Numerical Methods of Linear Algebra for Sparse Matrices

Lecture 10

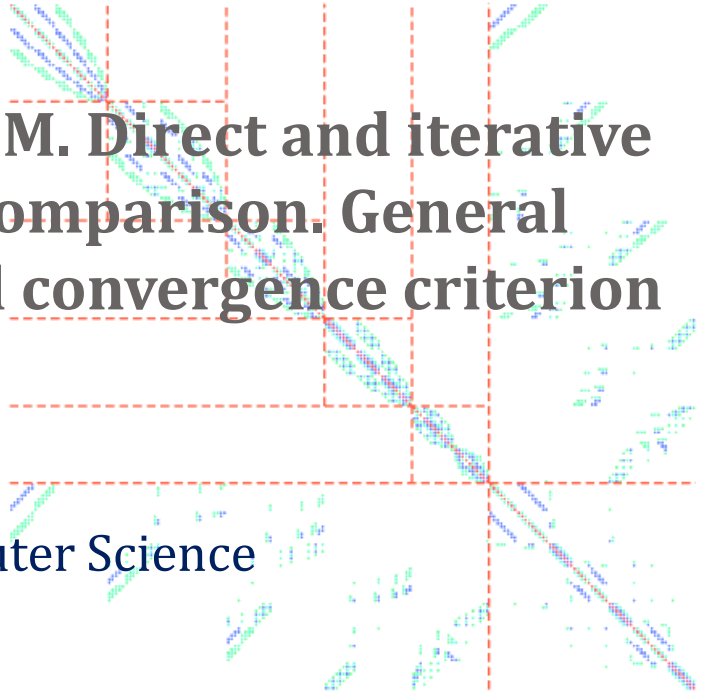
Discretization of PDE: overview of FEM. Direct and iterative methods for sparse linear systems: comparison. General formulation of iterative methods and convergence criterion

Anna Nasedkina

Department of Mathematical Modeling

Institute of Mathematics, Mechanics and Computer Science

Southern Federal University



Discretization of PDEs

Finite element method: overview and assembly process

Overview of finite element method

- Finite element method (FEM): unknown functions are approximated by piecewise-polynomial functions that are continuous on small elements of simple shape
- These functions are called shape functions, or test functions
- In FEM, we use weak formulation of the problem, which is based on Green's formula

Weak formulation of Poisson's equation

Consider 2D Poisson's equation

$$\begin{cases} -\Delta u(\underline{x}) = f(\underline{x}) \\ u(\underline{x})\big|_{\Gamma} = 0 \end{cases}$$

for $\underline{x} = (x_1, x_2)$ in a bounded open domain $\Omega \subset \mathbb{R}^2$,

$\Gamma = \partial\Omega$ is the boundary of Ω, $\overline{\Omega} = \Omega \cup \partial\Omega$ is the closed domain

$u(\underline{x}) = u(x_1, x_2)$ is a scalar *unknown function*, $f(\underline{x})$ is a known function

$\Delta = \dfrac{\partial^2}{\partial x_1^2} + \dfrac{\partial^2}{\partial x_2^2}$ is Laplacian: $\Delta u = \dfrac{\partial^2 u}{\partial x_1^2} + \dfrac{\partial^2 u}{\partial x_2^2}$

$\underline{n}$ is external unit vector normal to Γ(directed outwards)

To set a **weak formulation**, we multiply both parts of the equation by some function $v(\underline{x})$, which is at least twice differentiable, and integrate by Ω

Weak formulation of Poisson's equation

$-\Delta u = f \,\big|\, \cdot v$ and integrate by Ω, $\cdot$ is dot product (inner product)

$$-\int_{\Omega} \Delta u \cdot v \; d\underline{x} = \int_{\Omega} f \cdot v \; d\underline{x}$$

Apply Green's formula:

$$\int_{\Omega} \nabla v \cdot \nabla u \; d\underline{x} = -\int_{\Omega} v \cdot \Delta u \; d\underline{x} + \int_{\Gamma} v \cdot \frac{\partial u}{\partial \underline{n}} \; ds, \text{ where}$$

$$\nabla = \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2} \right) \text{ is the gradient in 2D:}$$

$$\nabla u = \left(\frac{\partial u(x_1, x_2)}{\partial x_1}, \frac{\partial u(x_1, x_2)}{\partial x_2} \right) \text{ is the vector-function, } \frac{\partial u}{\partial \underline{n}} = \nabla u \cdot \underline{n}$$

After applying Green's formula, we get:

$$\int_{\Omega} \nabla v \cdot \nabla u \; d\underline{x} - \int_{\Gamma} v \cdot \frac{\partial u}{\partial \underline{n}} \; ds = \int_{\Omega} f \cdot v \; d\underline{x}, \text{ where } \int_{\Gamma} v \cdot \frac{\partial u}{\partial \underline{n}} \; ds = 0 \text{ from BC, as } u\big|_{\Gamma} = 0$$

$$\int_{\Omega} \nabla v \cdot \nabla u \; d\underline{x} = \int_{\Omega} f \cdot v \; d\underline{x}$$

Weak formulation of Poisson's equation

$$\int_\Omega \nabla v \cdot \nabla u \; d\underline{x} = \int_\Omega f \cdot v \; d\underline{x}$$

Let's denote bilinear forms: $\int_\Omega \nabla v \cdot \nabla u \; d\underline{x} = a < u, v >$, $\int_\Omega f \cdot v \; d\underline{x} =< f, v >$

$a < u, v >=< f, v >$

In FEM we search for solution $u \in V$, such that $a < u, v >=< f, v >$ for $v \in V$

V is the subspace of low-degreee piecewise-polynomial functions, which are defined on small pieces ("finite elements") of the domain Ω

From BC $u\big|_\Gamma = 0 \;\Rightarrow\; v\big|_\Gamma = 0 \quad \forall v \in V$

Ω_h is approximation of Ω with finite elements K_i: $\Omega_h = \bigcup_{i=1}^{m} K_i$

V_h is the subspace that approximates V and consists of piecewise-linear functions defined on Ω_h, which vanish on Γ:

$V_h = \{\varphi \mid \varphi \text{ is continuos on } \Omega_h, \text{ linear on } K_i, \; \varphi\big|_{\Gamma_h} = 0\}$

Weak formulation of Poisson's equation

$$\int_\Omega \nabla v \cdot \nabla u \; d\underline{x} = \int_\Omega f \cdot v \; d\underline{x}$$

Let's denote bilinear forms: $\int_\Omega \nabla v \cdot \nabla u \; d\underline{x} = a < u, v >$, $\int_\Omega f \cdot v \; d\underline{x} =< f, v >$

$a < u, v >=< f, v >$

In FEM we search for solution $u \in V$, such that $a < u, v >=< f, v >$ for $v \in V$

V is the subspace of low-degreee piecewise-polynomial functions, which are defined on small pieces ("finite elements") of the domain Ω

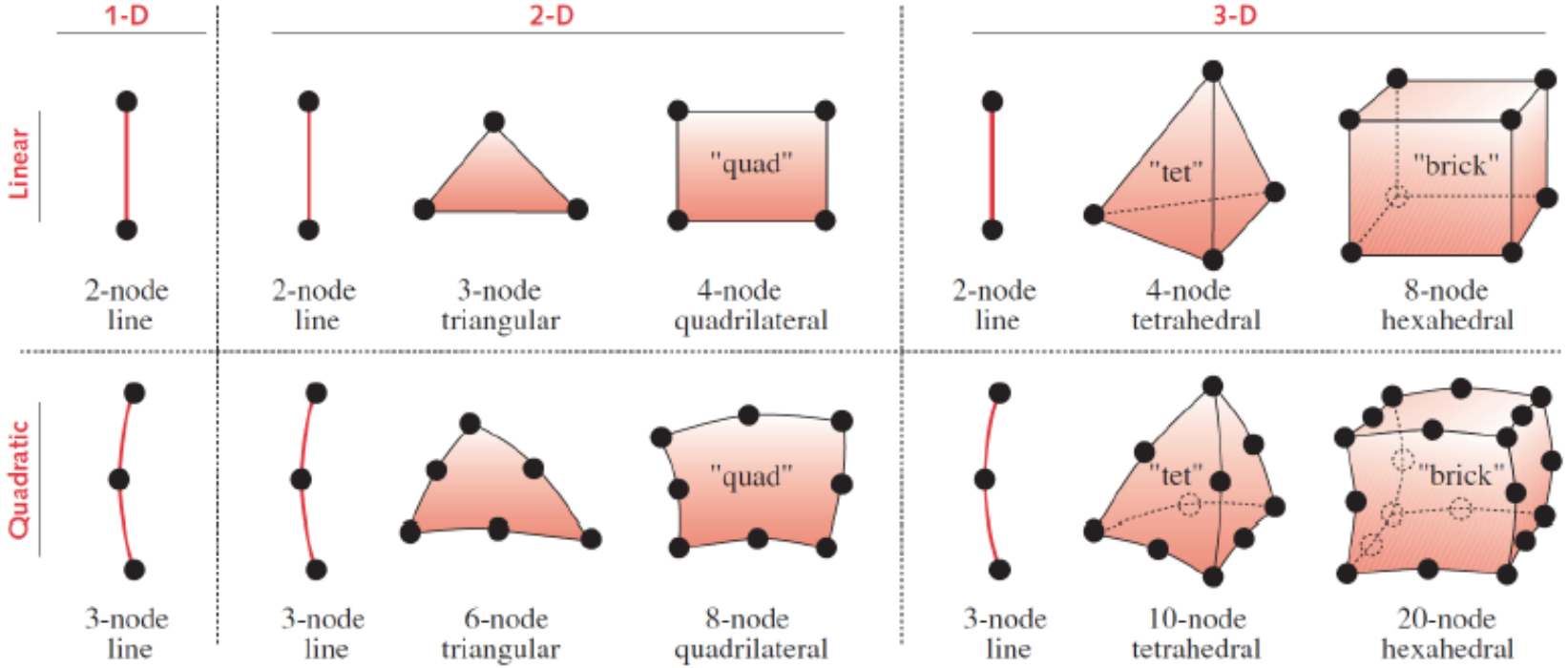
From BC $u\big|_\Gamma = 0 \;\Rightarrow\; v\big|_\Gamma = 0 \quad \forall v \in V$

Ω_h is approximation of Ω with finite elements K_i: $\Omega_h = \bigcup_{i=1}^{m} K_i$

V_h is the subspace that approximates V and consists of piecewise-linear functions defined on Ω_h, which vanish on Γ:

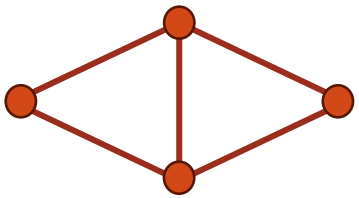
$V_h = \{\varphi \mid \varphi \text{ is continuos on } \Omega_h, \text{ linear on } K_i, \; \varphi\big|_{\Gamma_h} = 0\}$

Finite elements at glance

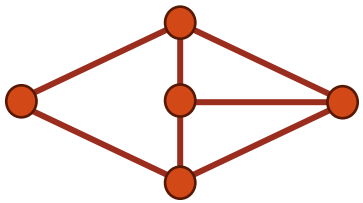


Rule to discretize the domain: adjacent elements should share the same nodes in the vertices; vertex of one element should no lie on the edge of another element

correct



wrong



Finite element approximation

Consider x_j are the nodes of triangulation, $j = \overline{1,n}$, n is the number of nodes

$\varphi_j(x_i) = \delta_{ij} = \begin{cases} 1, & x_i = x_j \\ 0, & x_i \neq x_j \end{cases}$; $\{\varphi_j\}_{j=1}^n$ is the basis in V_h, δ_{ij} is Kronecker delta

$\forall \varphi \in V_h \;\; \varphi(x) = \sum_{j=1}^{n} \xi_j \varphi_j(x)$ is represented by linear combination

We search for approximate solution $u \in V_h$, such that

$a < u, v > = < f, v >$ for $v \in V_h \;\Rightarrow\;$ find $u \in V_h$:

$a < u, \varphi_i > = < f, \varphi_i >, \; i = \overline{1,n}$

Thus we can search for solution in the form $u(x) = \sum_{j=1}^{n} \xi_j \varphi_j(x)$

$a < \sum_{j=1}^{n} \xi_j \varphi_j, \varphi_i > = < f, \varphi_i >, \; i = \overline{1,n}$

Finite element approximation

$$a < \sum_{j=1}^{n} \xi_j \varphi_j, \varphi_i >=< f, \varphi_i >, \; i = \overline{1,n}$$

$$\sum_{j=1}^{n} \xi_j a < \varphi_i, \varphi_j >=< f, \varphi_i >$$

Denote $\alpha_{ij} = a < \varphi_i, \varphi_j >, \; \beta_i =< f, \varphi_i >,$ we get a linear system:

$$\sum_{j=1}^{n} \alpha_{ij} \xi_j = \beta_i, \text{ where } \xi_j \text{ are unknowns, } i, j = \overline{1,n}$$

In matrix form: $Ax = b$

Matrix $A = \{\alpha_{ij}\}$, right-hand side vector $b = \{\beta_i\}$,

vector of unknowns $x = \{\xi_j\}, \; i, j = \overline{1,n}$

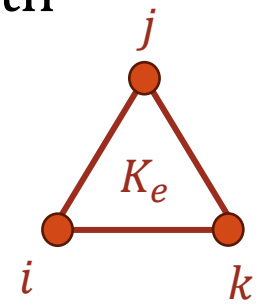
In FEM matrix A is built by assembly process (summing up the contributions of all elements): $a < \varphi_i, \varphi_j >= \sum_K a_K < \varphi_i, \varphi_j >$

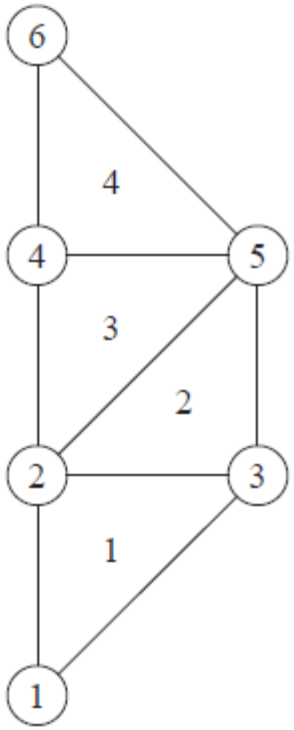
$a_K < \varphi_i, \varphi_j >$ is zero unless the nodes i and j are both vertices of K

Assembly process in FEM

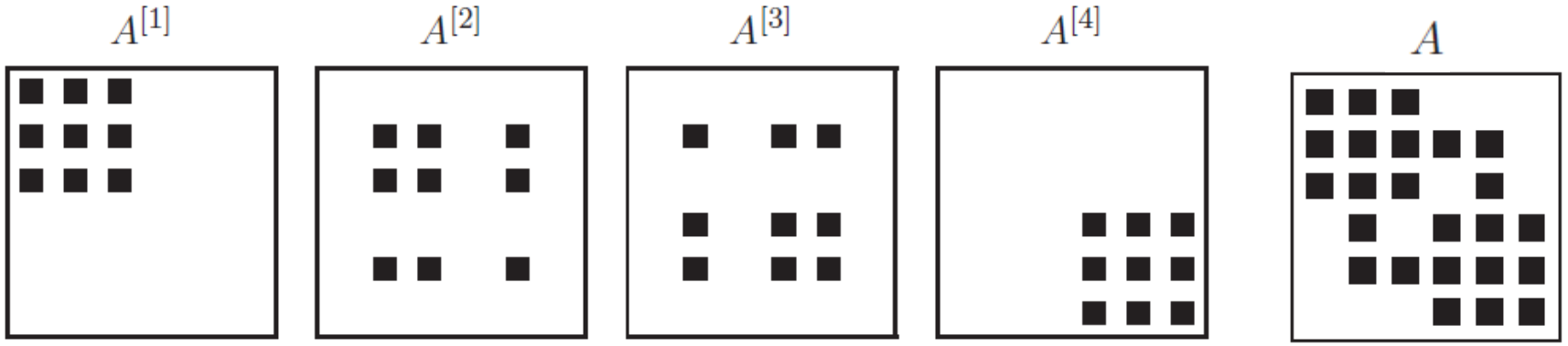
Element stiffness matrix for triangle with the nodes i, j, k

$$A_K = \begin{pmatrix} a_K(\phi_i,\phi_i) & a_K(\phi_i,\phi_j) & a_K(\phi_i,\phi_k) \\ a_K(\phi_j,\phi_i) & a_K(\phi_j,\phi_j) & a_K(\phi_j,\phi_k) \\ a_K(\phi_k,\phi_i) & a_K(\phi_k,\phi_j) & a_K(\phi_k,\phi_k) \end{pmatrix}$$

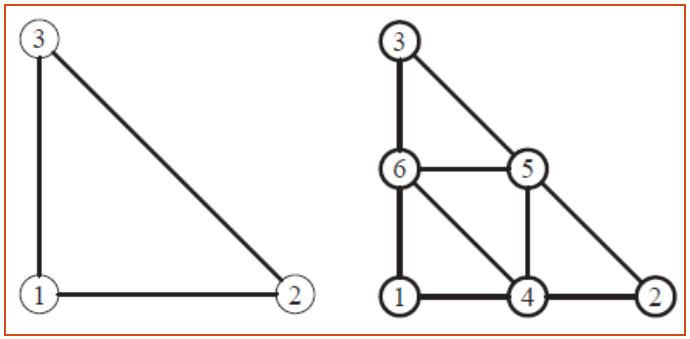




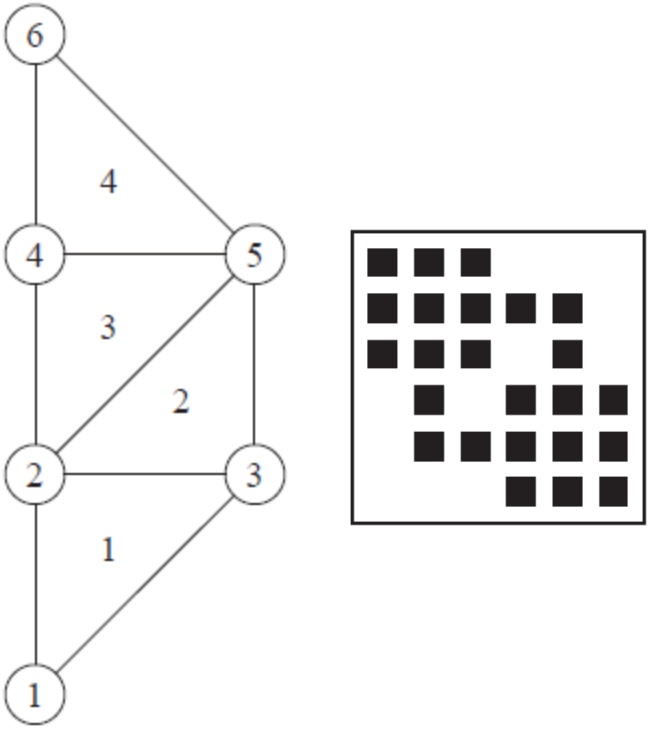
Assembly process $A = \sum_{e=1}^{nel} A^{[e]} \qquad A^{[e]} = P_e A_{K_e} P_e^T$



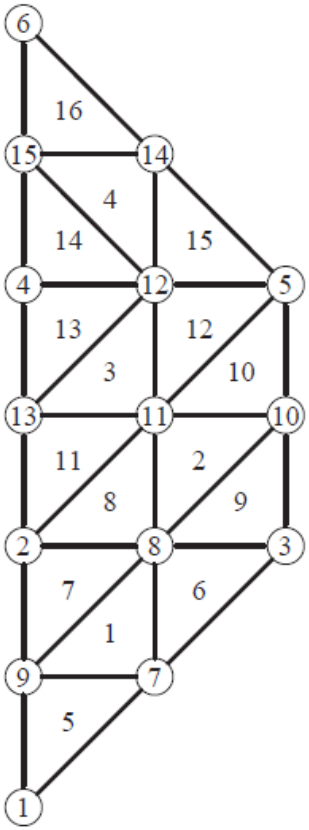
Mesh refinement in finite element method

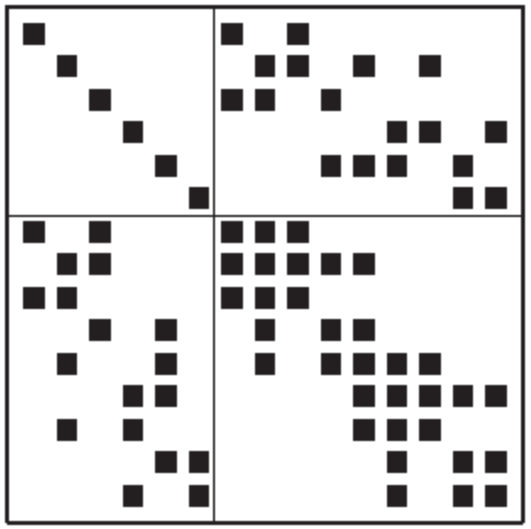


Original mesh and assembled matrix



Refined mesh and assembled matrix





Direct and iterative methods for sparse linear systems

Direct methods vs Iterative methods

Gaussian elimination

Gaussian elimination with partial pivoting, sparse version

Direct methods vs Iterative methods

- Exact solution in finite number of steps
- Do not preserve structure of the matrix
- Not suitable for large sparse matrices
  - **Modification: sparse direct methods** (try to preserve matrix sparsity)
- "Black-box" performance is typical

- Approximate solution in finite number of iterations
- Preserve structure of the matrix
- Suitable for large sparse matrices
- Do not always have "black-box" performance

Examples of direct and iterative methods

- Direct methods
  - Gaussian elimination with partial pivoting (based on LU-factorization)
  - Cholessky factorization
- Iterative methods
  - Classic iterative methods
    - Jacobi, Gauss-Seidel, SOR (Successive Over Relaxation), SSOR (Symmetric successive over relaxation), etc.
  - Projection methods
    - **One-dimensional projection methods**
    - Steepest Descent, Minimal Residual Iteration, Residual Norm Steepest Descent
    - **Multi-dimensional projection methods: Krylov subspace methods**
    - FOM (Full Orthogonalization Method), GMRES (Generalized Minimal Residual), CG (Conjugate Gradient), BiCG (Biconjugate gradient), etc.
    - Preconditioned methods

Review of direct methods: Gaussian elimination

- LU-factorization, if it exists for a given matrix, is not unique and is defined up to at least n degrees of freedom
- Steps of direct method (Gaussian elimination)

1. Factorize A

$$Ax = b, \quad A = LU$$

$$LUx = b, \quad Ux = y$$

$$Ly = b$$

2. Solve the system by forward substitution

$$Ly = b, \quad y = L^{-1}b$$

3. Solve the system by backward substitution

$$Ux = y, \quad x = U^{-1}y = U^{-1}L^{-1}y$$

Gaussian elimination with partial pivoting: direct sparse method

- PLU-factorization is possible for any matrix: A=PLU, where P is permutation matrix
- Steps of direct sparse method (Gaussian elimination with partial pivoting)

1. Preordering (find P)
2. Symbolic factorization PA=LU
3. Numerical factorization
4. Forward and backward substitution

$$Ax = b, \quad PA = LU, \quad A = P^{-1}LU$$

$$P^{-1}LUx = b, \quad Ly = z, \quad Ux = y$$

$$P^{-1}z = b \Rightarrow z = Pb$$

$$Ly = z \Rightarrow y = L^{-1}z$$

$$Ux = y \Rightarrow x = U^{-1}y$$

Classical iterative methods for linear systems

Formulation of classical iterative method

Convergence of iterative methods

Formulation of classical iterative method

Idea of an iterative method to solve the system $Ax = b$:

1. Take *initial guess* $x^{(0)}$

2. Apply iterative process until convergence

$x^{(k+1)} = G_k x^{(k)} + g_k$, where

$x^{(k)}$ is *approximate solution* at k-th iteration

G_k is iteration (transition) matrix, g_k is iteration vector

At every iteration $x^{(k)}$ is improved by the modification of one or several components of it, until convergence is reached.

This is called a *relaxation step*: $\left\| x^{(k+1)} - x^{(k)} \right\| < \varepsilon$

Also the goal is to make the norm of the residual vector $\|r_k\| = \left\| b - Ax^{(k)} \right\|$ smaller at every iteration.

Formulation of classical iterative method

Iterative process with splitting matrix Q_k

$$x^{(k+1)} = x^{(k)} + Q_k^{-1}(b - Ax^{(k)}) = x^{(k)} + Q_k^{-1} r_k$$

Stationary process: iterations do not depend on k: $x^{(k+1)} = Gx^{(k)} + g$

Relation between transition matrix G_k and splitting matrix Q_k :

$$x^{(k+1)} = x^{(k)} + Q_k^{-1}(b - Ax^{(k)}) = x^{(k)} + Q_k^{-1} b - Q_k^{-1} Ax^{(k)} =$$

$$= Q_k^{-1} b + (I - Q_k^{-1} A)x^{(k)} \Rightarrow G = I - Q^{-1} A, \quad g = Q^{-1} b$$

Iterations $x^{(0)}$, $x^{(1)}$, $x^{(2)}$,....

$$x^{(1)} = Gx^{(0)} + g,$$

$$x^{(2)} = Gx^{(1)} + g,$$

$$x^{(3)} = Gx^{(2)} + g,$$

...

$$x^{(k+1)} = Gx^{(k)} + g.$$

Error at k-th iteration

$$\varepsilon^{(k)} = x^{(k)} - x^{(\infty)}$$

$$\varepsilon^{(k)} = G^k \varepsilon^{(0)}$$

$\varepsilon^{(0)} = x^{(0)} - x^{(\infty)}$ is initial error

$x^{(\infty)}$ is exact solution

Convergence of iterative methods

For an iterative process $x^{(k+1)} = Gx^{(k)} + g$ it is required that iterations converge $x^{(k)} \to x^*$, where x^* is the solution of the system $Ax = b$

Spectral radius of transition matrix G: $\rho(G) = \max\limits_{i=1,\ldots,n} \left|\lambda_i(G)\right|$, where λ_i is the eigenvalue of G

Sufficient condition of convergence :

if $\|G\| < 1$, then iterations $x^{(k)}$ converge and matrix $I - G$ is nonsingular

Necessary and sufficient condition of convergence :

$\rho(G) < 1 \Leftrightarrow$ iterations $x^{(k)}$ converge for any initial guess $x^{(0)}$