1. Let $A$ be a real symmetric $n \times n$ matrix. A powerful algorithm for finding a specific eigenpair of $A$ starting from an initial approximation for the eigenvector is the Rayleigh quotient iteration. The Rayleigh quotient is a function $Q : \mathbb{R}^n \to \mathbb{R}$ defined by

\[ Q(x) := \frac{x^\top Ax}{x^\top x}. \]

The Rayleigh quotient iteration algorithm is written below. The norm $\|\cdot\|$ in it is the 2-norm.

**Input** An initial guess for the desired eigenvector: $w \in \mathbb{R}^n$. A symmetric $n \times n$ matrix $A$. Tolerance $\text{tol}$.

**Initialization** Normalize the approximation to eigenvector: $v = w/\|w\|$;
Compute approximation to eigenvalue: $\lambda = v^\top Av$;
Compute the residual: $r = Av - \lambda v$;

**The main body**

\[
\text{while } \|r\| > \text{tol} \text{ do}
\]
1: Solve $(A - \lambda I)w = v$ for $w$;
2: Normalize: $v = w/\|w\|$;
3: Update $\lambda$: $\lambda = v^\top Av$;
4: Recompute the residual: $r = Av - \lambda v$;
end

**Algorithm 1:** Rayleigh quotient iteration

**Remark** The Rayleigh quotient iteration converges cubically, i.e., the order of convergence is 3. This is a rare luck!

(a) (2 pts) Show that if $\{\lambda, v\}$ is an eigenpair of $A$ then $\{ (\lambda - \mu)^{-1}, v \}$ is an eigenpair of $(A - \mu I)^{-1}$.

(b) (3 pts) Verify that if $v$ is an eigenvector of $A$ then: (i) $Q(v) = \lambda$, the corresponding eigenvalue, and (ii) $\nabla Q(v) = 0$, i.e., $v$ is a stationary point of $Q(x)$.

(c) (5 pts) In step 1 of the Rayleigh quotient algorithm, we solve the system $(A - \lambda I)w = v$. The matrix $(A - \lambda I)$ is close to singular if $\lambda$ is close to an actual eigenvalue of $A$. However, step (1) is not ill-conditioned. Explain why using your knowledge of condition numbers.

(d) (5 pts) Let $A$ be a $N \times N$ matrix of the form

\[
A := \begin{bmatrix}
2 & -1 \\
-1 & 2 & -1 \\
& & \ddots & \ddots & \ddots \\
& & & -1 & 2
\end{bmatrix}.
\]

This matrix often arises in finite difference methods: $h^{-2}A$ ($h = 1/N$ is the step in space) is the approximation to the second derivative operator.
The eigenvalues and eigenvectors of $A$ are well-known: for $k = 1, 2, \ldots, N$,

$$
\lambda_k = -4 \left( \sin \frac{\pi k}{2N + 2} \right)^2, \quad v_k = \left[ \sin \left( \frac{\pi k}{N + 1} \right), \sin \left( \frac{2\pi k}{N + 1} \right), \ldots, \sin \left( \frac{N\pi k}{N + 1} \right) \right]^T.
$$

Set $N = 100$. This matrix $A$ is set up in Matlab using the commands (see Matlab’s help on `spdiags` if you are not familiar with this command):

```matlab
N = 100;
e = ones(N,1);
A = spdiags([e,-2*e,e],-1:1,N,N);
```

Pick the initial approximation for an eigenvector:

```matlab
x = linspace(0,1,N)';
v = x.*(1-x); % '.*' means that the multiplication is performed entrywise
```

Implement the Rayleigh quotient algorithm to compute an eigenpair starting from this approximation. Display the found eigenvalue with 15 digits after comma (format `%.15e`).

To which eigenvalue of $A$ did the algorithm converge? How many iterations were performed?

Do the same task for $v = x(1-x)(x-1/2)$ and $v = x(1-x)(x-1/3)(x-2/3)$.

Submit your code and its printout.

2. (10 pts) Problem 1 in Chapter 5 in `BindelGoodman.pdf` (page 123) posted on ELMS. Hint: to do (a), count flops in reducing the $n \times (2n)$ matrix $[A, I]$ to $[U, L^{-1}]$ using Gaussian elimination.