Numerical Methods for Physicists

9. Eigenvalue problems

Titus Beu
University "Babes-Bolyai"
Department of Theoretical and Computational Physics
Cluj-Napoca, Romania

Eigenvalue problems

- Bibliography
- Introduction
- Matrix diagonalization by similarity transformations
- Jacobi's method
- Systems of linear equations with tridiagonal matrix

Eigenvalue problems
Bibliography

- B.P. Demidovich şi I.A. Maron, *Computational Mathematics* (MIR Publishers, Moskow, 1981).
- R.A. Horn şi C.R. Johnson, *Matrix Analysis* (Cambridge University Press, Cambridge, 1985).
- R.L. Burden şi J.D. Faires, *Numerical Analysis, Third Edition* (Prindle, Weber & Schmidt, Boston, 1985).
- W.H. Press, S.A. Teukolsky, W.T. Vetterling şi B.P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing, Second Edition* (Cambridge University Press, Cambridge, 1992).
- Beu, T. A., Numerical Calculus in C, Third Edition (MicroInformatica Publishing House, Cluj-Napoca, 2004).

Eigenvalue problems
Introduction

- **Eigenvalue problem** of a linear operator $\mathcal{A}$:

$$\mathcal{A}\mathbf{x} = \lambda\mathbf{x}$$

λ – eigenvalue, $\mathbf{x}$ – eigenvector

- **Matrix eigenvalue problem** with $\mathbf{A} = [a_{ij}]_{nn}$, $\mathbf{x} = [x_i]_n$ – in a representation in R^n

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

or

$$(\mathbf{A} - \lambda\mathbf{E}) \cdot \mathbf{x} = 0$$

or

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Eigenvalue problems
Introduction

- **Characteristic (secular) equation**:

$$\det(\mathbf{A} - \lambda\mathbf{E}) = 0$$

$\det(\mathbf{A} - \lambda\mathbf{E})$ – **characteristic determinant (polynomial)**

- Characteristic equation in matrix form:

$$\begin{vmatrix} a_{11} - \lambda & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} - \lambda & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} - \lambda \end{vmatrix} = 0$$

- **Solution**: n real or complex roots $\lambda_1, \lambda_2, \ldots \lambda_n$ – **spectrum** of matrix $\mathbf{A}$
- Replacing $\lambda = \lambda_j$ in the eigenvalue equation $\rightarrow \mathbf{x}^{(j)}$ (corresponding eigenvector)

Eigenvalue problems
Matrix diagonalization by similarity transformations

- $\mathbf{X}$ – matrix having the eigenvectors $\mathbf{x}^{(j)}$ as columns
- $\mathbf{\Lambda}$ – diagonal matrix having the eigenvalues $\lambda_1, \lambda_2, \ldots \lambda_n$ on the diagonal

$$\mathbf{X} = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & \cdots & x_1^{(n)} \\ x_2^{(1)} & x_2^{(2)} & \cdots & x_2^{(n)} \\ \vdots & \vdots & \ddots & \vdots \\ x_n^{(1)} & x_n^{(2)} & \cdots & x_n^{(n)} \end{bmatrix}, \quad \mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}$$

- **Modal equation** – compilation of all n eigenvalue equations:

$$\mathbf{A} \cdot \mathbf{X} = \mathbf{X} \cdot \mathbf{\Lambda}$$

- **Theorem**: if the modal matrix $\mathbf{X}$ corresponding to matrix $\mathbf{A}$ is formed of linearly independent columns, then $\mathbf{X}^{-1}$ exists and the modal equation can be written:

$$\mathbf{X}^{-1} \cdot \mathbf{A} \cdot \mathbf{X} = \mathbf{\Lambda}$$

Eigenvalue problems
Matrix diagonalization by similarity transformations

- **Similar matrices** – associated to an operator relative to different basis sets
- **Theorem**: Two matrices with real elements $\mathbf{A}, \mathbf{B} \in \mathbf{M}_{\mathrm{R}}^{n\times n}$ are **similar** if and only if there exists a **non-singular** matrix $\mathbf{S} \in \mathbf{M}_{\mathrm{R}}^{n\times n}$ such that:

$$\mathbf{B} = \mathbf{S}^{-1} \cdot \mathbf{A} \cdot \mathbf{S}$$

- Such operation is called **similarity transformation**
- **Theorem**: Two **similar matrices** have **identical eigenvalues**. The corresponding eigenvectors result from one another by means of the similarity transformation.
- Matrices $\mathbf{A}$ and $\mathbf{\Lambda}$ are similar as per modal equation $\mathbf{\Lambda} = \mathbf{X}^{-1} \cdot \mathbf{A} \cdot \mathbf{X}$
- Matrices similar to a diagonal matrix are called **diagonalizable**
- **Importance of diagonalization**:
  if a transformation can be found which diagonalizes a matrix $\mathbf{A}$:
  - Eigenvalues of $\mathbf{A}$ – main diagonal of the transformed matrix $\mathbf{\Lambda}$
  - Eigenvectors of $\mathbf{A}$ – columns of the transformation matrix $\mathbf{X}$.

Eigenvalue problems
Jacobi's method

- **Real symmetric matrices** – eigenvectors are real and orthogonal
- **Orthogonal transformation matrix**:

$$\mathbf{R} \cdot \mathbf{R}^{T} = \mathbf{R}^{T} \cdot \mathbf{R} = \mathbf{E} \quad \text{or} \quad \mathbf{R}^{-1} = \mathbf{R}^{T}$$

- Diagonalization by the similarity transformation:

$$\mathbf{R}^{T} \cdot \mathbf{A} \cdot \mathbf{R} = \mathbf{\Lambda}$$

- **Basic ideas**:
  - Successive similarity transformations to annul symmetric non-diagonal elements
  - Transformations destroy previous zeroes – reduce overall the non-diagonal part
  - Product of successive transformations – modal matrix (eigenvectors on columns)

Eigenvalue problems
Jacobi's method

- **Example**: 2×2 matrix
- Orthogonal transformation matrix – **2D rotation**:

$$\mathbf{R} = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix}$$

- Effect: rotation of angle φ of the basis vectors
- φ can be adjusted such that $\mathbf{R}$ diagonalizes $\mathbf{A}$
- **Orthogonal transformation** (conserves symmetry):

$$\mathbf{A}' = \mathbf{R}^T \cdot \mathbf{A} \cdot \mathbf{R}$$

$$\begin{cases} a'_{11} = a_{11}\cos^2\varphi + 2a_{21}\sin\varphi\cos\varphi + a_{22}\sin^2\varphi \\ a'_{22} = a_{11}\sin^2\varphi - 2a_{21}\sin\varphi\cos\varphi + a_{22}\cos^2\varphi \\ a'_{21} = a_{21}(\cos^2\varphi - \sin^2\varphi) + (a_{22} - a_{11})\sin\varphi\cos\varphi = a'_{12} \end{cases}$$

Eigenvalue problems
Jacobi's method

- Vanishing condition for non-diagonal elements of $\mathbf{A}'$:

$$\cot^2\varphi + \frac{a_{22} - a_{11}}{a_{21}}\cot\varphi - 1 = 0$$

- Solution:

$$\tan\varphi = \left[\frac{a_{11} - a_{22}}{2a_{21}} \pm \sqrt{\left(\frac{a_{11} - a_{22}}{2a_{21}}\right)^2 + 1}\right]^{-1}$$

- Required values: $\cos\varphi = (1 + \tan\varphi)^{-1/2}, \quad \sin\varphi = \tan\varphi\cos\varphi$
- **Eigenvalues and eigenvectors**: **exact diagonalization** (1 orthogonal transformation)

$$\lambda_1 = a'_{11}, \quad \mathbf{x}^{(1)} = \begin{bmatrix} \cos\varphi \\ \sin\varphi \end{bmatrix}$$

$$\lambda_2 = a'_{22}, \quad \mathbf{x}^{(2)} = \begin{bmatrix} -\sin\varphi \\ \cos\varphi \end{bmatrix}$$

Eigenvalue problems
Jacobi's method

- **Generalization**: $n{\times}n$ matrix
- Orthogonal transformation matrix – **2D rotation**:

$$\mathbf{R}(i,j) = \begin{array}{cl} \begin{bmatrix} 1 & \vdots & & \vdots & 0 \\ \cdots & \cos\varphi & \cdots & -\sin\varphi & \cdots \\ & \vdots & \ddots & \vdots & \\ \cdots & \sin\varphi & \cdots & \cos\varphi & \cdots \\ 0 & \vdots & & \vdots & 1 \end{bmatrix} & \begin{array}{l} \\ \text{line } i \\ \\ \text{line } j \\ \\ \end{array} \\ \text{column } i \quad \text{column } j & \end{array}$$

- **Orthogonal transformation** – annuls elements a_{ij}' and a_{ji}' :

$$\mathbf{A}' = \mathbf{R}^T(i,j)\cdot\mathbf{A}\cdot\mathbf{R}(i,j)$$

- Only elements of $\mathbf{A}'$ on lines and columns i and j differ from those of $\mathbf{A}$

Eigenvalue problems
Jacobi's method

- Detailed operations:

$$\tilde{\mathbf{A}} = \mathbf{A}\cdot\mathbf{R}(i,j)$$

or

$$\begin{bmatrix} & \tilde{a}_{1i} & & \tilde{a}_{1j} & \\ & \vdots & & \vdots & \\ \cdots & \tilde{a}_{ki} & \cdots & \tilde{a}_{kj} & \cdots \\ & \vdots & & \vdots & \\ & \tilde{a}_{ni} & & \tilde{a}_{nj} & \end{bmatrix} = \begin{bmatrix} & a_{1i} & & a_{1j} & \\ & \vdots & & \vdots & \\ \cdots & a_{ki} & \cdots & a_{kj} & \cdots \\ & \vdots & & \vdots & \\ & a_{ni} & & a_{nj} & \end{bmatrix} \cdot \begin{bmatrix} 1 & \vdots & & \vdots & 0 \\ \cdots & \cos\varphi & \cdots & -\sin\varphi & \cdots \\ & \vdots & \ddots & \vdots & \\ \cdots & \sin\varphi & \cdots & \cos\varphi & \cdots \\ 0 & \vdots & & \vdots & 1 \end{bmatrix}$$

- New relevant elements on **columns i and j**:

$$\begin{cases} \tilde{a}_{ki} = a_{ki}\cos\varphi + a_{kj}\sin\varphi, \qquad k = 1,2,\ldots,n \\ \tilde{a}_{kj} = -a_{ki}\sin\varphi + a_{kj}\cos\varphi \end{cases}$$

Eigenvalue problems
Jacobi's method

- Detailed operations:

$$\mathbf{A}' = \mathbf{R}^T(i,j)\cdot\tilde{\mathbf{A}}$$

or

$$\begin{bmatrix} & & \vdots & & \\ a'_{i1} & \cdots & a'_{ik} & \cdots & a'_{in} \\ & & \vdots & & \\ a'_{j1} & \cdots & a'_{jk} & \cdots & a'_{jn} \\ & & \vdots & & \end{bmatrix} = \begin{bmatrix} 1 & \vdots & & \vdots & 0 \\ \cdots & \cos\varphi & \cdots & \sin\varphi & \cdots \\ & \vdots & \ddots & \vdots & \\ \cdots & -\sin\varphi & \cdots & \cos\varphi & \cdots \\ 0 & \vdots & & \vdots & 1 \end{bmatrix} \cdot \begin{bmatrix} & & \vdots & & \\ \tilde{a}_{i1} & \cdots & \tilde{a}_{ik} & \cdots & \tilde{a}_{in} \\ & & \vdots & & \\ \tilde{a}_{j1} & \cdots & \tilde{a}_{jk} & \cdots & \tilde{a}_{jn} \\ & & \vdots & & \end{bmatrix}$$

- New relevant elements on **lines *i* and *j***:

$$\begin{cases} a'_{ik} = \tilde{a}_{ik}\cos\varphi + \tilde{a}_{jk}\sin\varphi, \quad k = 1,2,\ldots,n \\ a'_{jk} = -\tilde{a}_{ik}\sin\varphi + \tilde{a}_{jk}\cos\varphi \end{cases}$$

Eigenvalue problems
Jacobi's method

- Modified elements of matrix $\mathbf{A}'$:

$$\begin{cases} a'_{ik} = a'_{ki} = a_{ik}\cos\varphi + a_{jk}\sin\varphi, \quad k = 1,2,\ldots,n \\ a'_{jk} = a'_{kj} = -a_{ik}\sin\varphi + a_{jk}\cos\varphi, \quad k \neq i,j \\ a'_{ii} = a_{ii}\cos^2\varphi + 2a_{ji}\sin\varphi\cos\varphi + a_{jj}\sin^2\varphi \\ a'_{jj} = a_{ii}\sin^2\varphi - 2a_{ji}\sin\varphi\cos\varphi + a_{jj}\cos^2\varphi \\ a'_{ij} = a'_{ji} = a_{ji}(\cos^2\varphi - \sin^2\varphi) + (a_{jj} - a_{ii})\sin\varphi\cos\varphi \end{cases}$$

- Vanishing condition for non-diagonal elements of $\mathbf{A}'$:

$$\cot^2\varphi + \frac{a_{jj} - a_{ii}}{a_{ji}}\cot\varphi - 1 = 0$$

- Solution:

$$\tan\varphi = \left[\frac{a_{ii} - a_{jj}}{2a_{ji}} \pm \sqrt{\left(\frac{a_{ii} - a_{jj}}{2a_j i}\right)^2 + 1}\right]^{-1}$$

- Required values: $\cos\varphi = (1 + \tan\varphi)^{-1/2}, \quad \sin\varphi = \tan\varphi\cos\varphi$

Eigenvalue problems
Jacobi's method

- Sequence of similar matrices with identical eigenvalues:

$$\mathbf{A}_0 = \mathbf{A}, \quad \mathbf{A}_l = \mathbf{R}_l^T \cdot \mathbf{A}_{l-1} \cdot \mathbf{R}_l, \quad l = 0,1,2,\ldots$$

- Sequence of orthogonal matrices:

$$\mathbf{X}_0 \equiv \mathbf{R}_0 \equiv \mathbf{E}, \quad \mathbf{X}_l = \mathbf{R}_0 \cdot \mathbf{R}_1 \cdots \mathbf{R}_l, \quad l = 0,1,2,\ldots$$

- Sequence of similar matrices:

$$\mathbf{A}_0 = \mathbf{A}, \quad \mathbf{A}_l = \mathbf{X}_l^T \cdot \mathbf{A} \cdot \mathbf{X}_l, \quad l = 0,1,2,\ldots$$

- In the limit:

$$\lim_{l\to\infty} \mathbf{A}_l = \mathbf{\Lambda}, \quad \lim_{l\to\infty} \mathbf{X}_l = \mathbf{X}$$

Eigenvalue problems
Jacobi's method

- **Recurrence for modal matrix**:

$$\mathbf{X}_l = \mathbf{X}_{l-1} \cdot \mathbf{R}_l(i,j)$$

- Modified elements:

$$\begin{cases} x'_{ki} = x_{ki}\cos\varphi + x_{kj}\sin\varphi, & k = 1,2,\ldots,n \\ x'_{kj} = -x_{ki}\sin\varphi + x_{kj}\cos\varphi \end{cases}$$

- **Convergence criterion**:

$$\max_{i\neq j} | a'_{ij} | \le \varepsilon$$

- **Stability** – rotation of basis vectors by minimum angle ($\varphi \le \pi/4$): Maximization of absolute value of denominator

$$\tan\varphi = \operatorname{sign}\left(\frac{a_{ii} - a_{jj}}{2a_{ji}}\right)\left[\left|\frac{a_{ii} - a_{jj}}{2a_{ji}}\right| + \sqrt{\left(\frac{a_{ii} - a_{jj}}{2a_{ji}}\right)^2 + 1}\right]^{-1}$$

Eigenvalue problems
Jacobi's method

```
//===========================================================================
int Jacobi(float **a, float **x, float d[], int n)
//---------------------------------------------------------------------------
// Solves the eigenvalue problem of a real symmetric matrix
// a  - real symmetric matrix (lower triangle is destroyed)
// x  - modal matrix: eigenvectors on columns (output)
// d  - vector of eigenvalues
// n  - order of matrix a
// Error flag: 0 - normal execution
//             1 - exceeded max. no. of iterations
//---------------------------------------------------------------------------
{
   const float eps = 1e-30;                           // precision criterion
   const int itmax = 50;                             // max no. of iterations
   float aii, aji, ajj, amax, c, s, t;
   int i, it, j, k;

   for (i=1; i<=n; i++) {                        // initializes modal matrix
      for (j=1; j<=n; j++) x[i][j] = 0.0;        // with unit matrix and
      x[i][i] = 1.0;                             // eigenvalues with diagonal
      d[i] = a[i][i];                            // elements
   }
```

Titus Beu 2011

```
   for (it=1; it<=itmax; it++) {                         // Loop of iterations
      amax = 0.0;
      for (j=2; j<=n; j++)
         for (i=1; i<=(j-1); i++) {                 // store diagonal elements
            aii = d[i]; ajj = d[j]; aji = fabs(a[j][i]);
            if (amax < aji) amax = aji;     // store max. non-diagonal element
            if (aji > eps) {                               // perform rotation
               c = 0.5*(aii-ajj)/a[j][i];
               t = 1.0/(fabs(c) + sqrt(1.0+c*c));                   // tangent
               if (c < 0.0) t = -t;                                    // sign
               c = 1.0/sqrt(1.0+t*t); s = c*t;                     // cos, sin
               for (k=1; k<=(i-1); k++) {                   // columns k=1,i-1
                  t       = a[i][k]*c + a[j][k]*s;
                  a[j][k] = a[j][k]*c - a[i][k]*s; a[i][k] = t;
               }
               for (k=(i+1); k<=(j-1); k++) {             // columns k=i+1,j-1
                  t       = a[k][i]*c + a[j][k]*s;
                  a[j][k] = a[j][k]*c - a[k][i]*s; a[k][i] = t;
               }
               for (k=(j+1); k<=n; k++) {                   // columns k=j+1,n
                  t       = a[k][i]*c + a[k][j]*s;
                  a[k][j] = a[k][j]*c - a[k][i]*s; a[k][i] = t;
               }
               for (k=1; k<=n; k++) {                // transform modal matrix
                  t       = x[k][i]*c + x[k][j]*s;
                  x[k][j] = x[k][j]*c - x[k][i]*s; x[k][i] = t;
               }
               t = 2 * s * c * a[j][i];
               d[i] = aii*c*c + ajj*s*s + t;             // update eigenvalues
               d[j] = aii*s*s + ajj*c*c - t;
               a[j][i] = 0.0;
            }
         }
      if (amax<=eps) return 0;                            // check convergence
   }
   printf("Jacobi: max. no. of iterations exceeded !\n");
   return 1;
}
```

Titus Beu 2011