Solutions to Examination in Scientific Computing

1. (a) Let \( v = x - y \) for \( x \) and \( y \) such that \( x^T x = y^T y \). The matrix

\[
P = I - 2 \frac{vv^T}{v^Tv}
\]

is a Householder matrix and

\[
P = x - 2 \left( \frac{(x - y)(x - y)^T}{x^Ty - 2x^Ty} \right) x = x - 2 \frac{(x - y)(x^T x - y^T x)}{2x^Ty - 2x^Ty}
\]

\[
x = x - (x - y) = y.
\]

(b) If

\[
A = \begin{pmatrix}
a_{11} & \ldots & a_{1n} \\
\vdots & \ddots & \vdots \\
a_{n1} & \ldots & a_{nn}
\end{pmatrix}
\]

and

\[
P_1 = I - 2 \frac{vv^T}{v^Tv}
\]

where \( v = x - y \), \( x = (a_{11}, \ldots, a_{n1})^T \) and \( y = (\|x\|_2, 0, \ldots, 0)^T \), then

\[
P_1 A = \begin{pmatrix}
\|x\|_2 & * & \ldots & * \\
0 & \ddots & \vdots & \\
0 & \ddots & A_2 \\
0 & \ddots & \ddots & \ddots
\end{pmatrix}.
\]

Similarly, if

\[
P_2 = \begin{pmatrix}
1 & 0 & \ldots & 0 \\
0 & \ddots & \vdots & \\
0 & \ddots & \ddots & \ddots \\
0 & \ddots & \ddots & \ddots
\end{pmatrix}
\]

where \( \hat{P}_2 \) is a Householder matrix satisfying

\[
\hat{P}_2 A_2 = \begin{pmatrix}
* & * & \ldots & * \\
0 & \ddots & \vdots & \\
0 & \ddots & A_3 \\
0 & \ddots & \ddots & \ddots
\end{pmatrix},
\]

then

\[
P_2 P_1 A = \begin{pmatrix}
* & * & \ldots & * \\
0 & * & \ldots & * \\
0 & \ddots & \ddots & \ddots \\
0 & \ddots & \ddots & \ddots
\end{pmatrix},
\]

Repeating gives \( A = P_1 \ldots P_{n-1} R = QR \), where \( Q = P_1 \ldots P_{n-1} \) is unitary and \( R \) is upper triangular.

(c) Let \( A = QR \) be a \( QR \)-factorisation. Since multiplication with a unitary matrix does not change the 2-norm,

\[
\|Ax - b\|_2 = \|Q^H(QRx - b)\|_2 = \|Rx - Q^Hb\|_2.
\]
2. (a) On standard form, the difference scheme reads

\[ \frac{u_j^{n+1} - u_j^n}{k} = D_+u_j^n + \frac{k}{2}D_-u_j^n, \]

A Taylor expansion shows that

\[ \frac{u_j^{n+1} - u_j^n}{k} = u_t + \frac{k}{2}u_{tt} + \frac{k^2}{6}u_{ttt} + O(k^3) \]

\[ D_+u_j^n = u_x + \frac{h}{2}u_{xx} + \frac{h^2}{2}u_{xxx} + O(h^3) \]

\[ D_+D_-u_j^n = u_{xx} + \frac{h^2}{12}u_{xxxx} + O(h^4) \]

giving a truncation error

\[ \tau(k, h) = \frac{u_j^{n+1} - u_j^n}{k} - D_+u_j^n - \frac{k}{2}D_-u_j^n \]

\[ = u_t - u_x + \frac{k}{2}(u_{tt} - u_{xx}) + \frac{k^2}{6}u_{ttt} - \frac{h}{2}u_{xx} + O(k^3) + O(h^3). \]

From the PDE follows that \( u_t = u_x \) and \( u_{tt} = u_{xx} = u_{tx} \) giving

\[ \tau(k, h) = O(k^2) + O(h). \]

(b) The discrete Fourier transform of the difference equation is \( \hat{u}_m^{n+1} = \hat{g}_m \hat{u}_m^n \), where

\[ \hat{g}_m = 1 + \lambda(e^{2\pi im/N} - 1) = 1 + \lambda \left( \cos \frac{2\pi m}{N} + i \sin \frac{2\pi m}{N} - 1 \right). \]

The scheme is stable if and only if \(|\hat{g}_m| \leq 1\) for all \( m \) and \( N \). Let \( \mu = 2\pi m/N \) and note that \( 0 \leq \mu < 2\pi \). Then

\[ |\hat{g}|^2 = (1 - \lambda + \lambda \cos \mu)^2 + \lambda^2 \sin^2 \mu = 1 - 2\lambda + 2\lambda^2 - 2\lambda(\lambda - 1) \cos \mu. \]

Extremums are assumed when

\[ \frac{d}{d\mu} |\hat{g}|^2 = 2\lambda (\lambda - 1) \sin \mu = 0. \]

Since \( \lambda > 0 \), this is true if \( \lambda = 1, \mu = 0 \), or \( \mu = \pi \). If \( \lambda = 1, |\hat{g}|^2 = \cos^2 \mu + \sin^2 \mu = 1 \). If \( \mu = 0, |\hat{g}|^2 = 1 \). Finally, if \( \mu = \pi, |\hat{g}|^2 = (1 - 2\lambda)^2 \), which is less than or equal to one if and only if \( 0 \leq \lambda \leq 1 \). To conclude, the approximation is stable if and only if \( 0 < \lambda \leq 1 \).

3. (a) Jacobi’s method uses \( M = -D^{-1}(L + U) \) and \( c = D^{-1}b \), where \( L \) is the lower triangular, \( D \) the diagonal, and \( U \) the upper triangular part of \( A \). One finds that

\[ M = \begin{pmatrix}
0 & -1/3 \\
-1/3 & 0 & -1/3 \\
& & \ddots & \ddots \\
& & \ddots & 0 & -1/3 \\
& & & -1/3 & 0 & -1/3 \\
& & & & -1/3 & 0
\end{pmatrix} \]

and

\[ c = \begin{pmatrix}
1 \\
1 \\
\vdots \\
1
\end{pmatrix}. \]

(b) Jacobi’s method converges since \( A \) is strictly diagonally dominant.
4. (a) Let $V$ be the space

$$V = \{ v \mid v \text{ cont on } [0,1], v' \text{ piecewise cont on } [0,1], v(0) = v(1) = 0 \}.$$ 

If $u$ solves the ODE and $v \in V$, it follows from integration by parts that

$$\int_0^1 v(x)f(x) \, dx = \int_0^1 v(x)u''(x) \, dx = [v(x)u'(x)]_0^1 - \int_0^1 v'(x)u'(x) \, dx$$

Since $v(0) = v(1) = 0$, the boundary terms vanish. The variational formulation is: Find $u \in V$ such that

$$- \int_0^1 v'(x)u'(x) \, dx = \int_0^1 v(x)f(x) \, dx, \quad \forall v \in V.$$

(b) Let $V_h$ be given by

$$V_h = \{ v \mid v \text{ cont on } [0,1], v \text{ linear on } [x_j, x_{j+1}], v(0) = v(1) = 0 \},$$

where $x_j = jh$, $j = 0, \ldots, N$, and $h = 1/(N+1)$. The piecewise linear functions $\phi_j(x) \in V_h$, $j = 1, \ldots, N$, satisfying

$$\phi_j(x_i) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

are a basis in $V_h$. Thus, any function $u_h \in V_h$ can be written

$$u_h(x) = \sum_{j=1}^N c_j \phi_j(x),$$

for some constants $c_j$. The FEM is given by: Find $u_h \in V_h$ such that

$$- \int_0^1 v_h'(x)u_h'(x) \, dx = \int_0^1 v_h(x)f(x) \, dx, \quad \forall v_h \in V_h,$$

or equivalently,

$$- \sum_{j=1}^N c_j \int_0^1 \phi'_j(x)\phi'_j(x) \, dx = \int_0^1 \phi_i(x)f(x) \, dx, \quad i = 1, \ldots, N.$$

This is a large system of equations $Ac = b$, where

$$A_{i,j} = - \int_0^1 \phi'_i(x)\phi'_j(x) \, dx \quad \text{and} \quad b_i = \int_0^1 \phi(x)f(x) \, dx.$$

Since $\phi_i(x) = 0$ when $x < x_{i-1}$ or $x > x_{i+1}$, the coefficients $A_{i,j}$ are different from zero only when $i = j$ or $i - j = \pm 1$. Then,

$$A_{j,j} = - \int_0^1 (\phi'_j(x))^2 \, dx = \ldots = -\frac{2}{h},$$

and

$$A_{j,j+1} = A_{j-1,j} = - \int_0^1 \phi'_j(x)\phi'_{j+1}(x) \, dx = \ldots = \frac{1}{h}.$$ 

If $f(x) = 1$,

$$b_j = \int_0^1 \phi_j(x) \, dx = \ldots = h.$$
5. (a) According to the given algorithm and since $F$ is unitary

$$x = Fy = FA^{-1}z = FA^{-1}F^Hb = FA^{-1}F^{-1}b = (FAF^{-1})^{-1}b = C^{-1}b.$$ 

(b) Step (a) consists of a matrix-vector multiplication with a dense matrix. For each element in the vector $z$, $N$ multiplications and $N-1$ additions is performed. Since $z$ contains $N$ elements, the algorithm requires a total of $O(N^2)$ arithmetic operations. Step (b) consists of a matrix-vector multiplication with a diagonal matrix. This requires only $O(N)$ arithmetic operations. Step (c) is similar to (a) and requires $O(N^2)$ arithmetic operations. In total, the algorithm requires $O(N^2)$ arithmetic operations.

(c) Noting that the discrete Fourier transform can be computed by multiplication with the Fourier matrix $F$, one realizes that step (a) can be computed by FFT, which is an $O(N \log_2 N)$ algorithm. Similarly, step (c) can be computed by inverse FFT. The FFT algorithm uses the fact that the transform can be computed by combining results from transforms of two vectors of half length. Repeating recursively saves a lot of arithmetic operations.