Problem 1 $y_n - y_{n-2} = h(f_n - 3f_{n-1} + 4f_{n-2})$

(a) This is an implicit method since y_n is defined implicitly above

(b) The polynomials associated with this method are:

$$p(z) = z^2 - 1$$
$$q(z) = z^2 - 3z + 4$$

<u>stability</u>:

The roots of p are ± 1, both of magnitude 1, but simple
$\Rightarrow$ <u>method is stable</u>.

<u>Consistency</u>.

$p(1) = 0$

$p'(1) = 2$
$\|$
$q(1) = 1 - 3 + 4 = 2$

<u>method is consistent</u>

Since method is stable and consistent, it must be <u>convergent</u>.

Problem 2 A is assumed to be symmetric $(A = A^T)$

(a)

<u>Power method</u>

$v^{(0)}$ = some vector with $\|v^{(0)}\| = 1$
for $k = 1, 2, \dots$
$w^{(k)} = A v^{(k-1)}$
$v^{(k)} = w^{(k)} / \|w^{(k)}\|$
$\lambda^{(k)} = v^{(k)T} A v^{(k)}$

<u>QR algorithm</u>

$A^{(0)} = Q_0^* A Q_0 = (\diagdown\!\diagdown\!\diagdown)$
(reduction to tridiagonal form)

for $k = 1, 2, \dots$
$Q^{(k)} R^{(k)} = A^{(k-1)}$
$A^{(k)} = R^{(k)} Q^{(k)}$

eigenvalues are in diag $(A^{(k)})$

(6)

Fundamental differences between methods:

- power method converges to largest eigenvalue (and eigenvector) in magnitude
- QR gives all eigenvalues of the matrix.

Power method:

advantages:

- does not need matrix A, only how to apply A to a vector
- cheap for large matrices
- quadratic convergence to eigenvalue

disadvantages:

- only gives largest eigenvalue
- can have slow convergence if largest eigenvalue in magnitude is not simple

QR algorithm

advantages:

- gives all eigenvalues at once.

disadvantages

- can be expensive (specially reduction to tridiagonal form)
- needs shifts to accelerate convergence
- needs to store whole matrix.

Problem 3

(a) Taylor expanding $y(t_{i+1})$, $y(t_{i-1})$ around $y(t_i)$ we get

① $y(t_{i+1}) = y(t_i) + h y'(t_i) + \frac{h^2}{2} y''(t_i) + \frac{h^3}{6} y'''(t_i) + O(h^4)$

② $y(t_{i-1}) = y(t_i) - h y'(t_i) + \frac{h^2}{2} y''(t_i) - \frac{h^3}{6} y'''(t_i) + O(h^4)$

①+②:

$$y(t_{i+1}) + y(t_{i-1}) = 2y(t_i) + h^2 y''(t_i) + O(h^4)$$

$$\Rightarrow y''(t_i) = \frac{y(t_{i+1}) - 2y(t_i) + y(t_{i-1})}{h^2} + O(h^2),$$

(b)

$t_0 \quad t_1 \quad \dots \quad t_n \quad t_{n+1}$

The nodes t_1 and t_n are special cases because of the boundary conditions.

$$\left\{\begin{aligned}
&\frac{1}{h^2}\Big(y_2 - 2y_1 + \underbrace{y_0}_{=\alpha}\Big) = p_1 \frac{1}{2h}\Big(y_2 - \underbrace{y_0}_{=\alpha}\Big) + q_1 y_1 + r_1 \\
&\frac{1}{h^2}\left(y_{i+1} - 2y_i + y_{i-1}\right) = p_i \frac{1}{2h}\left(y_{i+1} - y_{i-1}\right) + q_i y_i + r_i \quad \text{for } i = 2 \dots n-1 \\
&\frac{1}{h^2}\Big(\underbrace{y_{n+1}}_{=\beta} - 2y_n + y_{n-1}\Big) = p_n \frac{1}{2h}\Big(\underbrace{y_{n+1}}_{=\beta} - y_{n-1}\Big) + q_n y_n + r_n
\end{aligned}\right.$$

(c) In system form we get: $AU = B$ with

$$A = L - D - Q$$

where

$$A = \frac{1}{h^2}\begin{bmatrix} -2 & 1 & & \\ 1 & -2 & 1 & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{bmatrix}$$

$$D = \frac{1}{2h}\begin{bmatrix} 0 & p_1 & & & \\ -p_2 & 0 & p_2 & & \\ & -p_3 & 0 & p_3 & \\ & & \ddots & \ddots & p_{n-1} \\ & & & -p_n & 0 \end{bmatrix} \qquad p_i = p(t_i)$$

$$Q = \begin{bmatrix} q_1 & & & \\ & q_2 & & \\ & & \ddots & \\ & & & q_n \end{bmatrix} \qquad q_i = q(t_i)$$

and

$$B = R - \frac{\alpha p_1}{2h} e_1 + \frac{\beta p_n}{2h} e_n - \frac{\alpha}{h^2} e_1 - \frac{\beta}{h^2} e_n$$

$$R = \begin{bmatrix} r_1 \\ \vdots \\ r_n \end{bmatrix}, \quad r_i = r(t_i)$$

or in other words:

$$B = \begin{bmatrix} r_1 - \frac{\alpha p_1}{2h} - \frac{\alpha}{h^2} \\ r_2 \\ \vdots \\ r_{n-1} \\ r_n + \frac{\beta p_n}{2h} - \frac{\beta}{h^2} \end{bmatrix}$$

Problem 4

(a) We simply multiply PDE by test function $v \in V$ and integrate:

$$\int_0^1 -u'' v \, dx = \int_0^1 f v \, dx \qquad \forall v \in V$$

$\parallel$ IBP

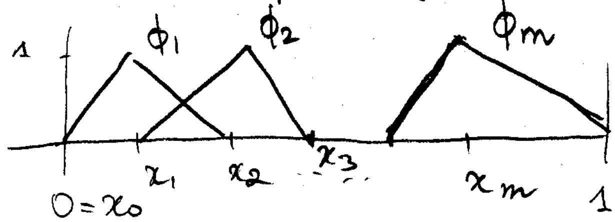
$$\underbrace{-u'v\Big|_0^1}_{=0 \text{ because } v \in V} + \int_0^1 u'v' \, dx$$

Thus if u solves PDE:

$$a(u,v) = (f,v) \quad \forall v \in V, \text{ where}$$

$$a(u,v) = \int_0^1 u'v' \, dx \text{ and } (f,v) = \int_0^1 f v \, dx.$$

(b) P1 finite elements means we use nodal values and the functions we approximate with are piecewise linear. So the finite dimensional subspace $V_h \subset V$ of functions we use is composed of "hat functions"



$V_h = \text{span}\{\phi_i\}$.

The Ritz Galerkin problem:

Find $u_h \in V_h$ s.t. $a(u_h, v_h) = (f, v_h) \quad \forall v_h \in V_h$.

produces the "best" approximation to the solution u in V_h (best in the sense of the energy norm) and is equivalent to solving the linear system:

$$AU = F, \text{ where } A_{ij} = a(\phi_i, \phi_j)$$
$$F_j = (f, \phi_j)$$
$$\text{and} \quad u_h = \sum_{i=1}^{n} U_i \phi_i$$

Here is some pseudocode to solve the problem above.

```
/* Global to local map */
i = [ 1    2
      2    3
      :    :
      m+1  m+2 ]

/* Assemble stiffness matrix and RHS */
for e = 1 .. m+1
  | A(i(e,:), i(e,:)) = A(i(e,:), i(e,:)) + A^e_loc
  | F(i(e,:))         = F(i(e,:)) + F^e_loc
```

Here $(A_{loc}^{e})_{ij} = \int_{I_e} (\phi_i^e)'(\phi_j^e)'\,dx$ = local stiffness matrix

$(F_{loc}^{e})_i = \int_{I_e} \phi_i^e f\,dx$ = local RHS

/* Take into account homog. Dirichlet B.C. */

$A(1,:) = 0;\ A(1,1) = 1;\ F(1) = 0;$

$A(m+1,:) = 0;\ A(m+1,m+1) = 1;\ F(m+1) = 0;$

/* Solve System */

U = LinearSolve (A, F);

Problem 5

(a) Method of lines simply means we discretize PDE first in space:

$0 = x_0 \quad x_1 \quad x_2 \quad \cdots \quad x_m \quad x_{m+1} = 1$

$$U_1'(t) = -\frac{a}{2h}\big(U_2(t) - \underbrace{U_0(t)}_{= U_{m+1}(t) \text{ by periodic B.C.}}\big)$$

$$U_i'(t) = -\frac{a}{2h}\big(U_{i+1}(t) - U_{i-1}(t)\big), \quad i = 2, \ldots, m$$

$$U_{m+1}'(t) = -\frac{a}{2h}\big(\underbrace{U_{m+2}(t)}_{= U_1(t) \text{ by periodic B.C.}} - U_m(t)\big)$$

⚠ be sure to use right discretization.

Therefore method of lines gives:

$$U'(t) = AU(t)$$

where

$$A = -\frac{a}{2h}\begin{bmatrix} 0 & 1 & & & -1 \\ -1 & 0 & 1 & & \\ & -1 & 0 & 1 & \\ & & & \ddots & 1 \\ 1 & & & -1 & 0 \end{bmatrix}$$

elements at corners are important for B.C.

and

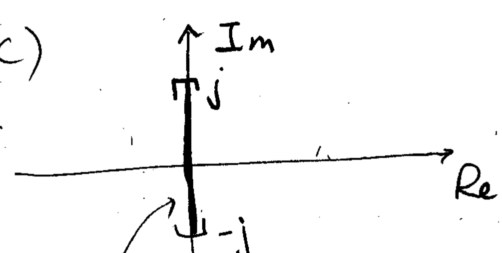
$$U(t) = \begin{bmatrix} U_1(t) \\ U_2(t) \\ \vdots \\ U_{m+1}(t) \end{bmatrix}$$

(b) Discretization in time gives:

$$\frac{U^{n+1} - U^{n-1}}{2k} = AU^n$$

$$U^{n+1} = U^{n-1} + 2k\,AU^n$$

(c)



Absolute stability region

For stability of system of ODEs we need to find k s.t.

$k\lambda_p(A) \in$ absolute stability region, $p = 1, \dots, m+1$

Since $\left| \operatorname{Im} \lambda_p(A) \right| \leq \frac{|a|}{h}$ ← ⚠ don't forget absolute values.

in order to have $|\operatorname{Im} k\lambda_p(A)| \leq 1$ we need

$$\frac{k\,|a|}{h} \leq 1 \iff \boxed{k \leq \frac{h}{|a|}}$$