Numerical Analysis - Exam

P1

a) Runge phenomenon is to describe extreme "polynomial wiggle" associated with high degree polynomial interpolation at evenly spaced points.

How to reduce: move some of the interpolation points towards the outside of the interval, where the function produce data can be better fit

Interpolation error: $\frac{\prod_{i=1}^{n}(x-x_i)}{n!} f^{n}(c) \quad x \in [-1,1]$

So we choose x_i $i \in \{1, \dots n\}$ to make ~~min of~~ $\max_{-1 \le x \le 1} |\prod (x - x_i)|$ as small as possible.

$\Rightarrow$ choose $x_i = \cos \frac{(2i-1)\pi}{2n}$ $\Rightarrow$ min value is $\frac{1}{2^{n-1}}$

Generalize to any $[a,b]$ interval

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

To prove this, we only need to prove $[-1, 1]$ interval

Let P_n monic polynomial.

T_n is chebyshev polynomial $(T_n(x) = \cos(n \arccos(x)))$

Assume $|P_n(x)| \le 1/2^{n-1}$

Since T_n alternate between -1 and 1 $n+1$ times

(It has ~~preeeee~~ roots at $\cos \frac{i\pi}{n}$ $i = 0, \dots n$)

$\Rightarrow P_n - T_n/2^{n-1}$ alternate positive and negative $n+1$ times

at these points

$\Rightarrow P_n - T_n/2^{n-1}$ crosses zeros at least n times.

$\Rightarrow P_n - T_n/2^{n-1}$ has at least n ~~pruus~~ roots

$\Rightarrow$ contradict ~~the~~ the degree difference is $\le n-1$

$\Rightarrow \max_{-1 \le x \le 1} |P_n| \ge 1/2^{n-1}$

$\Rightarrow$ for $[a,b]$ interval $x_i = \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\frac{(2i-1)\pi}{2n}\right)$

$\Rightarrow$ ~~min of max~~ $\left|\prod_{i=1}^{n} (x - x_i)\right| \le \left(\frac{b-a}{2}\right)^n / 2^{n-1}$

(2)

b) Runge - Kutta pairs is pairs of Runge - Kutta methods one of order p and another of order $p+1$.

Adapt ~~time~~ step's size

Define $e_{i+1} \approx |z_{i+1} - w_{i+1}|$ (z_{i+1} and w_{i+1} is 2 ~~new~~ estimation of Runge - Kutta methods)

estimated error

Let ε = error tolerance

If $|e_{i+1}| > |\varepsilon|$

$\Rightarrow h_{i+1} = h_i/2$ (we cut the step size in ~~two~~ half)

~~If~~ else if $|e_{i+1}| \le |\varepsilon|$

$h_{i+1} = h^*$ (Choose ~~an~~ a new appropriate step size base on h_i : $h^* = g(h_i)$.)

③

c) Stability of a finite difference method solution for the heat equation is when the error is amplified when the time step k is too large relative to spatial ~~and~~ step h

The necessary condition: CFL condition

for heat equation, we need

$$\sigma = \frac{Dk}{h^2} \leq 1$$

With
D: diffusion coefficient
k: time step (temporal step size)
h: spatial step

$\boxed{P2}$

a) Matlab code below.

b). $$\begin{bmatrix} 3 & 7 \\ 6 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -11 \end{bmatrix}$$

$$A = A_1 = \begin{bmatrix} 3 & 7 \\ 6 & 1 \end{bmatrix} \Rightarrow M_1 = \begin{bmatrix} 1 & 0 \\ -2 & 1 \end{bmatrix}$$

$$\Rightarrow M_1^{-1} = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}$$

~~2 row 1 - row 2~~
$\Downarrow$ row 2 − 2 row 1

$$A_2 = \begin{bmatrix} 3 & 7 \\ 0 & -13 \end{bmatrix} \quad b_2 = \begin{bmatrix} 1 \\ -13 \end{bmatrix}$$

$$\Rightarrow U = A_2 = \begin{bmatrix} 3 & 7 \\ 0 & -13 \end{bmatrix}; L = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}; b_2 = \begin{bmatrix} 1 \\ -13 \end{bmatrix}$$

④

Back substitution

- $Lc = b$

$$\Rightarrow \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -11 \end{bmatrix}$$

$$\Rightarrow \begin{cases} c_1 = 1 \\ 2c_1 + c_2 = -11 \end{cases} \Rightarrow \begin{cases} c_1 = 1 \\ c_2 = -13 \end{cases}$$

- $Ux = c$ for x

$$\Rightarrow \begin{bmatrix} 3 & 7 \\ 0 & -13 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -13 \end{bmatrix}$$

$$\Rightarrow \begin{cases} 3x_1 + 7x_2 = 1 \\ -13x_2 = -13 \end{cases}$$

$$\Rightarrow \begin{cases} x_1 = \cancel{1} -2 \\ x_2 = 1 \end{cases} \Rightarrow x = \begin{bmatrix} -2 \\ 1 \end{bmatrix}$$

$$\left[v_i \right] \quad \left[v_{i+1} = f(u_{i+1}) \right.$$

(5)

$\boxed{P3}$

a) The Jacobi methods apply ~~both~~ value calculated by the previous ~~sthep~~ step onto both L and U

Jacobi

x_0 = initial vector

$$x_{k+1} = D^{-1}\left(b - (L+U)\, x_k\right)$$

for $k = 0, 1, 2 \ldots$

The Gaussian-Seidel use the latest updated value ~~of~~ onto L and the value of previous ~~step~~ step

Gaussian - Seidel

x_0 = initial vector

$$x_{k+1} = D^{-1}\left(b - L\, x_{k+1} - U x_k\right)$$

for $k = 0, 1, 2 \ldots$

latest updated value of $x_{i\,k+1}$

$i = 1, 2 \ldots n$

(b) I run the matrices A, b ~~answers~~ with matlab and ~~they~~ it converge for both Jacobi and Gaussian Seidel.

(c) Check Matlab code

$\boxed{P4}$ ⑥

$$f(x) = f(w) + f'(w)(x-w) + \frac{f''(c_x)}{2}(x-w)^2$$

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

prove composite Midpoint Rule ~~apple~~.

$$\int_a^b f(x)\,dx = h\sum_{i=1}^{m} f(w_i) + \frac{(b-a)h^2}{24} f''(c)$$

$\forall [a,b]$ for a given h

~~Let m~~ $x_0 = a$, $x_m = b$, $h = x_m - x_{m-1} = \dots = x_1 - x_0$

~~On $[x_{i-1}, x_{i+1}]$ choose w_i~~ $h = \frac{b-a}{m}$

On $[x_{i-1}, x_i]$ choose $w_i = \frac{x_i - x_{i-1}}{2}$ mid point

$$\int_{x_{i-1}}^{x_i} f(x)\,dx = \int_{x_{i-1}}^{x_i} f(w_i) + f'(w_i)(x-w_i) + f''(c_{x_i})(x-w_i)^2$$

$$= \Big[x f(w_i) \cancel{\ } f'(w_i)\frac{(x-w_i)^2}{2} + f''(c_{x_i})\frac{(x-w_i)^3}{2\times 3} \Big]_{x_{i-1}}^{x_i}$$

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

$$+ f''(c_{x_i})\frac{\overbrace{(x_i-w_i)^3}^{+h^3/8} - \overbrace{(x_{i-1}-w_i)^3}^{-h^3/8}}{\cancel{2\times 3}\ 6}$$

$$= h f(W_i) + 0 + f''(c_{x_i})\frac{h^3}{24} \quad (1)$$

$$\int_a^b f(x)\,dx = \sum_{i=1}^{m} \int_{x_{i-1}}^{x_i} f(x)\,dx$$

$$\stackrel{(1)}{=} \sum_{i=1}^{m} h f(W_i) + f''(c_{x_i})\frac{h^3}{24}$$

$$= h\sum_{i=1}^{m} f(w_i) + \frac{h^3}{24}\sum_{i=1}^{m} f''(c_{x_i})$$

7

$$\int_a^b f(x)\,dx = h\sum_{i=1}^{m} f(w_i) + \frac{h^3}{24}\, m f''(c)$$

$$= h\sum_{i=1}^{m} f(w_i) + \frac{(b-a)h^2 f''(c)}{24}$$

$$\Rightarrow \int_a^b f(x)\,dx = h\sum_{i=1}^{m} f(w_i) + \frac{b-a}{24} h^2 f''(c)$$

$(a < c < b)$

$\boxed{P5}$ found in Matlab code below.

$\boxed{P6}$ $f(t,y)$ Lipschitz constant $L = 1$ for y

$$\begin{cases} y' = f(t,y) \\ y(0) = y_0 \\ t \text{ in } [0,T] \end{cases}$$

y_i at t_i approximated by one-step ODE.

local truncation error $e_i \le C h^{k+1}$

$C = 1,\ k \ge 0$

$\forall\ 0 \le t_i \le T$ global truncation error

$$g_i = |w_i - y_i| \le \frac{C h^k}{L}\left(e^{L t_i} - 1\right)$$

Estimate global truncation errors at $T = 10$ for

a) Euler's method with $e_i \le C h^2$

b) midpoint method with $e_i \le C h^3$

if at $T = 1$, 2 method have same local truncation error $g_1 = 0.1$, $h = 0.1$

8

a) We choose $h = 0.1$

~~$e_i \leq c h^2 = 1 \times 1^2 = 1$~~

~~$\Rightarrow e_i \leq 1$~~

for Euler's method

$$e_i \leq c h^2 \Rightarrow k = 1.$$

$$\Rightarrow g_i = |w_i - y_i| \leq \frac{c h^1}{L} (e^{L t_i} - 1).$$

with $c = 1, h = 0.1, L = 1, t_i = T$

$$g_i \leq \frac{1 \times 0.1^1}{1} (e^{1 \times 10} - 1)$$

$$\Rightarrow g_i \leq 0.1 (e^{10} - 1)$$

b) We choose $h = 0.1$

for Mid-point method

$$e_i \leq c h^3 \Rightarrow k = 2$$

$$g_i = |w_i - y_i| \leq c \frac{h^2}{L} (e^{L t_i} - 1)$$

with $c = 1, h = 0.1, L = 1, t_i = T$

$$\Rightarrow g_i \leq \frac{1 \times 0.1^2}{1} (e^{1 \times 10} - 1)$$

$$\Rightarrow g_i \leq 0.01 (e^{10} - 1)$$