Solution of Exercise 11

Question 2. We want to show that there is a positive β, independent of the mesh size h, such that

$$\min_{\mathbf{e}_H \in \Omega_H} \|\mathbf{e}^h - I_H^h \mathbf{e}^H\|_D^2 \le \beta \|\mathbf{e}^h\|_{A_h}^2.$$

for the 1D Poisson matrix

$$A_h = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{n\times n}, \qquad n \text{ odd.}$$

Here, I_H^h denotes *linear interpolation*.
The eigenvectors of $A_h$are given by

$$\boldsymbol{q}_k = (\sin\vartheta_k, \sin 2\vartheta_k, \ldots, \sin n\vartheta_k)^T, \quad \vartheta_k = \frac{k\pi}{n+1}, \quad k = 1, \ldots, n,$$

with associated eigenvalue

$$\lambda_k = 4\sin^2\frac{\vartheta_k}{2}.$$

To estimate β in the approximation property

$$\min_{\boldsymbol{e}_H \in \Omega_H} \|\boldsymbol{e}^h - I_H^h \boldsymbol{e}^H\|_D^2 \le \beta \|\boldsymbol{e}^h\|_{A_h}^2 \tag{AP}$$

we first set $\boldsymbol{e}^h = \boldsymbol{q}_k$, and $\boldsymbol{e}^H$ is the vector that interpolates $\boldsymbol{e}^h$ at all even-numbered nodes. At the odd-numbered nodes the difference is

$$\begin{aligned} e_j^h - \frac{1}{2}(e_{j-1}^h + e_{j+1}^h) &= \sin j\vartheta_k - \frac{1}{2}(\sin(j-1)\vartheta_k + \sin(j+1)\vartheta_k) \\ &= (1-\cos\vartheta_k)\sin j\vartheta_k = 2\sin^2\frac{\vartheta_k}{2}\sin j\vartheta_k \end{aligned}$$

So,

$$\|\boldsymbol{e}^h - I_H^h \boldsymbol{e}^H\|_D^2 = 2\sin^2\frac{\vartheta_k}{2}\sum_{j=1}^{(n+1)/2}\sin^2(2j-1)\frac{\vartheta_k}{2} = (n+1)\sin^2\frac{\vartheta_k}{2}.$$

The weight 2 before the sum stems from D. Likewise, since $\boldsymbol{q}_k$ is an eigenvector of A_h,

$$\|\boldsymbol{e}^h\|_{A_h}^2 = \boldsymbol{q}_k^T A_h \boldsymbol{q}_k = 2(n+1)\sin^2\frac{\vartheta_k}{2}.$$

So, we for $\boldsymbol{e}^h = \boldsymbol{q}_k$ inequality (AP) holds with $\beta = 1/2$. A general $\boldsymbol{e}^h$ can be represented as a linear combination of the eigenvectors,

$$\boldsymbol{e}^h = \sum_{k=1}^n \eta_k \boldsymbol{q}_k.$$

Thus,

$$\|\boldsymbol{e}^h\|_{A_h}^2 = \sum_{j,k=1}^n \eta_j\eta_k \boldsymbol{q}_j^T A_h \boldsymbol{q}_k = \sum_{j,k=1}^n \eta_j\eta_k 4\sin\frac{\vartheta_k}{2}\boldsymbol{q}_j^T\boldsymbol{q}_k = 2(n+1)\sum_{k=1}^n \eta_k^2 \sin\frac{\vartheta_k}{2}.$$

Similarly,

$$\|\boldsymbol{e}^h - I_H^h \boldsymbol{e}^H\|_D^2 = (n+1)\sum_{k=1}^{n} \eta_k^2 \sin^2 \frac{\vartheta_k}{2}.$$

So, inequality (AP) with $\beta = 1/2$ holds for general $\boldsymbol{e}^h$.

Question 3. The code `md2Dsolver` below generates the required matrices for the 2-grid method, ω and D for the damped Jacobi smoother, restriction R, prolongation P, fine and coarse grid matrices A_h and A_H.

```
%% Exercise 11.3 Solution
%% mg2Dsolver.m

global Ah AH LH P R D omega

%n=33;
N=(n-1)/2;

I1=speye(n);
Ah1=gallery('tridiag',n,-1,2,-1);
AH1=gallery('tridiag',N,-1,2,-1)/4;

Ah=kron(Ah1,I1) + kron(I1,Ah1);
AH=kron(AH1,eye(N)) + kron(eye(N),AH1);
LH=chol(AH,'lower');

I=kron(I1,I1);  D=4*I;

R=kron(abs(Ah1),abs(Ah1));

ind=reshape(1:n^2,n,n);
J=ind(2:2:n-1,2:2:n-1);% Indices of coarse grid points
J=J(:);

R=R(:,J); P=R;
R=R'/16; P=P/4;         % Restriction, prolongation

omega=4/5;              % Optimal 2D smoothing parameter
%%S=(I-omega*(D\Ah));   % Iteration matrix of smoother
%%T=I - P*(AH\R)*Ah;    % Iteration matrix of coarse grid correction
%%M=S*T*S;              % Iteration matrix for complete 2-grid preconditioning

xe = rand(n^2,1); b = Ah*xe; % exact solution, right-hand side

tic, [x,flag,relres,iter] = pcg(Ah,b,1e-6,500,D); etime=toc;
if (flag>0) flag, end
fprintf(['Jacobi preconditioner: n = %d, iter = %d, ',...
         'relres = %8.2e, time = %8.2e\n'],n,iter,relres,etime)

tic, [x,flag,relres,iter] = pcg(Ah,b,1e-6,50,@mfun1mfun); etime=toc;
if (flag>0) flag, end
fprintf(['2-grid preconditioner: n = %d, iter = %d, ',...
         'relres = %8.2e, time = %8.2e\n'],...
        n,iter,relres,etime)
return

function [x] = mfun1mfun(r)
%% preconditioner solves M x = r
```

```
global Ah AH LH P R D omega
n = length(r);

x = omega*(D\r);              % pre-smooting with x=0

rr = r - Ah*x;
x = x + P*(LH'\(LH\(R*rr))); % coarse-grid correction

rr = r - Ah*x;
x = x + omega*(D\rr);         % post-smooting

return
end
```

The iteration counts for PCG, preconditioned with (i) Jacobi and (ii) 2-grid solver with damped Jacobi smoother, are given in the table below.

	Jacobi precond.		2-grid precond.	
n	it. steps	time [s]	it. steps	time [s]
33	78	2.69e-02	7	3.29e-02
65	133	3.30e-02	7	7.77e-03
97	178	4.14e-02	7	1.35e-02
129	242	9.75e-02	7	2.97e-02

The stunning effect of the 2-grid preconditioner is the constant iteration count. Note that the solver is faster than Jacobi-preconditioned CG only for larger problem sizes.
In the approximation property the constant $\beta = 1/2$. The constant in the smoothing property is $\alpha = \omega(2\omega\rho(D^{-1}A_h)) = 0.32$ as $\rho(D^{-1}A_h)) \lessapprox 2$. So, the contraction factor in the theorem on Slide 11-34 is

$$\sqrt{1 - \frac{\alpha}{\beta}} = \sqrt{1 - 0.32/0.6} \approx 0.683.$$

The contraction rate refers to the A_h-norm of $S_h T_h^H$,

$$\max_{\boldsymbol{x}\neq\mathbf{0}} \left(\frac{\boldsymbol{x}^T T^T S^T A_h S T \boldsymbol{x}}{\boldsymbol{x}^T A_h \boldsymbol{x}} \right)^{1/2} = \max_{\boldsymbol{y}\neq\mathbf{0}} \frac{\|L_h^T S T L_h^{-T} \boldsymbol{y}\|_2}{\|\boldsymbol{y}\|_2} = \|L_h^T S T L_h^{-T}\|_2,$$

where $A_h = L_h L_h^T$ is the Cholesky factorization of A_h. The value of this norm is obtained in MATLAB by the command

```
max(svd(full(L'*S*T/L')))
```

The matrices S and T are given in the MATLAB code above. For $n = 65$ the value is 0.5995.