and we assume that we know

$$V_0 = V(x_0)$$
$$V_{N+1} = V(x_{N+1}) \quad \text{from boundary conditions}$$

get

$$\begin{aligned}
n=1 &\qquad -2V_1 + V_2 = h^2 \hat{f}_1 - V_0 \\
n=2 &\qquad V_1 - 2V_2 + V_3 = h^2 \hat{f}_2 \\
n=3 &\qquad V_2 - 2V_3 + V_4 = h^2 f_3 \\
&\qquad \vdots \\
n=N &\qquad V_{N-1} - 2V_N = h^2 \hat{f}_N - V_{N+1}
\end{aligned}$$

which is a matrix equation

$$\begin{pmatrix}
-2 & 1 & 0 & 0 & 0 & & & \cdots & 0 \\
1 & -2 & 1 & 0 & 0 & & & \cdots & 0 \\
0 & 1 & -2 & 1 & 0 & & & & 0 \\
0 & 0 & 1 & -2 & 1 & & & & 0 \\
\vdots & & & & & \ddots & & & \vdots \\
 & & & & & & 1 & -2 & 1 \\
0 & 0 & 0 & \cdots & & & 0 & 1 & -2
\end{pmatrix}
\begin{pmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ \vdots \\ V_{N-1} \\ V_N \end{pmatrix}
=
\begin{pmatrix} h^2 f_1 - V_0 \\ h^2 f_2 \\ h^2 f_3 \\ h^2 f_4 \\ \vdots \\ h^2 f_{N-1} \\ h^2 f_N - V_{N+1} \end{pmatrix}$$

or

$$A \vec{x} = \vec{b}$$

but with A a tridiagonal matrix

$\rightarrow$ need to find routine specialized for tridiagonal matrices

$\rightarrow$ avoid storing all the zeros

$\rightarrow$ faster

Note: In 2D we have $V_{m,n-1} + V_{m,n+1} + V_{m-1,n} + V_{m+1,n} - 4V_{m,n} = h^2 f_{m,n}$

Can map matrix of unknowns ($V_{m,n}$'s) to a vector $\tilde{V}_i$ via "dictionary order"

$$\begin{array}{lccccccccccc}
i = & 1 & 2 & \cdots & N & N+1 & N+2 & & 2N & \cdots & (N-1)N+1 & N^2 \\
(m,n) = & (1,1) & (1,2) & \cdots & (1,N) & (2,1) & (2,2) & \cdots & (2,N) & \cdots & (N,1) \cdots & (N,N)
\end{array}$$

$$\rightarrow \quad i = n + (m-1)N$$

Eigenvalue Problems

Many physics problems, especially from QM, can be expressed as eigenvalue problems

(1) $$A\vec{x} = \lambda\vec{x}$$

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \dots + a_{1N}x_N &= \lambda x_1 \\
a_{21}x_1 + a_{22}x_2 + \dots + a_{2N}x_N &= \lambda x_2 \\
\vdots \qquad\qquad\qquad & \quad \vdots \\
a_{N1}x_1 + a_{N2}x_2 + \dots + a_{NN}x_N &= \lambda x_N
\end{aligned}$$

like system of linear equations except RHS is unknown and solutions exist only for certain λ's (eigenvalues)

- sol$^{\underline{n}}$s are eigenvectors

(1) can be rewritten $(A - \lambda I)\vec{x} = 0$

For non-trivial solution ($\vec{x} \neq 0$), this implies

$\det(A - \lambda I) = 0$, which for an $N \times N$ matrix leads to an N-degree polynomial for λ

$$c_N \lambda^N + c_{N-1}\lambda^{N-1} + \dots + c_1\lambda + c_0 = 0$$

which has N sol$^{\underline{n}}$s for λ, ie N eigenvalues.

Finding all roots for an N-degree polynomial is hard for large N.

Rather than solving the difficult polynomial problem, iterative methods are used.

Example: Jacobi method for symmetric matrices

First, a few preliminaries

- a similarity transformation does not change eigenvalues

$$S = P^{-1} A P \quad \rightarrow \quad A = P S P^{-1}$$

$$A x = \lambda x$$
$$P S P^{-1} x = \lambda x$$
$$S (P^{-1} x) = \lambda (P^{-1} x)$$
$$S x' = \lambda x'$$

S and A have same eigenvalues. Eigenvector of S is $P^{-1} x$. Eigenvector of A is $P x'$.

- eigenvectors diagonalize the matrix

Let $Q = [\, v_1 \;\; v_2 \;\; \dots \;\; v_N \,]$ columns are eigenvectors of A

then $A Q = \Lambda Q$ where

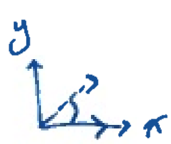
$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \ddots & \lambda_N \end{pmatrix}$$

$$\star \quad Q^{-1} A Q = Q^{-1} \Lambda Q = \Lambda$$

- eigenvectors of a symmetric matrix are orthogonal

$\star$ becomes $Q^T A Q = \Lambda$

- rotation matrices (rotate a vector in xy plane by θ counter-clockwise)



$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \checkmark$$

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -\sin\theta \\ \cos\theta \end{pmatrix} \checkmark$$

are orthogonal

$$\begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} c & s \\ -s & c \end{pmatrix} = \begin{pmatrix} c^2+s^2 & 0 \\ 0 & c^2+s^2 \end{pmatrix} = I$$

$$R R^T = I$$
$$\det(R) = 1$$

$$\begin{cases} c = \cos\theta \\ s = \sin\theta \end{cases}$$

okay, here we go!

Define Givens rotation matrix $G(i,j,\theta)$

$$\begin{array}{cc}
 & \begin{array}{ccccccccc} & & i & & & j & & & \end{array} \\
\begin{array}{c} \\ \\ i \\ \\ \\ j \\ \\ \\ \\ \end{array} &
\begin{pmatrix}
1 & 0 & 0 & & \cdots & & & & 0 \\
0 & 1 & 0 & & & & & & \\
0 & 0 & c & 0 & \cdots & -s & & & \\
 & & 0 & 1 & & & & & \\
 & & \vdots & & \ddots & & & & \\
 & & s & & & c & & & \\
 & & & & & & 1 & 0 & \\
 & & & & & & 0 & 1 & \\
0 & & & & & & & & 1
\end{pmatrix}
\end{array}$$

$g_{kk} = 1$ for $k \neq i, j$

$g_{ii} = g_{jj} = \cos\theta$

$g_{ij} = -\sin\theta$

$g_{ji} = \sin\theta$

$g_{\ell m} = 0$ otherwise

G will rotate each column in $\begin{pmatrix} A_{ii} & A_{ij} \\ A_{ji} & A_{jj} \end{pmatrix}$ ← submatrix of A

ccw by θ

E.g. G can be used to make $A'_{ji} = 0$

$$\begin{bmatrix} c & -s \\ s & c \end{bmatrix}\begin{bmatrix} A_{ii} & A_{ij} \\ A_{ji} & A_{jj} \end{bmatrix} = \begin{bmatrix} r & - \\ 0 & - \end{bmatrix}$$

$r = \sqrt{A_{ii}^2 + A_{ji}^2}$ (rotation preserves length of vector)

$$\left.\begin{array}{l} c A_{ii} - s A_{ji} = r \\ s A_{ii} + c A_{ji} = 0 \end{array}\right\} \rightarrow c = \frac{A_{ii}}{r}, \quad s = -\frac{A_{ji}}{r}$$

Note: This can be used repeatedly to reduce

$Ax = b$ to $Ux = b'$

or to decompose $A = QR$ (R - upper triangular, Q - orthogonal)

i.e. $G_1 A = A_1$ (A_1 has a lower element set to zero)

$\vdots$

$G_k G_{k-1} \cdots G_1 A = R$

$A = G_1^T \cdots G_{k-1}^T G_k^T R = QR$

Make a similarity transformation based on G

$A' = G A G^T$ $\quad$ ($A_{ji} = A_{ij}$ - symmetric
A' also symmetric)

$A'_{ii} = c^2 A_{ii} - 2cs A_{ij} + s^2 A_{jj}$

$A'_{jj} = s^2 A_{ii} + 2cs A_{ij} + c^2 A_{jj}$

key $\Rightarrow$ $A'_{ij} = A'_{ji} = (c^2 - s^2) A_{ij} + cs\,(A_{ii} - A_{jj})$

$A'_{ik} = A'_{ki} = c A_{ik} - s A_{jk}$ $\quad k \neq i,\ k \neq j$

$A'_{jk} = A'_{kj} = c A_{ik} + s A_{jk}$ $\quad k \neq i,\ k \neq j$

$A'_{kl} = A_{kl}$

(i^{th} and j^{th} columns and rows have changed)

choose $\cos\theta$ and $\sin\theta$ s.t. $A'_{ij} = 0$ (pick largest off-diagonal).

$\rightarrow \tan 2\theta = \dfrac{2A_{ij}}{A_{jj} - A_{ii}}$ $\quad$ ($\theta = \frac{\pi}{4}$ if $A_{ii} = A_{jj}$)

Algorithm:

0 → set Eigv = I

1 → find largest off-diagonal element, determine θ

2 → apply transformation to A, set Eigv = G . Eigv

3* → Is largest off-diagonal element small enough?

yes → converged - report eigenvalues (diagonals)
eigenvectors are rows of Eigv

no → back to 1

* G will approach I

show Mathematica workings