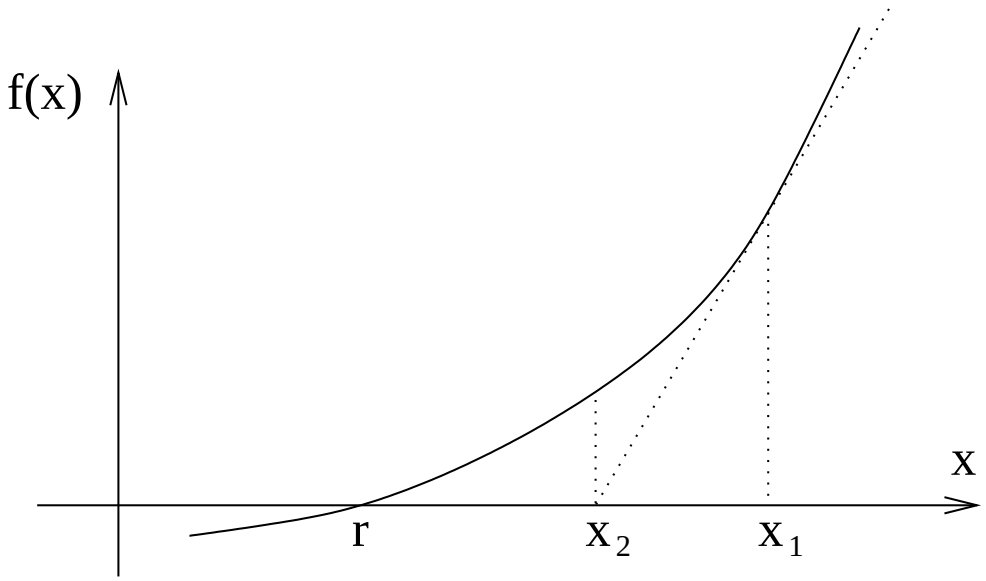

Figure: Illustrating the Newton-Raphson algorithm.

this means we have found a root we were not even near!!!

Problem arises whenever $|f'(x_n)|$ is small...the tangent is nearly horizontal, and intersects the origin far from the starting point.

Secant Method

Advantages: Fast (but slightly slower than Newton's method), no need to calculate $f'(x)$.

Disadvantages: Sometimes goes wild and gives wrong answer, need to make two initial guesses.

$$\text{Slope of secant:} \qquad m = \frac{f(x_2) - f(x_1)}{x_2 - x_1}$$

$$\text{Crosses the } x\text{-axis at: } \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{0 - f(x_2)}{x_3 - x_2} \quad \text{rearranging} \quad x_3 = x_2 - f(x_2)\left(\frac{x_2 - x_1}{f(x_2) - f(x_1)}\right)$$

The equation is therefore

$$x_{n+1} = x_n - f(x_n)\left(\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}\right)$$

and this can be used to generate x_4, x_5, x_6, etc. as before.

Bisection Method

Advantages: Very robust, always finds the root in the interval.

Disadvantages: Relatively slow to converge. Need starting points either side of root.

1. Start with points a, b known to be either side of the root r. This means that

$$f(a)f(b) \leq 0, \qquad \text{and if } f(a)f(b) = 0, \text{ then root } r \text{ is at } a \text{ or } b.$$

   Set $n = 1$.

2. If not, define

$$x_n = \frac{a+b}{2}, \qquad \text{and compute } f(x_n).$$

   If $f(x_n) = 0$ we have found the root. Stop!

3. Check sign of $f(a)f(x_n)$:

$$\begin{aligned} &\text{if } \ f(a)f(x_n) < 0 \quad \text{root is between } a \text{ and } x_n \text{ so set} \quad b = x_n, \quad n \to n+1., \\ &\text{if } \ f(a)f(x_n) > 0 \quad \text{root is between } x_n \text{ and } b \text{ so set} \quad a = x_n, \quad n \to n+1. \end{aligned}$$

4. Check $|a-b|$. If $|a-b| > \delta$ (tolerance) return to step 2. Otherwise, the final estimate for the root is

$$x_n = \frac{a+b}{2}.$$

How many iterations are needed? Let $L_0 = |b-a|$ be the initial interval size and δ be the tolerance. After N iterations the length of the interval is $L_N = 2^{-N}L_0$, as it has been chopped in half N times. Need to stop the iteration when

$$2^{-N}L_0 < \delta, \qquad 2^N > \frac{L_0}{\delta}, \qquad N > \log_2\left(\frac{L_0}{\delta}\right).$$

Numerical Differentation

Problem: Consider a dataset (x_1, y_1), (x_2, y_2), ..., (x_N, y_N) where the x_i may not be evenly spaced. The aim is to find good approximate expressions for the first and second derivatives. That is, we assume that there exists a smooth function $f(x)$ so that $y_i = f(x_i)$, and look for approximations to df/dx and d^2f/dx^2.

Formulae for Numerical Derivatives

First Derivatives

Starting with the definition

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

it is straightforward to make two approximations for $df/dx(x_i)$.

$$f'_+(x_i) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} \quad \text{and} \quad f'_-(x_i) = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}.$$

Notice that f'_+ is a right-sided derivative and f'_- is left-sided.

A better approximation is given by the average of f'_+ and f'_- (centred derivative)

$$f'_{(2)}(x_i) = \frac{1}{2}\left(f'_+ + f'_-\right).$$

Second Derivatives

The right-sided derivative $f'_+(x_i)$ equals the exact derivative of f at some point $\in [x_i, x_{i+1}]$. Can guess that this is the midpoint $(x_{i+1} + x_i)/2$, i.e.

$$f'_+(x_i) \approx \frac{df}{dx}\left(x = \frac{x_{i+1} + x_i}{2}\right)$$

Similarly we guess

$$f'_-(x_i) \approx \frac{df}{dx}\left(x = \frac{x_i + x_{i-1}}{2}\right)$$

From these we can write

$$\frac{d^2f}{dx^2}(x_i) \approx f''_{(2)} = \frac{f'_+(x_i) - f'_-(x_i)}{\frac{x_i+x_{i+1}}{2} - \frac{x_i+x_{i-1}}{2}}$$

Uniform Grid

Often we have a uniform grid in x, i.e. $x_i = x_0 + ih$. What do our forumlae above become?

$$f'_+(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h},$$

$$f'_+(x_i) = \frac{f(x_i) - f(x_{i-1})}{h},$$

$$f'_{(2)}(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h},$$

and

$$f''_{(2)}(x_i) = \frac{f'_+(x_i) - f'_-(x_i)}{h} = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2}.$$

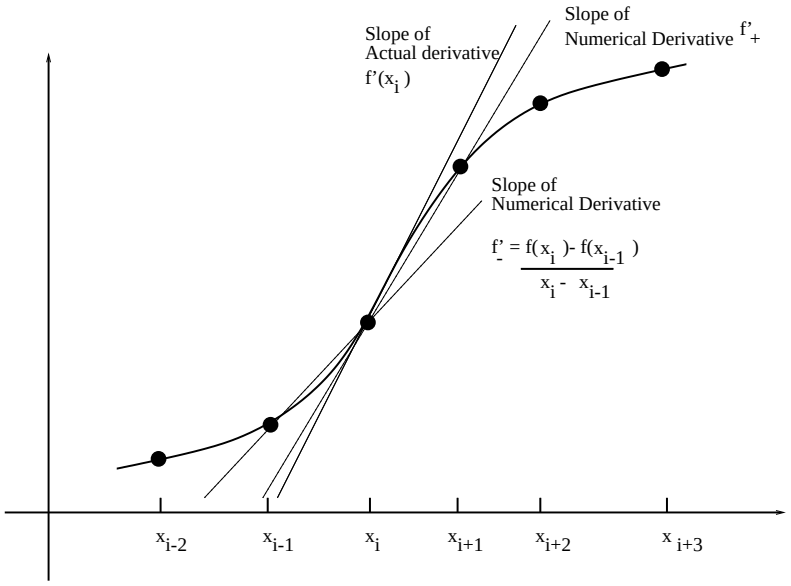

Figure: Illustrating left $f'_- = (f(x_i) - f(x_{i-1}))/(x_i - x_{i-1})$ and right $f'_+ = (f(x_{i+1}) - f(x_i))/(x_{i+1} - x_i)$ numerical derivatives.

How accurate are the above approximations?

Consider the simple function $f(x) = x^N$.

We have

$$f'(x) = Nx^{N-1}, \qquad f''(x) = N(N-1)x^(N-2).$$

Can compare with numerical derivatives. Take $x_0 = 0$, $x_0 = 1$, $x_0 = 2$. Then

$$\begin{aligned} f'_+(x_1) &= \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{2^N - 1^N}{1} = 2^N - 1. \\ f'_-(x_1) &= \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{1^N - 0^N}{1} = 1. \end{aligned}$$

whereas the exact value is $f'(x_1) = N$. This is equal to $f'_+(x_1)$ (and $f'_-(x_1)$) only for $N = 1$.

Right-sided (f'_+) and left-sided (f'_-) numerical derivatives are exact only for first order polynomials.

Try centred derivatives

$$f'_{(2)}(x_1) = \frac{f(x_2) - f(x_0)}{x_2 - x_1} = \frac{2^N - 0^N}{2} = 2^{N-1}.$$

This equals $f'(x_1)$ for both $N = 1$ and 2.

Centred ($f'_{(2)}$) numerical derivatives are exact for second order polynomials.

Comment: Can prove the above statements more generally by taking n$x_m - 1 = x_m - h$, x_m, and $x_{m+1} = x_m + h$.

Ordinary Differential Equations

Problem: The aim is to generate a numerical solution for the INITIAL VALUE PROBLEM consisting of the ordinary differential equation $(*)$ and the initial condition $(**)$

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

The numerical solution will be defined only at a discrete set of points (x_i, y_i), with $y_i \approx y(x_i)$ (where $y(x)$ is the true solution). Note that interpolation may be used to estimate the solution at an intermediate point.

Reminder: It is straightforward to find *analytic* solutions of (*) when $f(y,x)$ is separable ($f(y,x) = f_1(x)f_2(y)$, use SEPARATION) or $f(y,x)$ is linear in y, ($f(y,x) = F_1(x)y + F_2(x)$, use INTEGRATING FACTOR). Numerical solutions are often needed when $f(y,x)$ is nonlinear and non-separable.

Example 1: separation

$$\frac{dy}{dx} = x^2y^3 \qquad\qquad y(1) = 1.$$

$$\int \frac{dy}{y^3} = \int x^2 dx$$

$$-\frac{1}{2y^2} = \frac{1}{3}x^3 + c \qquad\qquad y(1) = 1, \quad \text{so } c = -\frac{5}{6}$$

$$y = \sqrt{\frac{3}{5 - 2x^3}}$$

Example 2: integrating factor

$$\frac{dy}{dx} = yx + x^3 \qquad\qquad y(0) = 1.$$

multiply equation by $q(x)$

$$q\frac{dy}{dx} - yqx = qx^3.$$

use definition of integrating factor

$$\frac{dq}{dx} = -qx, \qquad \text{solving} \qquad q(x) = \exp -x^2/2.$$

gives

$$q\frac{dy}{dx} + \frac{dq}{dx}y = \frac{d}{dx}(yq) = \frac{d}{dx}\left(ye^{-x^2/2}\right) = x^3e^{-x^2/2}.$$

integrating

$$ye^{-x^2/2} = -(2 + x^2)e^{-x^2/2} + c \qquad y(0) = 1, \quad \text{so } c = 2.$$

$$y = -2 - x^2 + 2e^{x^2/2}.$$

Method 1: Euler's Method

Aim is to get a solution on a regular grid $x_i = x_0 + ih$ (h is defined to be the stepsize). Simplest method is to use the RIGHT-SIDED numerical derivative defined above. That is, we replace $(*)$ with

$$\frac{y_{i+1} - y_i}{h} = f(y_i, x_i).$$

rearranging we get

$$y_{i+1} = y_i + hf(y_i, x_i) \qquad \text{with} \qquad y_0 = Y. \qquad \textbf{Euler's Method}$$

Euler's method is used to generate a sequence $\{y_i\}$ that gives the numerical solution of the equation at each point x_i (see diagram).

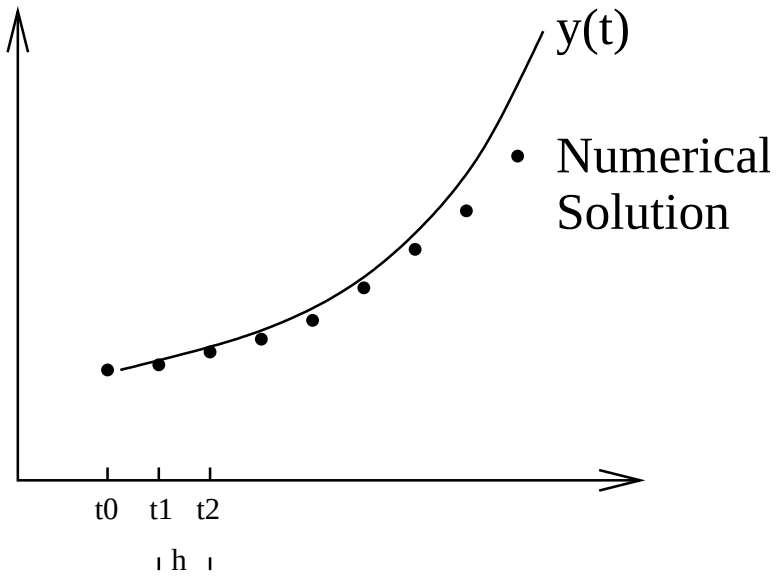

Figure: Illustrating the actual and numerical solutions of an ordinary differential equation.

Method 2: Midpoint Method, or Second-order Runge-Kutta Method (RK2)

As CENTRED numerical derivatives are more accurate than right-sided ones, these may be used to derive a more accurate method for solving the initial value problem $(*) + (**)$, i.e.

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

Problem here is that we don't know the numerical solution at $y_{i+1/2} \approx y(x_{i+1/2})$ (note that $x_{i+1/2} = x_i + h/2$). However, we can estimate this using Euler's method by setting

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

so explicitly, we have,

$$y_{i+1} = y_i + hf(y_i + \frac{h}{2} f(y_i, x_i), x_i + h/2) \quad y_0 = Y \qquad \textbf{Midpoint (RK2) Method}$$

This formula can be used to generate a sequence $\{y_i\}$, that can be shown to have higher accuracy in the limit of small h compared with Euler's method.

Systems of ODEs

Note that two coupled ordinary differential equations (+initial conditions)

$$\begin{aligned} \frac{dy}{dx} &= f(y, z, x), \qquad y(x_0) = Y \\ \frac{dz}{dx} &= g(y, z, x), \qquad z(x_0) = Z \end{aligned}$$

can be solved together straighforwardly (e.g. using Euler's method)

$$\begin{aligned} y_{i+1} &= y_i + hf(y_i, z_i, x_i), \qquad y_0 = Y \\ z_{i+1} &= z_i + hg(y_i, z_i, x_i), \qquad z_0 = Z. \end{aligned}$$

These generate sequences $\{y_i\}$ and $\{z_i\}$, the numerical solutions of the equations. The same principle can be extended to a system of N equations.

Higher order ODEs

Higher order differential equations may be solved by tranforming them into systems of first order equations like those above. We will illustrate this by considering the general second order equation (with 2 initial conditions)

$$\frac{d^2y}{dx^2} = f(\frac{dy}{dx}, y, x) \qquad y(x_0) = Y,\ \frac{dy}{dx}(x_0) = Z.$$

To solve this equation we set the auxillary variable $z = dy/dx$, and replace the above problem with two equivalent first order equations in y and z.

$$\begin{aligned} \frac{dy}{dx} &= z \qquad\quad y(x_0) = Y \\ \frac{dz}{dx} &= f(z, y, x) \qquad z(x_0) = Z. \end{aligned}$$

This is an example of a system of first order ODEs like those described above and can be solved in the usual way, e.g. using Euler's method

$$\begin{aligned} y_{i+1} &= y_i + hz_i, \qquad\quad y_0 = Y, \\ z_{i+1} &= z_i + hf(y_i, z_i, x_i), \qquad z_0 = Z. \end{aligned}$$

These generate sequences $\{y_i\}$ - the numerical solution of the equation ($y_i \approx y(x_i)$) and $\{z_i\}$, where $z_i \approx dy/dx\,(x_i)$.