1. Exercise 3 from chap. 4 of text [p. 127 os SIAM edition]

2. Exercise 8 from chap. 4 of text [p. 127 os SIAM edition] – questions (a) and (b) only.

3. Let $A$ be a non-singular matrix and assume that one step of a minimal residual iteration is performed. Call $x$ the current iterate, $x_{\text{new}}$ the new iterate, and $r$ and $r_{\text{new}}$ their associate residuals respectively.

(a) Show that
\[ \|r_{\text{new}}\|^2_2 = \|r\|^2_2 - \frac{(Ar,r)^2}{(Ar,Ar)} \]
where $\angle(x,y)$ represents the acute angle between $x$ and $y$. Does this show that the algorithm converges (for any nonsingular matrix)?

(b) Show the same formula as above (i.e., $\|r_{\text{new}}\|^2_2 = \|r\|^2_2 \sin^2 \angle(Ar,r)$) by using a purely geometric argument. One case when the algorithm converges is when $A$ is positive definite. Explain why. [short explanation – this is in the text!]

(c) Now consider the situation when $A$ is a normal matrix. Assume that all eigenvalues $\lambda_i$ of $A$ have a (strictly) positive real part. Then show that for any vector $r$ we have
\[ |(Ar,r)| \geq \mu_{\text{min}} \|r\|^2_2 \]
where $\mu_{\text{min}}$ is the smallest real part of the $\lambda_i$'s. Establish convergence of the algorithm in this case.

4. Consider the linear system $Ax = b$, where $A$ is a nonsingular matrix. We define a projection method which uses a two-dimensional space at each step. At a given step, we take $K = \text{Span}\{r, Ar\}$, where $r = b - Ax$ is the current residual, and $L = AK$.

(a) For a basis of $K$ we use the vectors
\[ p_1 = \frac{r}{\|Ar\|^2_2} \]
and the vector $p_2 = Ap_1 - \gamma p_1$ such that $Ap_2$ is orthogonal to $Ap_1$. Give the formula for computing $p_2$.

(b) Write the algorithm for performing the projection method described above.
(c) To which other method is this algorithm mathematically equivalent? When will the algorithm converge for any initial guess \( x_0 \)?

5. We consider a system \( Ax = b \) which arises from from a convection-diffusion equation and is generated from the script
\[
A = \text{fd3d}(nx, ny, nz, alpx, alpy, alpz, dshift)
\]
notations:
\[
x = 30; \quad ny = 40; \quad nz = 1; \quad alpx = 0.20; \quad alpy = -0.30; \quad alpz = 0.0; \quad dshift = 0.3
\]
The right-hand side is the vector \( b = Ae^* \), where \( e^* = [1, -1, 1, -1, \cdots, 1, -1]^T \). For initial guesses to the solvers, always take a zero vector.

(a) Write a matlab script to solve the linear system by using right-preconditioned FOM (non-restarted). You will need to modify the FOM script that will be posted in the class web-site to incorporate right-preconditioning. It is best to pass the preconditioner in a matlab struct (say PRE) which may contain the factors PRE.L and PRE.U (L, U) factors which are needed for the solve with \( M = LU \) (do not compute \( M = LU \) each time and then use back-slash to solve with it! Instead compute these before calling FOM). Always limit the number of FOM steps to a maximum of 300. The iteration should be stopped when the residual norm has been reduced by a factor of 1.e-07.

Next run FOM without preconditioning (’FOM-nopre’), with SSOR preconditioner (’FOM-ssor’), with the ILU(0) preconditioner, and then with the ILUT(0.05) preconditioner. For SSOR use \( \omega = 1 \). The ILU(0) factorization can be obtained from matlab as \( \text{ilu}(A) \) 1. The more elaborate ILU with threshold (ILUT) preconditioner, can be obtained by using \( \text{ilu} \) from matlab using the setting struct. Use a drop tolerance of 0.05 Show a new plot which now includes all four runs (no-pre, ssor, ilu(0), and ilu) on the same figure.

(b) Modify your FOM script so that it returns the last Hessenberg matrix obtained from the Arnoldi process. Compute the eigenvalues of the Hessenberg matrices for all four cases (no-pre, ssor, ilu(0), and ilu) in 4 small figures. The plots should show a small circle for each point (eigenvalue) in the complex plane.

(c) Any comments on the relations between the spectra obtained and the speeds of convergence?

---

1 Earlier versions of matlab: \( \text{luinc} \). Note \( \text{luinc} \) permutes for stability by default.