Study Material on Numerical Methods and Programming

Semester – IV (Hons.)

Instructor – Subhendu Saha

Department of Physics

The Jacobi Method

***Two assumptions made on Jacobi Method*:**

1.The system given by

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots a_{2n}x_n &= b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots a_{nn}x_n &= b_n \end{aligned}$$

Has a unique solution.

2.The coefficient matrix A has no zeros on its main diagonal, namely, $a_{11}, a_{22}, \ldots, a_{nn}$ are nonzeros.

Main idea of Jacobi

To begin, solve the 1st equation for x_1, the 2nd equation for x_2 and so on to obtain the rewritten equations:

$$x_1 = \frac{1}{a_{11}}(b_1 - a_{12}x_2 - a_{13}x_3 - \cdots a_{1n}x_n)$$

$$x_2 = \frac{1}{a_{22}}(b_2 - a_{21}x_1 - a_{23}x_3 - \cdots a_{2n}x_n)$$

$$\vdots$$

$$x_n = \frac{1}{a_{nn}}(b_n - a_{n1}x_1 - a_{n2}x_2 - \cdots a_{n,n-1}x_{n-1})$$

Then make an initial guess of the solution $\boldsymbol{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, \ldots x_n^{(0)})$. Substitute these values into the right hand side the of the rewritten equations to obtain the *first approximation*, $\left(x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, \ldots x_n^{(1)}\right)$.

This accomplishes one **iteration**.

In the same way, the *second approximation* $\left(x_1^{(2)}, x_2^{(2)}, x_3^{(2)}, \ldots x_n^{(2)}\right)$ is computed by substituting the first approximation's value $\left(x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, \ldots x_n^{(1)}\right)$ into the right hand side of the rewritten equations.

By repeated iterations, we form a sequence of approximations $\boldsymbol{x}^{(k)} = \left(x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, \ldots x_n^{(k)}\right)^t, \quad k = 1,2,3, \ldots$

The Jacobi Method. For each $k \geq 1$, generate the components $x_i^{(k)}$ of $\boldsymbol{x}^{(\boldsymbol{k})}$ from $\boldsymbol{x}^{(\boldsymbol{k-1})}$ by

$$x_i^{(k)} = \frac{1}{a_{ii}}\left[\sum_{\substack{j=1, \\ j \neq i}}^{n} (-a_{ij} x_j^{(k-1)}) + b_i\right], \qquad \text{for } i = 1,2,, \ldots n$$

Example. Apply the Jacobi method to solve

$$5x_1 - 2x_2 + 3x_3 = -1$$
$$-3x_1 + 9x_2 + x_3 = 2$$
$$2x_1 - x_2 - 7x_3 = 3$$

Continue iterations until two successive approximations are identical when rounded to three significant digits.

Solution

n	$k = 0$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$
$x_1^{(k)}$	0.000	-0.200	0.146	0.192			
$x_2^{(k)}$	0.000	0.222	0.203	0.328			
$x_2^{(k)}$	0.000	-0.429	-0.517	-0.416			

When to stop: 1. $\frac{||\boldsymbol{x}^{(k)}-\boldsymbol{x}^{(k-1)}||}{||\boldsymbol{x}^{(k)}||} < \varepsilon$; or $\left|\left|\boldsymbol{x}^{(k)}-\boldsymbol{x}^{(k-1)}\right|\right| < \varepsilon$. Here ε is a given small number . Another stopping criterion: $\frac{||\boldsymbol{x}^{(k)}-\boldsymbol{x}^{(k-1)}||}{||\boldsymbol{x}^{(k)}||}$

Definition 7.1 A **vector norm** on R^n is a function, $||\cdot||$, from R^n to R with the properties:

(i) $||\boldsymbol{x}|| \geq 0$ for all $\boldsymbol{x} \in R^n$

(ii) $\left|\left|\boldsymbol{x}\right|\right| = 0$ if and only if $\boldsymbol{x} = \boldsymbol{0}$

(iii) $\left|\left|\alpha\boldsymbol{x}\right|\right| = |\alpha|\left|\left|\boldsymbol{x}\right|\right|$ for all $\alpha \in R$ and $\boldsymbol{x} \in R^n$

(iv) $\left|\left|\boldsymbol{x}+\boldsymbol{y}\right|\right| \leq \left|\left|\boldsymbol{x}\right|\right| + ||\boldsymbol{y}||$ for all $\boldsymbol{x}, \boldsymbol{y} \in R^n$

Definition 7.2 The **Euclidean norm** l_2 and the **infinity norm** l_∞ for the vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^t$ are defined by

$$||\boldsymbol{x}||_2 = \left\{\sum_{i=1}^{n} {x_i}^2\right\}^{\frac{1}{2}}$$

and

$$||\boldsymbol{x}||_\infty = \max_{1 \le i \le n} |x_i|$$

Example. Determine the l_2 and l_∞ norms of the vector $x = (-1,1,-2)^t$.

Solution:

$$||\boldsymbol{x}||_2 = \sqrt{(-1)^2 + (1)^2 + (-2)^2} = \sqrt{6}.$$

$$||\boldsymbol{x}||_\infty = \max\{|-1|, |1|, |-2|\} = 2.$$

The Jacobi Method in Matrix Form

Consider to solve an $n \times n$ size system of linear equations $A\boldsymbol{x} = \boldsymbol{b}$ with

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \text{ and } \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \text{ for } \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}.$$

We split A into

$$A = \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{bmatrix} - \begin{bmatrix} 0 & \dots & 0 & 0 \\ -a_{21} & \dots & 0 & 0 \\ \vdots & & \ddots & \vdots \\ -a_{n1} & \dots & -a_{n,n-1} & 0 \end{bmatrix}$$
$$- \begin{bmatrix} 0 & -a_{12} & \dots & -a_{1n} \\ 0 & 0 & & \vdots \\ \vdots & \vdots & \ddots & -a_{n-1,n} \\ 0 & 0 & \dots & 0 \end{bmatrix} = D - L - U$$

Where $D = \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{bmatrix}$ $L = \begin{bmatrix} 0 & \dots & 0 & 0 \\ -a_{21} & \dots & 0 & 0 \\ \vdots & & \ddots & \vdots \\ -a_{n1} & \dots & -a_{n,n-1} & 0 \end{bmatrix}$,

$$U = \begin{bmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ 0 & 0 & & \vdots \\ \vdots & \vdots & \ddots & -a_{n-1,n} \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

$A\boldsymbol{x} = \boldsymbol{b}$ is transformed into $(D - L - U)\boldsymbol{x} = \boldsymbol{b}$.

$$D\boldsymbol{x} = (L + U)\boldsymbol{x} + \boldsymbol{b}$$

Assume D^{-1} exists and $D^{-1} = \begin{bmatrix} \frac{1}{a_{11}} & 0 & \cdots & 0 \\ 0 & \frac{1}{a_{22}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{a_{nn}} \end{bmatrix}$

Then

$$\boldsymbol{x} = D^{-1}(L + U)\boldsymbol{x} + D^{-1}\boldsymbol{b}$$

The matrix form of Jacobi iterative method is

$$\boldsymbol{x}^{(k)} = D^{-1}(L + U)\boldsymbol{x}^{(k-1)} + D^{-1}\boldsymbol{b} \qquad k = 1,2,3, \ldots$$

Define $T_j = D^{-1}(L+U)$ and $\boldsymbol{c} = D^{-1}\boldsymbol{b}$, Jacobi iteration method can also be written as

$$\boldsymbol{x}^{(k)} = T_j\boldsymbol{x}^{(k-1)} + \boldsymbol{c} \qquad k = 1,2,3,\ldots$$

The Gauss-Seidel Method

Main idea of Gauss-Seidel

With the Jacobi method, only the values of $x_i^{(k)}$ obtained in the kth iteration are used to compute $x_i^{(k+1)}$. With the Gauss-Seidel method, we use the new values $x_i^{(k+1)}$ as soon as they are known. For example, once we have computed $x_1^{(k+1)}$ from the first equation, its value is then used in the second equation to obtain the new $x_2^{(k+1)}$, and so on.

Example. Use the Gauss-Seidel method to solve

$$5x_1 - 2x_2 + 3x_3 = -1$$
$$-3x_1 + 9x_2 + x_3 = 2$$
$$2x_1 - x_2 - 7x_3 = 3$$

Choose the initial guess $x_1 = 0, x_2 = 0, x_3 = 0$

n	$k = 0$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$
$x_1^{(k)}$	0.000	-0.200	0.167				
$x_2^{(k)}$	0.000	0.156	0.334				
$x_2^{(k)}$	0.000	-0.508	-0.429				

The Gauss-Seidel Method. For each $k \geq 1$, generate the components $x_i^{(k)}$ of $\boldsymbol{x}^{(\boldsymbol{k})}$ from $\boldsymbol{x}^{(\boldsymbol{k-1})}$ by

$$x_i^{(k)} = \frac{1}{a_{ii}}\left[-\sum_{j=1}^{i-1}(a_{ij}x_j^{(k)}) - \sum_{j=i+1}^{n}(a_{ij}x_j^{(k-1)}) + b_i\right], \qquad \text{for } i = 1,2,,\ldots n$$

Namely,

$$a_{11}x_1^{(k)} = -a_{12}x_2^{(k-1)} - \cdots - a_{1n}x_n^{(k-1)} + b_1$$

$$a_{22}x_2^{(k)} = -a_{21}x_1^{(k)} - a_{23}x_3^{(k-1)} - \cdots - a_{2n}x_n^{(k-1)} + b_2$$

$$a_{33}x_3^{(k)} = -a_{31}x_1^{(k)} - a_{32}x_2^{(k)} - a_{34}x_4^{(k-1)} - \cdots - a_{3n}x_n^{(k-1)} + b_3$$

$$\vdots$$

$$a_{nn}x_n^{(k)} = -a_{n1}x_1^{(k)} - a_{n2}x_2^{(k)} - \cdots - a_{n,n-1}x_{n-1}^{(k)} + b_n$$

Matrix form of Gauss-Seidel method.

$$(D - L)\boldsymbol{x}^{(k)} = U\boldsymbol{x}^{(k-1)} + \boldsymbol{b}$$

$$\boldsymbol{x}^{(k)} = (D - L)^{-1}U\boldsymbol{x}^{(k-1)} + (D - L)^{-1}\boldsymbol{b}$$

Define $T_g = (D - L)^{-1}U$ and $\boldsymbol{c}_g = (D - L)^{-1}\boldsymbol{b}$, Gauss-Seidel method can be written as

$$\boldsymbol{x}^{(k)} = T_g\boldsymbol{x}^{(k-1)} + \boldsymbol{c}_g \qquad\qquad k = 1,2,3, \ldots$$

Convergence theorems of the iteration methods

Let the iteration method be written as

$$\boldsymbol{x}^{(k)} = T\boldsymbol{x}^{(k-1)} + \boldsymbol{c} \qquad \text{for each } k = 1,2,3, \ldots$$

Definition 7.14 The **spectral radius** $\rho(A)$ of a matrix A is defined by

$$\rho(A) = \max|\lambda|, \qquad \text{where } \lambda \text{ is an eigenvalue of } A.$$

Remark: For complex $\lambda = a + bj$, we define $|\lambda| = \sqrt{a^2 + b^2}$.

Lemma 7.18 If the spectral radius satisfies $\rho(T) < 1$, then $(I - T)^{-1}$ exists, and

$$(I - T)^{-1} = I + T + T^2 + \cdots = \sum_{j=0}^{\infty} T^j$$

Theorem 7.19 For any $\boldsymbol{x}^{(0)} \in R^n$, the sequence $\{\boldsymbol{x}^{(k)}\}_{k=0}^{\infty}$ defined by

$$\boldsymbol{x}^{(k)} = T\boldsymbol{x}^{(k-1)} + \boldsymbol{c} \qquad \text{for each } k \geq 1$$

converges to the unique solution of $\boldsymbol{x} = T\boldsymbol{x} + \boldsymbol{c}$ if and only if $\rho(T) < 1$.

Proof (only show $\rho(T) < 1$ is sufficient condition)

$\boldsymbol{x}^{(k)} = T\boldsymbol{x}^{(k-1)} + \boldsymbol{c} = T\left(T\boldsymbol{x}^{(k-2)} + \boldsymbol{c}\right) + \boldsymbol{c} = \cdots = T^k\boldsymbol{x}^{(0)} + (T^{k-1} + \cdots + T + I)\boldsymbol{c}$

Since $\rho(T) < 1$, $\lim_{k\to\infty} T^k\boldsymbol{x}^{(0)} = \mathbf{0}$

$$\lim_{k\to\infty} \boldsymbol{x}^{(k)} = \mathbf{0} + \lim_{k\to\infty} \left(\sum_{j=0}^{k-1} T^j\right)\boldsymbol{c} = (I - T)^{-1}\boldsymbol{c}$$

Definition 7.8 A **matrix norm** $||\cdot||$ on $n \times n$ matrices is a real-valued function satisfying

(i) $||A|| \geq 0$

(ii) $\big|\big|A\big|\big| = 0$ if and only if $A = 0$

(iii) $\big|\big|\alpha A\big|\big| = |\alpha|\big|\big|A\big|\big|$

(iv) $\big|\big|A + B\big|\big| \leq \big|\big|A\big|\big| + ||B||$

(v) $\big|\big|AB\big|\big| \leq \big|\big|A\big|\big|\,||B||$

Theorem 7.9. If $||\cdot||$ is a vector norm, the **induced** (or **natural**) **matrix norm** is given by

$$\big|\big|A\big|\big| = \max_{||\boldsymbol{x}||=1} ||A\boldsymbol{x}||$$

Example. $||A||_\infty = \max_{||\boldsymbol{x}||_\infty=1} ||A\boldsymbol{x}||_\infty$, the l_∞ induced norm.

$||A||_2 = \max_{||\boldsymbol{x}||_2=1} ||A\boldsymbol{x}||_2$, the l_2 induced norm.

Theorem 7.11. If $A = [a_{ij}]$ is an $n \times n$ matrix, then

$$\left||A|\right|_\infty = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}|$$

Example. Determine $\left||A|\right|_\infty$ for the matrix $A = \begin{bmatrix} 1 & 2 & -1 \\ 0 & 3 & -1 \\ 5 & -1 & 1 \end{bmatrix}$

Corollary 7.20 If $\left||T|\right| < 1$ for any natural matrix norm and $\boldsymbol{c}$ is a given vector, then the sequence $\{\boldsymbol{x}^{(k)}\}_{k=0}^{\infty}$ defined by $\boldsymbol{x}^{(k)} = T\boldsymbol{x}^{(k-1)} + \boldsymbol{c}$ converges, for any $\boldsymbol{x}^{(0)} \in R^n$, to a vector $\boldsymbol{x} \in R^n$, with $\boldsymbol{x} = T\boldsymbol{x} + \boldsymbol{c}$ **,** and the following error bound hold:

(i) $\left||\boldsymbol{x} - \boldsymbol{x}^{(k)}|\right| \le \left||T|\right|^k \left||\boldsymbol{x}^{(0)} - \boldsymbol{x}|\right|$

(ii) $\left|\left|\boldsymbol{x}-\boldsymbol{x}^{(k)}\right|\right| \leq \frac{||T||^{k}}{1-||T||}\left|\left|\boldsymbol{x}^{(1)}-\boldsymbol{x}^{(0)}\right|\right|$

Theorem 7.21 If A is strictly diagonally dominant, then for any choice of $\boldsymbol{x}^{(0)}$, both the Jacobi and Gauss-Seidel methods give sequences $\{\boldsymbol{x}^{(k)}\}_{k=0}^{\infty}$ that converges to the unique solution of $A\boldsymbol{x}=\boldsymbol{b}$.

Rate of Convergence

Corollary 7.20 (i) implies $\left|\left|\boldsymbol{x}-\boldsymbol{x}^{(k)}\right|\right| \approx \rho(T)^{k}\left|\left|\boldsymbol{x}^{(0)}-\boldsymbol{x}\right|\right|$

Theorem 7.22 (Stein-Rosenberg) If $a_{ij} \leq 0$, for each $i \neq j$ and $a_{ii} \geq 0$, for each $i = 1,2,\ldots,n$, then one and only one of following statements holds:

(i) $0 \leq \rho(T_g) < \rho(T_j) < 1$;
(ii) $1 < \rho(T_j) < \rho(T_g)$;
(iii) $\rho(T_j) = \rho(T_g) = 0$;
(iv) $\rho(T_j) = \rho(T_g) = 1$.

```
import numpy as np

ITERATION_LIMIT = 1000

# initialize the matrix
A = np.array([[10., -1., 2., 0.],
              [-1., 11., -1., 3.],
              [2., -1., 10., -1.],
              [0., 3., -1., 8.]])

# initialize the RHS vector
b = np.array([6., 25., -11., 15.])

print("System of equations:")

for i in range(A.shape[0]):
    row = ["{0:3g}*x{1}".format(A[i, j], j + 1)

    for j in range(A.shape[1])]
    print("[{0}] = [{1:3g}]".format(" + ".join(row), b[i]))

x = np.zeros_like(b)

for it_count in range(1, ITERATION_LIMIT):
    x_new = np.zeros_like(x)
    print("Iteration {0}: {1}".format(it_count, x))

    for i in range(A.shape[0]):
        s1 = np.dot(A[i, :i], x_new[:i])
        s2 = np.dot(A[i, i + 1:], x[i + 1:])
        x_new[i] = (b[i] - s1 - s2) / A[i, i]
    if np.allclose(x, x_new, rtol=1e-8):
        break
    x = x_new

print("Solution: {0}".format(x))
error = np.dot(A, x) - b
print("Error: {0}".format(error))
```

5.30 - Gram-Schmidt Orthogonalization Procedure

Given vectors $u_1, u_2, \ldots$ produce orthonormal vectors $q_1, q_2, \ldots$.

Let $v_1 = u_1$.
Let $v_2 = u_2 - \text{proj}_{v_1} u_2 = u_2 - \frac{u_2, v_1}{\|v_1\|^2} v_1$.
Let $v_3 = u_3 - \text{proj}_{v_1} u_3 - \text{proj}_{v_2} u_3 = u_3 - \frac{u_3, v_1}{\|v_1\|^2} v_1 - \frac{u_3, v_2}{\|v_2\|^2} v_2$.
etc. . .

Finally normalize all vectors $q_i = v_i / \|v_i\|^2$ for $i = 1, \ldots, n$.

5.31 Example - Orthogonalization

Find an orthonormal basis for the space spanned by the vectors $u_1 = (1, 2, 2)$ and $u_2 = (-4, 3, 2)$.

Clearly $v_1 = u_1$ and $q_1 = (1/3, 2/3, 2/3)$.

Now $v_2 = u_2 - \frac{u_2, v_1}{\|v_1\|^2} v_1 = (-4, 3, 2) - 2/3(1, 2, 2) = (-14/3, 5/3, 2/3)$.

And finally $q_2 = (-14/15, 5/15, 2/15)$.

CHECK: $q_1 \cdot q_2 = 0, q_1 \cdot q_1 = 1$ and $q_2 \cdot q_2 = 0$.

5.32 Solving $Ax = b$ with QR factorization

- Factor A into QR form where Q is an orthogonal matrix and R is upper triangular
- Rewrite the system $Ax = b$ as $QRx = b$ and invert matrix Q.
- Since $Q^{-1} = Q^T$ obtain $Rx = Q^T b$.
- Solve the resulting upper triangular system.

Note: to construct Q use Gram-schmid on the columns of A
Note: matrix A does not need to be square!
Note: to construct the upper triangular $R = Q^T A$.

5.33 Example

Use QR factorization to solve $\begin{pmatrix} 1 & -4 \\ 2 & 3 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -3 \\ 15 \\ 9 \end{pmatrix}$

From our previous example above in 5.31 we have already used Gram-Schmidt on the columns of A and therefore the matrix Q is given below together with the calculation of the matrix R,

$$Q = \begin{pmatrix} 1/3 & -14/15 \\ 2/3 & 5/15 \\ 2/3 & 2/15 \end{pmatrix}, \quad R = Q^T A = \begin{pmatrix} 3 & 2 \\ 0 & 5 \end{pmatrix}$$

We solve $R \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = Q^T \begin{pmatrix} -3 \\ 15 \\ 9 \end{pmatrix}$ has the solution $x_1 = 3.8, x_2 = 1.8.$

```
import numpy as np

"""
Applies the Gram-Schmidt method to A and returns Q and R, so Q*R = A.
"""

def gramschmidt(A):

    R = np.zeros((A.shape[1], A.shape[1]))
    Q = np.zeros(A.shape)

    for k in range(0, A.shape[1]):
        R[k, k] = np.sqrt(np.dot(A[:, k], A[:, k]))
        Q[:, k] = A[:, k]/R[k, k]

        for j in range(k+1, A.shape[1]):
            R[k, j] = np.dot(Q[:, k], A[:, j])
            A[:, j] = A[:, j] - R[k, j]*Q[:, k]

    return Q, R

cols = int(input('give number of columns : '))
rows = int(input('give number of rows : '))
A = np.random.rand(rows, cols)
Q, R = gramschmidt(A)

print("Q = \n%s" % Q)
print("R = \n%s" % R)
```

10.3 POWER METHOD FOR APPROXIMATING EIGENVALUES

In Chapter 7 we saw that the eigenvalues of an $n \times n$ matrix A are obtained by solving its characteristic equation

$$\lambda^n + c_{n-1}\lambda^{n-1} + c_{n-2}\lambda^{n-2} + \cdots + c_0 = 0.$$

For large values of n, polynomial equations like this one are difficult and time-consuming to solve. Moreover, numerical techniques for approximating roots of polynomial equations of high degree are sensitive to rounding errors. In this section we look at an alternative method for approximating eigenvalues. As presented here, the method can be used only to find the eigenvalue of A that is largest in absolute value—we call this eigenvalue the **dominant eigenvalue** of A. Although this restriction may seem severe, dominant eigenvalues are of primary interest in many physical applications.

Definition of Dominant Eigenvalue and Dominant Eigenvector

Let $\lambda_1, \lambda_2, \ldots,$ and λ_n be the eigenvalues of an $n \times n$ matrix A. λ_1 is called the **dominant eigenvalue** of A if

$$|\lambda_1| > |\lambda_i|, \quad i = 2, \ldots, n.$$

The eigenvectors corresponding to λ_1 are called **dominant eigenvectors** of A.

Not every matrix has a dominant eigenvalue. For instance, the matrix

$$A = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

(with eigenvalues of $\lambda_1 = 1$ and $\lambda_2 = -1$) has no dominant eigenvalue. Similarly, the matrix

$$A = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(with eigenvalues of $\lambda_1 = 2$, $\lambda_2 = 2$, and $\lambda_3 = 1$) has no dominant eigenvalue.

EXAMPLE 1 ***Finding a Dominant Eigenvalue***

Find the dominant eigenvalue and corresponding eigenvectors of the matrix

$$A = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}.$$

Solution From Example 4 of Section 7.1 we know that the characteristic polynomial of A is $\lambda^2 + 3\lambda + 2 = (\lambda + 1)(\lambda + 2)$. Therefore the eigenvalues of A are $\lambda_1 = -1$ and $\lambda_2 = -2$, of which the dominant one is $\lambda_2 = -2$. From the same example we know that the dominant eigenvectors of A (those corresponding to $\lambda_2 = -2$) are of the form

$$\mathbf{x} = t\begin{bmatrix} 3 \\ 1 \end{bmatrix}, \quad t \neq 0.$$

The Power Method

Like the Jacobi and Gauss-Seidel methods, the power method for approximating eigenvalues is iterative. First we assume that the matrix A has a dominant eigenvalue with corresponding dominant eigenvectors. Then we choose an initial approximation $\mathbf{x}_0$ of one of the dominant eigen*vectors* of A. This initial approximation must be a *nonzero* vector in R^n. Finally we form the sequence given by

$$\begin{aligned}
\mathbf{x}_1 &= A\mathbf{x}_0 \\
\mathbf{x}_2 = A\mathbf{x}_1 &= A(A\mathbf{x}_0) = A^2\mathbf{x}_0 \\
\mathbf{x}_3 = A\mathbf{x}_2 &= A(A^2\mathbf{x}_0) = A^3\mathbf{x}_0 \\
&\vdots \\
\mathbf{x}_k = A\mathbf{x}_{k-1} &= A(A^{k-1}\mathbf{x}_0) = A^k\mathbf{x}_0.
\end{aligned}$$

For large powers of k, and by properly scaling this sequence, we will see that we obtain a good approximation of the dominant eigenvector of A. This procedure is illustrated in Example 2.

EXAMPLE 2 ***Approximating a Dominant Eigenvector by the Power Method***

Complete six iterations of the power method to approximate a dominant eigenvector of

$$A = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}.$$

Solution We begin with an initial nonzero approximation of

$$\mathbf{x}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

We then obtain the following approximations.

	Iteration		*Approximation*
$\mathbf{x}_1 = A\mathbf{x}_0$	$\begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -10 \\ -4 \end{bmatrix}$	→	$-4\begin{bmatrix} 2.50 \\ 1.00 \end{bmatrix}$
$\mathbf{x}_2 = A\mathbf{x}_1$	$\begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} -10 \\ -4 \end{bmatrix} = \begin{bmatrix} 28 \\ 10 \end{bmatrix}$	→	$10\begin{bmatrix} 2.80 \\ 1.00 \end{bmatrix}$
$\mathbf{x}_3 = A\mathbf{x}_2$	$\begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} 28 \\ 10 \end{bmatrix} = \begin{bmatrix} -64 \\ -22 \end{bmatrix}$	→	$-22\begin{bmatrix} 2.91 \\ 1.00 \end{bmatrix}$
$\mathbf{x}_4 = A\mathbf{x}_3$	$\begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} -64 \\ -22 \end{bmatrix} = \begin{bmatrix} 136 \\ 46 \end{bmatrix}$	→	$46\begin{bmatrix} 2.96 \\ 1.00 \end{bmatrix}$
$\mathbf{x}_5 = A\mathbf{x}_4$	$\begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} 136 \\ 46 \end{bmatrix} = \begin{bmatrix} -280 \\ -94 \end{bmatrix}$	→	$-94\begin{bmatrix} 2.98 \\ 1.00 \end{bmatrix}$
$\mathbf{x}_6 = A\mathbf{x}_5$	$\begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} -280 \\ -94 \end{bmatrix} = \begin{bmatrix} 568 \\ 190 \end{bmatrix}$	→	$190\begin{bmatrix} 2.99 \\ 1.00 \end{bmatrix}$

Note that the approximations in Example 2 appear to be approaching scalar multiples of

$$\begin{bmatrix} 3 \\ 1 \end{bmatrix},$$

which we know from Example 1 is a dominant eigenvector of the matrix

$$A = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}.$$

In Example 2 the power method was used to approximate a dominant eigenvector of the matrix A. In that example we already knew that the dominant eigenvalue of A was $\lambda = -2$. For the sake of demonstration, however, let us assume that we do not know the dominant eigenvalue of A. The following theorem provides a formula for determining the eigenvalue corresponding to a given eigenvector. This theorem is credited to the English physicist John William Rayleigh (1842–1919).

Theorem 10.2
Determining an Eigenvalue from an Eigenvector

If $\mathbf{x}$ is an eigenvector of a matrix A, then its corresponding eigenvalue is given by

$$\lambda = \frac{A\mathbf{x} \cdot \mathbf{x}}{\mathbf{x} \cdot \mathbf{x}}.$$

This quotient is called the **Rayleigh quotient.**

Proof Since $\mathbf{x}$ is an eigenvector of A, we know that $A\mathbf{x} = \lambda\mathbf{x}$, and we can write

$$\frac{A\mathbf{x} \cdot \mathbf{x}}{\mathbf{x} \cdot \mathbf{x}} = \frac{\lambda\mathbf{x} \cdot \mathbf{x}}{\mathbf{x} \cdot \mathbf{x}} = \frac{\lambda(\mathbf{x} \cdot \mathbf{x})}{\mathbf{x} \cdot \mathbf{x}} = \lambda.$$

In cases for which the power method generates a good approximation of a dominant eigenvector, the Rayleigh quotient provides a correspondingly good approximation of the dominant eigenvalue. The use of the Rayleigh quotient is demonstrated in Example 3.

EXAMPLE 3 ***Approximating a Dominant Eigenvalue***

Use the result of Example 2 to approximate the dominant eigenvalue of the matrix

$$A = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}.$$

Solution After the sixth iteration of the power method in Example 2, we had obtained.

$$\mathbf{x}_6 = \begin{bmatrix} 568 \\ 190 \end{bmatrix} \approx 190 \begin{bmatrix} 2.99 \\ 1.00 \end{bmatrix}.$$

With $\mathbf{x} = (2.99, 1)$ as our approximation of a dominant eigenvector of A, we use the Rayleigh quotient to obtain an approximation of the dominant eigenvalue of A. First we compute the product $A\mathbf{x}$.

$$A\mathbf{x} = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix} \begin{bmatrix} 2.99 \\ 1.00 \end{bmatrix} = \begin{bmatrix} -6.02 \\ -2.01 \end{bmatrix}$$

Then, since

$$A\mathbf{x} \cdot \mathbf{x} = (-6.02)(2.99) + (-2.01)(1) \approx -20.0$$

and

$$\mathbf{x} \cdot \mathbf{x} = (2.99)(2.99) + (1)(1) \approx 9.94,$$

we compute the Rayleigh quotient to be

$$\lambda = \frac{A\mathbf{x} \cdot \mathbf{x}}{\mathbf{x} \cdot \mathbf{x}} \approx \frac{-20.0}{9.94} \approx -2.01,$$

which is a good approximation of the dominant eigenvalue $\lambda = -2$.

From Example 2 we can see that the power method tends to produce approximations with large entries. In practice it is best to "scale down" each approximation before proceeding to the next iteration. One way to accomplish this **scaling** is to determine the component of $A\mathbf{x}_i$ that has the largest absolute value and multiply the vector $A\mathbf{x}_i$ by the reciprocal of this component. The resulting vector will then have components whose absolute values are less than or equal to 1. (Other scaling techniques are possible. For examples, see Exercises 27 and 28.

EXAMPLE 4 ***The Power Method with Scaling***

Calculate seven iterations of the power method with *scaling* to approximate a dominant eigenvector of the matrix

$$A = \begin{bmatrix} 1 & 2 & 0 \\ -2 & 1 & 2 \\ 1 & 3 & 1 \end{bmatrix}.$$

Use $\mathbf{x}_0 = (1, 1, 1)$ as the initial approximation.

Solution One iteration of the power method produces

$$A\mathbf{x}_0 = \begin{bmatrix} 1 & 2 & 0 \\ -2 & 1 & 2 \\ 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 5 \end{bmatrix},$$

and by scaling we obtain the approximation

$$\mathbf{x}_1 = \tfrac{1}{5} \begin{bmatrix} 3 \\ 1 \\ 5 \end{bmatrix} = \begin{bmatrix} 0.60 \\ 0.20 \\ 1.00 \end{bmatrix}.$$

A second iteration yields

$$A\mathbf{x}_1 = \begin{bmatrix} 1 & 2 & 0 \\ -2 & 1 & 2 \\ 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 0.60 \\ 0.20 \\ 1.00 \end{bmatrix} = \begin{bmatrix} 1.00 \\ 1.00 \\ 2.20 \end{bmatrix}$$

and

$$\mathbf{x}_2 = \frac{1}{2.20} \begin{bmatrix} 1.00 \\ 1.00 \\ 2.20 \end{bmatrix} = \begin{bmatrix} 0.45 \\ 0.45 \\ 1.00 \end{bmatrix}.$$

Continuing this process, we obtain the sequence of approximations shown in Table 10.6.

TABLE 10.6

$\mathbf{x}_0$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$
$\begin{bmatrix} 1.00 \\ 1.00 \\ 1.00 \end{bmatrix}$	$\begin{bmatrix} 0.60 \\ 0.20 \\ 1.00 \end{bmatrix}$	$\begin{bmatrix} 0.45 \\ 0.45 \\ 1.00 \end{bmatrix}$	$\begin{bmatrix} 0.48 \\ 0.55 \\ 1.00 \end{bmatrix}$	$\begin{bmatrix} 0.51 \\ 0.51 \\ 1.00 \end{bmatrix}$	$\begin{bmatrix} 0.50 \\ 0.49 \\ 1.00 \end{bmatrix}$	$\begin{bmatrix} 0.50 \\ 0.50 \\ 1.00 \end{bmatrix}$	$\begin{bmatrix} 0.50 \\ 0.50 \\ 1.00 \end{bmatrix}$

From Table 10.6 we approximate a dominant eigenvector of A to be

$$\mathbf{x} = \begin{bmatrix} 0.50 \\ 0.50 \\ 1.00 \end{bmatrix}.$$

Using the Rayleigh quotient, we approximate the dominant eigenvalue of A to be $\lambda = 3$. (For this example you can check that the approximations of $\mathbf{x}$ and λ are exact.)

REMARK: Note that the *scaling factors* used to obtain the vectors in Table 10.6,

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$
↓	↓	↓	↓	↓	↓	↓
5.00	2.20	2.82	3.13	3.02	2.99	3.00,

are approaching the dominant eigenvalue $\lambda = 3$.

In Example 4 the power method with scaling converges to a dominant eigenvector. The following theorem tells us that a sufficient condition for convergence of the power method is that the matrix A be diagonalizable (and have a dominant eigenvalue).

Theorem 10.3
Convergence of the Power Method

If A is an $n \times n$ diagonalizable matrix with a dominant eigenvalue, then there exists a nonzero vector $\mathbf{x}_0$ such that the sequence of vectors given by

$$A\mathbf{x}_0, \quad A^2\mathbf{x}_0, \quad A^3\mathbf{x}_0, \quad A^4\mathbf{x}_0, \quad \ldots, \quad A^k\mathbf{x}_0, \quad \ldots$$

approaches a multiple of the dominant eigenvector of A.

Proof Since A is diagonalizable, we know from Theorem 7.5 that it has n linearly independent eigenvectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ with corresponding eigenvalues of $\lambda_1, \lambda_2, \ldots, \lambda_n$. We assume that these eigenvalues are ordered so that λ_1 is the dominant eigenvalue (with a corresponding eigenvector of $\mathbf{x}_1$). Because the n eigenvectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ are linearly independent, they must form a basis for R^n. For the initial approximation $\mathbf{x}_0$, we choose a nonzero vector such that the linear combination

$$\mathbf{x}_0 = c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + \cdots + c_n\mathbf{x}_n$$

has nonzero leading coefficients. (If $c_1 = 0$, the power method may not converge, and a different $\mathbf{x}_0$ must be used as the initial approximation. See Exercises 21 and 22.) Now, multiplying both sides of this equation by A produces

$$\begin{aligned} A\mathbf{x}_0 &= A(c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + \cdots + c_n\mathbf{x}_n) \\ &= c_1(A\mathbf{x}_1) + c_2(A\mathbf{x}_2) + \cdots + c_n(A\mathbf{x}_n) \\ &= c_1(\lambda_1\mathbf{x}_1) + c_2(\lambda_2\mathbf{x}_2) + \cdots + c_n(\lambda_n\mathbf{x}_n). \end{aligned}$$

Repeated multiplication of both sides of this equation by A produces

$$A^k\mathbf{x}_0 = c_1(\lambda_1{}^k\mathbf{x}_1) + c_2(\lambda_2{}^k\mathbf{x}_2) + \cdots + c_n(\lambda_n{}^k\mathbf{x}_n),$$

which implies that

$$A^k\mathbf{x}_0 = \lambda_1{}^k\left[c_1\mathbf{x}_1 + c_2\left(\frac{\lambda_2}{\lambda_1}\right)^k\mathbf{x}_2 + \cdots + c_n\left(\frac{\lambda_n}{\lambda_1}\right)^k\mathbf{x}_n\right].$$

Now, from our original assumption that λ_1 is larger in absolute value than the other eigenvalues it follows that each of the fractions

$$\frac{\lambda_2}{\lambda_1}, \quad \frac{\lambda_3}{\lambda_1}, \quad \ldots, \quad \frac{\lambda_n}{\lambda_1}$$

is less than 1 in absolute value. Therefore each of the factors

$$\left(\frac{\lambda_2}{\lambda_1}\right)^k, \quad \left(\frac{\lambda_3}{\lambda_1}\right)^k, \quad \ldots, \quad \left(\frac{\lambda_n}{\lambda_1}\right)^k$$

must approach 0 as k approaches infinity. This implies that the approximation

$$A^k\mathbf{x}_0 \approx \lambda_1{}^k c_1\mathbf{x}_1, \quad c_1 \neq 0$$

improves as k increases. Since $\mathbf{x}_1$ is a dominant eigenvector, it follows that any scalar multiple of $\mathbf{x}_1$ is also a dominant eigenvector. Thus we have shown that $A^k\mathbf{x}_0$ approaches a multiple of the dominant eigenvector of A.

The proof of Theorem 10.3 provides some insight into the rate of convergence of the power method. That is, if the eigenvalues of A are ordered so that

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_n|,$$

then the power method will converge quickly if $|\lambda_2| / |\lambda_1|$ is small, and slowly if $|\lambda_2| / |\lambda_1|$ is close to 1. This principle is illustrated in Example 5.

EXAMPLE 5 ***The Rate of Convergence of the Power Method***

(a) The matrix

$$A = \begin{bmatrix} 4 & 5 \\ 6 & 5 \end{bmatrix}$$

has eigenvalues of $\lambda_1 = 10$ and $\lambda_2 = -1$. Thus the ratio $|\lambda_2| / |\lambda_1|$ is 0.1. For this matrix, only four iterations are required to obtain successive approximations that agree when rounded to three significant digits. (See Table 10.7.)

TABLE 10.7

$\mathbf{x}_0$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$
$\begin{bmatrix} 1.000 \\ 1.000 \end{bmatrix}$	$\begin{bmatrix} 0.818 \\ 1.000 \end{bmatrix}$	$\begin{bmatrix} 0.835 \\ 1.000 \end{bmatrix}$	$\begin{bmatrix} 0.833 \\ 1.000 \end{bmatrix}$	$\begin{bmatrix} 0.833 \\ 1.000 \end{bmatrix}$

(b) The matrix

$$A = \begin{bmatrix} -4 & 10 \\ 7 & 5 \end{bmatrix}$$

has eigenvalues of $\lambda_1 = 10$ and $\lambda_2 = -9$. For this matrix, the ratio $|\lambda_2| / |\lambda_1|$ is 0.9, and the power method does not produce successive approximations that agree to three significant digits until sixty-eight iterations have been performed, as shown in Table 10.8.

TABLE 10.8

$\mathbf{x}_0$	$\mathbf{x}_1$	$\mathbf{x}_2$		$\mathbf{x}_{66}$	$\mathbf{x}_{67}$	$\mathbf{x}_{68}$
$\begin{bmatrix} 1.000 \\ 1.000 \end{bmatrix}$	$\begin{bmatrix} 0.500 \\ 1.000 \end{bmatrix}$	$\begin{bmatrix} 0.941 \\ 1.000 \end{bmatrix}$	$\cdots$ $\cdots$	$\begin{bmatrix} 0.715 \\ 1.000 \end{bmatrix}$	$\begin{bmatrix} 0.714 \\ 1.000 \end{bmatrix}$	$\begin{bmatrix} 0.714 \\ 1.000 \end{bmatrix}$

In this section we have discussed the use of the power method to approximate the *dominant* eigenvalue of a matrix. This method can be modified to approximate other eigenvalues through use of a procedure called **deflation.** Moreover, the power method is only one of several techniques that can be used to approximate the eigenvalues of a matrix. Another popular method is called the **QR algorithm.**

This is the method used in most computer programs and calculators for finding eigenvalues and eigenvectors. The algorithm uses the *QR*–factorization of the matrix, as presented in Chapter 5. Discussions of the deflation method and the *QR* algorithm can be found in most texts on numerical methods.

SECTION 10.3 ❑ EXERCISES

In Exercises 1–6, use the techniques presented in Chapter 7 to find the eigenvalues of the given matrix A. If A has a dominant eigenvalue, find a corresponding dominant eigenvector.

1. $A = \begin{bmatrix} 2 & 1 \\ 0 & -4 \end{bmatrix}$

2. $A = \begin{bmatrix} -3 & 0 \\ 1 & 3 \end{bmatrix}$

3. $A = \begin{bmatrix} 1 & -5 \\ -3 & -1 \end{bmatrix}$

4. $A = \begin{bmatrix} 4 & -5 \\ 2 & -3 \end{bmatrix}$

5. $A = \begin{bmatrix} 2 & 3 & 1 \\ 0 & -1 & 2 \\ 0 & 0 & 3 \end{bmatrix}$

6. $A = \begin{bmatrix} -5 & 0 & 0 \\ 3 & 7 & 0 \\ 4 & -2 & 3 \end{bmatrix}$

In Exercises 7–10, use the Rayleigh quotient to compute the eigenvalue λ of A corresponding to the given eigenvector $\mathbf{x}$.

7. $A = \begin{bmatrix} 4 & -5 \\ 2 & -3 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$

8. $A = \begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} -3 \\ 1 \end{bmatrix}$

9. $A = \begin{bmatrix} 1 & 2 & -2 \\ -2 & 5 & -2 \\ -6 & 6 & -3 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix}$

10. $A = \begin{bmatrix} 3 & 2 & -3 \\ -3 & -4 & 9 \\ -1 & -2 & 5 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} 3 \\ 0 \\ 1 \end{bmatrix}$

C In Exercises 11–14, use the power method with scaling to approximate a dominant eigenvector of the matrix A. Start with $\mathbf{x}_0 = (1, 1)$ and calculate five iterations. Then use $\mathbf{x}_5$ to approximate the dominant eigenvalue of A.

11. $A = \begin{bmatrix} 2 & 1 \\ 0 & -7 \end{bmatrix}$

12. $A = \begin{bmatrix} -1 & 0 \\ 1 & 6 \end{bmatrix}$

13. $A = \begin{bmatrix} 1 & -4 \\ -2 & 8 \end{bmatrix}$

14. $A = \begin{bmatrix} 6 & -3 \\ -2 & 1 \end{bmatrix}$

C In Exercises 15–18, use the power method with scaling to approximate a dominant eigenvector of the matrix A. Start with $\mathbf{x}_0 = (1, 1, 1)$ and calculate four iterations. Then use $\mathbf{x}_4$ to approximate the dominant eigenvalue of A.

15. $A = \begin{bmatrix} 3 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 2 & 8 \end{bmatrix}$

16. $A = \begin{bmatrix} 1 & 2 & 0 \\ 0 & -7 & 1 \\ 0 & 0 & 0 \end{bmatrix}$

17. $A = \begin{bmatrix} -1 & -6 & 0 \\ 2 & 7 & 0 \\ 1 & 2 & -1 \end{bmatrix}$

18. $A = \begin{bmatrix} 0 & 6 & 0 \\ 0 & -4 & 0 \\ 2 & 1 & 1 \end{bmatrix}$

C In Exercises 19 and 20, the given matrix A does not have a dominant eigenvalue. Apply the power method with scaling, starting with $\mathbf{x}_0 = (1, 1, 1)$, and observe the results of the first four iterations.

19. $A = \begin{bmatrix} 1 & 1 & 0 \\ 3 & -1 & 0 \\ 0 & 0 & -2 \end{bmatrix}$

20. $A = \begin{bmatrix} 1 & 2 & -2 \\ -2 & 5 & -2 \\ -6 & 6 & -3 \end{bmatrix}$

21. (a) Find the eigenvalues and corresponding eigenvectors of

$$A = \begin{bmatrix} 3 & -1 \\ -2 & 4 \end{bmatrix}.$$

(b) Calculate two iterations of the power method with scaling, starting with $\mathbf{x}_0 = (1, 1)$.

(c) Explain why the method does not seem to converge to a dominant eigenvector.

22. Repeat Exercise 21 using $\mathbf{x}_0 = (1, 1, 1)$, for the matrix

$$A = \begin{bmatrix} -3 & 0 & 2 \\ 0 & -1 & 0 \\ 0 & 1 & -2 \end{bmatrix}.$$

C **23.** The matrix

$$A = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}$$

has a dominant eigenvalue of $\lambda = -2$. Observe that $A\mathbf{x} = \lambda\mathbf{x}$ implies that

$$A^{-1}\mathbf{x} = \frac{1}{\lambda}\mathbf{x}.$$

Apply five iterations of the power method (with scaling) on A^{-1} to compute the eigenvalue of A with the smallest magnitude.

C **24.** Repeat Exercise 23 for the matrix

$$A = \begin{bmatrix} 2 & 3 & 1 \\ 0 & -1 & 2 \\ 0 & 0 & 3 \end{bmatrix}.$$

```
import numpy as np
import numpy.linalg as la

A = np.array([[1,   2,   3],
              [2, 4,  5],
              [3, 5,  -1]])

def eigenvalue(A, v):
    Av = A.dot(v)
    return v.dot(Av)

def power_iteration(A):
    n, d = A.shape
    v = np.ones(d) / np.sqrt(d)
    ev = eigenvalue(A, v)

    while True:
        Av = A.dot(v)
        v_new = Av / la.norm(Av)
        ev_new = eigenvalue(A, v_new)
        if np.abs(ev - ev_new) < 0.01:
            break
        v = v_new
        ev = ev_new
        return ev_new, v_new

ev_new, v_new = power_iteration(A)

print("Largest eigenvalue =", ev_new)
print("Corresponding eigenvector = \n%s" % v_new)
```