Linear systems.

A very simple example of linear system is the following one:

$$(1)\quad \begin{cases} 3x_1 + 2x_2 - x_3 = 5 \\ 2x_1 - x_2 = 6 \\ x_1 + x_2 - 5x_3 = 7 \end{cases}$$

the unknowns are 3: x_1, x_2, x_3, all at power 1.
the equations are 3.
this is a square system : n° of unknowns = n° of eq
but also rectangular systems can be found and written.
We will consider only square systems in this lesson.

(1) can be re-written in matrix form as follows:

set $A = \begin{bmatrix} 3 & 2 & -1 \\ 2 & -1 & 0 \\ 1 & 1 & -5 \end{bmatrix}$ $\underline{b} = \begin{bmatrix} 5 \\ 6 \\ 1 \end{bmatrix}$ $\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$

and (1) is equivalent to $A\underline{x} = \underline{b}$

where A is the matrix of the coeff.
$\underline{b}$ is the right hand side
$\underline{x}$ is the unknown vector.

In general $A \in \mathbb{R}^{n \times n}$ $\underline{x}, \underline{b} \in \mathbb{R}^{n \times 1}$ and

given $A \in \mathbb{R}^{n \times n}$, $\underline{b} \in \mathbb{R}^{n \times 1}$, we look for $\underline{x} \in \mathbb{R}^{n \times 1}$: (2)

$$A\underline{x} = \underline{b}. \quad (2)$$

(2) is well-posed (i.e. $\exists!$ solution $\underline{x}$)

if 1) A is not singular ($\det(A) \neq 0$)

$\Updownarrow$ (i)

2) A is invertible ($\exists A^{-1}: A \cdot A^{-1} = A^{-1} \cdot A = I$)

$\Updownarrow$ (ii)

3) $\operatorname{rank}(A) = n$ ($\operatorname{rank}(A)$ = maximum n° of linearly independent column vectors (or rows))

4) $A\underline{x} = \underline{0} \Rightarrow \underline{x} = \underline{0}$

5) $\operatorname{rank}(A, \underline{b}) = \operatorname{rank}(\underline{b})$

Example: Idraulic Network. ———————

From now on we will consider only non singular matrices so that (2) is well-posed.

We will see how to solve (2) in an efficient way without using Cramer rule.

Numerical methods to solve (2) are subdivided in two main categories:

Direct methods (DM) Iterative methods. (IM)

DM: A DM yields the exact solution of (2) (unless rounding errors) in a finite n° of operations.

These methods modify the matrix in order to transform

$A\underline{x} = \underline{b}$ in a simpler but equivalent system.

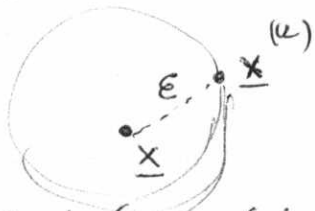
Simpler means triangular.

DMs include : Gaussian Elimination
LU factorization
Cholesky factorization
Multifrontal

IM: An IM yields the solution as a limit of a sequence
ie. given an initial guess $\underline{x}^{(0)}$, an IM builds a sequence $\{\underline{x}^{(k)}\}_{k\geq 0}$ with the aim that

$$\lim_{k\to\infty} \underline{x}^{(k)} = \underline{x} \quad \text{(the exact solution)}.$$

Thus, the exact solution can be reached only at the limit, in practice we will stop our sequence when we will estimate that $\underline{x}^{(k)}$ is sufficiently close to $\underline{x}$, eg. when $\|\underline{x}^{(k)} - \underline{x}\| \leq \varepsilon$, ε = given tolerance
Iterative methods yield rounding errors like direct methods, moreover there is the error ε due to the stopping test.



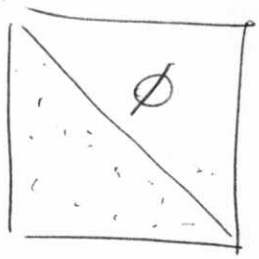
IMs include : classical methods (Jacobi, Gauss-Seidel Richardson)
modern methods (Conjugate gradient GMRES, Krylov methods)

Direct methods

1) The case of triangular matrices.

$L \in \mathbb{R}^{n\times n}$ is said lower triangular if

$$L_{ij} = 0 \quad \forall i<j$$



$U \in \mathbb{R}^{n\times n}$ is said upper triangular if

$$U_{ij} = 0 \quad \forall i>j.$$

A triangular matrix is non-sing iff all the diagonal entries are non null.

As a matter of fact $\lambda_i(A) = a_{ii}$ if A is triang

and $\det(A) = \prod_{i=1}^{n} \lambda_i(A) = \prod_{i=1}^{n} a_{ii}$.

A triangular system can be solved in n^2 floating point operation by using substitution.

1.1) $L\underline{x} = \underline{b}$ (lower)

$$L = \begin{bmatrix} \ell_{11} & & & \\ \ell_{21} & \ell_{22} & & \\ \ell_{31} & \ell_{32} & \ell_{33} & \\ & & & \ell_{nn} \end{bmatrix}$$

$$x_1 = b_1/\ell_{11}$$

$$\ell_{21}x_1 + \ell_{22}x_2 = b_2 \rightarrow x_2 = \frac{b_2 - \ell_{21}x_1}{\ell_{22}}$$

$$\ell_{31}x_1 + \ell_{32}x_2 + \ell_{33}x_3 = b_3 \rightarrow x_3 = \frac{b_3 - \ell_{31}x_1 - \ell_{32}x_2}{\ell_{33}}$$

$$\vdots$$

forward substitution

for $i = 1 : n$

$$x_i = \left(b_i - \sum_{j=i+1}^{n} \ell_{ij}x_j\right)/\ell_{ii}$$

Note bb:

$x = L \backslash b$ (if L is lower tr. \ invokes forward subs)

(5)

1.2) by $U\underline{x} = \underline{b}$ U upper triangular

$$U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ 0 & 0 & u_{33} & u_{34} \\ & \varnothing & & u_{nn} \end{bmatrix}$$

$$x_n = b_n / u_{nn}$$

$$u_{n-1,n-1} x_{n-1} + u_{n-1,n} x_n = b_{n-1}$$

$$x_{n-1} = \frac{b_{n-1} - u_{n-1,n} x_n}{u_{n-1,n-1}}$$

backward substitution

for $i = n:-1:1$

$$x_i = \left(b_i - \sum_{j=1}^{i-1} u_{ij} x_j\right) / u_{ii}$$

Matlab: $x = U \backslash b$ (if U is upper \ invokes backward subs)

If A is a generic matrix

we try to transform $A\underline{x} = \underline{b}$ in an equivalent Triangular system

or in a couple of tri. syst. 20'

LU factorization

Given $A \in \mathbb{R}^{n \times n}$ non sing

We look for $L, U \in \mathbb{R}^{n \times n}$ L lower triang, U upper triang

such that $L \cdot U = A$.

Once L and U have been computed

it holds: $Ax = \underline{b} \iff LU\underline{x} = \underline{b} \iff L(U\underline{x}) = \underline{b}$

$A = LU$ associativity of matrix product

$$\iff \begin{cases} L\underline{y} = \underline{b} \\ U\underline{x} = y \end{cases}$$

solve

1) How to compute L and U ?

2) Given A, L and U are unique ? NO

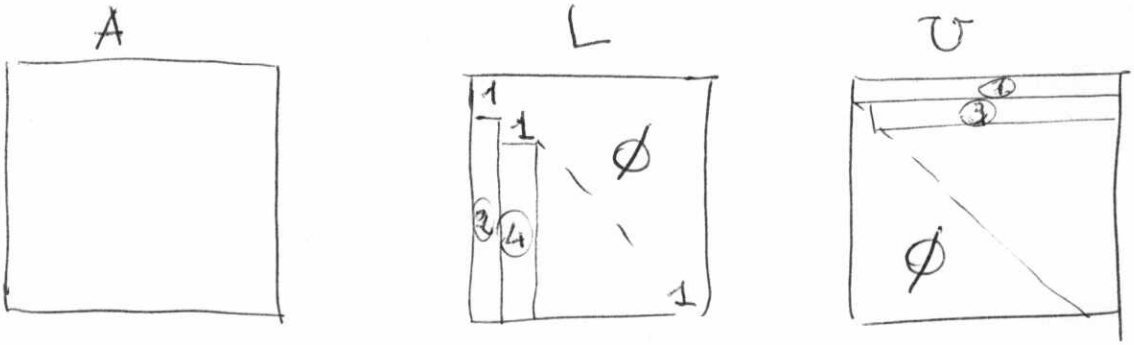
(think, e.g. to the factorization of a n°

$$12 = 3 \cdot 4$$
$$= 2 \cdot 6$$

But, if we fix n entries in L (or in U)
then $\exists ! L, U : LU = A$.

In general we take $L_{ii} = 1 \quad i = 1, \dots, n$ -

the LU factorization with $L_{ii} = 1$ is called also Gauss-factorization.



by exploiting the relations

$$A = L \cdot U$$
$$(A_{ij}) = (LU)_{ij} = \sum_{k=1}^{n} L_{ik} U_{kj} \quad \text{for } i = 1:n, \; j = 1:n$$

the computational cost is $\frac{2}{3} n^3$ + low order Terms

Matlab:

```
A=[3,2,-1; 2,-1,0; 1,1,-5]; b=[5;6;1]
[L,U]=lu(A);
y=L\b; x=U\y;
```

Gaussian Elimination (GEM)

Given $A\underline{x} = \underline{b}$, this method modifies A and $\underline{b}$ in $(n-1)$ steps in order to obtain an upper triangular system

$\tilde{A}\underline{x} = \tilde{\underline{b}}$ equivalent to $A\underline{x} = \underline{b}$.

(i.e. with the same solution)

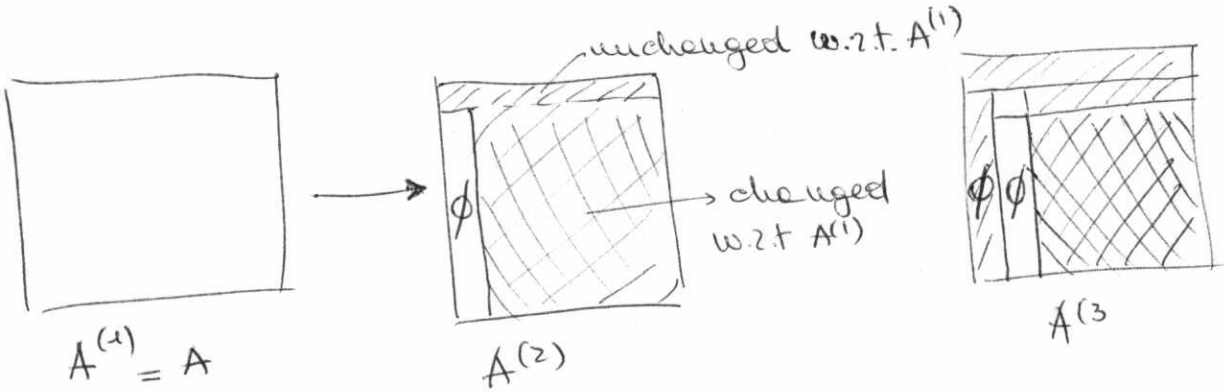
This method exploits 3 simple rules:

1) if we multiply an equation by a non-null real number we obtain a system equivalent to the original one

2) if we replace an eq with the linear combination of it with another eq of the system we obtain a system equivalent to the original one

3) by changing the order of the eqs we obtain a system equiv to the original one.

The idea is to transform A in $\tilde{A}$ upper triang and $\underline{b}$ in $\tilde{\underline{b}}$ so that $Ax = b$ and $\tilde{A}x = \tilde{b}$ are equiv.



$$\left[\begin{array}{ccc} \mathbf{2} & 0 & 3 \\ 2 & 2 & 2 \\ 3 & 6 & 4 \end{array}\right] \quad \left[\begin{array}{c} 5 \\ 6 \\ 13 \end{array}\right]$$

Row 2 ← Row2 − 2/2 Row1

Row 3 ← Raw3 − 3/2 Row 1

we compute the multipliers $m_{21} = \frac{2}{2} = \frac{a_{21}}{a_{11}}$

$m_{31} = 3/2 = \frac{a_{31}}{a_{11}}$

$$\begin{bmatrix} 2 & 0 & 3 \\ 2 & 2 & 2 \\ 3 & 6 & 4 \end{bmatrix} \begin{bmatrix} 5 \\ 6 \\ 13 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 0 & 3 \\ 0 & 2 & -1 \\ 0 & 6 & -1/2 \end{bmatrix} \begin{bmatrix} 5 \\ 1 \\ 11/2 \end{bmatrix}$$

(1)

$4 - \frac{3}{2} \cdot 3 = -\frac{1}{2}$

$13 - \frac{3}{2} \cdot 5 = \frac{11}{2}$

(2)

$m_{32} = \frac{a_{32}}{a_{22}} = 3$

$R3 \leftarrow R3 - m_{32} R2$

(3)

$$\begin{bmatrix} 2 & 0 & 3 \\ 0 & 2 & -1 \\ 0 & 0 & 5/2 \end{bmatrix} \begin{bmatrix} 5 \\ 1 \\ 5/2 \end{bmatrix}$$

$-\frac{1}{2} - 3 \cdot (-1) = \frac{5}{2}$

$\frac{11}{2} - 3 \cdot 1 = \frac{5}{2}$

$$\tilde{A} = \begin{bmatrix} 2 & 0 & 3 \\ 0 & 2 & -1 \\ 0 & 0 & 5/2 \end{bmatrix} \quad \tilde{b} = \begin{bmatrix} 5 \\ 1 \\ 5/2 \end{bmatrix}$$

```
for k = 1 : n-1
    for i = k+1 : n
        m_ik = a_ik / a_kk
        for j = k+1 : n
            a_ij = a_ij - m_ik · a_kj
        b_i = b_i - m_ik b_k
```

GEM costs exactly as LU + back + for $= \frac{2}{3} n^3$ + back

Pivoting

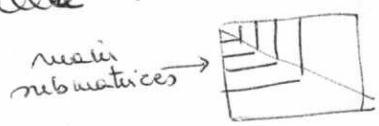
Ex $A = \begin{bmatrix} 1 & 1 & 3 \\ 2 & 2 & 2 \\ 3 & 6 & 4 \end{bmatrix}$ $\underline{b} = \begin{bmatrix} 5 \\ 6 \\ 13 \end{bmatrix}$

$A^{(2)} = \begin{bmatrix} 1 & 1 & 3 \\ 0 & 0 & -1 \\ 0 & 3 & -5 \end{bmatrix}$ $\underline{b}^{(2)} = \begin{bmatrix} 5 \\ -4 \\ -2 \end{bmatrix}$

$A^{(3)}$ $R_3 \leftarrow R_3 - \frac{3}{0} R_2$ it is not possible

?

Prop: If A is non singular, the LU fact $\exists$! (or equiv GEM terminates) iff. all main submatrices of A are non singular.



But we know that if A is non sing, then $\exists$! solution $\underline{x}$.
In order to compute it we have to change the algorithm by using the exchange of rows.
Eg. if we change $R_2 \leftrightarrow R_3 \Rightarrow$ $A^{(2)}$ is upper triang and we can solve the system.

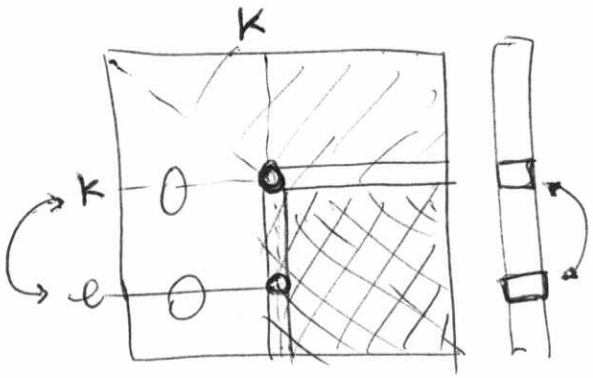
Then the pivoting is a technique to exchange rows of a system in order to guarantee the termination of GEM (or LU) $\forall$ non-sing matrix.
In practice pivoting is performed almost always even if all submatrices are non-sing.

As a matter of fact $m_{ik} = \frac{a_{ik}}{a_{kk}}$

are responsible of rounding errors propagations

and the smaller they are the smaller is the propagation of rounding errors is.

At step k:



we exchange R_k with R_e where

e: $$|a_{ek}| = \max_{k \le i \le n} |a_{ik}| \quad .$$

In Matlab:

$$[L, U, P] = lu(A)$$

P is a permutation matrix which stores the indeces of pivoted rows. such that

$$P \cdot A = LU$$

$$\Rightarrow \quad A\underline{x} = \underline{b}$$

$$\hookleftarrow \underbrace{PA}_{=} \underline{x} = P\underline{b}$$

$$\underbrace{(LU)}_{\underline{y}} \underline{x} = P\underline{b}$$

$$\begin{cases} L\underline{y} = P\underline{b} & \underline{y} = L \setminus (P * \underline{b}) \\ U\underline{x} = \underline{y} & \underline{x} = U \setminus \underline{x} \end{cases}$$

GEM with pivoting is implemented in matlab in the command

$x = A \setminus b$ (where A is full)

When A is symmetric and positive definite
(i.e. $A^T = A$ and $\underline{x}^T A \underline{x} > 0 \quad \forall \underline{x} \in \mathbb{R}^n, \underline{x} \neq \underline{0}$)

LU fact can be replaced by Cholesky factorization.
which is cheaper, it costs about $\frac{n^3}{3}$.

Cholesky looks for a matrix R upper triang:

$$A = R^T R$$

$$A\underline{x} = \underline{b}$$
$$\Updownarrow$$
$$R^T \underbrace{R\underline{x}}_{\underline{y}} = \underline{b} \iff \begin{cases} R^T \underline{y} = \underline{b} \\ R\underline{x} = \underline{y} \end{cases}$$

Matlab

```
A = [ ... ] ; b = [ ... ]
R = chol (A);
y = (R')\b ;
x = R\y
```

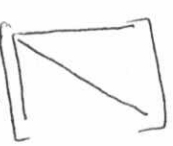
When A is sym. if we can store only $\frac{1}{2}$ matrix
(either the upper or lower triang)

A sparse matrix has a lot of zero entries

A matrix is sparse if the null entries prevail

A matrix is full if the non-null entries prevail.

When a sparse matrix is factorized, it can occur that L and U become full, it depends on the structure of A.

eg. $A =$  $L =$  $U =$ 

sparse FILL-IN full

Condition number

even if $A \in \mathbb{R}^{n\times n}$ $\lambda_i(A) \in \mathbb{C}$

$\forall A \in \mathbb{R}^{n\times n}$ SPD $\quad \|A\|_2 = \sqrt{\rho(A^TA)}$ where $\rho(A) = \max_{1\le i\le n} |\lambda_i(A)|$ is the spectral radius

if A is SPD $\Rightarrow \|A\|_2 = \rho(A)$ (since $\lambda_i(A) \in \mathbb{R}^+$)

it is a norm for matrices

$\forall$ non singular A:

Def $K_2(A) = \|A\|_2 \cdot \|A^{-1}\|_2$ condition number of A

if A is SPD $K_2(A) = \dfrac{\max \lambda_i(A)}{\min \lambda_i(A)}$

$K_2(A) \ge 1$ and $K_2(A) = 1$ if $A = I$ or A is ortogonal $(A^{-1} = A^T)$

matrices with $K_2(A)$ large / small are (named) clubbed

(ill-conditioned / well-conditioned)

What is the meaning of $K_2(A)$.

$K_2(A)$ gives a measure of the propagation of errors on the data during the solution of the system $A\underline{x} = \underline{b}$.

$\hat{A}, \hat{\underline{b}}$ perturbed $\quad \hat{A}\hat{\underline{x}} = \hat{\underline{b}}$

$$\frac{\|\hat{\underline{x}} - \underline{x}\|_2}{\|\underline{x}\|_2} \le K_2(A) \cdot \left[\frac{\|\hat{A} - A\|_2}{\|A\|_2} + \frac{\|\hat{\underline{b}} - \underline{b}\|_2}{\|\underline{b}\|_2} \right]$$

error on the solution — err on the data

$K_2(A)$ is the amplification factor of the errors

if $e_d \sim 10^{-16}$ and $K_2(A) \sim 10^8 \Rightarrow e_x \sim 10^{-8}$

we loose 8 orders of magnitude on the accuracy of the sol.

This is called a-priori estimate, since the error can be predicted, before solving the system, only by knowing the data.

The gradient method

Let $A \in \mathbb{R}^{n\times n}$ sym and pos def. , $\underline{b} \in \mathbb{R}^n$

Define $\phi(\underline{x}) = \frac{1}{2}\underline{x}^T A \underline{x} - \underline{x}^T \underline{b}$ $\qquad \phi : \mathbb{R}^n \longrightarrow \mathbb{R}$

ϕ is a convex function, it is $C^2(\mathbb{R}^n)$
and it has a unique stationary point ($\nabla\phi(x) = 0$)
this stat. point is a global minimizer ~~point~~.
Let $\underline{x}^*$ the ~~minimum point~~ minimizer. of ϕ:

$$\phi(\underline{x}^*) = \min_{\underline{x}\in\mathbb{R}^n} \phi(\underline{x})$$

We evaluate

$$\nabla\phi(\underline{x}) = A\underline{x} - \underline{b}$$

therefore $\underline{x}^*$ is such that $\nabla\phi(\underline{x}^*) = 0 \iff A\underline{x}^* = \underline{b}$

i.e. the ! solution of $A\underline{x} = \underline{b}$
is the! global minimizer of ϕ.

We look for the minimizer of ϕ.

The problem of solving $A\underline{x} = \underline{b}$ has been changed in a minimization problem.

~~Idea~~ To look for the minimizer of ϕ we build a sequence $\{\underline{x}^{(k)}\}_{k\geq 0}$ with the aim that $\underline{x}^{(k)} \xrightarrow[k\to\infty]{} \underline{x}^*$

Given an initial guess $\underline{x}^{(0)} \in \mathbb{R}^n$
we define $\underline{x}^{(1)}$ such that $\phi(\underline{x}^{(1)}) < \phi(\underline{x}^{(0)})$
and we exploit the knowledge of $\nabla\phi(\underline{x}^{(0)})$
As a matter of fact $-\nabla\phi(\underline{x}^{(0)})$ is the direction along ~~with~~ which the function ϕ ~~has~~ the decreases most rapidly

$-\nabla\phi(\underline{x}^{(0)}) = \underline{b} - A\underline{x}^{(0)} = \underline{r}^{(0)}$ residual at step 0
it measures how much the distance from the solution is.

we want to move along $\underline{r}^{(0)}$, in the sense that the new point $\underline{x}^{(1)}$ will be chosen on the line

$$\underline{x}^{(\alpha)} = \underline{x}^{(0)} + \alpha \underline{r}^{(0)} \quad (\alpha \in \mathbb{R})$$

When α spans $\mathbb{R}$, $\underline{x}^{(\alpha)}$ spans the line passing at $\underline{x}^{(0)}$ with the direction of $\underline{r}^{(0)}$.

Among all possible $\alpha \in \mathbb{R}$ we choose the value α_1 such that $\phi(\underline{x}^{(\alpha_1)}) = \min_{\alpha \in \mathbb{R}} \phi(\underline{x}^{(\alpha)})$

Since $\phi(\underline{x}^{(\alpha)})$ is a parabola, α_k is computed exactly by deriving ϕ and asking that $\phi'(\alpha) = 0$ alphak.m

we found $\alpha_1 = \dfrac{(\underline{r}^{(0)})^T \underline{r}^{(0)}}{\underline{r}^{(0)T} A \underline{r}^{(0)}}$

and $\underline{x}^{(1)} = \underline{x}^{(0)} + \alpha_1 \underline{r}^{(1)}$.

Once $\underline{x}^{(1)}$ is computed, we iterate the process:
we compute the new $-\nabla\phi(\underline{x}^{(1)}) = \underline{r}^{(1)}$
the new value of α_2 ....
until the convergence is reached.

What means convergence?

we have $\lim_{k\to\infty} \underline{x}^{(k)} = \underline{x}^* \iff \lim_{k\to\infty} \|\underline{x}^{(k)} - \underline{x}^*\| = 0$

\# $\|\underline{x}^{(k)} - \underline{x}^*\|$ is a measure of the convergence

but $\underline{x}^*$ is unknown and we cannot evaluate $\|\underline{x}^{(k)} - \underline{x}^*\|$

We use estimators of the true error $\|x^{(k)} - x^*\|$

such as $\|x^{(k+1)} - x^{(k)}\|$ or $\frac{\|r^{(k)}\|}{\|b\|}$

and we stop when the estimator is less than a given tolerance ε.

the gradient method reads.

given ε

given $x^{(0)}$, $r^{(0)} = b - A x^{(0)}$

for $k = 0, 1,$ until conv

$$\alpha_k = \frac{(r^{(k)})^T r^{(k)}}{r^{(k)T} A r^{(k)}}$$

$$x^{(k+1)} = x^{(k)} + \alpha_k r^{(k)}$$

$$r^{(k+1)} = r^{(k)} - \alpha_k A r^{(k)}$$

$$err = \|r^{(k+1)}\| / \|b\|$$

dis_xk.m

storie1.m

thm If A is spd $\Rightarrow$ grad method converges to x^* $\forall\, x^{(0)} \in \mathbb{R}^n$

and the convergence rate depends on $K_2(A)$: the smaller $K_2(A)$ is and the faster the convergence is to a fixed tolerance ε.

The convergence is quite slow

the gradient method can move along parallel directions. In the example

$\underline{r}^{(1)} \parallel \underline{r}^{(3)} \parallel \underline{r}^{(5)} \dots$

$\underline{r}^{(2)} \parallel \underline{r}^{(4)} \parallel \underline{r}^{(6)} \dots$

We want a method that does not ~~come back~~ use the same direction more than once.

With this request the convergence to $\underline{x}^*$ is guaranteed ~~almost~~ at most in n steps (where n = size(A))

the ~~gra~~ new method is called

CONJUGATE GRADIENT

and, given $x^{(k)}$, $\underline{x}^{(k+1)} = \underline{x}^{(k)} + \alpha_k \underline{p}^{(k)}$

where $\underline{p}^{(k)}$ is called descent direction.

It coincides with the residual ($\underline{r}^{(k)} = -\nabla \phi$) only at the first step

while during the successive steps

$$\underline{p}^{(k+1)} = \underline{r}^{(k+1)} - \beta_k \underline{p}^{(k)} \quad , \beta_k = \frac{\underline{r}^{(k+1)T} \underline{r}^{(k+1)}}{\underline{r}^{(k)T} \underline{r}^{(k)}}$$

With this choice it holds:

$$\begin{cases} (r^{(k+1)})^T p^{(j)} = 0 & j = 0, \dots, k \\ p^{(k+1)T} A p^{(j)} = 0 & j = 0, \dots, k \end{cases}$$

Thm: if A is spd the convergence is guaranteed $\forall \underline{x}^{(0)} \in \mathbb{R}^n$ at most in n steps in exact arithmetic and the convergence rate is greater than that of gradient method.

CG:

$x^{(0)}$ given

$r^{(0)} = b - A x^{(0)}$; $p^{(0)} = r^{(0)}$

for $k = 0, 1, \ldots$ until conv

$$\alpha_k = \frac{p^{(k)T} r^{(k)}}{p^{(k)T} A p^{(k)}}$$

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$$

$$r^{(k+1)} = r^{(k)} - \alpha_k A p^{(k)}$$

$$\beta_k = \left(r^{(k+1)T} r^{(k+1)}\right) / \left(r^{(k)T} r^{(k)}\right)$$

$$p^{(k+1)} = r^{(k+1)} + \beta_k p^{(k)}$$

When A is not SPD it is possible to extend the definition of CG method and we have Bi-CGstab, CGS, BiCG

Another method on a similar spirit is GMRES.

Given $\phi(\underline{x}) = \frac{1}{2}\underline{x}^T A \underline{x} - \underline{x}^T \underline{b}$ $\quad \underline{x} \in \mathbb{R}^n, A \in \mathbb{R}^{n \times n}, \underline{b} \in \mathbb{R}^n$

$$\nabla\phi(\underline{x}) = \begin{bmatrix} \frac{\partial \phi}{\partial x_1}(\underline{x}) \\ \frac{\partial \phi}{\partial x_2}(\underline{x}) \\ \vdots \\ \frac{\partial \phi}{\partial x_n}(\underline{x}) \end{bmatrix}$$

Given $\underline{x}, \underline{y} \in \mathbb{R}^n$: $(\underline{x}, \underline{y}) = \sum_{i=1}^{n} x_i y_i = \underline{y}^T \underline{x} = \underline{x}^T \underline{y}$

$\nabla\phi(\underline{x})$ is computed by this formula :

let $\underline{y} \in \mathbb{R}^n$

$$(\nabla\phi(\underline{x}), \underline{y}) = \underline{y}^T \nabla\phi(\underline{x}) = \lim_{h \to 0} \frac{\phi(\underline{x} + h\underline{y}) - \phi(\underline{x})}{h} =$$

$$= \lim_{h \to 0} \frac{\frac{1}{2}(\underline{x}+h\underline{y})^T A (\underline{x}+h\underline{y}) - (\underline{x}+h\underline{y})^T \underline{b} - \frac{1}{2}\underline{x}^T A \underline{x} + \underline{x}^T \underline{b}}{h}$$

$$= \lim_{h \to 0} \frac{\frac{1}{2}\underline{x}^T A \underline{x} + \frac{h}{2}\underline{x}^T A \underline{y} + \frac{h}{2}\underline{y}^T A \underline{x} + \frac{h^2}{2}\underline{y}^T A \underline{y} - \underline{x}^T \underline{b} - h\underline{y}^T \underline{b} - \frac{1}{2}\underline{x}^T A \underline{x} + \underline{x}^T \underline{b}}{h}$$

$$= \lim_{h \to 0} \frac{\frac{h^2}{2}\underline{y}^T A \underline{y} + h\underline{y}^T A \underline{x} - h\underline{y}^T \underline{b}}{h}$$

$$= \underline{y}^T (A\underline{x} - \underline{b}) = (A\underline{x} - \underline{b}, \underline{y})$$

$$\Rightarrow (\nabla\phi(\underline{x}), \underline{y}) = (A\underline{x} - \underline{b}, \underline{y}) \qquad \forall \underline{y} \in \mathbb{R}^n$$

$$\Rightarrow \nabla\phi(\underline{x}) = A\underline{x} - \underline{b} = -\underline{r}(\underline{x})$$

↑ residual.