Numerical Differentiation and Integration

Problem Statement: If the values of a function f are given at a few points, say, x_0, x_1, $\cdots$, x_n, we attempt to estimate a derivative $f'(c)$ or an integral $\int_a^b f(x)dx$.

- Basics of Numerical Differentiation
- Richardson Extrapolation
- Basics of Numerical Integration
- Quadrature Formulas
  - Trapezoidal Rule
  - Simpson's $\frac{1}{3}$ Rule
  - ♡ Simpson's $\frac{3}{8}$ Rule
  - ♡ Boole's Rule

Basics of Numerical Differentiation

$$f'(x) = limit_{h\to 0}\frac{f(x+h)-f(x)}{h} \tag{1}$$

$$f'(x) \approx \frac{1}{h}[f(x+h)-f(x)] \tag{2}$$

$$f(x+h) = f(x)+hf'(x)+\frac{h^2}{2}f''(\xi) \tag{3}$$

$$f'(x) = \frac{1}{h}[f(x+h)-f(x)]-\frac{h}{2}f''(\xi) \tag{4}$$

$$f'(x) \approx \frac{1}{2h}[f(x+h)-f(x-h)] \tag{5}$$

$$f(x+h) = f(x)+hf'(x)+\frac{h^2}{2}f(x)+\frac{h^3}{3!}f^{(3)}(\xi) \tag{6}$$

$$f(x-h) = f(x)-hf'(x)+\frac{h^2}{2}f(x)-\frac{h^3}{3!}f^{(3)}(\tau) \tag{7}$$

$$f'(x) = \frac{1}{2h}[f(x+h)-f(x-h)]-\frac{h^2}{12}[f^{(3)}(\xi)+f^{(3)}(\tau))] \tag{8}$$

Examples:

1. $f(x) = \cos(x)$, evaluate $f'(x)$ at $x = \pi/4$ with $h = 0.01$, $h = 0.005$

2. $g(x) = ln(1+x)$, evaluate $g'(x)$ at $x = 1$ with $h = 0.01$, $h = 0.005$.

3. $t(x) = tan^{-1}x$, evaluate $t'(x)$ at $x = \sqrt{2}$ with $h = 0.01$, $h = 0.005$.

- True Solutions

1. $f'(\pi/4) = -\sin(\pi/4) = -\frac{1}{\sqrt{2}} = -0.707106781$

2. $g'(1) = \frac{1}{1+1} = 0.500000000$

3. $t'(\sqrt{2}) = \frac{1}{1+(\sqrt{2})^2} = \frac{1}{3} = 0.333333333$

Approximation by Taylor Expansion

Theorem 1: Assume that $f \in C^3[a,b]$ and $x-h, x, x+h \in [a,b]$ with $h > 0$. Then

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h} = D_0(h) \tag{9}$$

Furthermore, $\exists c \in [a,b]$ such that

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h} + E_1(f,h) \tag{10}$$

where $E_1(f,h) = -\frac{h^2 f^{(3)}(c)}{6} = O(h^2)$ is called the truncation error.

Theorem 2: Assume that $f \in C^5[a,b]$ and $x \mp 2h, x \mp h, x \in [a,b]$ with $h > 0$. Then

$$f'(x) \approx \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h} = D_1(h) \tag{11}$$

Furthermore, $\exists c \in [a,b]$ such that

$$f'(x) \approx \frac{-f_2 + 8f_1 - 8f_{-1} + f_{-2}}{12h} + E_2(f,h) \tag{12}$$

where $E_2(f,h) = -\frac{h^4 f^{(5)}(c)}{30} = O(h^4)$

h	By Theorem 1	By Theorem 2	Richardson
0.1	-0.716161095	-0.717353703	
0.01	-0.717344150	-0.717356108	
0.001	-0.717356000	-0.717356167	
0.0001	-0.717360000	-0.717360833	
$f'(0.8)$	-0.717356091	-0.717356091	-0.717356091

Table 1: Approximating the derivative of $f(x) = \cos(x)$ at $x = 0.8$

Richardson's Extrapolation

Recall that

$$f'(x_0) \approx D_0(h) + Ch^2, \quad f'(x_0) \approx D_0(2h) + 4Ch^2 \tag{13}$$

Then

$$f'(x_0) \approx \frac{4D_0(h) - D_0(2h)}{3} \approx \frac{-f_2 + 8f_1 - 8f_{-1} + f_{-2}}{12h} = D_1(h) \tag{14}$$

Similarly,

$$f'(x_0) = \frac{-f_2 + 8f_1 - 8f_{-1} + f_{-2}}{12h} + \frac{h^4 f^{(5)}(\xi)}{30} \approx D_1(h) + Ch^4 \tag{15}$$

$$f'(x_0) = \frac{-f_4 + 8f_2 - 8f_{-2} + f_{-4}}{12h} + \frac{h^4 f^{(5)}(\tau)}{30} \approx D_1(2h) + 16Ch^4 \tag{16}$$

Then

$$f'(x_0) \approx \frac{16D_1(h) - D_1(2h)}{15} \tag{17}$$

Richardson's Extrapolation Theorem: Let $D_{k-1}(h)$ and $D_{k-1}(2h)$ be two approximations of order $O(h^{2k})$ for $f'(x_0)$, such that

$$f'(x_0) = D_{k-1}(h) + c_1 h^{2k} + c_2 h^{2k+2} + \cdots \tag{18}$$

$$f'(x_0) = D_{k-1}(2h) + 2^{2k} c_1 h^{2k} + 2^{2k+2} c_2 h^{2k+2} + \cdots \tag{19}$$

Then an improved approximation has the form

$$f'(x_0) = D_k(h) + O(h^{2k+2}) = \frac{4^k D_{k-1}(h) - D_{k-1}(2h)}{4^k - 1} + O(h^{2k+2}) \tag{20}$$

Differentiation Approximation Algorithms

- *Algorithm: Differentiation Using Limits*

  Generate the numerical sequence

$$f'(x) \approx D_j = \frac{f(x+2^{-j}h) - f(x-2^{-j}h)}{2 \times (2^{-j}h)} \quad for \quad j = 0, 1, \cdots, n$$

until $|D_{n+1} - D_n| \geq |D_n - D_{n-1}|$ or $|D_n - D_{n-1}| < tol$, a user-specified tolerance, which attempts to find the best approximation $f'(x) \approx D_n$.

- *Algorithm: Differentiation Using Extrapolation*

  To approximate $f'(x)$ by generating a table of $D(j,k)$ for $k \leq j$ using $f'(x) \approx D(n,n)$ as the final answer. The $D(j,k)$ entries are stored in a lower-Δ matrix. The first column is

$$D(j,0) = \frac{f(x+2^{-j}h) - f(x-2^{-j}h)}{2 \times (2^{-j}h)}$$

and the elements in row j are

$$D(j,k) = D(j,k-1) + \frac{D(j,k-1) - D(j-1,k-1)}{4^k - 1} \quad for \quad 1 \leq k \leq j.$$

Matlab Codes for Richardson Extrapolation

```
%
% Script file: richardson.m
% Example: format long
%          richardson(@cos,0.8,0.00000001,0.00000001)
%          richardson(@sinh,1.0,0.00001,0.00001)
%
% Richardson Extrapolation for numerical differentiation
% P.333 of John H. Mathews, Kurtis D. Fink
% Input f is the function input as a string 'f'
%       delta is the tolerance error
%       toler is the tolerance for the relative error
% Output D is the matrix of approximate derivatives
%        err is the error bound
%        relerr is the relative error bound
%        n is the coordinate of the best approximation
%
function [D, err, relerr, n]=richardson(f,x,delta,toler)
err=1.0;
relerr=1.0;
h=1.0;
j=1;
D(1,1)=(feval(f,x+h)-feval(f,x-h))/(2*h);
while (relerr>toler & err>delta & j<12)
    h=h/2;
    D(j+1,1)=(feval(f,x+h)-feval(f,x-h))/(2*h);
    for k=1:j
        D(j+1,k+1)=D(j+1,k)+(D(j+1,k)-D(j,k))/(4^k-1);
    end
    err=abs(D(j+1,j+1)-D(j,j));
    relerr=2*err/(abs(D(j+1,j+1))+abs(D(j,j))+eps);
    j=j+1;
end
[n n]=size(D);
```

Basics of Numerical Integration

- $\int_0^\infty \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx = \frac{1}{2}$
- $\int_0^2 \sqrt{1 - \frac{x^2}{4}} dx = \frac{\pi}{2}$
- $\int_0^\pi sin(x) dx = 2$
- $\int_1^4 \sqrt{x} dx = \frac{14}{3}$
- $erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2/2} dt,\ 0 < x < \infty,\ \ erf(1) = ?$
- $\eta(x) = \int_0^x \sqrt{1 + cos^2 t} dt,\ 0 < x < \pi,\ \ \eta(\pi) = ?$
- $\xi(x) = \int_0^x \frac{sin t}{t} dt,\ 0 < x < \infty,\ \ \xi(1) = ?$
- $\psi(x) = \int_0^x \sqrt{1 + \frac{t^2}{4(4-t^2)}} dt,\ 0 < x < 2,\ \ \psi(2) = ?$
- $\phi(x) = \int_0^x \frac{t^3}{e^t - 1} dt,\ 0 < x < \infty,\ \ \phi(5) = 4.8998922$

Quadrature Formulas

The general approach to numerically compute the definite integral $\int_a^b f(x)dx$ is by evaluating $f(x)$ at a finite number of sample and find an interpolating polynomial to approximate the integrand $f(x)$.

Definition: Suppose that $a = x_0 < x_1 < x_2 < \cdots < x_n = b$. A formula of the form

$$Q[f] = \sum_{i=0}^{n} \beta_i f(x_i) = \beta_0 f(x_0) + \beta_1 f(x_1) + \cdots + \beta_n f(x_n) \tag{21}$$

such that

$$\int_a^b f(x)dx = Q[f] + E[f]$$

is called a quadrature formula. The term $E[f]$ is called the truncation error for integration. The values $\{x_j\}_{j=0}^n$ are called the quadrature nodes, and $\{\beta_j\}_{j=0}^n$ are called the weights.

- *Closed Newton-Cotes Quadrature Formula*

  Assume that $x_i = x_0 + i \cdot h$ are equally spaced nodes and $f_i = f(x_i)$. The first four closed Newton-Cotes quadrature formula are

$$(trapezoidal\ rule) \int_{x_0}^{x_1} f(x)dx \approx \frac{h}{2}(f_0 + f_1) \tag{22}$$

$$(Simpson's\ \frac{1}{3}\ rule) \int_{x_0}^{x_2} f(x)dx \approx \frac{h}{3}(f_0 + 4f_1 + f_2) \tag{23}$$

$$(Simpson's\ \frac{3}{8}\ rule) \int_{x_0}^{x_3} f(x)dx \approx \frac{3h}{8}(f_0 + 3f_1 + 3f_2 + f_3) \tag{24}$$

$$(Boole's\ rule) \int_{x_0}^{x_4} f(x)dx \approx \frac{2h}{45}(7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4) \tag{25}$$

Approximating Integrals by Trapezoidal and Simpson's Rules

Example: The arc length of a curve $f(x) = \frac{2}{3}x^{3/2}$ between $x \in [0,1]$ can be computed by

$$\alpha = \int_0^1 \sqrt{1+x}dx = \frac{2}{3}(2\sqrt{2}-1) \;=\; \mathbf{1.21895142}$$

• **Trapezoidal Rule**

$$\alpha \approx \frac{h}{2}[f(x_0) + 2f(x_1) + 2f(x_2) + \cdots + 2f(x_{n-1}) + f(x_n)]$$

where $0 = x_0 < x_1 < x_2 < \cdots < x_{n-1} < x_n = 1, \;\; h = \triangle x_i = x_{i+1} - x_i \;\; \forall \;\; 0 \le i \le n-1$

$$\alpha \approx 1.21894\mathbf{4654}, \;\; when\ n = 50; \;\; \alpha \approx 1.21895\mathbf{020}, \;\; when\ n = 100; \;\; \alpha^* = 1.21895\mathbf{142}$$

• **Simpson's 1/3 Rule**

$$\alpha \approx \frac{h}{3}[f_0 + 4f_1 + 2f_2 + 4f_3 + 2f_4 + \cdots + 2f_{2m-2} + 4f_{2m-1} + f_{2m}]$$

where $0 = x_0 < x_1 < x_2 < cdots < x_{n-1} < x_{2m} = 1, \;\; h = \triangle x_i = x_{i+1} - x_i \;\; \forall \;\; 0 \le i \le 2m-1$

$$\alpha \approx 1.21895\mathbf{133}, \;\; when\ n = 12; \;\; \alpha \approx 1.21895\mathbf{140}, \;\; when\ n = 20; \;\; \alpha^* = 1.21895\mathbf{142}$$

Matlab Codes for Simpson's 1/3 Rule

```
%
% Simpson.m - Simpson's 1/3 Rule for \sqrt(1+x)
%
format long
n=20;
h=1/n;
x0=0;  x1=x0+h;  x2=x1+h;
s=0;
for i=0:2:n-2,
  f0=sqrt(1+x0);
  f1=sqrt(1+x1);
  f2=sqrt(1+x2);
  s=s+f0+4*f1+f2;
  x0=x2;
  x1=x2+h;
  x2=x2+h+h;
end
s=h*s/3.0;
'Simpson Approximated Arc length is', s
```