Numerical linear algebra – iterative methods

- comparison of direct (see the winter semester) and iterative methods of solution $Ax=b$

  direct methods (LU decomposition, QR factorization)

  + finite number of steps $\sim n^3$
  + multiple use of the decomposition for several right-hand sides b
  – demanding on memory, even for sparse matrices, for LU decomposition is not usually sparse

  iterative methods

  + many methods need only subroutine providing $A \cdot x$, for sparse matrices it can take only $O(n)$ operations sometimes it is not necessary to store A
  + sometimes only a few iterations is enough to achieve required accuracy $\Rightarrow$ they can be faster than direct methods
  – convergence is not guaranteed for general matrices usually only for special types of matrices
  – it is usually necessary to use preconditioners to get an efficient method, which are often taylored to the given problem

Note: in the floating-point arithmetics, even direct methods do not give accurate results due to round-off errors $\Rightarrow$ iterative methods are used to get more accurate results

- there are two basic classes of iterative methods:
  - stationary iterative methods (Jacobi, Gauss-Seidl, SOR(ω))
    - usually not very efficient, but basic blocks of some other very efficient methods, e.g. multigrid
  - non-stationary iterative methods based on Krylov subspaces (conjugated gradients, GMRES)

- **stationary iterative methods** (see Demmel for details)

- basic idea: write $A = M - K$ in such a way that M is regular and M^{-1} is simple to calculate

and iterate $Ax = Mx - Kx = b$

$$x = M^{-1}Kx + M^{-1}b = Rx + c$$

using simple iterations

$$\boxed{x_{m+1} = Rx_m + c}$$ starting with a suitable choice x_0

if R and c are constant during iterations, we call the method stationary

- convergence depends on properties of the matrix R

by subtracting $x = Rx + c$, which is valid for a solution x of $Ax = b$

we get $$e_{m+1} = x_{m+1} - x = R(x_m - x) = Re_m$$

and thus $$\|x_{m+1} - x\| \leq \|R\| \|x_m - x\| \leq \|R\|^{m+1} \|x_0 - x\|$$

if $\|R\| = \max_{x \neq 0} \frac{\|Rx\|}{\|x\|}$ is the operator norm for the chosen vector norm $\|\cdot\|$

it follows that $x_m \to x$, i.e. $\|x_m - x\| \to 0$ if $\underline{\|R\| < 1}$

- in general, it can be shown that

$\underline{x_m \text{ converges to the solution of } Ax = b}$

for an arbitrary x_0 and right-hand side b

$\underline{\text{if and only}}$ if the $\underline{\text{spectral radius}}$ of the matrix R satisfies the condition

$$\boxed{\varrho(R) \equiv \max_{\substack{\text{eigenvalues} \\ \text{of } R}} |\lambda| < 1}$$

- $\underline{\text{speed of convergence}}$ $$\boxed{r(R) = -\log_{10} \varrho(R)}$$

it roughly says how many correct decimal digits we get in one iteration, for $\|x_{m+1} - x\| \leq \varrho(R) \|x_m - x\|$

and thus $$\log_{10} \|x_m - x\| - \log_{10} \|x_{m+1} - x\| \geq r(R)$$

- goal of iterative methods

1) to choose $M-K$ in such a way that $M^{-1}Kx = Rx$ and $M^{-1}b = c$ is relatively easy to compute (e.g. M can be diagonal or triangular)

2) and, at the same time, in such a way that $\varrho(R)$ is as small as possible

- unfortunately, these two conditions are contradictory:

consider two extreme cases:

1) $M = I$ is optimal for $M^{-1} = I$ but $\varrho(R)$ is then typically greater than one

2) $M = A$ and thus $K = 0 \Rightarrow \varrho(R) = 0$ but A^{-1} is usually difficult

- a compromise is necessary

- Jacobi method

basic idea: if we have an approximation x_m after m iterations we update j-th element from j-th equation and for all other elements of x we use old values:

$$x_{m+1,j} = \frac{1}{a_{jj}} \left(b_j - \sum_{k \neq j} a_{jk} x_{m,k} \right)$$

for $a_{jj} \neq 0$, otherwise we have to reorder equations

- it is actually decomposition

$$A = M - K = D - (\tilde{L} + \tilde{U})$$

D: diagonal elements

$\tilde{L}$, $\tilde{U}$: elements of A above ($\tilde{U}$) and below ($\tilde{L}$) the diagonal taken with opposite sign

and thus $R_{Jac} = D^{-1}(\tilde{L} + \tilde{U})$

$c_{Jac} = D^{-1} b$

example: consider matrix

$$A = \underbrace{\begin{pmatrix} 2 & -1 & 0 & & \\ -1 & 2 & -1 & 0 & \\ & -1 & 2 & \ddots & \\ 0 & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix}}_{n} \Bigg\} n$$

which appears in discretization of 1D Poisson eq. on an equidistant grid

$$\left.\frac{d^2 f}{dx^2}\right|_{x_j} \approx \frac{f(x_{j+1}) - 2f(x_j) + f(x_{j-1})}{h^2}$$

for which we get

$$D^{-1} = \begin{pmatrix} \frac{1}{2} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{2} \end{pmatrix} \quad \text{and} \quad R_{Jac} = \begin{pmatrix} 0 & \frac{1}{2} & & & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ & \frac{1}{2} & 0 & \ddots & \\ & & \ddots & \ddots & \frac{1}{2} \\ 0 & & & \frac{1}{2} & 0 \end{pmatrix} \; n$$

it can be shown, that

$$\varrho(R_{Jac}) = \cos\frac{\pi}{n+1} \approx 1 - \frac{\pi^2}{2(n+1)^2}$$

and for $n \to \infty$ $\varrho(R_{Jac}) \to 1$ and this method converges slower and slower for a denser and denser grid

(e.g. for $n = 100$ is $\varrho(R_{Jac}) = 0{,}999516\ldots$)

Gauss - Seidel method

basic idea: modify Jacobi method by using already updated values of elements $x_1, \ldots x_{j-1}$ for x_j

$$x_{m+1,j} = \frac{1}{a_{jj}} \left(b_j - \sum_{k=1}^{j-1} a_{jk} x_{m+1,k} - \sum_{k=j+1}^{n} a_{jk} x_{m,k} \right)$$

($x_{m+1,k}$: updated values; $x_{m,k}$: old values)

this can be rewritten in the matrix form

$$(D - \tilde{L})\, x_{m+1} = \tilde{U} x_m + b$$

and thus $R_{GS} = (D - \tilde{L})^{-1} \tilde{U}$, $c_{GS} = (D - \tilde{L})^{-1} b$

- on the contrary to Jacobi method, here the order of equations is important, sometimes we can get faster convergence by reordering of variables and equations

example: again consider

$$A = \begin{pmatrix} 2 & -1 & 0 & & \\ -1 & 2 & -1 & 0 & \\ & \ddots & \ddots & \ddots & \\ 0 & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix} \Rightarrow R_{GS} = \underbrace{\begin{pmatrix} 2 & & & \\ -1 & 2 & & 0 \\ & \ddots & \ddots & \\ & & -1 & 2 \end{pmatrix}}_{D - \tilde{L}}^{-1} \underbrace{\begin{pmatrix} 0 & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & 0 \end{pmatrix}}_{\tilde{U}} =$$

$$= \begin{pmatrix} \frac{1}{2} & & & 0 \\ \frac{1}{4} & \ddots & & \\ \vdots & \ddots & \ddots & \\ \frac{1}{2^n} & \cdots & \frac{1}{4} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 0 & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & 0 \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{2} & & 0 \\ & \frac{1}{4} & \ddots & \\ & & \ddots & \frac{1}{2} \\ 0 & \frac{1}{2^n} & & \frac{1}{4} \end{pmatrix}$$

and for $n = 100$ we now get $\varrho(R_{GS}) = 0{,}999033$ which is $< \varrho(R_{Jac})$ but just slightly

- SOR(ω) = successive overrelaxation method

- modification of the Gauss-Seidel method: instead of $x_{m+1,j}$ we use a weighted average with $x_{m,j}$

$$x_{m+1,j} = (1-\omega)\, x_{m,j} + \frac{\omega}{a_{jj}} \left(b_j - \sum_{k=1}^{j-1} a_{jk} x_{m+1,k} - \sum_{k=j+1}^{n} a_{jk} x_{m,k} \right)$$

(weights: $(1-\omega)$ and $\frac{\omega}{a_{jj}}$)

or in the matrix form

$$(D - \omega \tilde{L})\, x_{m+1} = [(1-\omega) D + \omega \tilde{U}]\, x_m + \omega b$$

and thus

$$R_{SOR(\omega)} = (D - \omega \tilde{L})^{-1} [(1-\omega) D + \omega \tilde{U}]$$

$$c_{SOR(\omega)} = (D - \omega \tilde{L})^{-1} \omega b$$

for $\omega > 1$ we have overrelaxation ← this one works

for $\omega = 1$ (G-S method) relaxation

and for $\omega < 1$ underrelaxation

- in general, it can be shown (see for example the book by Demmel) that $\varrho(R_{SOR(\omega)}) \geq |\omega - 1|$ and thus a necessary condition for convergence is $0 < \omega < 2$

moreover if A is symmetric matrix, which is positive definite ($y^T A x = x^T A y$) ($x^T A x > 0$ for all $x \neq 0$)

then $\varrho(R_{SOR(\omega)}) < 1$ for $0 < \omega < 2$

and thus SOR(ω) (and also G-S method for $\omega = 1$) converges

and an optimal choice of ω is

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \varrho(R_{Jac})^2}}$$

note that it depends on the spectral radius of the Jacobi method

and in this case

$$\varrho(R_{SOR(\omega)}) = \left(\frac{\varrho(R_{Jac})}{1 + \sqrt{1 - \varrho(R_{Jac})^2}} \right)^2$$

example: again consider

$$A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & \ddots & \ddots \\ 0 & \ddots & -1 \\ & -1 & 2 \end{pmatrix}, \quad \text{we had } \varrho(R_{Jac}) \simeq 1 - \frac{\pi^2}{2(n+1)^2} \dots$$

and thus $\omega_{opt} \simeq \frac{2}{1 + \frac{\pi}{n+1}}$ (it goes to 2 for $n \to \infty$!)

and $\varrho(R_{SOR(\omega)}) \simeq 1 - \frac{2\pi}{n+1}$ $\Rightarrow$ for $n = 100$ $\varrho(R_{SOR(\omega)}) \simeq 0{,}94\dots$

$\Rightarrow$ much faster convergence

- in general, we do not know $\varrho(R_{Jac})$, but at least we can estimate ω even though SOR(ω) is reasonably fast only for ω close to ω_{opt}

for example we can find an estimate for ω on a coarse grid or from estimation of speed of convergence of Jacobi

- notes on convergence of Jacobi, G-S and SOR(ω) methods

  - in general, convergence is not guaranteed
  - it depends on properties of the matrix A,

    for example it can be shown (see again Dennel) that

    Jacobi and G-S are convergent and $\varrho(R_{GS}) < \varrho(R_{Jac})$

    if A is diagonally dominant (strongly) $|a_{ii}| > \sum_{j \neq i} |a_{ij}|$ for all i

    or A is weakly diagonally dominant and irreducible

    ($|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|$ for all i and at least for one i we have $>$ instead $\geq$)

    (irreducibility means that we cannot reorder rows and columns to get $\left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline 0 & A_{22} \end{array}\right)$)

  - not always G-S is faster that Jacobi etc.