Numerical methods in Linear Algebra

$\underline{A} \in \mathbb{R}^{m\times n}$ or $\mathbb{C}^{m\times n}$ matrix m rows n columns

$\underline{x} \in \mathbb{R}^m = \mathbb{R}^{m\times 1}$ (or $\mathbb{C}$) vector (column vector)

Common problems:

<u>Linear equations</u>: $\underline{A}\,\underline{x} = \underline{b}$, given $\underline{\underline{A}}, \underline{b}$

$\underline{x} = ?$

<u>Eigen value problem</u>

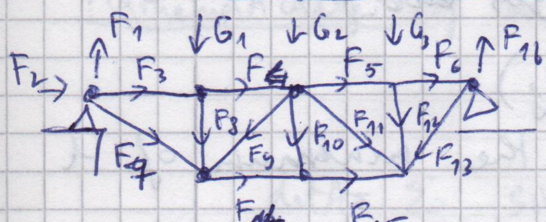
given $\underline{\underline{A}} \in \mathbb{R}^{n\times n}$ (or $\mathbb{C}$)

find $\lambda \in \mathbb{R}$ (or $\mathbb{C}$) $\underline{x}_\lambda \in \mathbb{R}^n$ (or $\mathbb{C}^n$)

such that $\underline{\underline{A}}\,\underline{x}_\lambda = \lambda \cdot \underline{x}_\lambda$

give $\underline{\underline{U}}$ such that $\underline{\underline{U}}\,\underline{\underline{A}}\,\underline{\underline{U}}^{-1}$ = diagonal

<u>A Couple Physics Examples:</u> Bridges in 2D



Assumptions:
1. No torque at the junctions
2. rods take also no torque only pushing and pulling
3. External forces: G_1, G_2, G_3; F_1, F_2, F_{16} unknown.
4. leftmost junction can't move rightmost can slide left-right. (no external force)

$K = 8$ junctions

$R = 13$ rods

$b = 3$ forces from support.

$R + b$ unknowns

$d \cdot K$ eqs. $d = 2$

Statically determined if $R + b = dK$

We want F_i as a function of G_1, G_2, G_3

In all junctions we can write $\sum F = 0$

e.g. $\underline{F}_1 + \underline{F}_2 + \underline{F}_3 + \underline{F}_7 = 0$

$$F_1\begin{pmatrix}0\\1\end{pmatrix} + F_2\begin{pmatrix}1\\0\end{pmatrix} + F_3\begin{pmatrix}1\\0\end{pmatrix} + F_7\begin{pmatrix}1/\sqrt{2}\\-1/\sqrt{2}\end{pmatrix} = 0$$

etc. $\sum_j A_{ij} F_j = G_i'$

Example 2, QM $\hat{H}|\psi\rangle = E|\psi\rangle$

Take an orthonormal basis $|i\rangle$ $i = 1, 2 \dots$ $\langle i|j\rangle = \delta_{ij}$

$$|\psi\rangle = \sum_i x_i |i\rangle$$

$$H|\psi\rangle = \sum_i x_i \hat{H}|i\rangle = E|\psi\rangle = E\sum_i x_i |i\rangle \quad / \langle j| \text{ from left}$$

$$\sum_i x_i \langle j|\hat{H}|i\rangle = E\sum x_i \langle j|i\rangle = E x_j$$

Truncate $i = 1 \dots \infty \to i = 1 \dots N$

$$\sum_i H_{ji} x_i = E x_j \quad \text{Eigenvalue problem for } H_{ji}$$

Example 3,

Classical Many-body problem,

e.g. vibrational spectrum of a molecule

Coordinates $\underline{\varphi}_i$ (positon, angles ~~etc.~~)

Potential energy $U(\varphi_1 \dots \varphi_j)$

wlog: $\varphi_i = 0$ $U(0, 0 \dots 0) = 0$ is the minimum of U

Taylor-expansion around ~~momentum~~ 0:

$$U(\varphi_i) = U(0, \dots 0) + \underbrace{\left(\sum_i \frac{\partial U}{\partial \varphi_i}\right)}_{0 \text{ at min.}} \cdot \varphi_i + \frac{1}{2}\sum_{i,j} \underbrace{\left(\frac{\partial}{\partial \varphi_i}\frac{\partial}{\partial \varphi_j} U\right)}_{\text{at } \varphi = 0 \;=\; A_{ij}} \underline{\varphi}_i \underline{\varphi}_j$$

$$\simeq \frac{1}{2} A_{ij} \varphi_i \varphi_j$$

Kinetic energy: $T(\dot{\varphi}_1 \dots \dot{\varphi}_n) \simeq \frac{1}{2}\sum M_{jk} \dot{\varphi}_j \dot{\varphi}_k$

Similarly "Mass matrix"

Looking for small displacements

Taylor-exp. is justified

Euler-Lagrange eqs: $\frac{\partial \mathcal{L}}{\partial \varphi_k} - \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\varphi}_k} = 0$ with $\mathcal{L} = T - U$

$$\sum_j A_{kj}\varphi_j + M_{kj}\ddot{\varphi}_j = 0$$

A, M symmetric

Ansatz $\varphi_j = a_j \cdot e^{i\omega t}$

$$\sum_j \left(A_{kj} \cdot a_j - M_{kj} \cdot a_j \cdot \omega^2\right) = 0 \qquad \left[\underline{\underline{A}} - \underline{\underline{M}} \cdot \omega^2\right] \underline{a} = 0$$

$\det M \neq 0$ otherwise system not „confined" $\Rightarrow M^{-1}$ exist

$\underline{\underline{M}}^{-1} \underline{\underline{A}} \; \underline{a} = \omega^2 \underline{a}$ Eigenvalue problem
Inverse of M needed

1st Topic | System of Linear equations

$\underline{\underline{A}} \; \underline{x} = \underline{b}$ $\quad \underline{\underline{A}} \in \mathbb{R}^{n \times n}$ (or $\mathbb{C}$ ~ later too)

$\underline{b} \in \mathbb{R}^n \quad b \neq 0$

$\underline{x} = ?$

Mathematically: if $\det \underline{\underline{A}} \neq 0 \Rightarrow$ Cramer's rule

$$x_i = \frac{\det A_i}{\det A} \qquad A_i = \text{i-th column is replaced with } \underline{b}$$

$$\det A = \sum_{p \in P_n} \operatorname{sign}(p) \, A_{1p_1} A_{2p_2} \cdots A_{np_n}$$

$n!$ terms $\Rightarrow$ quickly becomes impractical for numerics

numerically : direct methods

give an exact solution (rounding errors are a problem!)

LU decomposition (Gauss elimination) , QR decomposition

$O(N^3)$ algorithms

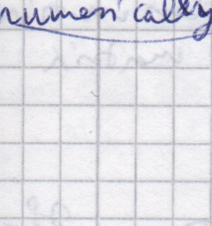
Iterative methods

give an estimate for the solution
improved using many iterations

e.g. Gauss-Seidel , successive overrelaxation (SOR),

Conjugate gradient (CG)

Sparse matrices



have lots of zeroes

Typically $O(N)$ nonzero $\Rightarrow$ Calculating $\underline{A} \cdot \underline{x}$ is cheap

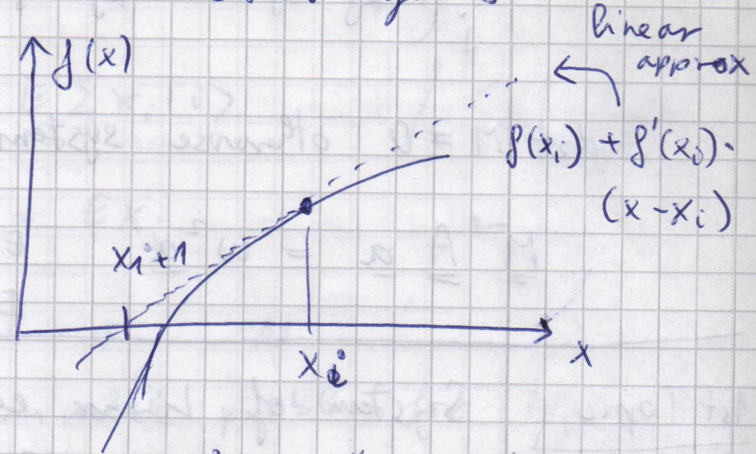
well suited for iterative methods

Example 4. Newton's algorithm for root finding

$f(x) = 0$ $\quad f$ some non-linear function. $\quad x = ?$

Iterative procedure: 1. x_i is our latest guess

2. calculate $f(x_i)$
3. done if $f(x_i) < \varepsilon$
4. $$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$
5. goto 1



if initial guess is close enough, f is "smooth"

$$\Delta_{i+1} \sim \Delta_i^2 \qquad \Delta_i = x_i - x$$

exact solution

Many dimensions:

$$f_k(x_j) = 0 \qquad x_j = ? \qquad k, j = 1 \ldots N$$

Iterative proc.:

1. $x_j^{(i)}$ is latest guess
2. calc $f_k(x_j^{(i)})$
3. done if $< \varepsilon \quad \forall k$
4. $x_e^{(i+1)} = x_e^{(i)} - (D^{-1})_{ek} f_k(\underline{x}^{(i)})$

linear approx:

$$f_k(x_e^{(i)}) + \underbrace{\frac{\partial f_k}{\partial x_e}}_{D_{ke}}(x_e - x_e^{(i)})$$

derivative matrix

Need to calculate the inverse of $D_{ke} = \frac{\partial f_k}{\partial x_e}$ for every iteration

Gauss elimination

$\underline{\underline{A}} \cdot \underline{x} = \underline{b}$ $\quad A \in \mathbb{R}^{n \times n}$ $\quad$ ~~n~~ equations

2x2 example

$$a_{11} x_1 + a_{12} x_2 = b_1$$

$$a_{21} x_1 + a_{22} \cdot x_2 = b_2 \quad \leftarrow \text{ add first eq } \times -\frac{a_{21}}{a_{11}}$$

$$\left(a_{22} - \frac{a_{12} \cdot a_{21}}{a_{11}}\right) \cdot x_2 = b_2 \Rightarrow \text{substitute } x_2 \text{ in first eq} \Rightarrow \text{done}$$

Generally

$$\begin{matrix} A_{11} & \cdots & A_{1n} & b_1 \\ A_{21} & \cdots & A_{2n} & b_2 \\ \vdots & & \vdots & \\ A_{n1} & & A_{nn} & b_n \end{matrix}$$

$\Downarrow$ adding first row $\cdot \left(-\frac{A_{j1}}{A_{11}}\right)$ to jth row

$$\begin{matrix} A_{11} & \cdots & A_{1n} & b_1 \\ 0 & A'_{22} & \cdots A'_{2n} & b'_2 \\ \vdots & \vdots & & \\ 0 & A'_{n2} & A'_{nn} & b'_n \end{matrix}$$

Linear system with $n-1$ variables

do elimination $n-1$ times:

$$\left(\begin{matrix} \tilde{A}_{11} & \cdots & \cdots & \tilde{A}_{1n} \\ 0 & \tilde{A}_{22} & \cdots & \\ \vdots & 0 & \ddots & \\ 0 & & 0 & \tilde{A}_{nn} \end{matrix} \middle| \begin{matrix} \tilde{b}_1 \\ \\ \vdots \\ \tilde{b}_n \end{matrix}\right)$$

Back substitution $\cancel{x_N = \frac{\tilde{b}_N}{\tilde{A}_{NN}}}$ $\quad x_n = \frac{\tilde{b}_n}{\tilde{A}_{nn}}$

$$x_{n-1} = \left(\tilde{b}_{n-1} - \tilde{A}_{n-1,n} x_n\right) \cdot \frac{1}{\tilde{A}_{n-1,n-1}}$$

$$x_j = \frac{1}{\tilde{A}_{jj}} \left(\tilde{b}_j - \sum_{k=j+1}^{n} \tilde{A}_{jk} x_k\right)$$

We can use several RHS.

$$\underline{\underline{A}} \cdot \underline{x}^{(i)} = \underline{b}^{(i)} \qquad i = 1 \ldots N_b$$

$$\underline{\underline{A}} \cdot \underline{\underline{X}} = \underline{\underline{B}}$$

We can use Gauss elimination as before

$n \times n \quad n \times N_b \quad n \times N_b$

Choose $\underline{\underline{B}} = \mathbb{1} \qquad N_b = n \qquad \Rightarrow \underline{\underline{X}} = \underline{\underline{A}}^{-1}$

Determinant: det does not change by adding a factor · row to an other row

$$\det A = \det \tilde{A} = \prod_{i=1}^{n} \tilde{A}_{ii} \quad \text{as } \tilde{A} \text{ is upper triagonal}$$

Pivoting: e.g. $A = \begin{pmatrix} 0 & 5 & 6 \\ -2 & 1 & 3 \\ 4 & 5 & 2 \end{pmatrix}$

Gauss elimination fails at the first step!

$\Rightarrow$ change the order of eqs.

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \times A = \begin{pmatrix} -2 & 1 & 3 \\ 0 & 5 & 6 \\ 4 & 5 & 2 \end{pmatrix}$$

Best stability properties if we take the largest element always by swaping rows

(so here $\begin{pmatrix} 4 & 5 & 2 \\ 0 & 5 & 6 \\ -2 & 1 & 3 \end{pmatrix}$)

(in practice one notes in a permutation vector which swaps took place)

$$p = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} \rightarrow \begin{pmatrix} 3 \\ 2 \\ 1 \\ 4 \end{pmatrix} \qquad \tilde{M}_{ij} = \tilde{M}'(p(i), j)$$

For the determinant we need ~~to~~ to count number of swaps $\det M = \text{sign} \cdot \det \tilde{M}'$

Sources of errors

Floating point representation on the comp:

$1.b_1 b_2 \text{———} b_{52} \cdot 2^M$ for double precision

M has 11 bits

$M = e - 1023$

Machine ε : $1 + \varepsilon = 1$

double : $2^{-53} \approx 1.11 \cdot 10^{-16}$

float : $2^{-24} \approx 5.96 \cdot 10^{-8}$

(look at hypothetical 4 decimal digit computer.)

Common sources of error: Subtracting nearly equal

$1.234 \cdot 10^0 - 1.233 \cdot 10^0 = 1.000 \cdot 10^{-3}$

$1.235 \cdot 10^0 - 1.233 \cdot 10^0 = 2.000 \cdot 10^{-3}$

0.1% error in input $\Rightarrow$ 100% error in output

Adding large number to a small

$0.1234 \cdot 10^3 + 0.1200 \cdot 10^{-2} = 0.1234 \cdot 10^3$