Math 611 Numerical Discretization

Professor - Dr. Constantin Bacuta

Spring 2019

Contents

1 Interpolation

Main Idea- Approximate complicated functions by simple functions such as polynomials, piecewise polynomials, etc.

Example 1.

$$e^{x^2} \approx 1 + x^2 + \frac{x^4}{2}$$

1.1 Polynomial Interpolation

Let $\{x_0, \dots, x_n\}$ be $n+1$ real numbers and $\{y_0, \dots, y_n\}$ are the associated function values, i.e. $f(x_i) = y_i$. We can build a polynomial of degree n such that $P(x_i) = y_i$, $\forall i \in \{0, 1, \dots, n\}$.

$$\begin{aligned} P(x_0) &= a_0 + a_1 x_0 + \cdots + a_n x_0^n \\ P(x_1) &= a_0 + a_1 x_1 + \cdots + a_n x_1^n \\ &\vdots \\ P(x_n) &= a_0 + a_1 x_n + \cdots + a_n x_n^n \end{aligned}$$

We get an $(n+1) \times (n+1)$ linear system. This system will be of the form $V_n \mathbf{a} = \mathbf{y}$ where V_n is the Vandermonde matrix and $\mathbf{a}$ is the vector of coefficients.

$$V_n = \begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & & & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix}$$

One property of the Vandermonde is the determinant is relatively easy to compute.

$$det(V_n) = \prod_{0 \le i < j \le n} (x_j - x_i)$$

Note that the determinant is non zero provided that $x_i \neq x_j$ when $i \neq j$. With this we have that there exists a unique polynomial that will interpolate the data at the given nodes.

1.2 Lagrange form

Given data $\{x_0, \dots, x_n\}$, $\{y_0, \dots, y_n\}$ with x_i distinct, we define the following functions.

$$L_{0,n} = \frac{(x - x_1)(x - x_2) \dots (x - x_n)}{(x_0 - x_1)(x_0 - x_2) \dots (x_0 - x_n)}$$

$$L_{1,n} = \frac{(x - x_0)(x - x_2) \dots (x - x_n)}{(x_1 - x_0)(x_1 - x_2) \dots (x_1 - x_n)}$$

In general

$$L_{i,n} = \frac{(x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)}$$

Note that $L_{i,n}(x_j) = \delta_{ij}$. Therefore we have

$$P_n(x) = \sum_{i=0}^{n} L_{i,n}(x) y_i$$

Example 2. The linear interpolant for two points $\{x_0, x_1\}$, $\{y_0, y_1\}$.

$$P_1(x) = y_0 \frac{x - x_1}{(x_0 - x_1)} + y_1 \frac{x - x_0}{x_1 - x_0}$$

The quadratic interpolant on $x = \{-1, 0, 1\}$, $f(x) = \{2, 1, 3\}$ is

$$\begin{aligned} P_2(x) &= \frac{(x)(x-1)}{(-1)(-1-1)}2 + \frac{(x-1)(x+1)}{(0+1)(0-1)}1 + \frac{(x+1)(x)}{(1+1)(1+-)}3 \\ &= \frac{3}{2}x^2 + \frac{1}{2}x + 1 \end{aligned}$$

In Matlab:

```
x = [-1 0 1];
y = [2 1 3];
p = polyfit(x,y,2)
```

Theorem 1. Let $f \in C^{(n+1)}([a, b])$ and let $x_0, x_1, \ldots, x_n \in [a, b]$ be distinct and let $P = P_n = \prod_n(f)$ be the interpolant on $x_0, \ldots, x_n$. Then for every $x \in [a, b]$, $\exists \xi_x \in (a, b)$ such that

$$f(x) - P(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{i=0}^{n} (x - x_i) \tag{1}$$

Proof. If $x = x_i$ for $i = 0, 1, \ldots, n$ then both sides of (1) are zero. Assume that $x \neq x_i$ for $i = 0, 1, \ldots, n$. Define

$$F(t) = f(t) - P(t) - \lambda \omega_{n+1}(t)$$

where λ is a constant such that $F(x) = 0$. Then

$$F(x) = f(x) - P(x) - \lambda \omega_{n+1}(x) \qquad \Rightarrow \qquad \lambda = \frac{f(x) - P(x)}{\omega_{n+1}(x)}$$

Clearly $F \in C^{(n+1)}([a,b])$, $F(x) = 0$ and $F(x_i) = 0$ for $i = 0, 1, \dots, n$. By Rolle's theorem

$$F' \text{ has n+1 distinct roots in } (a,b)$$
$$F'' \text{ has n distinct roots in } (a,b)$$
$$\vdots$$
$$F^{(n+1)} \text{ has at least one root in } (a,b),\ \xi_x$$

so $F^{(n+1)}(\xi_x) = 0$. Note that

$$F^{(n+1)}(t) = f^{(n+1)}(t) - P^{(n+1)}(t) - \lambda\omega_{n+1}^{(n+1)}(t)$$

but as P is degree less than or equal to n and the n^{th} derivative of ω_{n+1} is $(n+1)!$. Hence

$$0 = F^{(n+1)}(\xi_x) = f^{(n+1)}(\xi_x) + \lambda(n+1)! \tag{2}$$

Therefore from (1) and (2) we have

$$f^{(n+1)}(\xi_x) = \frac{(f(x) - P(x))(n+1)!}{\omega_{n+1}(x)}$$

□

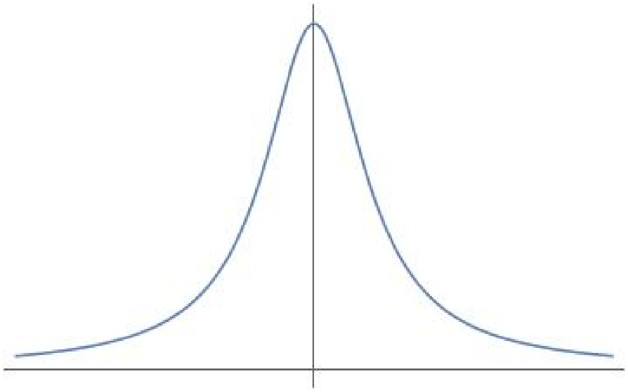
Corollary 1.

$$|f(x) - P_n(x)| \le \frac{1}{(n+1)!} \sup_{x\in[a,b]} \left|f^{(n+1)}(x)\right| \sup_{x\in[a,b]} \left|\omega_{n+1}(x)\right|$$

Example 3. Let $f : [0,1] \to \mathbb{R}$, where $f(x) = \sin(x)$ and $x_0, \dots, x_9 \in [0,1]$ be distinct. Prove that $|\sin(x) - P_n(x)| \le \frac{1}{10!} < 2.8 \times 10^{-7}$.
Solve using Corollary 1.

$$|\sin(x) - P_9(x)| \le \underbrace{\sup_{x\in[0,1]} |\sin^{10}(x)|}_{\le 1} \underbrace{\sup_{x\in[0,1]} |\omega_{10}(x)|}_{\le 1} \frac{1}{(9+1)!} \le \frac{1}{10!}$$

Main Idea: If f is a smooth given function on $[a,b]$ and we increase the number of uniformly distributed nodes in $[a,b]$. Then $P_n(x) - f(x)$ can be very large for some values $x \in [a,b]$. Take Runge's example $f(x) = (1+x^2)^{-1}$ for $x \in [-5,5]$.



Note It can be proved that for $|x| > 3.63$ with $x \neq x_i$ that

$$\lim_{n\to\infty} |f(x) - P_n(x)| \to \infty$$

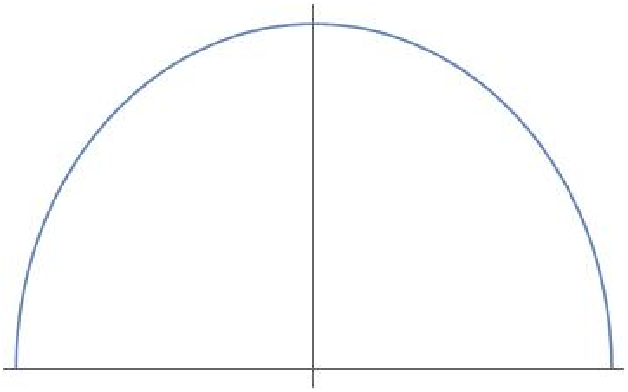
How do we fix this?

- Don't use too many equally spaced nodes on a fixed interval $[a,b]$.

- Use nodes such that $\sup_{x\in[a,b]} |\omega_{n+1}(x)|$ is minimal.

1.3 Chebyshev Points

For $[0, 1]$ we discretize using equal length arcs on a semi circle.



so

$$x_i = \cos\left(\frac{i\pi}{n}\right), \qquad i = 0, 1, \dots, n$$

To get corresponding points on $[a, b]$ we use the image of the linear function $\varphi : [-1, 1] \to [a, b]$ where $\varphi(-1) = a$ and $\varphi(1) = b$. Use the interpolant to get

$$\varphi(x) = \frac{x-1}{-1-1}a + \frac{x+1}{1--1}b = \frac{a+b}{2} + \frac{b-a}{2}x$$

Therefore, the Chebyshev nodes for $[a, b]$ are given by

$$x_i = \frac{a+b}{2} + \frac{b-a}{2}\cos\left(\frac{i\pi}{n}\right), \qquad i = 0, 1, \dots, n$$

1.4 Newton's Form

Given $\{x_0, x_1, \dots, x_n\}$ and $\{y_0, y_1, \dots, y_n\}$ where $y_i = f(x_i)$ we look for the interpolant P_n written

$$P_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n(x - x_0)\dots(x - x_{n-1})$$

where we determine a_i by imposing condition of $P_n(x_i) = y_i$. With this we get

$$a_0 = f(x_0)$$

$$a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

$$a_2 = \frac{\frac{f(x_2)-f(x_1)}{x_2-x_1} - \frac{f(x_1)-f(x_0)}{x_1-x_0}}{x_2 - x_0}$$

Define the divided difference.

$$f[x_i] = f(x_i) \qquad \text{order zero}$$

$$f[x_i, x_j] = \frac{f[x_j] - f[x_i]}{x_j - x_i} \qquad \text{order one}$$

$$f[x_i, x_j, x_k] = \frac{f[x_j, x_k] - f[x_i, x_j]}{x_k - x_i} \qquad \text{order two}$$

$$\vdots$$

$$f[x_0, x_1, \dots, x_k] = \frac{f[x_1, \dots, x_k] - f[x_0, \dots, x_{k-1}]}{x_k - x_0} \qquad \text{order k}$$

Claim 1. $a_k = f[x_0, x_1, \dots, x_k]$ for any $k+1$ points.

Proof. By induction clearly $a_0 = f[x_0]$. Need to prove that $a_{k+1} = f[x_0, \dots, x_{k+1}]$. Let b_k be the coefficient of x^k in the interpolant Q_k for the $(k+1)$ points $x_1, x_2, \dots, x_{k+1}$. Using the induction hypothesis we have that $b_k = f[x_1, x_2, \dots, x_{k+1}]$. We claim that the interpolant is given by

$$P_{k+1}(x) = Q_k(x) + \frac{x - x_{k+1}}{x_{k+1} - x_0}(Q_k(x) - P_k(x)) = \tilde{P}_{k+1}$$

Note

- P_{k+1} is a polynomial of degree less than or equal to $k+1$.

- $\tilde{P}_{k+1}(x_0) = Q_k(x_0) - (Q_k(x_0) - P_k(x_0)) = f(x_0)$.

- $\tilde{P}_{k+1}(x_i) = f(x_i)$ for $i = 1, \dots, k$.

- $\tilde{P}_{k+1}(x_{k+1}) = f(x_{k+1})$

Therefore from the claim =, the coefficient of x^{k+1} in P_{k+1} i.e. a_{k+1} is the coefficient of x^{k+1} in $\tilde{P}_{k+1}$ which is

$$a_{k+1} = \frac{1}{x_{k+1} - x_0}(f[x_1, \dots, x_{k+1}] - f[x_0, \dots, x_k]) = f[x_0, \dots, x_{k+1}]$$

□

Therefore we have the Newton form of the interpolant is given by

$$P_n = \sum_{k=0}^{n} a_k(x - x_0)\dots(x - x_{k-1})$$

Example 4. Divided difference table for $n = 3$.

x_0	$f[x_0]$	$f[x_0, x_1]$	$f[x_0, x_1, x_2]$	$f[x_0, x_1, x_2, x_3]$
x_1	$f[x_1]$	$f[x_1, x_2]$	$f[x_1, x_2, x_3]$	
x_2	$f[x_2]$	$f[x_2, x_3]$		
x_3	$f[x_3]$			

Progression from left to right, bottom to top. In Matlab we use dividif(x,y) and the diagonal will be the a_k.

1.5 Hermite Interpolation

Main Idea Given the nodes $x_0, \dots, x_n \in [a, b]$ and a smooth function $f : [a, b] \to \mathbb{R}$, look for a polynomial of least degree that interpolates $f, f', \dots, f^{(n)}$.

$$P(x_i) = f(x_i) = y_i, \qquad i = 0, 1, \dots, n \tag{3}$$

$$P'(x_i) = f'(x_i) = y_i^(1), \qquad i = 0, 1, \dots, n \tag{4}$$

So we have $n+2$ condition, so look for a polynomial of degree less than or equal to $2n+1$.

Theorem 2. There exists a unique polynomial of degree less than or equal to $2n+1$ that satisfies (3) and (4).

Consider the Lagrange form of P_{2k+1}.

$$L_i = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}, \qquad L_i(x_j) = \delta_{ij}$$

For Hermite, look for polynomials A_i, B_i of degree less than or equal to $2n+1$ such that

$$A_i(x_j) = \delta_{ij}, \qquad A_i'(x_j) = 0 \tag{5}$$

$$B_i(x_j) = 0, \qquad B_i'(x_j) = \delta_{ij} \tag{6}$$

Assuming that we have A_i, B_i that satisfy (5) and (6) then

$$P_{2n+1}^H(x) = \sum_{i=0}^{n} A_i(x) f(x_i) + \sum_{i=0}^{n} B_i(x) f'(x_i)$$

where

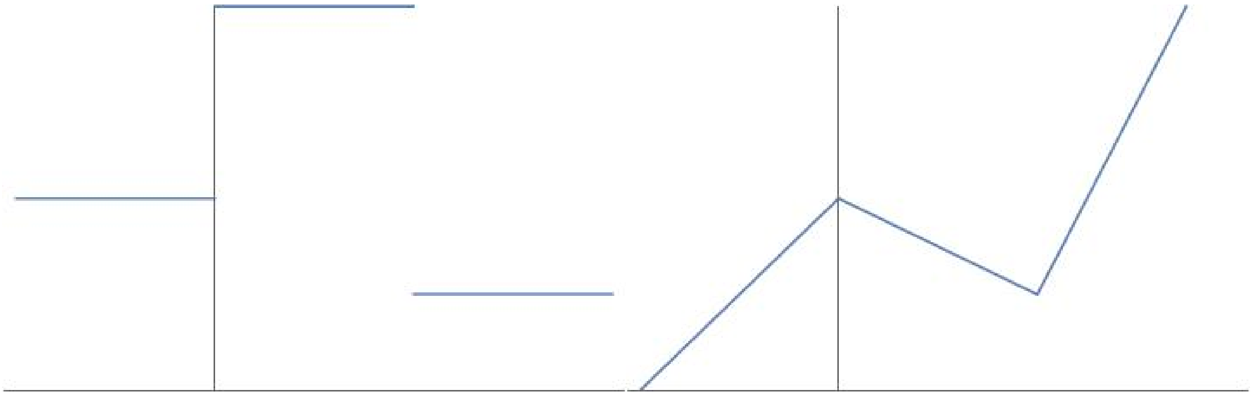
$$A_i(x) = (1 - 2(x - x_i) L_i'(x_i)) L_i^2(x)$$
$$B_i(x) = (x - x_i) L_i^2(x)$$

2 Splines

Definition 1. Let $a = x_0 < x_1 < \cdots < x_n = b$. A function $S = S_k(x)$ is a spline of degree k relative to the nodes if

- On each $[x_{i-1}, x_i]$ S is a polynomial of degree less than or equal to k.
- $S \in C^{(k-1)}([a, b])$

Example 5. $k = 0$ and $k = 1$



If $S(x) = a_j x + b_j$ for $x \in [x_{j-1}, x_j]$ $j = 1, \ldots, n$ then we need to impose that S is continuous at $x_1, \ldots, x_{n-1}$.

$$S(x_j - 0) = S(x_j + 0) = y_j, \qquad j = 1, \ldots, n$$

$$S(x) = \frac{x_j - x}{h_j} y_j + \frac{x - x_{j-1}}{h_j} y_j$$

where $h_j = x_j - x_{j-1}$. S is uniquely determined.

2.1 Cubic Splines

Let $a = x_0 < x_1 < \dots < x_n = b$. S is a cubic spline if S is a polynomial of degree less than or equal to 3 on each $[x_{k-1}, x_k]$ and $S \in C^2$.

If we denote $M_k = S''(x_k)$ and $f_k = S(x_k)$ and assume that S is a cubic spline then S'' is a continuous piecewise linear function(spline of order 1). By previous example we have

$$S''(x) = M_{k-1}\frac{x_k - x}{h_k} + M_k\frac{x - x_{k-1}}{h_k}$$

Integrating twice yields

$$S(x) = M_{k-1}\frac{(x_k - x)^3}{6h_k} + M_k\frac{(x - x_k)^3}{6h_k} + C_{k-1}(x - x_{k-1}) + \tilde{C}_{k-1}$$

where C and $\tilde{C}$ are determined by imposing $S(x_{k-1}) = f_{k-1}$ and $S_k = f_k$. Sovling for these will give

$$\tilde{C}_{k-1} = f_{k-1} - \frac{M_{k-1}h_k^2}{6}$$

$$C_{k-1} = f_k - \frac{f_{k-1}}{h_k} - \frac{h_k(M_k - M_{k-1})}{6}$$

As S is a spline we also have that S' is continuous at interior nodes, i.e. $S'(x_k - 0) = S'(x_k + 0)$ which leads to

$$\frac{h_k}{6}M_{k-1} + \frac{h_k}{3}M_k + \frac{f_k - f_{k-1}}{h_{k+1}} = -\frac{h_{k+1}}{3}M_k - \frac{h_{k+1}}{6}M_{k+1} + \frac{f_{k+1} - f_k}{h_{k+1}} = S'(x_k)$$

which is a relation in M_{k-1}, M_k, M_{k+1}. We put this in matrix form using μ, λ, d.

$$\begin{bmatrix} \mu_1 & 2 & \lambda_1 & 0 & \dots & 0 \\ 0 & \mu_2 & 2 & \lambda_2 & \dots & 0 \\ \vdots & & \ddots & \ddots & & \\ 0 & \dots & & \mu_{n-1} & 2 & \lambda_{n-1} \end{bmatrix} \begin{bmatrix} M_0 \\ M_1 \\ \ddots \\ M_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \ddots \\ d_{n-1} \end{bmatrix}$$

Notice that we require two more conditions $2M_0 + \lambda_0 M_0 = d_0$ and $\mu_n M_{n-1} + 2M_n = d_n$ with $0 \leq \lambda_0, \mu_n \leq 1$ and d_0, d_n are given.

2.1.1 Natural Spline

One choice is to force $\lambda_0 = \mu_n = d_0 = d_n = 0$. This is know as the natural spline and this forces $S''(a) = S''(b) = 0$.

2.1.2 Not-a-knot Spline

Impose S''' be continuous at x_1, x_{n-1}. Note that since S, S', S'', S''' are continuous at x_1 then there exists a unique polynomial of degree less than or equal to 3 with prescribed values $P(x_1), P'(x_1), P''(x_1), P'''(x_1)$. Thus S is the same polynomial of degree 3 on $[x_0, x_2]$ and similarly on $[x_{n-2}, x_n]$. From the computational point of view

$$S''' = -\frac{M_{k-1}}{h_k} + \frac{M_k}{h_k} = \frac{M_k - M_{k-1}}{h_k}, \qquad x \in [x_{k-1}, x_k]$$

Imposing S''' to be continuous at x_1, x_{n-1} results in

$$P(x_1) = \frac{M_2 - M_1}{h_2} \qquad P(x_{n-1}) = \frac{M_n - M_{n-1}}{h_n}$$

Matlab To do splines in Matlab use spline().

```
x = [x_0,...,x_n]
y = [y_0,...,y_n]

pp = spline(x,y)
% builds info about not-a-knot spline that interpolates data pp.coef nx4
% matrix with coefficients on each n intervals

%To get evaluations on grid points z = a:h:b the
s = ppval(x,y,z)
%or
s = spline(x,y,z)
```

If you want a spline that has $f'(a)$ and $f'(b)$ given, then replace the y vector with $[f'(a), y, f'(b)]$.

2.2 Properties of Spline

Theorem 1. (Minimum norm peroperty) Let $f \in C^2([a,b])$ and S the natural cubic spline interpolating f on $a = x_0 < x_1 < \cdots < x_n = b$. Then

$$\int_a^b (S''(x))^2 \, dx \leq \int_a^b (f''(x))^2 \, dx$$

with equality iff $f(x) = S(x)$.

Theorem 2. Let $f \in C^2([a,b])$ i.e. $|\int_a^b f \, dx| < \infty$. Let S_f be the spline interpolating f on $a = x_0 < x_1 < \cdots < x_n = b$ such that $S_f'(a) = f'(a)$ and $S_f'(b) = f'(b)$, then

$$\int_a^b (f'' - S_f'')^2 \, dx \leq \int_a^b (f'' - S'')^2 \, dx$$

for any cubic spline S interpolating $(x, f(x))$.

2.3 Parametric Spline

Main idea Use the spline interpolant for $(t, x(t))$ denoted S_x and likewise for $(t, y(t))$ denoted S_y. This gives us a parametric spline $S = (S_x(t), S_y(t))$. Note that S depends upon the parametrization of the curve and also on the magnitude $(x'(t))^2 + (y'(t))^2$. To minimize $(x')^2 + (y')^2$ we use the arclength parametrization

$$S = S(t) = \int_a^t \sqrt{x'(\tau)^2 + y'(\tau)^2} \, d\tau$$

Matlab Given $a = t_0 < t_1 < \cdots < t_n = b$ define $0 = s_0 < s_1 < \cdots < s_n$ by

```
s_0 = 0;
for i = 0:1:n
    s_{i+1} = s_i + sqrt((x_{i+1}-x_i)^2 + (y_{i+1}-y_i)^2)
end
h = mesh size
Sh = 0:h:s_n
Sxh = spline(s,x,sh)
Syh = spline(s,y,sh)
```

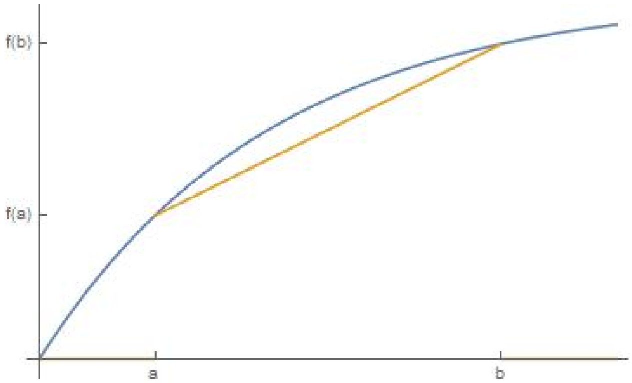
Note that matlab indexing begins at 1 so to implement the loop, change the indexing.

3 Numerical Integration

Goal: Approximate $\int_a^b f(x)\,dx$ when $\int f(x)\,dx$ is not available.
Main Idea: Approximate f by the interpolant P.

$$\int_a^b f(x)\,dx \approx \int_a^b P(x)\,dx$$

3.1 Trapezoid Formula



Use the linear interpolant on $\{a,b\}$, $\{f(a), f(b)\}$ to get

$$P_1(x) = \frac{x-b}{a-b}a + \frac{x-a}{b-a}b$$

with error

$$f(x) = P_1(x) + \frac{(x-a)(x-b)}{2}f''(c_x)$$

So,

$$\int_a^b f(x)\,dx = \underbrace{\int_a^b P_1(x)\,dx}_{I_1(f)}\underbrace{\int_a^b \frac{(x-a)(x-b)}{2}f''(c_x)\,dx}_{E_1(f)}$$

The approximating gives

$$\int_a^b f(x)\,dx \approx \int_a^b P_1(x)\,dx = (b-a)\frac{f(a)+f(b)}{2}$$

We can then get error $E_1(f)$ in terms of $h = b-a$.

$$\begin{aligned}
E_1(f) &= \int_a^b \frac{(x-a)(x-b)}{2}f''(c_x)\,dx \\
&= f''(c_x)\int_a^b \frac{(x-a)(x-b)}{2}\,dx \\
&= -f''(\xi)\frac{(b-a)^3}{12} \\
&= -f''(\xi)\frac{h^3}{12}
\end{aligned}$$

This will give the trapezoid rule.

$$\int_a^b f(x)\,dx = \underbrace{(b-a)\frac{f(a)+f(b)}{2}}_{I_1(f)} - \underbrace{f''(\xi)\frac{(b-a)^3}{12}}_{E_1(f)}, \qquad \xi \in [a,b], f \in C^2$$

The trapezoid quadrature is given by just $I_1(f)$. Note that if $|f''(x)| \leq M$ on $[a,b]$ then $E_1(f) = O(h^3)$. Also note that the quadrature is exact for polynomials of degree less than or equal to 1.

3.2 (Cavaleri) Simpson Formula

We use $P_2(x)$ which is the order 2 interpolant of f on $\{a = x_0, x_1, x_2 = b\}$.

$$f(x) = P_2(x) + \frac{(x-x_0)(x-x_1)(x-x_2)}{3!} f''(c_x)$$

Simpsons rule is then given by

$$\int_a^b f(x)\,dx = \underbrace{\int_a^b P_2(x)\,dx}_{I_2(f)} + \underbrace{\int_a^b \frac{(x-x_0)(x-x_1)(x-x_2)}{6} f'''(c_x)\,dx}_{E_2(f)}$$

As before we can compute $E_2(f)$ in terms of h.

$$E_2(f) = -\frac{h^5 f^{(4)}(\xi)}{90}$$

If $f^{(4)}$ is bounded then error is $O(h^5)$. Thus, the quadrature is given by

$$\begin{aligned}\int_a^b f(x)\,dx &\approx I_2(f)\\ &= \frac{h}{3}(y_0 + 4y_1 + y_2)\\ &= \frac{b-a}{6}(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b))\end{aligned}$$

Simpson's quadrature is exact for polynomials of degree less than or equal to 3.

Example 6. Use Simpson quadrature to approximate $\int_1^2 \ln(x)\,dx$ and estimate error. $\{1, 3/2, 2\}$ and $\{0, \ln(3/2), \ln(2)\}$. So

$$\begin{aligned}\int_1^2 \ln(x)\,dx &\approx I_2(\ln(x))\\ &= \frac{0.5}{3}(4\ln(3/2) + \ln(2))\\ &= \frac{1}{6}(\ln(9/4) + \ln(2))\\ &\approx 0.3858\end{aligned}$$

For an error bound we use $f(x) = \ln(x)$ then $f^{(4)} = -6x^{-4}$ so

$$|E_2(f)| \leq |h^5/90| \max |f^{(4)}(x)| = 6|h^5/90| = 0.0021$$

3.3 General Quadrature and degree of precision/exactness

If $a \leq x_0 < x_1 < \cdots < x_n \leq b$ are $(n+1)$ distince points in $[a, b]$ and $w_0, w_1, \ldots, w_n$ are real numbers then a general (Lagrange) interpolation quadrature is of the form

$$I_n(f) = \sum_{k=0}^n w_k f(x_k)$$

For Hermite quadrature we require another set of weights $w_i^{(1)}$.

$$H_n(f) = \sum_{k=0}^n w_k f(x_k) + \sum_{k=0}^n w_k^{(1)} f'(x_k)$$

Definition 2. The degree of precision/exactness of a general quadrature $I_n(f)$ is r(positive integer) if

$$I_n(f) = I(f)$$

for all polynomials with degree less than or equal to r and $I_n(f) \neq I(f)$ for at least one polynomial of degree $r+1$. For example Trapezoid has D.O.E. 1 and Simpson has D.O.E. 3. Note that we only need to check for $1, x, x^2, \ldots$.

Example 7. Find the D.O.E. for $Q(f) = (b-a)f((a+b)/2)$.

$$\int_a^b 1\,dx = b-a \qquad Q(1) = b-a$$

$$\int_a^b x\,dx = \frac{b^2-a^2}{2} \qquad Q(x) = (b-a)((a+b)/2) = \frac{b^2-a^2}{2}$$

$$\int_a^b x^2\,dx = \frac{b^3-a^3}{3} \qquad Q(x^2) = (b-a)((a+b)/2)^2$$

Note the last line is not equal so we have D.O.E. of 1.

Example 8. For $\int_{-1}^1 f\,dx \approx Af(-1) + Bf(0) + Cf(1)$ find A, B, C such that the D.O.E. is at least 2. What is the D.O.E.? Use the method of undetermined coefficients.

$$f = 1 \qquad \int_{-2}^2 1\,dx = 4 = A+B+C$$

$$f = x \qquad \int_{-2}^2 x\,dx = 0 = -A+C$$

$$f = x^2 \qquad \int_{-2}^2 x^2\,dx = 16/3 = A+C$$

Solving for A, B, C gives $A = C = 8/3$ and $B = -4/3$. To get the D.O.E. we check $x^3, \ldots$. $f = x^3$ will work but x^4 will not. Therefore D.O.E. is 3.

3.4 Composite Formulas

Given $a \le x_0 < x_1 < \cdots < x_n \le b$ on each panel $[x_k, x_{k+1}]$ apply the same quadrature formula.

3.4.1 Composite Trapezoidial Rule(CTR/CTF)

Given $\{a = x_0, x_1, \ldots, x_m = b\}$ and $\{y_0, \ldots, y_m\}$ uniformly distributed points with size $h = (b-a)/m = x_i - x_{i-1}$ then on each $[x_i, x_{i+1}]$ apply the trapezoid rule.

$$\int_{x_i}^{x_{i+1}} f(x)\,dx = \frac{h}{2}(y_i + y_{i+1}) - \frac{h^3}{12}f''(c_{i+1}), \qquad c_{i+1} \in (x_i, x_{i+1})$$

Take the sum.

$$\begin{aligned}
\int_a^b f(x)\,dx &= \sum_{i=0}^{m-1} \int_{x_i}^{x_{i+1}} f(x)\,dx \\
&= \frac{h}{2}(y_0 + y_1) + \frac{h}{2}(y_1 + y_2) + \cdots + \frac{h}{2}(y_{m-1} + y_m) \\
&- \frac{h^3}{12}(f''(c_1) + \cdots + f''(c_m)) \\
&= \frac{h}{2}(y_0 + y_m + 2(y_1 + \cdots + y_{m-1})) - \frac{h^3}{12}\left(\underbrace{\frac{f''(c_1) + \cdots + f''(c_m)}{m}}_{=f''(c)\in[a,b]\text{by MVT}} \frac{b-a}{h}\right)
\end{aligned}$$

Therefore CTR is given by

$$\int_a^b f(x)\,dx = \underbrace{\frac{h}{2}\left(y_0 + y_m + 2\sum_{i=1}^{m-1} y_i\right)}_{I_{1,m}(f)} - \underbrace{\frac{(b-a)h^2}{12} f''(c)}_{E_{1,m}(f)}$$

3.4.2 Order of approximation

Assume that $I(f) \approx Q(f)$ and $E(f) = |I(f) - Q(f)| = O(h^\alpha)$. We find α by considering a function f such that the integral is available. Choose $h = (b-a)/2^n$, $n = 1, 2, \dots, 10$. Compute $E(h)/E(h/2) \approx 2^\alpha$. Therefore $\log_2(E(2^n)/E(2^{n+1})) \approx \alpha$.

Composite Simpson Rule(CSR)

Idea Use the quadratic interpolant on each pair of subintervals.
Split $[a, b]$ in $2m$ subintervals, $h = (b-a)/2m$ $[x_i, x_{i+2}]$ is a panel. A generic panel is given by

$$[x_{2i}, x_{2i+1}, x_{2i+2}]$$

where $x_i = a + hi$, $i = 0, \dots, 2m$. In the same fashion as CTR we have

$$\int_{x_{2i}}^{x_{2i+2}} f(x)\,dx = \frac{h}{3}(y_{2i} + 4y_{2i+1} + y_{2i+2}) - \frac{h^5 f^{(4)}(c_{i+1})}{90}, \qquad i = 0, 1, \dots, m-1$$

Sum these up to get

$$\int_a^b f(x)\,dx = \frac{h}{3}\left(y_0 + y_{2m} + 4\sum_{i=1}^{m} y_{2i-1} + 2\sum_{i=1}^{m-1} y_{2i}\right) - \frac{(b-a)h^4 f^{(4)}(c)}{180}$$

The D.O.E. for CSR is 3.

Example 9. Find the number of panels necessary for CSR to approximate $\int_0^\pi \sin^2(x)\,dx$ with 8 decimal places. Need to find m or $h = (b-a)/2m$.

$$E(h) = \frac{(b-a)h^4}{180}|f^{(4)}(c)| \leq 8\frac{(b-a)h^4}{180} < 0.5 \times 10^{-8}$$

3.5 Gaussian Quadrature

Goal: Approximate $\int_a^b f(x)\,dx$ using the best placement of n nodes $a \leq x_1 < x_2, \cdots < x_n \leq b$.
Findings:
Gauss - By special placement of the nodes then the D.O.E. can be increased.
Legendre - The special nodes are the roots of the Legendre polynomials for $[-1, 1]$.

3.5.1 Legendre Polynomials

The Legendre polynomials are given by

$$q_0(x) = 1 \qquad q_1(x) = x$$

$$q_n(x) = \frac{2n-1}{n} q_{n-1}(x) - \frac{n-1}{n} q_{n-2}(x)$$

Roots for $n = 2, 3$. When $n = 2$ we have $q_2(x) = 1/2(3x^2 - 1)$ by simple calculation use the formula above. Thus $q_2(x) = 0$ when $x = \pm 1/\sqrt{(}3)$. For $n = 3$ the roots are $x = 0$ and $x = \pm\sqrt{3/5}$.
Facts

- Rodrigues' formula - $q_j(x) = \frac{1}{2^j j!}\frac{d^j}{dx^j}[(x^2-1)^j]$

- q_i are orthogonal in $L^2[-1,1]$, i.e.

$$\int_{-1}^{1} q_i q_j \, dx = \begin{cases} 0 & i \neq j \\ \frac{2}{2j+1} & i = j \end{cases}$$

- q_n has degree n and exactly n roots in $(-1,1)$

3.5.2 The quadrature

Let $x_1, \dots, x_n$ be the roots of $q_n(x)$ on $(-1,1)$. Define the weights $w_1, \dots, w_n$ by using the cardinal function:

$$w_j = \int_{-1}^{1} L_j(x) \, dx$$

The **Gaussian quadrature** (GQ) with n nodes is

$$\int_{-1}^{1} f(x) \, dx \approx \sum_{i=1}^{n} w_i f(x_i) = \int_{-1}^{1} P_n(x) \, dx$$

Where P_n is the interpolant on the Gaussian nodes.

Example 10. $n = 2$ $q_2 = 1/2(3x^2 - 1)$ and roots are $\pm 1/\sqrt{3}$.

$$\int_{-1}^{1} f(x) \, dx \approx w_1 f(-1\sqrt{3}) + w_2 f(1/\sqrt{3})$$

$$w_1 = \int_{-1}^{1} \frac{x - x_2}{x_1 - x_2} \, dx \qquad w_2 = \int_{-1}^{1} \frac{x - x_1}{x_2 - x_1} \, dx$$

To get w_i we can do the method above or use the method of undetermined coefficients and Impose that G_2 is exact for $f = 1, x$. Using the method of undetermined coefficients we have

$$\int_{-1}^{1} 1 \, dx = 2 \Rightarrow w_1 + w_2 = 2$$

$$\int_{-1}^{1} x \, dx = 0 \Rightarrow w_1(-1/\sqrt{3}) + w_2(1/\sqrt{3}) = 0$$

These conditions give $w_1 = w_2 = 1$. It is easily shown by direct computation that G_2 is exact for $f = x^2$ and $f = x^3$ but not x^4. Thus D.O.E. for G_2 is 3. Likewise the D.O.E. for G_3 is 5.

Question 1. *How do we approximate $\int_a^b f(x) \, dx$ using GQ?*

To do this we find the linear function $\varphi : [-1,1] \to [a,b]$ such that $\varphi(-1) = a$ and $\varphi(1) = 1$. Use the linear interpolant which is given by $\varphi(t) = 1/2(b-a)t + 1/2(b+a)$. Transforming the integral we have

$$\int_a^b f(x) \, dx = \int_{-1}^{1} f\left(\frac{(b-a)}{2} t + \frac{(b+a)}{2}\right) \frac{b-a}{2} \, dt$$

Theorem 1. The D.O.E. of G_n is $2n - 1$.

Proof. We want to show that $\int_{-1}^{1} f(x)\,dx = G_n(f)$ for any polynomial of degree less than or equal to $2n-1$. Let f be such a polynomial. We then use long division of polynomial for f and q_n(n^{th} legendre polynomial) to get

$$f(x) = Q(x)q_n(x) + R(x) \tag{7}$$

where $\deg(R) < \deg(q_n) = n$ which implies that $\deg(R) \leq n-1$. Note that $f(x_k) = R(x_k)$ for $k = 1, 2, \ldots, n$ which gives $G_n(f) = G_n(R)$. Now as the $q_0, \ldots, q_{n-1}$ are a basis for polynomials of degree less than or equal to $n-1$ then $Q(x) = \sum \alpha_j q_j$ for some $\alpha_j \in \mathbb{R}$. Thus from (7) we have

$$\int_{-1}^{1} f(x)\,dx = \int_{-1}^{1} Q(x)q_n(x)\,dx + \int_{-1}^{1} R(x)\,dx$$

and

$$\int_{-1}^{1} Q(x)q_n(x)\,dx = \int_{-1}^{1} \sum_{k=0}^{n-1} \alpha_k q_k(x)q_n(x)\,dx = \sum_{k=0}^{n-1} \alpha_k \int_{-1}^{1} q_k(x)q_n(x)\,dx = 0$$

Gathering this together we have

$$\int_{-1}^{1} f(x)\,dx = \int_{-1}^{1} R(x)\,dx$$

and as $\deg(R) \leq n-1$ then $G_n(R)$ is exactly $G_n(f)$. All that remains is to show there exists a polynomial of degree 2n such that G_n is not exact. This is left as an exercise to the reader. □

Corollary 2. The weights of G_n are positive.

Proof. Note that by definition

$$w_j = \int_{-1}^{1} l_j(x)\,dx$$

and as $\deg(l_j) = n-1$ then we can apply Theorem 1 on l_j^2.

$$\begin{aligned}
0 &< \int_{-1}^{1} l_j^2(x)\,dx \\
&= G_n(l_j^2) \\
&= \sum_{i=1}^{n} w_i l_j^2(x_i) \\
&= \sum_{i=1}^{n} w_i l_j(x_i) \\
&= \int_{-1}^{1} l_j(x)\,dx
\end{aligned}$$

□

3.6 Adaptive Quadrature

Main Idea: Use composite rules with unequal subintervals.
Motivation: Reduce error and use fewer evaluations.
Let $f : [a,b] \to \mathbb{R}$ and $a = x_0 < x_1 < \cdots < x_m = b$ with $h_i = x_i - x_{i-1}$ for $i = 1, \ldots, m$. On each $[x_{i-1}, x_i]$ Simpson's rules gives

$$S_{[x_{i-1},x_i]}(f) = \frac{h_i}{6}(f(x_{i-1}) + 4f\left(\frac{x_i + x_{i-1}}{2}\right) + f(x_i))$$

$$E_i(f) = ch_i^5 f^{(4)}(c_i), \qquad c_i \in (x_{i-1}, x_i)$$

Our global error is given by $|E_1 + \cdots + E_m| \leq |E_1| + \cdots + |E_m|$. if we impose

$$\frac{|E_i|}{h_i} < \frac{\epsilon}{b-a} \tag{8}$$

then we have

$$\sum_{i=1}^{m} |E_i| < \frac{\epsilon}{b-a} \sum_{i=1}^{m} h_i = \epsilon$$

To impose (8) we need a computable estimator $\tilde{E}_i$ such that $E_i \approx \tilde{E}_i$ and $E_i \leq \tilde{E}_i$. If we assume that $\tilde{E}_i$ is available then the main idea for an adaptive quadrature is given by the following steps.

1. Start with a course partition of $[a, b]$.

2. For each subinterval (α, β) chech if $\tilde(E)_{(\alpha,\beta)} < \epsilon \frac{\beta-\alpha}{b-a}$. If it is then we use this partition. If not, then refine to $[\alpha, (\alpha+\beta)/2] \cup [(\alpha+\beta)/2, \beta]$ and repeat.

To find $\tilde{E}_{(\alpha,\beta)} \approx E_{(\alpha,\beta)}$ we define

$$\tilde{S}_{[\alpha,\beta]} = S_{[\alpha,(\alpha+\beta)/2]} + S_{[(\alpha+\beta)/2,\beta]} \tag{9}$$

Then

$$\int_{\alpha}^{\beta} f(x)\,dx = S_{[\alpha,\beta]} + ch^5 + O(h^7) \tag{10}$$

As we can divide the integral into subintervals we have

$$\int_{\alpha}^{\beta} f(x)\,dx = \tilde{S}_{[\alpha,\beta]} + (1/16)ch^5 + O(h^7) \tag{11}$$

Eliminating c from (10) and (11) we have

$$\int_{\alpha}^{\beta} f(x)\,dx - \tilde{S}_{[\alpha,\beta]} = \frac{1}{15}(\tilde{S}_{[\alpha,\beta]} - S_{[\alpha,\beta]}) + O(h^7)$$

If h is small then $\tilde{E}_{[\alpha,\beta]} = \frac{1}{15}(\tilde{S}_{[\alpha,\beta]} - S_{[\alpha,\beta]})$ is a good estimator.
Matlab To implement in Matlab we use adaptsimp.m on the course webpage.

```
I = adapts(f,a,b,epsilon)
    input f,[a,b],epsilon tolerance
    output approximation for integral
```

3.7 Multidimensional Quadrature

When we are in dimension $d = 2$ we assume that $\Omega = [0,1] \times [0,1]$ and $F : \Omega \to \mathbb{R}$ is continuous. If we consider the two general quadratures

$$\int_0^1 f(x)\,dx \approx Q_1(f) := \sum_{i=0}^{m} A_i f(x_i)$$

$$\int_0^1 f(y)\,dy \approx Q_2(f) := \sum_{j=0}^{n} B_j f(y_j)$$

where A_i, B_j are weights and x_i, y_j with are distinct nodes then

$$\begin{aligned}\int_{\Omega} F(x,y)\,d\Omega &= \int_0^1 \int_0^1 F(x,y)\,dx\,dy \\ &\approx \int_0^1 \sum_{j=0}^{n} B_j F(x, y_j)\,dx \\ &\approx \sum_{i=0}^{m} \sum_{j=0}^{n} A_i B_j F(x_i, y_j)\end{aligned}$$

Note that if $\Omega = [a,b] \times [c,d]$ then we can always transform to $[0,1] \times [0,1]$ as we did with the Gaussian Quadrature. If we have a more complicated domain such as $\Omega = (x,y)$ where $a \leq x \leq b$ and $\phi_1 \leq y \leq \phi_2$ then we have

$$\int_{\phi_1}^{\phi_2} F(x,y)\,dy \approx (\phi_2(x) - \phi_1(x)) \sum_{j=0}^{n} B_j F(x, \tilde{y}_{j,x})$$

where

$$\tilde{y}_{j,x} = (1 - y_j)\phi_1(x) + y_j \phi_2(x)$$

Getting the transformation for x we have

$$\int_{\Omega} F(x,y)\,d\Omega \approx \sum_{i=0}^{m} \sum_{j=0}^{n} A_i B_j (b-a)(\phi_2(\tilde{x}_i)\phi_1(\tilde{x}_i)) F(\tilde{x}_i, \tilde{y}_{j,\tilde{x}_i})$$

where $\tilde{x}_i$ is defined in the same fashion as $\tilde{y}$.

3.8 2D-Composite quadrature

Assume that $\Omega \subset \mathbb{R}^2$ is a polygonal domain. We then split Ω into a union of triangle such that

$$\overline{\Omega} = \bigcup_{K_i \in \tau_h} K_i$$

and

$$K_i \cap K_j = \begin{cases} \emptyset \\ \text{vertex} \\ \text{edge} \end{cases}$$

Define $\tau_j = \{K_i : K_i \in \overline{\Omega}\}$. A regular mesh is defined to be one such that for all $K \in \tau_h$, $\operatorname{diam}(K) \leq h$ and $\rho \geq c_1 h$ where ρ is the radius of the inscribed circle of K. If the area of K is approximately ch^2 then we have a Quasi-uniform mesh.

3.8.1 Linear Interpolation for triangles

Let $K = [z_1, z_2, z_3]$ where $z_i = (x_i, y_i)$. If $f : K \to \mathbb{R}$ is continuous then there exists a unique $P_1(x,y) = ax + by + c$ such that $P(z_i) = f(z_i)$. If $f(z_i) = P(z_i) = ax_i + by_i = c$ then

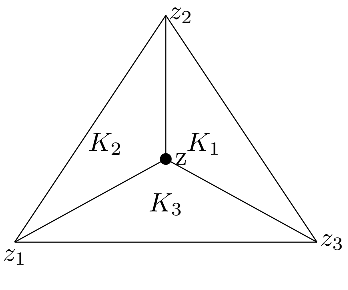
$$\Delta K = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 - x_1 & y_2 - y_1 & 0 \\ x_3 - x_1 & y_3 - y_1 & 0 \end{vmatrix} = 2\text{area}(K) \neq 0$$

3.8.2 Lagrange nodal function for the triangle

We wish to look for $\lambda_1, \lambda_2, \lambda_3$ lean function of (x,y) such that $\lambda_i(z_j) = \delta_{ij}$. If we assume that we have them then $P_1(x,y) = \lambda_1 f(z_1) + \lambda_2 f(z_2) + \lambda_3 f(z_3)$. Consider

$$\lambda_1(x,y) = \frac{1}{\Delta K}\begin{vmatrix} 1 & 1 & 1 \\ x & x_2 & x_3 \\ y & y_2 & y_3 \end{vmatrix} \qquad \lambda_2(x,y) = \frac{1}{\Delta K}\begin{vmatrix} 1 & 1 & 1 \\ x_1 & x & x_3 \\ y_1 & y & y_3 \end{vmatrix} \qquad \lambda_3(x,y) = \frac{1}{\Delta K}\begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x \\ y_1 & y_2 & y \end{vmatrix}$$

These give rise to what is know as Barycentric coordinates. A simple geometric interpretation is provided below.



Here we have $z = (x, y)$ and $\lambda_1(x, y) = \text{area}(K_1)/\text{area}(K)$, $\lambda_2(x, y) = \text{area}(K_2)/\text{area}(K)$, and $\lambda_3(x, y) = \text{area}(K_3)/\text{area}(K)$. Notice that $\lambda_1 + \lambda_2 + \lambda_3 = 1$. We call $(\lambda_1, \lambda_2, \lambda_3)$ the Barycentric coordinates with respect to $K = [x_1, z_2, z_3]$.

To get the Cartesian coordinates from the Barycentric coordinates we use $1, x, y$ as a basis for $\mathbb{P}_1$. Therefore we have $(x, y) = z = \lambda_1 z_1 + \lambda_2 z_2 + \lambda_3 z_3$. Note that this can be done in three dimensions, however we then use tetrahedra. Say we now wish to integrate over the triangle $K = [z_1, z_2, z_3]$ and we have the interpolant P_1. Then

$$\begin{aligned}\int_K f(x, y)\, dx\, dy &\approx \int_K P_1(x, y)\, dx\, dy \\ &= \int_K \lambda_1 f(z_1) + \lambda_2 f(z_2) + \lambda_3 f(z_3)\, dx\, dy \\ &= |K|(P(z_1) + P(z_2) + P(z_3)) \\ &= |K| f(z_B)\end{aligned}$$

where we use the fact that the integral of $\lambda_i = |K|/3$ and z_B is the average of the vertices. This quadrature is know as the Barycenter Quadrature (BR). Note that BR has D.O.E. 1.

The Midpoint Rule (MR) We use the same K as before however now we denote z_{ij} to the the midpoint of the edge connecting z_i and z_j. With this we have

$$\int_K f(x, y)\, dx\, dy \approx \frac{|K|}{3}(f(z_{12}) + f(z_{13}) + f(z_{23}))$$

and the D.O.E. for MR is 2.

Theorem 2. If Q is a quadrature for $\int_K f(x, y)\, dx\, dy$ with D.O.E. r, then

$$\left| \int_K f(x, y)\, dx\, dy - Q(f) \right| \leq ch^{r+1} \int_K \sum_{|\alpha|=r+1} |D^\alpha f|\, dx\, dy$$

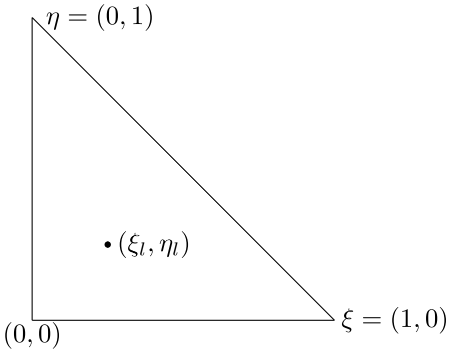
Here D is the differential operator and the sum will be over all partial derivative of order α.

For a composite rule we apply the methods above for all K. This results in

$$\int_\Omega f(x, y)\, d\Omega \approx \sum_{K \in \tau_K} |K| f(z_B)$$

3.8.3 General Quadrature on triangles

For this we use the reference triangle $\hat{K}$ which has vertices at $(0, 0), (0, 1), (1, 0)$.



Here (ξ_l, η_l) is some point in $\hat{K}$ for $l = 1, \dots, m$ and we have corresponding weights. Witht this we have

$$\int_{\hat{K}} g(\xi, \eta)\, d\xi\, d\eta \approx \sum_{l=1}^{m} w_l g(\xi_l, \eta_l)$$

The question remains of how to obtain a quadrature corresponding to a generic triangle $[z_1, z_2, z_3]$? For this we use an affine transformation. We have the existence of a unique affine transformation $F_K : \hat{K} \to K$ where $F_K(0,0) = z_1$, $F_k(1,0) = z_2$ and $F_k(0,1) = z_3$. Using the transformation that can easily be obtained we have that for a continuous $f : K \to \mathbb{R}$,

$$\int_K f(x,y)\, dx\, dy = \int_{\hat{K}} f(F_K(\xi\eta)) 2|K|\, d\xi\, d\eta$$

3.9 Singular Integrals

The first case we have are jump discontinuities. In this case if $f : [a,b] \to \mathbb{R}$ and is continuous everywhere except for a jump at $c \in (a,b)$ then simply split the integral and apply quadrature to the two new integrals. For the second case say we wish to determine I where

$$I = \int_a^b \frac{\varphi(x)}{(x-a)^\mu}\, dx, \qquad 0 < \mu < 1$$

and $\varphi(a) \neq 0$ and is smooth and bounded by M on $[a,b]$. For this we choose $\epsilon \in (0, b-a)$ and divide the integral.

$$I = \underbrace{\int_a^{a+\epsilon} \frac{\varphi(x)}{(x-a)^\mu}\, dx}_{I_1} + \underbrace{\int_{a+\epsilon}^{b} \frac{\varphi(x)}{(x-a)^\mu}\, dx}_{I_2}$$

For I_2 simply apply a composite quadrature such that $|I_2 - CQ(f)| < \delta$. For I_1 we Taylor expand to get

$$\varphi(x) = \varphi(a) + \sum_{k=1}^{p} \frac{\varphi^{(k)}(a)}{k!}(x-a)^k + \frac{\varphi^{(p+1)}(\xi_x)}{(p+1)!}(x-a)^{p+1}$$

and call the truncated expansion φ_p. Inserting this into I_1 we have

$$I_1 = \underbrace{\int_a^{a+\epsilon} \frac{\varphi_p(x)}{(x-a)^\mu}\, dx}_{\tilde{I}_1} + \underbrace{\int_a^{a+\epsilon} \frac{\varphi^{(p+1)}(\xi)(x-a)^{p+1}}{(p+1)!(x-a)^\mu}\, dx}_{E_1}$$

Note that $\tilde{I}_1$ is computable. Now assume that $|\varphi^{(p+1)}(x)| \leq M_p$ on $[a, a+\epsilon]$. Then

$$\begin{aligned}|E_1| &\leq \frac{M_p}{(p+1)!} \int_a^{a+\epsilon} (x-a)^{p+1-\mu}\, dx \\ &= \frac{M_p \epsilon^{p+2-\mu}}{(p+1)!(p+2-\mu)}\end{aligned}$$

In order to be within some tolerance we pick p or ϵ such that $|E_1| \leq \delta/2$. This will give us the desired result of $I \approx I_1 + CQ(f)$.

4 Finite Difference

We turn to a well know definition

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

which we use to approximate the first derivative

$$f'(x) \approx \frac{f(x+h)-f(x)}{h}$$

This formula is very easily obtained from Taylor expanding $f(x+h)$.

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

and a quick rearrangement will give

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

If we drop the second derivative term then we get what is called the **Two point forward difference**. It should be immediate that the error for this formula will be $E(h) = -f''(\xi)h/2$, and if $f''(\xi)$ is bounded on $[x, x+h]$ then $E(h) = O(h)$.

Centered Difference Formula for f'

If we now consider two points $x \pm h$ and Taylor expand $f(x \pm h)$ we have

$$f(x+h) = f(x) + hf'(x) + f''(x)\frac{h^2}{2} + f'''(c_1)\frac{h^3}{6}$$
$$f(x-h) = f(x) - hf'(x) + f''(x)\frac{h^2}{2} - f'''(c_2)\frac{h^3}{6}$$

We can then isolate $f'(x)$ by subtracting the second from the first and dividing by $2h$. This will give the centered difference formula:

$$f'(x) = \frac{f(x+h)-f(x-h)}{2h} + E(h)$$
$$E(h) = -\frac{f'''(c)h^2}{6} = O(h^2)$$

Centered difference for f''

Repeating the same process but now expanding to order 4 we have

$$f(x+h) = f(x) + hf'(x) + f''(x)\frac{h^2}{2} + f'''(x)\frac{h^3}{6} + \frac{f^{(4)}(c_1)h^4}{24}$$
$$f(x-h) = f(x) - hf'(x) + f''(x)\frac{h^2}{2} - f'''(x)\frac{h^3}{6} + \frac{f^{(4)}(c_2)h^4}{24}$$

Adding these two and subtracting $2f(x)$ and then dividing by h^2 will give with a little rearrangement

$$f''(x) = \frac{f(x+h)+f(x-h)-2f(x)}{h^2} - E(h)$$
$$E(h) = \frac{h^2(f^{(4)}(c_1)+f^{(4)}(c_2))}{24} = O(h^2)$$

General procedure, given points $x+ih$ we expand $f(x+ih)$ and take linear combination to eliminate all derivative except the desired one.

5 Numerical solutions of IVP's

5.1 The Cauchy Problem

$$\begin{cases} y'(t) = f(t,y) \\ y(t_0) = y_0 \\ t \in I \end{cases} \tag{12}$$

Here I is an interval containing t_0. For this problem $f(t,y) : I \times \mathbb{R} \to \mathbb{R}$, t_0, and y_0 are given and we wish to find $y(t)$.

5.1.1 Local Existence and Uniqueness

Assume that $R = J \times Y$ is a rectangular neighborhood of (t_0, y_0) and $t_0 \in J \subseteq I$ such that f is Lipschitz on R,i.e.

$$|f(t, y_1) - f(t, y_2)| \leq L|y_1 - y_2| \tag{13}$$

and assume that $M = \max_{(t,y)\in R} |f(t, y)| < \infty$ and $J = [a, b]$ and $Y = [c, d]$.

Theorem 1. If f is Lipschitz on R then (12) admits a unique solution on $(t_0 - r_0, t_0 + r_0)$ where $r_0 < \min\{(b - a), (d - c)/M, 1/L\}$

Theorem 2. (Gloval existence) If (13) is satisfied on an infinite strip $I \times (-\infty, \infty)$ then (12) has a unique solution on I.

We note an important property of (13) being satisfied when $|\partial f/\partial y| \leq L$ for all $(t, y) \in R$.

Example 11.

$$\begin{cases} y'(t) = y + e^t \\ y(t_0) = y_0 \\ t \in I = [-1, \infty) \end{cases}$$

In this example $f(t, y) = y + e^t$. Therefore

$$\frac{\partial f}{\partial y} = 1 \qquad \Rightarrow \qquad |f(t, y_1) - f(t, y_2)| \leq |y_1 - y_2| \qquad \forall (t, y_1)(t, y_2)$$

Thus, by the global existence theorem we have a unique solution and it is given by

$$y(t) = te^t + y_0 e^t$$

Example 12.

$$\begin{cases} y'(t) = 1 + y^2 \\ y(t_0) = 0 \\ t \in I = [-\pi/2, \pi/2] \end{cases}$$

For this we have no global solution as $\partial f/\partial y = 2y$

Liapunov Stability of (12)
Assume that (12) has a unique soluiton on I and I is bounded. Let $\delta_0 \in \mathbb{R}$ and $\delta = \delta(t)$ be a continuous function on I and consider the perturbed problem.

$$\begin{cases} z'(t) = f(t, z) + \delta(t) \\ z(t_0) = y_0 + \delta_0 \\ t \in I \end{cases} \tag{14}$$

Definition 3. The IVP (12) is **stable** on I if for any $\epsilon > 0$ small enough such that (14) has a unique solution on I and for any $\delta_0 < \epsilon$ and $|\delta(t)| < \epsilon$, $\forall t \in I$, there exists $c > 0$ independent of ϵ such that $|y(t) - z(t)| < c\epsilon$ for all $t \in I$.

Definition 4. If I contains $+\infty$ we say that (12) is **asymptotically stable** if (12) is stable for any bounded subinterval of I and $\lim_{t\to\infty} |y(t) - z(t)| = 0$.

Lemma 1. (Gronwall) Assume p, q, φ are functions defined on $[t_0, t_0 + T]$ such that

1. p is integrable

2. g, φ are continuous

3. For all $t_0 \leq t_1 < t_2 \leq T$, we have $p(t_1) < p(t_2)$

4. $\varphi(t) \leq g(t) + \int_{t_0}^{t} p(z)\varphi(z)\, dx$ for all $t \in [t_0, t_0 + T]$

then we have

$$\varphi(t) \leq g(t) e^{\int_{t_0}^{t} p(z)\, dz} \tag{15}$$

Theorem 3. If f is uniformly Lipschitz on I, then the IVP (12) is stable.

Proof. We need to check the requirements for stability where the $z(t)$ in the definition is the solution to (14). Let $\epsilon > 0$ be fixed and small enough, and δ, δ_0 satisfying $|\delta_0| < \epsilon$, $|\delta(t)| < \epsilon$ for all $t \in [t_0, t_0 + T]$. Let $w(t) = z(t) - y(t)$. Then from (12) and (14) we have $w'(t) = f(t, z(t)) - f(t, y(t)) + \delta(t)$. If we then integrate on $[t_0, t]$ and use the Fundamental Theorem of Calculus we get

$$w(t) - w(t_0) = \int_{t_0}^{t} f(s, z(s)) - f(s, y(s))\, ds + \int_{t_0}^{t} \delta(s)\, ds$$

Note that $w(t_0) = z(t_0) - y(t_0) = \delta_0$. Therefore,

$$|f(s, z(s)) - f(s, y(s))| \leq L|z(s) - y(s)| = L|w(s)|$$

We also have that $|\delta(s)| < \epsilon$ on $[t_0, t]$. Thus,

$$\begin{aligned} |w(t)| &\leq |\delta_0| + L\int_{t_0}^{t} |w(s)|\, ds + \epsilon(t - t_0) \\ &\leq (1 + (t - t_0))\epsilon + L\int_{t_0}^{t} |w(s)|\, ds \end{aligned}$$

We can now apply Gronwall's lemma with $\varphi(t) = |w(t)|$, $p(t) = L$, and $g(t) = (1 + (t - t_0))\epsilon$. This will result in

$$\begin{aligned} |w(t)| &\leq (1 + (t - t_0))\epsilon e^{\int_{t_0}^{t} L\, dT} \\ &= (1 + (t - t_0))\epsilon e^{L(t - t_0)} \\ &\leq (1 + T)e^{cT}\epsilon \\ &= c\epsilon \end{aligned}$$

□

5.2 Numerical Approach

Given the Cauchy problem (12) we discretize t where $t_n = t_0 + nh$ and $h = T/n$. At each point in time t_n we will approximate $y(t_n)$ by u_n.

5.2.1 Forward Euler's Method

For this method we use the Expansion of $y(t_n + h) = y(t_n) + hy'(t_n) + O(h^2)$. With this we have

$$u_0 = y_0$$

$$u_1 = y(t_1) = y(t_0 + h) = u_0 + hf(t_0, u_0)$$

$$\vdots$$

$$u_{n+1} = u_n + hf(t_n, u_n)$$

This method above is know as Forward Euler's Method.

5.2.2 Backward Euler

This is a one step implicit method and is given by

$$u_{n+1} = u_n + hf_{n+1} = u_n + hf(t_{n+1}, u_{n+1})$$

This method will generally result in a system that we need to solve at each time step.

5.2.3 Crank-Nicolson

This is again implicit.

$$u_{n+1} = u_n + \frac{h}{2}(f_n + f_{n+1})$$

5.2.4 Heun's Method

$$u_{n+1} = u_n + \frac{h}{2}(f_n + f(t_{n+1}, u_n + hf_n))$$

5.3 Analysis of One step methods

The general form of a one step method is given by

$$u_{n+1} = u_n + h\phi(t_n, u_n, f_n, h) \tag{16}$$

where ϕ is known as the increment function. For Forward Euler $\phi = f_n$ and for $C - N$ we have $\phi = 1/2(f_n + f_{n+1})$.

5.3.1 Local Truncaiton Error (LTE)

When working with Forward Euler we have $y(t_0 + h) = y_1 = y_0 = hf(t_0, y_0) + \epsilon_1$. If we continue then we have $y_{n+1} = y_n + hf(t_n, y_n) + \epsilon_{n+1}$. Here ϵ_{n+1} is the residual at the step t_{n+1} when we replace u_n with y_n in (16). We will impose that the exact solution "satisfies" the numerical method. Therefore in general

$$y_{n+1} = y_n + h\phi(t_n, y_n, f(t_n, y_n), h) + \epsilon_{n+1} \tag{17}$$

If we rewrite (17) to model the (12) then we have

$$\frac{y_{n+1} - y_n}{h} = \phi(t_n, y_n, f(t_n, y_n), h) + \tau_{n+1}(h)$$

where $\tau_{n+1}(h) = \epsilon_{n+1}/h$ is the local truncation error and $\tau(h) = \max_{0 \le n \le N-1} |\tau_{n+1}(h)|$ is the global truncation error. For example in Forward Euler we have

$$\frac{y_{n+1} - y_n}{h} = f(t_n, y_n) + \tau_{n+1}(h)$$

where Taylor series will give that $\tau_{n+1}(h) = O(h)$.

Definition 5. A method given by (16) is **consistent** if LTE goes to zero as $h \to 0$

$$\lim_{n \to \infty} \tau_{n+1}(h) = 0$$

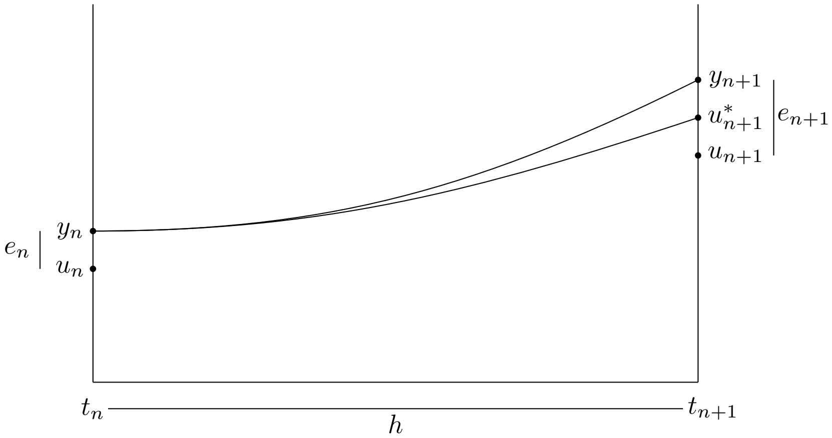
Definition 6. If $\tau(h)$ is $O(h^p)$ as $h \to 0$ then we say the method is order p. Some examples are Euler's method is order 1 and Heun/C-N are order 2.

5.4 Convergence

Definition 7. A method is said to be **convergent** if $|u_n - y_n| < c(h)$ and $\lim_{n\to\infty} c(h) = 0$ or $\lim_{n\to\infty} |u_n - y_n| = 0$.

We also that a method is convergent with order p if $c(h) = ch^p$. As we move forward we will use $e_n = y_n - u_n$ for ease of notation.

Theorem 4. Forward Euler is first order convergent if we assume Lipschitz condition $|f(t, y_1) - f(t, y_2)| < L|y_1 - y_2|$ and $y \in C^2([t_0, t_0 + T])$.



In this picture we have $u^*_{n+1} = y_n + hf(t_n, y_n)$.

Proof. This proof will simply be a progression to get to the desired result.

$$\begin{aligned}
|e_{n+1}| &= |y_{n+1} - u_{n+1}| \\
&= |y_{n+1} - u^*_{n+1} + u^*_{n+1} - u_{n+1}| \\
&= |h\tau_{n+1}(h) + y_n + hf(t_n, u_n) - (u_n + hf(t_n, u_n))| \\
&= |y_n - u_n + h(f(t_n, y_n) - f(t_n, u_n))| \\
&\leq |e_n| + hL\left|\frac{y_n - u_n}{h}\right| \\
&= |e_n|(1 + hL)
\end{aligned}$$

Therefore $|e_{n+1}| \leq h|\tau_{n+1}| + |e_n|(1+hL)$. Thus, if we begin at $|e_1|$ and continue to $|e_{n+1}|$ we get a succession of multiplication by factors of $(1 + hL)$.

$$\begin{gathered}
|e_1| \leq h\tau(h) \\
|e_2| \leq h\tau(h) + |e_1|(1 + hL) \\
\vdots \\
|e_n| \leq h\tau(h) + |e_{n-1}|(1 + hL) \\
|e_{n+1}| \leq h\tau(h) + |e_n|(1 + hL)
\end{gathered}$$

This then implies that

$$\begin{aligned}
|e_{n+1}| &\leq h\tau((1 + hL) + \cdots + (1 + hL)^n) \\
&= \frac{(1 + hL)^{n+1} - 1}{L}\tau(h)
\end{aligned}$$

Now we use the simple trick of $1 + x \leq e^x$ to get that $(1 + hL)^{n+1} \leq e^{hl(n+1)} = e^{LT}$. Now as $\tau(h) = \max|\tau_{n+1}(h)| \leq \max|y''(t_n)/2| := M/2$ then we have the desired result of

$$|e_{n+1}| \leq \frac{M(e^{LT} - 1)}{2L}h$$

□

5.5 Absolute Stability of one step methods

We start with the Cauchy problem (12) along with the general one step method (16). We need to check if $|u_n|$ remains bounded as $t_n \to \infty$. For the formal definition we turn to a common test problem.

$$\begin{cases} y'(t) = \lambda y(t) \\ y(0) = 1 \\ t \in [0, \infty) \end{cases} \tag{18}$$

Here we have $\lambda \in \mathbb{C}$ with $\Re(\lambda) < 0$. Clearly we have that the exact solution is given by $y(t) = e^{\lambda t}$ and if $\lambda = \lambda_0 + i\lambda_1$ then $y(t) = e^{\lambda_0 t}(\cos(\lambda_1 t) + i \sin(\lambda_1 t))$ which them implies that $|y(t)| = e^{\lambda_0 t} \to 0$ if and only if $\lambda_0 < 0$.

Definition 8. A numerical method (M) for approximating (18) is **absolutely stable** if

$$|u_n| \to 0 \quad \text{as} \quad t_n \to \infty \tag{19}$$

Definition 9. The **region of absolute stability** of M is the subset of $\mathbb{C}$ denoted $\mathcal{A}_M$ where

$$\mathcal{A}_M := \{z \in \mathbb{C} : z = \lambda h \text{ is stable}\}$$

An important thing to note is that if λ is fixed then knowing $\mathcal{A}_M$ will give the range of h for which the method is stable. Please also know that the choice of the test problem (18) is ideal as we can easily adapt it to matrix systems $y' = Ay$.

5.5.1 Absolute stability for Forward Euler

For Forward Euler $u_{n+1} = u_n + hf(t_n, u_n)$ we have $u_0 = 1$ via the test problem (18). We also have $f(t, y) = \lambda y$. Substituting this into the method we have

$$u_{n+1} = u_n + h\lambda u_n = u_n(1 + h\lambda)$$

Now bringing in the intial condition we have

$$u_1 = (1 + h\lambda)$$

$$u_2 = u_1(1 + h\lambda) = (1 + h\lambda)^2$$

$$\vdots$$

$$u_n = (1 + h\lambda)^n$$

It should be immediate that in order for $|u_n| \to 0$, we require $|1 + h\lambda| < 1$. Therefore we have

$$\mathcal{A}_E = \{z \in \mathbb{C} : |1 + z| < 1\}$$

For example, if $\lambda \in \mathbb{R}$ and $\lambda = \lambda_0 < 0$ then we have $|1 + h\lambda_0| < 1$ which is equivalent to $-2 < h\lambda_0 < 0$. Therefore, for absolute stability we require $h < -2/\lambda_0$.

Example 13. If we have the test problem (18) and $\lambda = -20$ then what is an h that will work? From the note above we know that $h < -2/\lambda_0 = -2/-20 = 1/10$. Thus, we can take $h = 1/11$ as $u_n = (1-20/11)^n = (-9/11)^n \to 0$.

5.5.2 Absolute stability for Backward Euler

$$u_{n+1} + u_n + hf(t_{n+1}, u_{n+1})$$

Again as before we have $u_0 = 1$, $f(t, y) = \lambda y$ which will in turn give

$$u_{n+1} = u_n + h\lambda u_{n+1}$$

Rearranging will yield

$$u_{n+1} = u_n(1 - h\lambda)^{-1}$$

and we can get from the start as we did in Forward Euler that:

$$u_{n+1} = (1 + h\lambda)^{-n}$$

Thus,

$$\mathcal{A}_{BE} = \{z \in \mathbb{C} : |1 - z| > 1\}$$

5.5.3 Absolute stability for Crank-Nicolson

Using the same test problem it can be shown that

$$u_{n+1} = \left(\frac{1 + h\lambda/2}{1 - h\lambda/2}\right)^n$$

which in turn gives

$$\mathcal{A}_{CN} = \{z \in \mathbb{C} : \left|\frac{2 + z}{2 - z}\right| < 1\}$$

Please note that with a little algebra this region can be shown to be equivalent to $\{z \in \mathbb{C} : \Re(z) < 0\}$.

5.5.4 Absolute stability for Heun

$$\mathcal{A}_H = \{z \in \mathbb{C} : |1 + z + z^2/2| < 1\}$$

5.5.5 Category

Backward Euler and Crank-Nicolson are what is know as **A-Stable** as their region of stability is the whole left half plane of $\mathbb{C}$. As for Forward Euler and Heun, we call these methods **conditionally stable** as they have requirements on the choice of h.

6 Difference Equations

6.1 Simple example

We begin with a simple example that many have hear of before, the Fibonacci sequence. This sequence is defined by

$$u_0 = 0 \qquad u_1 = 1 \qquad u_n = u_{n-1} + u_{n-2}$$

Therefore the first few terms are $0, 1, 1, 2, 3, 5, 8, 13, 21, \ldots$. We would like to find a function that when given an n value will give us u_n without having to go through the whole sequence to determine it. To do this we will treat the relation above as a homogeneous differential equation with constant coefficients.

$$u_{n+2} - u_{n+1} - u_n = 0$$

Here the change of indices is simply for convince. This resembles a Euler/Cauchy-Euler differential equation, so we turn to the characteristic polynomial.

$$r^2 - r - 1 = 0$$

Solving this will give $r = (1 \pm \sqrt{5})/2$. Therefore our solution is given by

$$u_n = c_1\left(\frac{1-\sqrt{5}}{2}\right)^n + c_2\left(\frac{1+\sqrt{5}}{2}\right)^n$$

As in differential equations we turn to u_0 and u_1 to determine $c_{1,2}$.

$$u_0 = c_1 + c_2 = 0 \qquad u_1 = c_1\left(\frac{1-\sqrt{5}}{2}\right) + c_2\left(\frac{1+\sqrt{5}}{2}\right) = 1$$

Solving for the constants will give $c_2 = -c_1 = 1/\sqrt{5}$. Thus,

$$u_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right]$$

6.2 General Form

The set up for the general form is giving by

$$u_{n+k} + \alpha_{k-1}u_{n+k-1} + \cdots + \alpha_1 u_{n+1} + \alpha_0 u_n = \varphi_{n+k}$$

For this we assume that $\alpha_0 \neq 0$, the remaining α are constants and we are given $u_0, \dots, u_{k-1}$. As in differential equations, we fall into two cases, either $\varphi = 0$ or $\varphi \neq 0$.

When $\varphi = 0$ we are in the homogeneous case. For this we consider the characteristic equation $r^k + \alpha_{k-1}r^{k-1} + \cdots + \alpha_1 r + \alpha_0 = 0$. If $r_{k-1}, \dots, r_1, r_0$ are the distinct roots of the characteristic equation then we have

$$u_n = \gamma_0 r_0^n + \gamma_1 r_1^n + \cdots + \gamma_{k-1} r_{k-1}^n$$

Please note that in the case of a repeated root, we can fix the issue. If r^* is a repeated root then we simply use $\gamma_i r_i^{*n} + \gamma_{i+1} n r_i^{*n}$.

When we have $\varphi \neq 0$ then we proceed as we would in differential equations and consider $u_n = u_h + u_p$ where u_h is the solution to the homogeneous equation, and u_p is the particular solution. In order to find u_p we make a guess of the solution form and determine coefficients. For example, if $\varphi_{n+k} = c^n Q(n)$ where $Q(n)$ is a polynomial of degree p in n then we try $u_p = c^n(b_p n^p + \cdots + b_1 n + b_0)$.

7 Linear Multistep Methods

We again use the Cauchy Problem (12) which is given again for reference.

$$\begin{cases} y' = f(t,y) \\ y(t_0) = y_0 \\ t \in [t_0, t_0 + T] \end{cases} \tag{12}$$

and here we take $u_0, \dots, u_N$ to be approximations of $y_0, \dots, y_N$. For notation we will use $t_n = t_0 + nh$, $h = T/N$, and $f_n = f(t_n, u_n)$.

7.1 Adams-Bashforth (AB2)

This method will be an order two multistep method. To get the scheme we will assume that u_{n-1} and u_n are available. In order to determine $u_{n+1} \approx y(t_{n+1})$ we will integrate the Cauchy Problem given by (12).

$$\begin{aligned} y(t_{n+1}) - y(t_n) &= y(t)\Big|_{t_n}^{t_{n+1}} \\ &= \int_{t_n}^{t_{n+1}} y'(t)\,dt \\ &= \int_{t_n}^{t_{n+1}} f(t, y(t))\,dt \\ &\approx \int_{t_n}^{t_{n+1}} \frac{t - t_n}{t_{n-1} - t_n} f(t_{n-1}, y(t_{n-1})) + \frac{t - t_{n-1}}{t_n - t_{n-1}} f(t_n, y(t_n))\,dt \\ &= \frac{3h}{2} f(t_n, y(t_n)) - \frac{h}{2} f(t_{n-1}, y(t_{n-1})) \end{aligned}$$

This will give us the scheme for AB2

$$u_{n+1} = u_n + \frac{h}{2}(3f_n - f_{n-1})$$

Please note that we can also have use a quadrature to approximate $\int f(t, y(t))\,dt$. One such choice is to use Simpson's Rule which will ultimately result in the implicit method given by

$$u_{n+1} = u_n + \frac{h}{3}(f_{n-1} + 4f_n + f_{n+1})$$

7.2 General Form

If we assume that $u_0, \ldots, u_p$ are available then a general linear multistep method (LMS) is given by

$$u_{n+1} = \sum_{j=0}^{p} a_j u_{n-j} + h\left(b_{-1} f_{n+1} \sum_{j=0}^{p} b_j f_{n-j}\right) \tag{20}$$

The method above is classified as a $p+1$ step method if a_p or b_p are not equal to zero. In the case of $b_{-1} \neq 0$ then we have an implicit scheme and explicit if it zero.

7.3 Local Truncation Error

We will impose that the exact solution $y = y(t)$ satisfies (20) and divide through by h. This results in

$$\begin{aligned} h\tau_{n+1}(h) &= y_{n+1} - \sum_{j=0}^{p} a_j y_{n-j} - h \sum_{j=-1}^{p} b_j y'_{n-j} \\ &= y(t_n + h) - \sum_{j=0}^{p} a_j y(t_n - jh) - h \sum_{j=-1}^{p} b_j y'(t_n - jh) \end{aligned}$$

If we recall that $\tau(h) = \max_n |\tau_n(h)|$ then (20) is **consistent** if $\tau(h) \to 0$ as $h \to 0$. Also (20) is order q if $\tau(h) = ch^q$.

Example 14. For AB2 we have $u_{n+1} = u_n + \frac{h}{2}(3f_n - f_{n-1})$. Using the general form we have

$$\begin{aligned} p &= 1 & b_{-1} &= 0 \\ a_0 &= 1 & b_0 &= 3/2 \\ a_1 &= 0 & b_1 &= -1/2 \end{aligned}$$

Therefore for the LTE we have

$$\begin{aligned} h\tau_{n+1}(h) &= y_{n+1} - y_n - h\Big(\frac{3}{2}y'_n - \frac{1}{2}y'_{n-1}\Big) \\ &= y(t_n + h) - y(t_n) - h\Big(\frac{3}{2}y'(t_n) - \frac{1}{2}y'(t_n - h)\Big) \end{aligned}$$

Utilizing Taylor Series we have

$$y(t_n + h) = y(t_n) + hy'(t_n) + \frac{h^2}{2}y''(t_n) + \frac{h^3}{6}y'''(c_n)$$

$$y'(t_n - h) = y'(t_n) - hy''(t_n) + \frac{h^2}{2}y'''(\tilde{c}_n)$$

Substituting these into the equation for LTE we have

$$\begin{aligned} h\tau_{n+1}(h) &= hy'(t_n) + \frac{h^2}{2}y''(t_n) + \frac{h^3}{6}y'''(c_n) - h\Big(\frac{3}{2}y'(t_n) - \frac{1}{2}(y'(t_n) - hy''(t_n) + \frac{h^2}{2}y'''(\tilde{c}_n))\Big) \\ &= h^3\Big(\frac{1}{6}y'''(c_n) + \frac{1}{4}y'''(\tilde{c}_n)\Big) \\ &\leq Mh^3 \end{aligned}$$

Thus, we have that $\tau_{n+1} \leq Mh^2$.

7.4 Building Adams methods

Assume that $u_0, \ldots, u_p$ are available. Further take

$$u_{n+1} = u_n + h\sum_{j=-1}^{p} b_j f_{n-j}$$

The question remains of how to obtain values for $b_j's$. The first option is to approximate $f(t, y(t))$ by interpolation on $\{t_{n-p}, \ldots, t_n\}$. In this case we will end up with $b_{-1} = 0$ and get what is classified as an Adams-Bashforth (AB) method. Some examples are included below.

$$p = 1 \quad \Rightarrow \quad u_{n+1} = u_n + \frac{h}{2}(3f_n - f_{n-1})$$

$$p = 2 \quad \Rightarrow \quad u_{n+1} = u_n + \frac{h}{12}(23f_n - 16f_{n-1} + 5f_{n-2})$$

The other case we have to approximate by the interpolant on $\{t_{n-p-1}, \ldots, t_n, t_{n+1}\}$. This will result in an implicit method which is classified as Adams-Moulton (AM). Again some examples are included.

$$p = 0 \quad \Rightarrow \quad u_{n+1} = u_n + \frac{h}{2}(f_n + f_{n+1})$$

$$p = 1 \quad \Rightarrow \quad u_{n+1} = u_n + \frac{h}{12}(5f_{n+1} + 8f_n - f_{n-1})$$

7.5 Backward Difference Method

As in Adams methods, we will interpolate but for this method we interpolate now $y(t)$ on the points $\{t_{n-p-1}, \ldots, t_n, t_{n+1}\}$. This will provide coefficients for a scheme as before.

7.6 Analysis of LMS

$$u_{n+1} = \sum_{j=0}^{p} a_j u_{n-j} + h \sum_{j=-1}^{p} b_j f_{n-j} \tag{20}$$

Theorem 1. (20) is consistent if and only if

$$\sum_{j=0}^{p} a_j = 1 \tag{21}$$

and

$$\sum_{j=-1}^{p} b_j - \sum_{j=0}^{p} j a_j = 1 \tag{22}$$

Moreover, if $y \in C^{q+1}([t_0, t_0 + T])$ then $\tau(h) = O(h^q)$ if and only if

$$\sum_{j=0}^{p} (-j)^i a_j + i \sum_{j=-1}^{p} (-j)^{i-1} b_j = 1, \qquad i = 2, 3, \ldots, q \tag{23}$$

Note that when $i = 1$ (23) becomes (22).

Definition 10. The **1st Characteristic polynomial** of (20) is given by

$$\rho(r) = r^{p+1} - \sum_{j=0}^{p} a_j r^{p-j}$$

Definition 11. The **2nd Characteristic polynomial** of (20) is given by

$$\sigma(r) = b_{-1} r^{p+1} + \sum_{j=0}^{p} b_j r^{p-j}$$

Definition 12. The **Characteristic polynomial** of (20) is given by

$$\Pi(r) = \rho(r) - z\sigma(r)$$

where $z = h\lambda$.

Definition 13. (20) is **convergent** if for any choice of $u_0, \ldots, u_p$ such that

$$\max_{0 \le i \le p} |u_i - y_i| \to 0 \quad \text{as} \quad h \to 0$$

we have that

$$\max_{n \ge p} |u_n - y_n| \le c(h) \to 0 \quad \text{as} \quad h \to 0$$

Definition 14. (20) is **zero stable** or satisfies the root condition if

1. Every root of the $\rho(r)$ has modulus less than or equal to 1.

2. All roots of $\rho(r)$ of modulus one are simple roots.

Theorem 2. If (20) is convergent then it is zero stable.

Proof. Assume that (20) is convergent and let r be a root of ρ. Consider the IVP $y' = 0$, $y(0) = 0$. Then $f(t, y) = y(t) = 0$. Since $f(t, y) = 0$ then the sequence $(u_n)_{n \geq p}$ produced by (20) satisfies

$$u_{n+1} = \sum_{j=0}^{p} a_j u_{n-j}$$

Note that the characteristic equation is exactly $\rho(r) = 0$. For case one we let r be a simple root of ρ. We define $u_i = hr^i$ for $i = 0, \ldots, p$. Clearly $\max |u_i - y_i| \to 0$ as $h \to 0$ for $0 \leq i \leq p$. Thus we have the base step for induction covered, so we consider $u_{p+1} = hr^{p+1}$. Then we have

$$\begin{aligned} u_{p+1} &= \sum_{j=0}^{p} a_j u_{p-j} \\ &= \sum_{j=0}^{p} a_j h r^{p-j} \\ &= h \sum_{j=0}^{p} a_j r^{p-j} \\ &= hr^{p+1} \end{aligned}$$

Now by the convergence assumption we have $\lim_{N\to\infty} |u_N| = 0$ where $h = 1/N$. Direct substitution will result in $\lim_{N\to\infty} |r|^N/N = 0$. This implies that $|r| \leq 1$. For the case of r being a double root we apply the same argument for $u_i = hir^i$ for $i = 0, 1, \ldots, p + 1$.

□

Theorem 3. If (20) is convergent, then it is consistent.

Proof. Assume that (20) is convergent and consider the IVP given by

$$\begin{cases} y' = 0 \\ y(0) = 1 \end{cases}$$

Since we have that $f(t, y) = 0$ then

$$u_{n+1} = \sum_{j=0}^{p} a_j u_{n-j}$$

To prove (21) take $u_0 = \cdots = u_p = 1$. Then we have simply that

$$u_{p+1} = \sum_{j=0}^{p} a_j u_{p-1} = \sum_{j=0}^{p} a_j$$

Now for the assumption of convergence we have

$$\lim_{h\to 0} |u_{p+1} - 1| = 0 \qquad \Rightarrow \qquad \sum_{j=0}^{p} a_j = 1$$

To prove (22) we use the IVP given by

$$\begin{cases} y' = 1 \\ y(0) = 0 \end{cases}$$

□

Theorem 4. (20) is convergent if and only if it is consistent and zero stable.

An important consequence of this theorem is that determination of convergence is reduced to checking algebraic conditions.

First Dahlquist barrier - The highest order of a linear multistep method is $p+1$ if p is odd and $p+2$ if p is even.

Example 15. Prove that the Midpoint scheme given below is convergent.

$$u_{n+1} = u_{n-1} + 2hf_n$$

First check consistency.

$$p = 1 \quad a_0 = 0 \quad a_1 = 1$$

$$b_{-1} = 0 \quad b_0 = 2 \quad b_1 = 0$$

Clearly the sum (21) and (22) are met and so it is consistent. For zero stability we look at the roots of $\rho(r) = r^2 - a_0 r - a_1 = r^2 - 1$. So the roots are ± 1 and both are simple. Thus we have zero stability. By the theorem above we have convergence of the scheme.

7.7 Absolute Stability for LMS

For this we refer back to the test problem

$$\begin{cases} y'(t) = \lambda y(t) \\ y(0) = 1 \\ t \in [0, \infty) \end{cases} \tag{18}$$

Definition 15. (20) is A-stable if $|u_n| \to 0$ as $t_n \to \infty$. The region of A-stability is given by

$$\mathcal{A} = \{z \in \mathbb{C} : z = h\lambda \quad \text{s.t. (20) stable on (18)}\}$$

Example 16. Consider AB2

$$u_{n+1} = u_n + h/2(3f_n - f_{n-1})$$

If we consider the test problem then we have $f_n = \lambda u_n$. Thus

$$u_{n+1} = u_n + h/2(3\lambda u_n - \lambda u_{n-1})$$

Transforming to z and with a little algebra we obtain

$$u_{n+1} = (1 + 3z/2)u_n - zu_{n-1}/2$$

Note that the characteristic equation $\Pi(r) = r^2 - (1 + 3z/2)r - z/2$

For absolute stability we require that the roots have modulus less than 1. Solving yields

$$r_{1,2} = 1/2((1 + 3z/2) \pm \sqrt{1 + z + 9z^2/4})$$

7.8 Runge-Kutta (RK)

We will again consider the Cauchy Problem (12). The motivation for the use of RK methods is their high accuracy, self starting and easy to implement nature.

Main Idea - We find u_{i+1} from u_i using repeated evaluations of f.

One example of and RK method we have seen already is the Trapezoid/Heun's method. This method is a result of using the trapezoid quadrature on (12).

$$u_{n+1} = u_n + \frac{h}{2}\Big(f(t_n, u_n) + f(t_n + h, u_n + hf(t_n, u_n))\Big)$$

Putting this in the form that RK methods are presented in gives

$$\begin{aligned} K_1 &= f(t_n, u_n) \\ K_2 &= f(t_n + h, u_n + hK_1) \\ u_{n+1} &= u_n + \frac{h}{2}(K_1 + K_2) \end{aligned}$$

7.8.1 General RK methods

We can generalize to the form given by

$$u_{n+1} = u_n + h \underbrace{F(t_n, u_n, h, f)}_{\text{Increment function}} \tag{24}$$

where F is defined as

$$F = \sum_{i=1}^{s} b_i K_i$$

$$K_i = f\left(t_n + c_i h, u_n + h \sum_{j=1}^{s} a_{ij} K_j\right)$$

$$s - \text{number of stages}$$

With this form we have a convenient and compact way of writing RK methods known as **Butcher Tables**. These are given in the form

$$\begin{array}{c|ccc} c_1 & a_{11} & \dots & a_{1s} \\ c_2 & a_{21} & \dots & a2s \\ \vdots & \vdots & & \vdots \\ c_s & a_{s1} & \dots & a_{ss} \\ \hline & b_1 & \dots & b_s \end{array}$$

For Heun's method the Butcher Table is

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \end{array}$$

Another important example is RK4 which has the following Butcher Table.

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array}$$

and written explicitly

$$\begin{aligned} K_1 &= f(t_n, u_n) \\ K_2 &= f(t_n + h/2, u_n + hK_1/2) \\ K_3 &= f(t_n + h/2, u_n + hK_2/2) \\ K_2 &= f(t_n + h, u_n + hK_3) \\ u_{n+1} &= u_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \end{aligned}$$

An important note is that the sum over i of a_{ij} is equivalently c_i. We also have that the method is explicit given that $a_{ij} = 0$ for all $j \geq i$.

Now recall the definition of LTE and consistency but now adapted for the general RK form (24). The LTE at t_{n+1} is $\tau_{n+1}(h)$ defined by

$$h\tau_{n+1}(h) = y_{n+1} - y_n - hF(t_n, y_n, h, f)$$

and the method is consistent if $\tau(h) = \max |\tau_n(h)| \to 0$ as $h \to 0$. Below are a couple key facts for RK methods.

1. An RK method is consistent if and only if $\sum_{i=1}^{s} b_i = 1$.

2. The order of an RK method with s stages is in general less than or equal to s.

3. An explicit RK method with $s > 5$ stages is order strictly less than s.

4. For $s \in \{1, 2, 3, 4\}$ there are explicit RK methods of order $\{1, 2, 3, 4\}$ respectively.

8 Systems of ODE's

We will frequently reference the following problem.

$$\begin{cases} y'(t) = F(t, y) \\ y(t_0) = y_0 \in \mathbb{R}^d \\ t \in [t_0, T] \\ F : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d \end{cases} \tag{25}$$

If we are in the linear case then we have that this problem reduces to $y' = Ay$ where A is a real $d \times d$ matrix. Further assuming that A has d distinct eigenvalues and a basis of eigenvectors then we can explicitly write the solution.

$$y(t) = \sum_{j=1}^{d} c_j e^{\lambda_j t} v_j$$

In this solution c_j are any constants and further, if $\Re(\lambda_j) < 0$ for $j = 1, \ldots, d$ then $|y(t)| \to 0$ as $t \to \infty$.

8.1 Decoupling

Taking A as it is above, then if we have a basis of eigenvectors of A, $\{v_j\}_{j=1}^{d}$ then we write Q to be the matrix where the i^{th} column is v_i. Now as these are eigenvectors then $AQ = Q\Lambda$ where Λ is a diagonal matrix that has eigenvalues along the diagonal.

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & & \\ \vdots & & \ddots & \\ 0 & & & \lambda_d \end{bmatrix}$$

Solving for A results in $A = Q\Lambda Q^{-1}$. Plugging this into the linear system we have $y'(t) = Q\Lambda Q^{-1} y$. With this define $z = Q^{-1}y$ which then implies that $y = Qz$. Differentiating both side with respect to t gives the last component of $y' = Qz'$. Gathering all the known information

$$y' = Q\Lambda Q^{-1} y \qquad \Rightarrow \qquad y' = Q\Lambda z$$

$$y' = Qz' \qquad \Rightarrow \qquad Q\Lambda z = Qz'$$

Taking the last equation we reduce it to $z' = \Lambda z$. As Λ is diagonal then we have decoupled the system into d ODE's of the form $z_j' = \lambda_j z_j$.

It is important to note that a method to solve the linear system is A-stable if each of the decoupled problems is A-stable.

8.2 Stiff Problems

Consider the ODE system below.

$$y'(t) = Ay(t) + \varphi(t) \tag{26}$$

Here A is a $d \times d$ matrix. Assume that $\lambda_1, \ldots, \lambda_d$ are distinct eigenvalues of A and $\Re\lambda_j < 0$ for $j = 1, \ldots, d$ with corresponding eigenvectors $v_1, \ldots, v_d$. Stiffness of a problem arises when looking at the eigenvalues. We first consider the following example.

Example 17.

$$y' = Ay$$
$$y(0) = [3, 0]^T$$

where

$$A = \begin{bmatrix} -33.4 & 66.6 \\ 33.3 & -66.7 \end{bmatrix}$$

If we apply a numerical scheme with a bounded region of stability $\mathcal{A}_M$ then the choice of step size depends on the eigenvalues of A. One can calculate that the eigenvalues of $\lambda_1 = -100$ and $\lambda_2 = -1/10$. Therefore we will have one solution that has fast decay and likewise one solution with slow decay. An example of restriction is if we use Euler's method then we require $h < -2/\lambda_j$ for $j = 1, 2$. One may note that the eigenvalues are quite far apart. This gives rise to the following classification.

Definition 16. (26) is said to be stiff if

1. $\Re(\lambda_j) < 0$
2. $\min |\Re(\lambda_j)| << \max |\Re(\lambda_j)|$

Along with this definition we can now define the **stiffness quotient** or the degree of stiffness.

$$S := \frac{\max |\Re(\lambda_j)|}{\min |\Re(\lambda_j)|}$$

9 Two Point BVP

We begin with the model problem where we are given $f = f(x)$ and $x \in (0, 1)$ and we wish to find $u = u(x)$ such that

$$-u''(x) = f(x) \qquad u(0) = u(1) = 0 \tag{27}$$

To build a numerical scheme we begin with a uniform discretization of the domain where $h = 1/n$ and $x_j = hj$. The end goal is to approximate $u(x_j)$ for all the interior points $x_1, \dots, x_{n-1}$. We are only concerned with interior points as we will assume that $u_0 = u_n = 0$ from the given boundary conditions. We turn to the finite difference approximation for the second derivative.

$$u''(x_j) \approx \frac{u(x_j + h) - 2u(x_j) + u(x_j - h)}{h^2}$$

Using the approximation we impose that the following holds true.

$$\begin{cases} u_0 = 0 \\ -\frac{u(x_j - h) - 2u(x_j) + u(x_j + h)}{h^2} = f(x_j) \\ u_n = 0 \end{cases}$$

If we examine the resulting equations themselves

$$-\frac{1}{h^2}(u_0 - 2u_1 + u_2) = f_1$$

$$-\frac{1}{h^2}(u_1 - 2u_2 + u_3) = f_2$$

$$\vdots$$

$$-\frac{1}{h^2}(u_{n-2} - 2u_{n-1} + u_n) = f_{n-1}$$

Note that $u_0 = u_n = 0$, therefore it should be immediate that this can be put into a matrix system.

$$\frac{1}{h^2} A_{FD} u = f$$

where $A_{FD} = \text{tridiag}(-1, 2, 1)$ and $u = (u_1, \dots, u_{n-1})^T$ and likewise for f. If we wish to solve this we need to answer weather A_{FD} is invertible or not. Let $x \in \mathbb{R}^{n-1}$ and consider $x^T A_{FD} x$.

$$\begin{aligned} x^T A_{FD} x &= \sum_{j=1}^{n-1} \sum_{i}^{n-1} a_{ij} x_i x_j \\ &= 2 \sum_{i=1}^{n-1} x_i^2 - 2 \sum_{j=2}^{n-1} x_{j-1} x_j \\ &= \sum_{i=2}^{n-1} (x_{i-1} - x_i)^2 + x_1^2 + x_{n-1}^2 \end{aligned}$$

Now as this is a sum of non negative terms them the inner product $x^T A_{FD} x \geq 0$ and only zero when $x = 0$. Therefore A_{FD} is symmetric positive definite and thus invertible. With this guaranteed invertiblity we let $u = (u_1, \dots, u_{n-1})^T$ be the unique solution of

$$\begin{cases} A_{FD} u = f \\ u(0) = u(1) = 0 \end{cases} \tag{28}$$

We can interpret u as a function on $[0, 1]$ given by the piecewise linear interpolant for

x	x_0	x_1	$\dots$	x_{n-1}	x_n
u	0	u_1	$\dots$	u_{n-1}	0

To proceed we will define two spaces.

$$V_h = \{v : v \in \mathbb{R}^{n-1}\}$$

$$V_h^0 = \{v \in V_h : v_0 = v_n = 0\} \subset V_h$$

We will also define the discrete second order differential operator on $w \in V_h$ by

$$(\mathcal{L}_h w)(x_j) = -\frac{w_{j-1} - 2w_j + w_{j+1}}{h^2}$$

Using this operator we can write an equivalent form for (28).

$$\begin{cases} (\mathcal{L}_h u)(x_j) = f(x_j) & j = 1, \dots, n-1 \\ u \in V_h^0 \end{cases} \tag{29}$$

9.1 Stability Analysis

To do stability analysis on (29) we will consider the **Energy Method**. For this we will define the inner product on V_h

$$(w, v)_h = h \sum_{k=0}^{n} c_k w_k v_k$$

where $c_0 = c_n = 1/2$ and $c_1 = \cdots = c_{n-1} = 1$. This inner product will induce a norm given by

$$||v||_h^2 = h \sum_{k=0}^{n} c_k v_k^2$$

An important note to make is that $\mathcal{L}_h$ is symmetric and positive definite on V_h^0.

Definition 17. The **Energy Norm** on V_h^0 is defined by

$$|||v|||_h = \left(h \sum_{j=0}^{n-1} \left(\frac{v_{j+1} - v_j}{h} \right)^2 \right)^{1/2} = (\mathcal{L}_h v, v)^{1/2}$$

Lemma 2. (Poincare-Type Inequality) The following inequality holds.

$$||v||_h \leq \frac{1}{\sqrt{2}} |||v|||_h \qquad \forall v \in V_h^0$$

Returning to (29) we have $\mathcal{L}_h u = f$, so clearly it must remain true that

$$(\mathcal{L}_h u, v)_h = (f, v)_h \qquad \forall v \in V_h^0$$

and in particular as $u \in V_h^0$ then we get

$$\begin{aligned} |||u|||_h^2 &= (\mathcal{L}_h u, u)_h \\ &= (f, u)_h \\ &\leq ||f||_h ||u||_h \end{aligned}$$

At this point we now use Lemma 2(Poincare) on the energy norm of u.

$$|||u|||_h^2 \leq \frac{1}{\sqrt{2}} ||f||_h |||u|||_h \quad \Rightarrow \quad |||u|||_h \leq \frac{1}{\sqrt{2}} ||f||_h$$

This is stability in the energy norm. It is also worth noting that $\mathcal{L}_h$ is injective as $|||u|||_h \leq \frac{1}{\sqrt{2}} ||\mathcal{L}_h u||_h$ and if $\mathcal{L}_h u = 0$ then $|||u||_h = 0$ i.e. $u = 0$. Now as V_h^0 is finite dimensional then the operator $\mathcal{L}_h$ is invertible. If we instead consider the infinity norm $|| \cdot ||_{h,\infty}$ in the expected manor then as $\mathcal{L}_h$ is invertible it can be proved that

$$c||w||_{h,\infty} \leq ||\mathcal{L}_h w||_{h,\infty} \tag{30}$$

and in fact the best constant we can have is $c = 8$.

9.2 Convergence

If we have $u_h \to u$ then $||u_h - u||_\alpha = O(h^\alpha)$ then our goal in the following sections will be to determine α.

9.2.1 Consistency

For this we assume $f \in C^2([0,1])$, and $u \in C^n([0,1])$ where u is the solution of (27) and u_h is the solution of (29). In this case the local truncation error is given by the grid function

$$\tau_h(x_j) = (\mathcal{L}_h u)(x_j) - f(x_j) = \frac{u(x_j - h) - 2u(x_j) + u(x_j + h)}{h^2} - f(x_j)$$

Utilizing the Taylor expansion for $u(x_j \pm h)$ up to order 4 and the difference equation $-u'' = f$ we get that

$$\begin{aligned}
\tau_h(x_j) &= \frac{h^2}{12} u^{(4)}(\xi_j) \qquad \xi_j \in (x_{j-1}, x_{j+1}) \\
&\Downarrow \\
|\tau_h(x_j)| &\leq \frac{h^2}{12} \max_{x \in [0,1]} |u^{(4)}(x)| \\
&= \frac{h^2}{12} \max_{x \in [0,1]} |f''(x)| \\
&= \frac{h^2}{12} ||f''||_\infty
\end{aligned}$$

The key equation we will use is given below

$$||\tau_h||_{h,\infty} \leq \frac{h^2 ||f''||_\infty}{12} \tag{31}$$

Returning back to convergence, we define $e = u - u_h$ at the grid points. Clearly we have $e(0) = e(1) = 0$ and $e(x_j) = u(x_j) - u_j$. If we consider the action of $\mathcal{L}_h$ on e then notice that

$$\begin{aligned}
(\mathcal{L}_h e)(x_j) &= (\mathcal{L}_h u)(x_j) - (\mathcal{L}_h u_h)(x_j) \\
&= \frac{u(x_j - h) - 2u(x_j) + u(x_j + h)}{h^2} - \frac{u_{j-1} - 2u_j + u_{j+1}}{h^2} \\
&= \frac{u(x_j - h) - 2u(x_j) + u(x_j + h)}{h^2} - f(x_j) \\
&= \tau_h(x_j)
\end{aligned}$$

Upon combination of (30) and (31) we have

$$\begin{aligned}
||e||_{h,\infty} &\leq \frac{1}{8} ||\mathcal{L}_h e||_{h,\infty} \\
&= \frac{1}{8} ||\tau_h(u_h)||_{h,\infty} \\
&\leq \frac{h^2}{96} ||f''||_\infty
\end{aligned}$$

Therefore we arrive at

$$||u(x_j) - u_j||_{h,\infty} \leq \frac{h^2}{96} ||f''||_\infty$$

9.3 Neumann Boundary Conditions

We present now the same boundary value problem, but now with general Neumann boundary conditions on one end.

$$\begin{cases} -u''(x) = f(x) \\ u'(0) = \sigma \\ u(1) = \beta \end{cases} \tag{32}$$

To handle the Neumann type boundary condition we have three options immediate options as to how we proceed.

9.3.1 Forward Difference

As one may have expected, we will use the forward finite difference formula for the first derivative.

$$u'(0) \approx \frac{u(h) - u(0)}{h}$$

Upon substitution of the B.C. yields

$$\frac{u(h) - u(0)}{h} = \sigma$$

With this equation we proceed as before where we set up a system of $n - 1$ equations. One downfall is that this approximation is order one and if we are looking for higher order this can be a major road block.

9.3.2 Ghost Point

As mentioned in the previous section of text, the desired order can dictate the method that we deal with Neumann B.C.'s. If we now consider introducing a new unknown $u_{-1} = u(-h)$, this will allow us to use an order two centered difference formula.

$$u'(0) \approx \frac{u(h) - u(-h)}{2h}$$

Upon substitution of the B.C. yields

$$\frac{u_1 - u_{-1}}{2h} = \sigma$$

As we do not have a value of u_{-1} we wish to eliminate this from the set of equations. To do this we note that the following is true.

$$u''(0) \approx \frac{u-1 - 2u_0 + u_1}{h^2} = f_0$$

Solving for u_{-1} in the first and substituting it into the second gives

$$\frac{-u_0 + u_1}{h^2} = \frac{f_0}{2} + \frac{\sigma}{h}$$

9.3.3 Finite difference

If we wish to preserve the order of a method we are using then we simple derive a forward difference formula to the order we desire using $u(0), u(h), u(2h), \ldots$.

10 Heat Equation

Given the heat equation we wish to find $u = u(x,t)$ such that

$$\begin{cases} \frac{\partial u}{\partial t} + Lu = f & x \in (0,1), t > 0 \\ u(0,t) = u(1,t) = 0 & t > 0 \\ u(x,0) = u_0(x) & x \in [0,1] \end{cases} \tag{33}$$

where

$$L = -\mu \frac{\partial^2}{\partial x^2} \qquad \mu > 0$$

and f is the external force term. In the case that $f = 0$ then we can explicitly write the solution.

$$u(x,t) = \sum_{n=1}^{\infty} c_n e^{-n^2\pi^2 t} \sin(n\pi x)$$

$$c_n = 2\int_0^1 u_0(x)\sin(n\pi x)\,dx$$

To those that are interested, this will come from separation of variables. If $f \neq 0$ then things are more complicated. However we do have this general fact

$$E(t) = \int_0^1 u^2(x,t)\,dx$$

then

$$E(t) \leq e^{-\gamma t}E(0) + \frac{1}{\gamma}\int_0^t e^{\gamma(s-t)}F(s)\,ds$$

where

$$F(t) = \int_0^1 f^2(x,t)\,dx \qquad \gamma = \frac{\mu}{C_p^2}$$

10.1 Method of Lines

The main idea of the Method of Lines (MOL) is to use only the spatial discretization and then solve the time continuous system. For this we let $x_i = hi$ where $h = 1/n$. Then $u(x_i,t)$ is defined on a line and is a function of only t. This is true for $i = 1,\dots,n-1$. For convenience of notation we will denote $u(x_i,t)$ by $u_i(t)$. Now returning to the heat equation we can use a second order approximation for L

$$Lu \approx -\frac{\mu}{h^2}(u_{i-1}(t) - 2u_i(t) + u_{i+1}(t))$$

We now take this approximation an plug it into (33).

$$u_i'(t) - \frac{\mu}{h^2}(u_{i-1}(t) - 2u_i(t) + u_{i+1}(t)) = f_i(t) \qquad i = 1,\dots,n-1$$

Now again for convenience we write

$$f(t) = [f_1(t), f_2(t), \dots, f_{n-1}(t)]^T$$
$$u(t) = [u_1(t), u_2(t), \dots, u_{n-1}(t)]^T$$
$$u_0 = [u_0(x_1), u_0(x_2), \dots, u_0(x_{n-1})]^T$$

It should be immediate that we can rewrite (33) now as the following

$$\begin{cases} u'(t) = -\frac{\mu}{h^2}A_{FD}u(t) + f(t) \\ u(0) = u_0 \end{cases} \tag{34}$$

where $A_{FD} = \text{tridiag}_{n-1}(-1,2,-1)$ as in section 9.

10.1.1 Properties of A_{FD}

It is worth looking at properties for A_{FD} as the heat equation has many real world applications. If we define

$$\theta_j = \frac{j\pi}{n}$$

and

$$v_j = [\sin(\theta_j), \sin(2\theta_j), \dots, \sin((n-1)\theta_j)]^T$$

It can be easily checked using basic trig identities that $A_{FD}v_j = (2 - 2\cos(\theta_j))v_j$. Thus, v_j are eigenvectors with corresponding eigenvalues $\lambda_j = 2 - 2\cos(\theta_j) = 4\sin^2(\theta_j/2)$. Therefore the spectrum of $-\frac{\mu}{h^2}A_{FD}$ is

$$\{-\mu\lambda_j/h^2 : j = 1,\dots,n-1\}$$

If we look at the stiffness quotient $S = \lambda_{n-1}/\lambda_1$ for this problem then

$$\begin{aligned} S &= \frac{\sin^2((n-1)\pi/2n)}{\sin^2(\pi/2n)} \\ &\approx \frac{4n^2}{\pi^2} \end{aligned}$$

Thus, as we increase the number of mesh points, then the stiffness of the problem increases which can cause issue. To demonstrate this, consider using Forward Euler for solving this problem. As the region of absolute stability if given by $|z+1| < 1$ then we require

$$\lambda_j \Delta t > -2$$

for all λ_j which then implies that

$$\Delta t < \frac{h^2}{2\mu \sin^2((n-1)\pi/2n)}$$

Note that as $h \to 0$ then we have extreme restriction on Δt.

10.2 θ-method

For the θ-method we use the spatial discretization used in the MOL but now we also use a uniform time discretization $t^k = k\Delta t$. For notation we use

$$\begin{aligned} u^k &= [u_1(t^k), u_2(t^k), \ldots, u_{n-1}(t^k)]^T \\ f^k &= [f_1(t^k), f_2(t^k), \ldots, f_{n-1}(t^k)]^T \end{aligned}$$

With this we define the scheme to be

$$\begin{aligned} \frac{u^{k+1} - u^k}{\Delta t} &= -\frac{\mu}{h^2} A_{FD}(\theta u^{k+1} + (1-\theta)u^k) + \theta f^{k+1} + (1-\theta) f^k \\ u^0 &= [u_0(x_1), u_0(x_2), \ldots, u_0(x_{n-1})]^T \end{aligned}$$

Notice that when $\theta = 0$ we get back Forward Euler and $\theta = 1$ gives Backward Euler. A quick rearrangement yields

$$(I + \frac{\mu\theta\Delta t}{h^2} A_{FD})u^{k+1} = (I - \frac{\mu\Delta t(1-\theta)}{h^2} A_{FD})u^k + g^{k+1}$$

where

$$g^{k+1} = \Delta t(\theta f^{k+1} + (1-\theta) f^k)$$

If we let $A = \frac{\mu\Delta t}{h^2} A_{FD}$ then

$$u^{k+1} = (I + \theta A)^{-1}((I - (1-\theta)A)u^k + g^{k+1})$$