The conjugate gradient method

Christian Kuschel and Florian Schornbaum
December 2015
Chair for System Simulation

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

What is this class about?

Central topic today:

Solve $\boldsymbol{Ax} = \boldsymbol{b}$ with conjugate gradient method (CG)

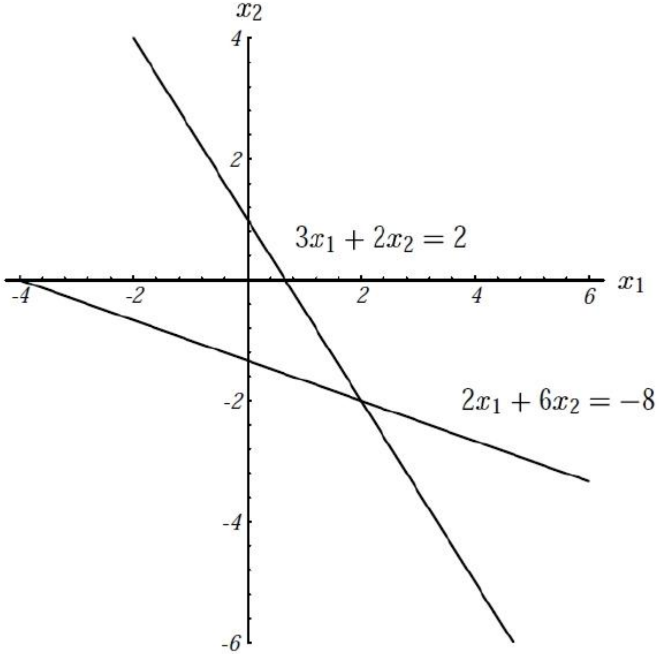
- Known solvers:
  - Gaussian elimination (direct)
  - Jacobi / Gauß-Seidel method (iterative)

What can CG do better?

Let’s have a look at the problem first

Geometric interpretation

$$\begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix} \cdot \boldsymbol{x} = \begin{pmatrix} 2 \\ -8 \end{pmatrix}$$



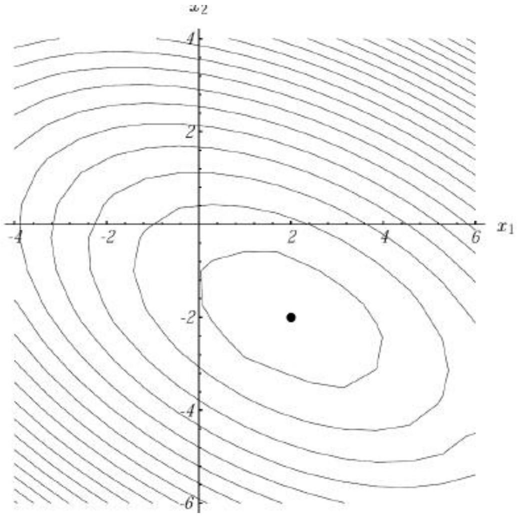
Solution to an optimisation problem

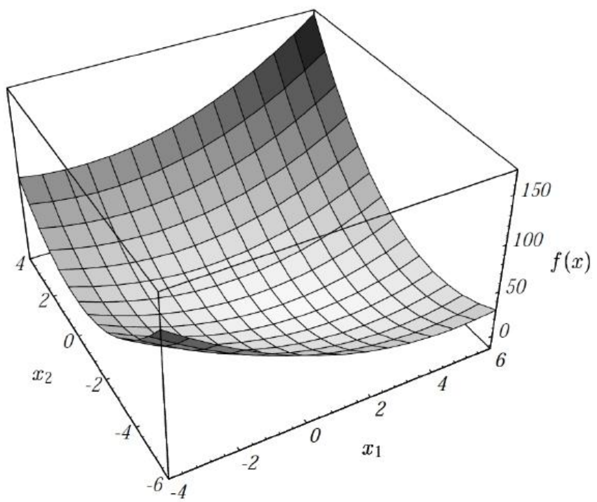
Consider following quadratic optimisation problem:

$$\min f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\boldsymbol{x} + \boldsymbol{c}$$

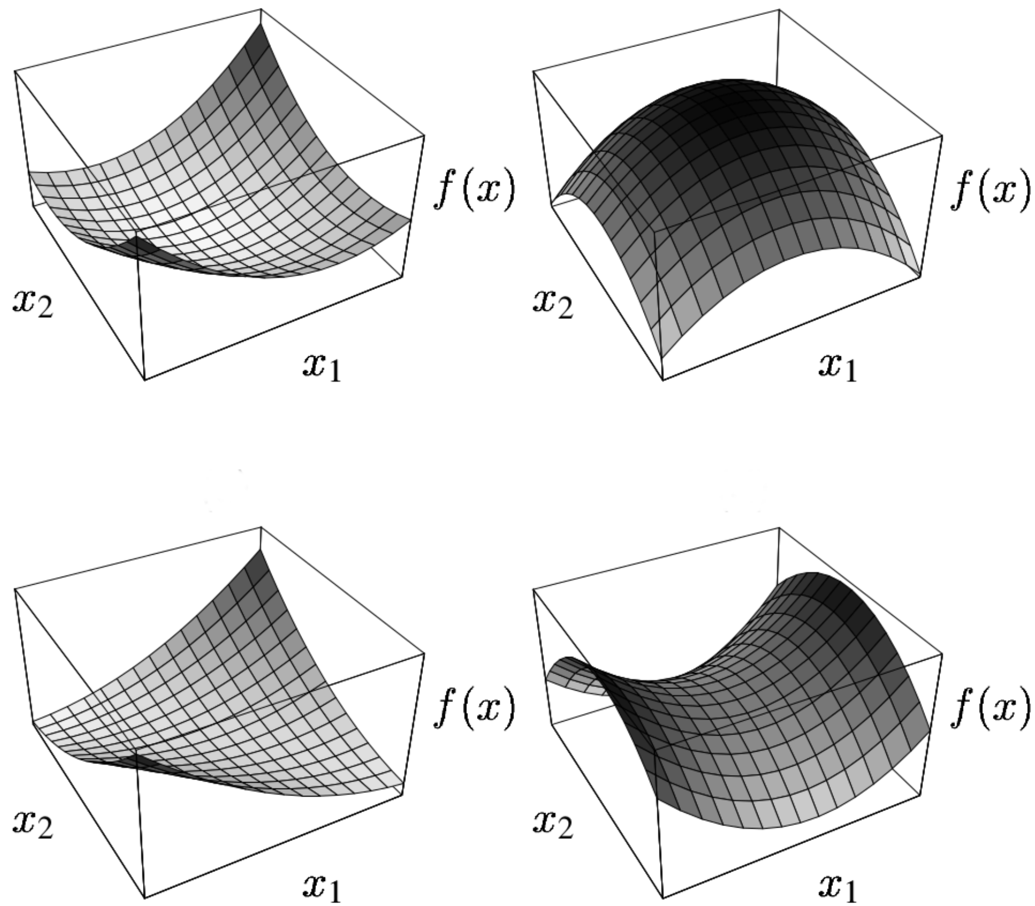
Assumptions: $\boldsymbol{A}$ is symmetric positive definite

- Symmetry: $\boldsymbol{A}^T = \boldsymbol{A}$
- Positive definiteness: $\boldsymbol{v}^T \boldsymbol{A}\boldsymbol{v} > 0 \quad \forall \boldsymbol{v} \neq \boldsymbol{0}$





Definiteness of the matrix



Solving the optimisation problem
Or: Motivating assumptions for A

- Let $\boldsymbol{x}^*$ be an optimum
- It holds that $\nabla f(\boldsymbol{x}^*) = \mathbf{0}$:

$$\begin{aligned} f(\boldsymbol{x}) &= \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\boldsymbol{x} + \boldsymbol{c} \\ \nabla f(\boldsymbol{x}^*) &= \frac{1}{2}\boldsymbol{A}^T\boldsymbol{x}^* + \frac{1}{2}\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b} \overset{!}{=} \mathbf{0} \end{aligned}$$

- Validate optimum with Taylor series expansion around $\boldsymbol{x}^*$:

$$f(\boldsymbol{x}^* + \boldsymbol{d}) = f(\boldsymbol{x}^*) + \underbrace{\nabla f(\boldsymbol{x}^*)^T\boldsymbol{d}}_{=0} + \underbrace{\frac{1}{2}\boldsymbol{d}^T\nabla^2 f(\boldsymbol{\xi})\boldsymbol{d}}_{>0 \text{ f. } \boldsymbol{d}\neq\mathbf{0}} > f(\boldsymbol{x}^*)$$

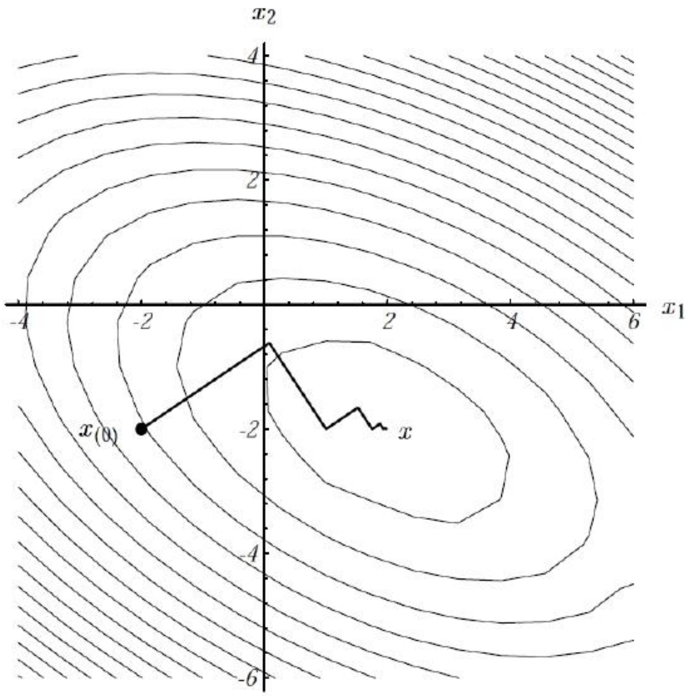
Properties of the optimisation problem

- $\boldsymbol{A}$ is pos. definite $\rightarrow$ f is strictly convex $\rightarrow$ unique minimum
- If f is smooth, so minimisation is global convergent

Method of steepest descent

Fundamental idea

- Choose (arbitrary) initial solution
- Follow function in descent direction
- Obtain updated solution
- Repeat until optimum found



Line search: Search direction

Considerations

- Descent in a direction which decreases the function value
- Any descent direction works
- We aim for a maximal decrease $\rightarrow$ steepest descent

Let $\boldsymbol{d}_i$ be a descent direction, the maximum decrease is obtained if

$$\boldsymbol{d}_i = \min_{\boldsymbol{d}_i \neq \boldsymbol{0}} \frac{\boldsymbol{d}_i^T \nabla f(\boldsymbol{x}_i)}{||\boldsymbol{d}_i|| \cdot ||\nabla f(\boldsymbol{x}_i)||}$$

Minimum is obviously obtained if $\boldsymbol{d}_i = -\nabla f(\boldsymbol{x}_i)$.

Line search: Step size

Considerations

- Following a descent direction takes us off the function's slope
- How far should we follow the descent direction?
- We want to enforce a decrease of the function value
- We want the algorithm to terminate eventually

Let $\boldsymbol{d}_i := -\nabla f(\boldsymbol{x}_i)$

Compute updated approximation

- Find a new point for which $f(\boldsymbol{x}_{i+1}) < f(\boldsymbol{x}_i)$ holds
- Choose α_i: $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i \cdot \boldsymbol{d}_i$ such that $f(\boldsymbol{x}_{i+1}) < f(\boldsymbol{x}_i)$

Naïve line search

Algorithm

1. Set $\alpha_i = 10^{-10}$ (i.e. small)
2. While $f(\boldsymbol{x}_i + \alpha_i \cdot \boldsymbol{d}_i) > f(\boldsymbol{x}_i)$: $\alpha_i \leftarrow \alpha_i + \alpha_i$
3. Set $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i \cdot \boldsymbol{d}_i$

Problem

- While-loop is likely to terminate after first iteration
- Only small decrease in f are achieved $\rightarrow$ Slow (or no!) convergence

Naïve line search

Algorithm

1. Set $\alpha_i = 10^{-10}$ (i.e. small)
2. While $f(\boldsymbol{x}_i + \alpha_i \cdot \boldsymbol{d}_i) > f(\boldsymbol{x}_i)$: $\alpha_i \leftarrow \alpha_i + \alpha_i$
3. Set $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i \cdot \boldsymbol{d}_i$

Problem

- While-loop is likely to terminate after first iteration
- Only small decrease in f are achieved $\rightarrow$ Slow (or no!) convergence

Improvement: Backtracking

1. Set $\alpha_i = 0.5$
2. While $f(\boldsymbol{x}_i + \alpha_i \cdot \boldsymbol{d}_i) > f(\boldsymbol{x}_i)$: $\alpha_i \leftarrow \alpha_i \cdot \alpha_i$
3. Set $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i \cdot \boldsymbol{d}_i$

A sufficient decrease in f is guaranteed $\rightarrow$ convergent

Optimal line search

Compute α_i which minimises $f(\boldsymbol{x}_{i+1}) = f\left(\boldsymbol{x}_i + \alpha_i \cdot \boldsymbol{d}_i\right)$
Equate first derivative to zero:

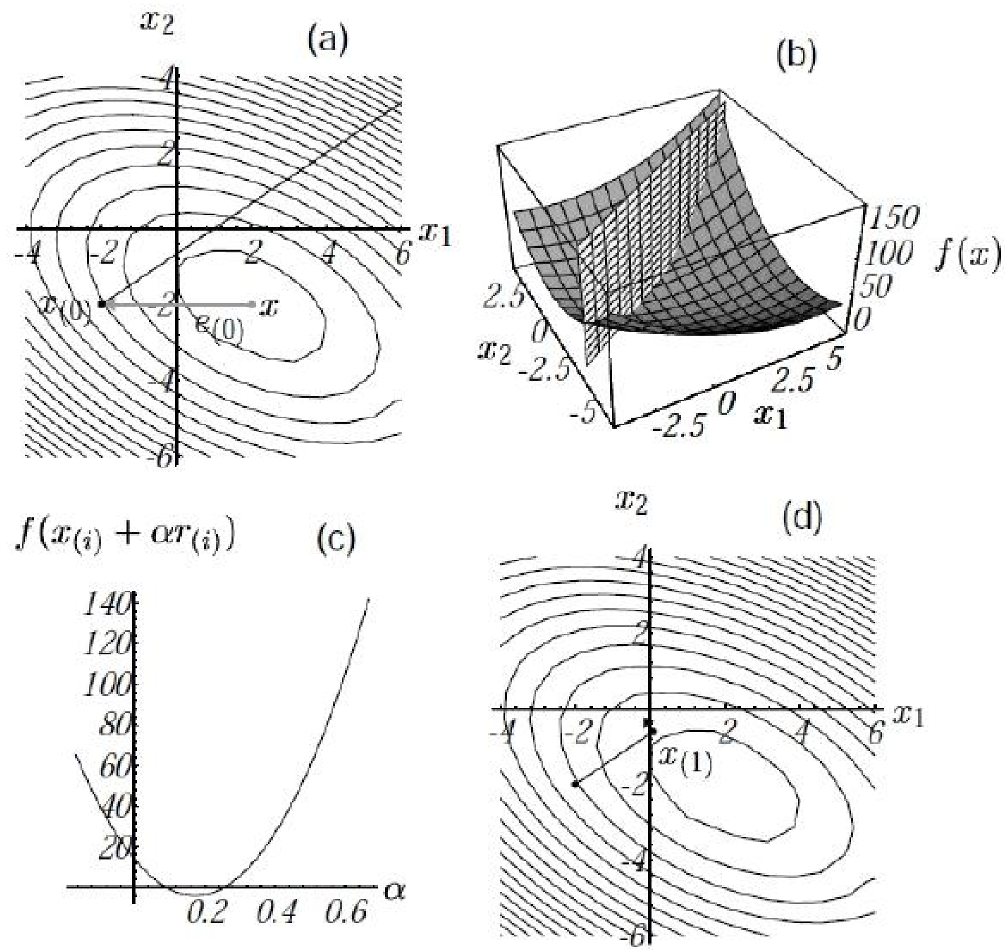
$$\begin{aligned}
\frac{\partial}{\partial \alpha_i} f(\boldsymbol{x}_{i+1}) &= \nabla f(\boldsymbol{x}_{i+1})^T \cdot \frac{\partial}{\partial \alpha_i} \boldsymbol{x}_{i+1} \\
&= \left(\boldsymbol{A}\boldsymbol{x}_{i+1} - \boldsymbol{b}\right)^T \cdot \boldsymbol{d}_i \\
&= \left(\boldsymbol{A}\boldsymbol{x}_i + \alpha_i \boldsymbol{A}\boldsymbol{d}_i - \boldsymbol{b}\right)^T \cdot \boldsymbol{d}_i \\
&= \left(\alpha_i \boldsymbol{A}\boldsymbol{d}_i - \boldsymbol{r}_i\right)^T \cdot \boldsymbol{d}_i \stackrel{!}{=} 0
\end{aligned}$$

So optimal α_i:

$$\alpha_i = \frac{\boldsymbol{r}_i^T \boldsymbol{d}_i}{\boldsymbol{d}_i^T \boldsymbol{A}\boldsymbol{d}_i}$$

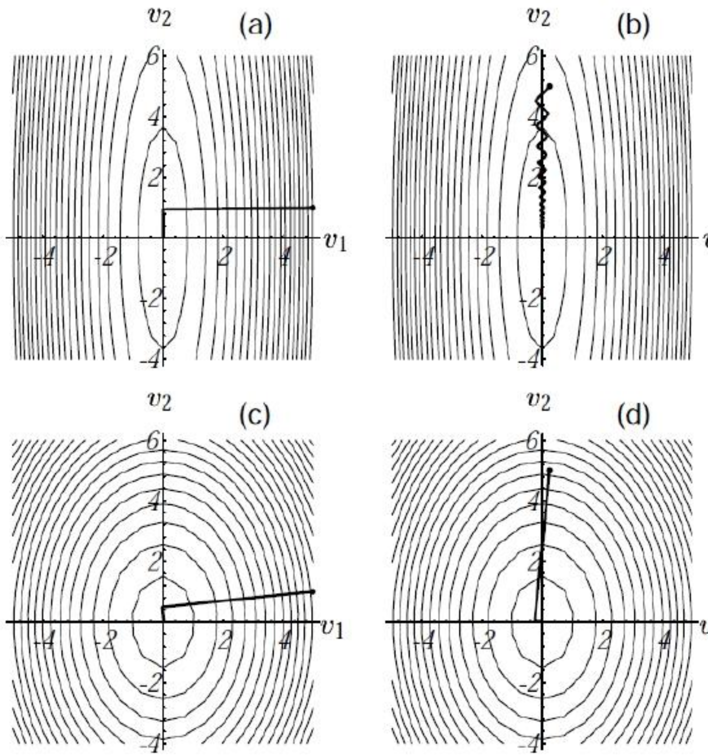
with residual $\boldsymbol{r}_i := \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_i = -\nabla f(\boldsymbol{x}_i) = \boldsymbol{d}_i$.

Geometric interpretation



Convergence and properties

- Each iteration reduces the error $\boldsymbol{e}_i = \boldsymbol{x}_i - \boldsymbol{x}^*$ w.r.t. energy norm
- Energy norm: $||\boldsymbol{e}_i||_A^2 = \boldsymbol{e}_i^T \boldsymbol{A} \boldsymbol{e}_i$
- Upper bound estimate: $||\boldsymbol{e}_{i+1}||_A \leq \frac{\kappa-1}{\kappa+1} ||\boldsymbol{e}_i||_A$



Improvement thoughts

- Perform descent not in similar directions
- Descent in N orthogonal directions $\boldsymbol{d}_i$ $(1 \leq i \leq N)$
- Compute minimum for each $\boldsymbol{d}_i$: error is orthogonal to descent direction
- Compute $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i \cdot \boldsymbol{d}_i$ with property:

$$\boldsymbol{d}_i^T \boldsymbol{e}_{i+1} = 0 \Leftrightarrow \boldsymbol{d}_i^T (\boldsymbol{e}_i + \alpha_i \boldsymbol{d}_i) = 0$$

$$\Rightarrow \alpha_i = -\frac{\boldsymbol{d}_i^T \boldsymbol{e}_i}{\boldsymbol{d}_i^T \boldsymbol{d}_i}$$

Improvement thoughts

- Perform descent not in similar directions
- Descent in N orthogonal directions $\boldsymbol{d}_i$ $(1 \leq i \leq N)$
- Compute minimum for each $\boldsymbol{d}_i$: error is orthogonal to descent direction
- Compute $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i \cdot \boldsymbol{d}_i$ with property:

$$\boldsymbol{d}_i^T \boldsymbol{e}_{i+1} = 0 \Leftrightarrow \boldsymbol{d}_i^T (\boldsymbol{e}_i + \alpha_i \boldsymbol{d}_i) = 0$$

$$\Rightarrow \alpha_i = -\frac{\boldsymbol{d}_i^T \boldsymbol{e}_i}{\boldsymbol{d}_i^T \boldsymbol{d}_i}$$

Problem: We do not know $\boldsymbol{e}_i = \boldsymbol{x}_i - \boldsymbol{x}^*$

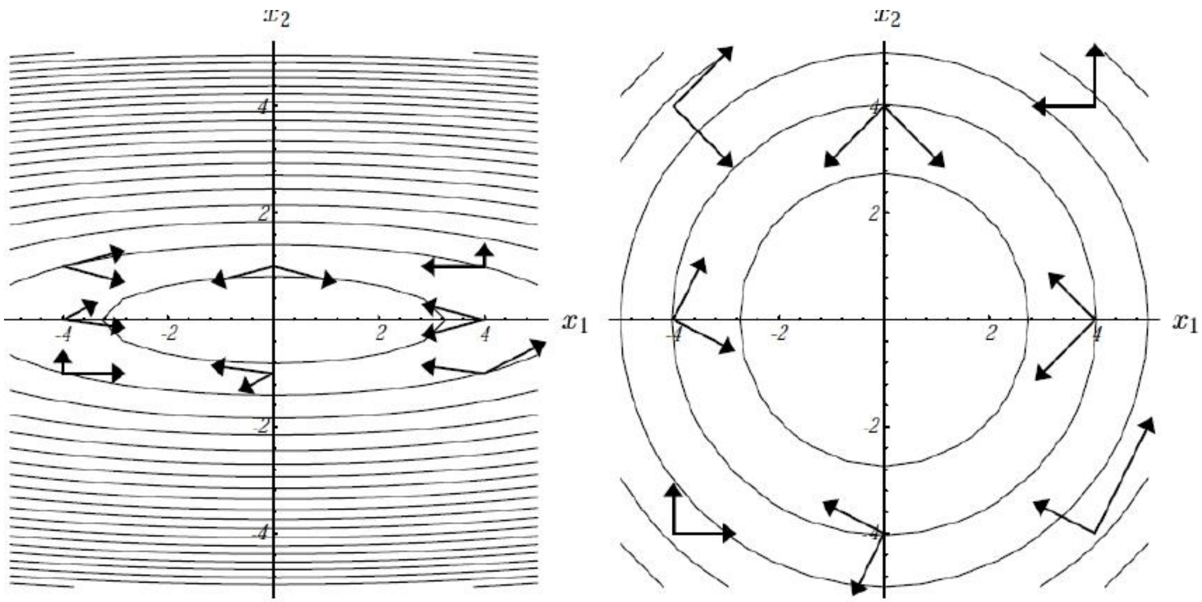
Let's start over: Line search

$$
\begin{aligned}
\frac{\partial}{\partial \alpha_i} f(\boldsymbol{x}_{i+1}) &= \nabla f(\boldsymbol{x}_{i+1})^T \cdot \frac{\partial}{\partial \alpha_i} \boldsymbol{x}_{i+1} \\
&= (\boldsymbol{A}\boldsymbol{x}_{i+1} - \boldsymbol{b})^T \cdot \boldsymbol{d}_i \\
&= (\boldsymbol{A}\boldsymbol{x}_{i+1} - \boldsymbol{A}\boldsymbol{x}^*)^T \cdot \boldsymbol{d}_i \\
&= \boldsymbol{e}_{i+1}^T \boldsymbol{A}\boldsymbol{d}_i \stackrel{!}{=} 0
\end{aligned}
$$

$\boldsymbol{e}_{i+1}$ and $\boldsymbol{d}_i$ are A-conjugate (like in optimal line search before)

Conjugate directions

- A-conjugacy: $\boldsymbol{u}^T \boldsymbol{A} \boldsymbol{v} = 0$
- "$\boldsymbol{u}$ and $\boldsymbol{v}$ are orthogonal with respect to $\boldsymbol{A}$"



Select α_i such that search directions are A-conjugate

Use $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i \cdot \boldsymbol{d}_i$ to express $\boldsymbol{e}_{i+1}$:

$$\begin{aligned} \boldsymbol{e}_{i+1} &= \boldsymbol{x}_i + \alpha_i \cdot \boldsymbol{d}_i - \boldsymbol{x}^* \\ &= \boldsymbol{e}_i + \alpha_i \cdot \boldsymbol{d}_i \end{aligned}$$

Plug this in equation from previous slide:

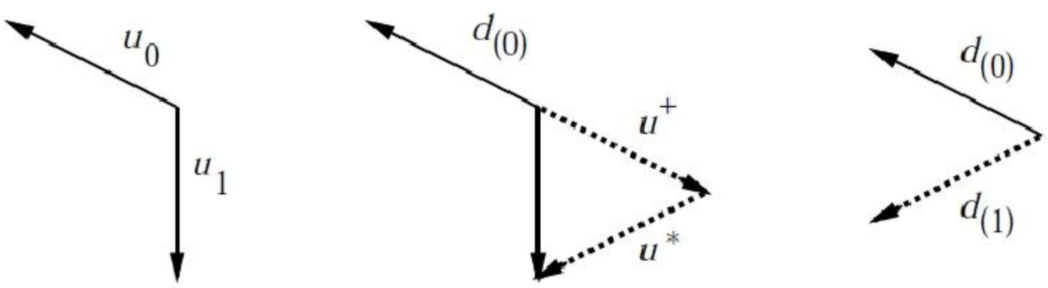
$$\begin{aligned} \boldsymbol{e}_{i+1}^T \boldsymbol{A}\boldsymbol{d}_i &= (\boldsymbol{e}_i + \alpha_i \cdot \boldsymbol{d}_i)^T \boldsymbol{A}\boldsymbol{d}_i \\ &= \boldsymbol{e}_i^T \boldsymbol{A}\boldsymbol{d}_i + \alpha_i \boldsymbol{d}_i^T \boldsymbol{A}\boldsymbol{d}_i \stackrel{!}{=} 0 \end{aligned}$$

$$\Rightarrow \alpha_i = -\frac{\boldsymbol{e}_i^T \boldsymbol{A}\boldsymbol{d}_i}{\boldsymbol{d}_i^T \boldsymbol{A}\boldsymbol{d}_i} = \frac{\boldsymbol{r}_i^T \boldsymbol{d}_i}{\boldsymbol{d}_i^T \boldsymbol{A}\boldsymbol{d}_i}$$

Optimal α_i for A-conjugate search directions is determined without $\boldsymbol{x}^*$

Computing conjugate directions

- Task: Find N search directions s.t. $\boldsymbol{d}_i^T \boldsymbol{A} \boldsymbol{d}_j = 0, \forall i \neq j$
- Conjugate Gram-Schmidt process:



Choose linear independent vectors and construct A-conjugate directions

Conjugate Gram-Schmidt process in formulas

Choose N linear independent vectors $\boldsymbol{u}_i$

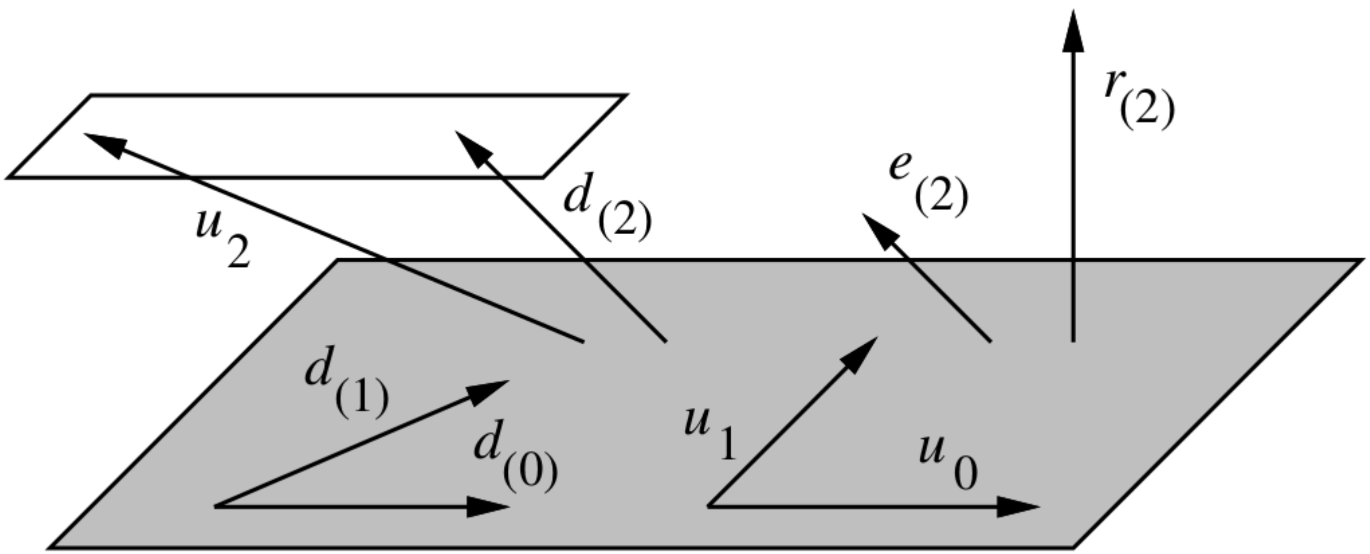
$$\boldsymbol{d}_i := \boldsymbol{u}_i + \sum_{j=0, i>j}^{i-1} \beta_{ij} \boldsymbol{d}_j$$

$$\begin{aligned}
\boldsymbol{d}_i^T \boldsymbol{A} \boldsymbol{d}_k &= \boldsymbol{u}_i^T \boldsymbol{A} \boldsymbol{d}_k + \sum_{j=0}^{i-1} \beta_{ij} \boldsymbol{d}_j^T \boldsymbol{A} \boldsymbol{d}_k \\
0 &= \boldsymbol{u}_i^T \boldsymbol{A} \boldsymbol{d}_k + \beta_{ik} \boldsymbol{d}_k^T \boldsymbol{A} \boldsymbol{d}_k \\
&\Rightarrow \beta_{ik} = -\frac{\boldsymbol{u}_i^T \boldsymbol{A} \boldsymbol{d}_k}{\boldsymbol{d}_k^T \boldsymbol{A} \boldsymbol{d}_k}
\end{aligned}$$

Issues

- Complexity of $\mathcal{O}(N^3)$
- Each search direction must be stored

Geometric interpretation of what will be discussed



Error/residual are either not contained in span$\{\boldsymbol{d}_0, \ldots, \boldsymbol{d}_{i-1}\}$ or 0

Orthogonality of error and residual

- The residual $\boldsymbol{r}_{i+1}$ is orthogonal on $\mathcal{D}_i$
- The error $\boldsymbol{e}_{i+1}$ is A-conjugate on $\mathcal{D}_i$

$$-\boldsymbol{d}_i^T \boldsymbol{A} \boldsymbol{e}_j = -\sum_{j=i}^{N-1} \delta_j \boldsymbol{d}_i^T \boldsymbol{A} \boldsymbol{d}_j \quad \Leftrightarrow \quad 0 = \boldsymbol{d}_i^T \boldsymbol{r}_j, i < j$$

$$\boldsymbol{d}_i^T \boldsymbol{r}_j = \boldsymbol{u}_i^T \boldsymbol{r}_j + \sum_{k=0}^{i-1} \beta_{ik} \boldsymbol{d}_k^T \boldsymbol{r}_j \quad \Leftrightarrow \quad 0 = \boldsymbol{u}_i^T \boldsymbol{r}_j, i < j$$

Conjugate gradient method idea

Idea:

- Use conjugate directions
- Use residual to compute new search directions: $\boldsymbol{u}_i = \boldsymbol{r}_i$

Conjugate Gram-Schmitt process revisited:

$$\beta_{ik} = -\frac{\boldsymbol{u}_i^T \boldsymbol{A}\boldsymbol{d}_k}{\boldsymbol{d}_k^T \boldsymbol{A}\boldsymbol{d}_k} = -\frac{\boldsymbol{r}_i^T \boldsymbol{A}\boldsymbol{d}_k}{\boldsymbol{d}_k^T \boldsymbol{A}\boldsymbol{d}_k}$$

$$\begin{aligned} \boldsymbol{r}_i^T \boldsymbol{r}_{k+1} &= \boldsymbol{r}_i^T \boldsymbol{r}_k - \alpha_k \boldsymbol{r}_i^T \boldsymbol{A}\boldsymbol{d}_k \\ \alpha_k \boldsymbol{r}_i^T \boldsymbol{A}\boldsymbol{d}_k &= \boldsymbol{r}_i^T \boldsymbol{r}_k - \boldsymbol{r}_i^T \boldsymbol{r}_{k+1} \\ &= 0 \text{ for } i \neq k \wedge i \neq k+1 \end{aligned}$$

$$\Rightarrow \beta_{ik} = \frac{\boldsymbol{r}_i^T \boldsymbol{r}_i}{\alpha_{i-1} \boldsymbol{d}_{i-1}^T \boldsymbol{A}\boldsymbol{d}_{i-1}} = \frac{\boldsymbol{r}_i^T \boldsymbol{r}_i}{\boldsymbol{r}_{i-1}^T \boldsymbol{r}_{i-1}} \quad \text{for } i = k+1$$

Redefine β_{ik} as β_i

Conjugate gradient method

Procedure

$$\boldsymbol{d}_0 = \boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$$

For $i = 1$ to N:

$$\begin{aligned}
\alpha_i &= \frac{\boldsymbol{r}_i^T \boldsymbol{r}_i}{\boldsymbol{d}_i^T \boldsymbol{A} \boldsymbol{d}_i} \\
\boldsymbol{x}_{i+1} &= \boldsymbol{x}_i + \alpha_i \boldsymbol{d}_i \\
\boldsymbol{r}_{i+1} &= \boldsymbol{r}_i - \alpha_i \boldsymbol{A} \boldsymbol{d}_i \\
\beta_{i+1} &= \frac{\boldsymbol{r}_{i+1}^T \boldsymbol{r}_{i+1}}{\boldsymbol{r}_i^T \boldsymbol{r}_i} \\
\boldsymbol{d}_{i+1} &= \boldsymbol{r}_{i+1} - \beta_{i+1} \boldsymbol{d}_i
\end{aligned}$$

Error analysis

Define error as linear combination of all (A-conjugate) search directions:

$$\boldsymbol{e}_0 := \sum_{i=0}^{N-1} \delta_i \boldsymbol{d}_i$$

Plug in A-conjugacy definition:

$$\boldsymbol{e}_0^T \boldsymbol{A} \boldsymbol{d}_j = \sum_{i=0}^{N-1} \delta_i \boldsymbol{d}_i^T \boldsymbol{A} \boldsymbol{d}_j = \delta_j \boldsymbol{d}_j^T \boldsymbol{A} \boldsymbol{d}_j$$

Solve for δ_j:

$$\delta_j = \frac{\boldsymbol{e}_0^T \boldsymbol{A} \boldsymbol{d}_j}{\boldsymbol{d}_j^T \boldsymbol{A} \boldsymbol{d}_j} \overset{A\text{-conjugacy}}{=} \frac{\boldsymbol{e}_j^T \boldsymbol{A} \boldsymbol{d}_j}{\boldsymbol{d}_j^T \boldsymbol{A} \boldsymbol{d}_j} = -\alpha_j$$

$\Rightarrow$ Error is reduced component by component

Error representation

Let $\mathcal{D}_i = \text{span}\{\boldsymbol{d}_0, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_{i-1}\}$ (Krylov subspace)
The error in the energy norm is minimal:

$$||e_i||_A = \sum_{j=i}^{N-1} \sum_{k=i}^{N-1} \delta_j \delta_k \boldsymbol{d}_j^T \boldsymbol{A} \boldsymbol{d}_k = \sum_{j=i}^{N-1} \delta_j^2 \boldsymbol{d}_j^T \boldsymbol{A} \boldsymbol{d}_j$$

Properties of CG method

- Convergence:

$$||\boldsymbol{e}_i||_A \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i ||\boldsymbol{e}_0||_A$$

- Number of iterations:

$$\text{CG: } i \leq \left\lceil \frac{\sqrt{\kappa}}{2}\ln\left(\frac{2}{\varepsilon}\right)\right\rceil \qquad \text{SD: } i \leq \left\lceil \frac{\kappa}{2}\ln\left(\frac{1}{\varepsilon}\right)\right\rceil$$

- Memory complexity: $\mathcal{O}(m)$ (m number of non-zero entries in $\boldsymbol{A}$)
- Computational complexity:
  - CG: $\mathcal{O}(m\sqrt{\kappa})$
  - Steepest descent: $\mathcal{O}(m\kappa)$

Initialisation and stopping criteria

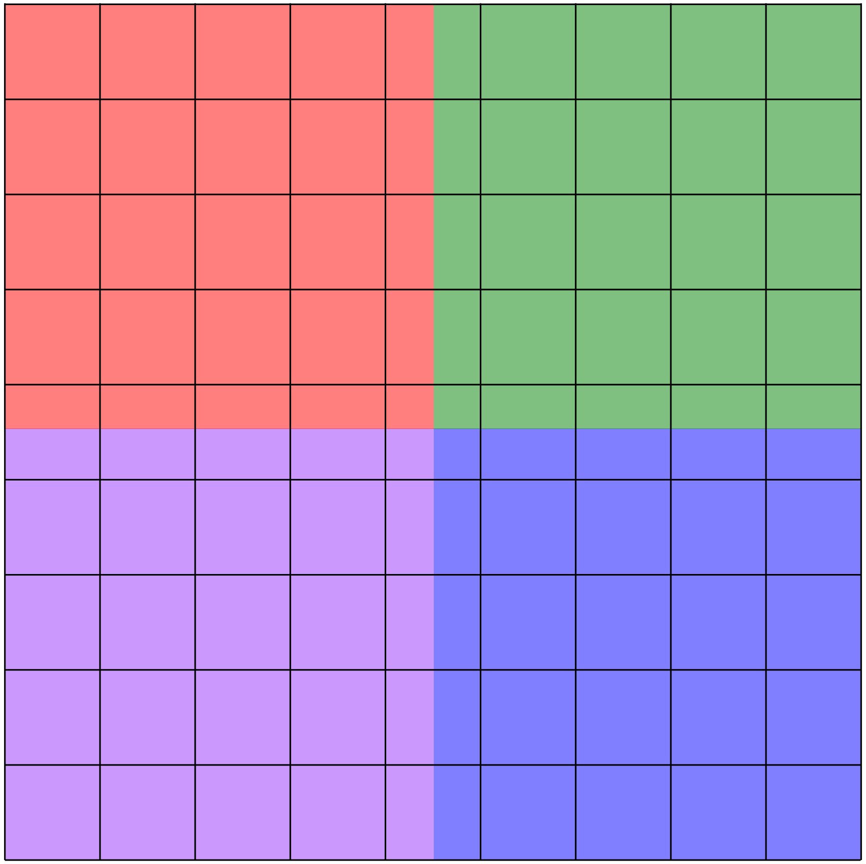
Initialisation

- If initial guess available, use it
- If no information available, start at arbitrary point (e.q. $\boldsymbol{x}_0 = \mathbf{0}$)
- If $\boldsymbol{A}$ is symmetric, positive definite, CG is globally convergent

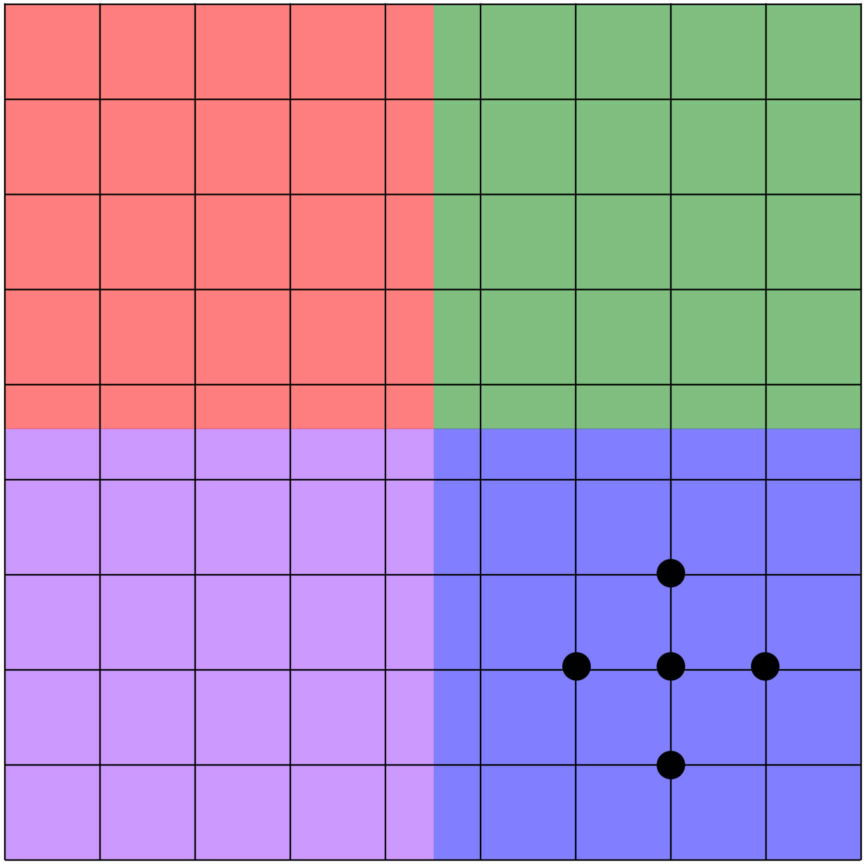
Stopping criteria

- After N iterations exact solution was computed
- Stop when $||\boldsymbol{r}_i|| \leq \varepsilon ||\boldsymbol{r}_0||$

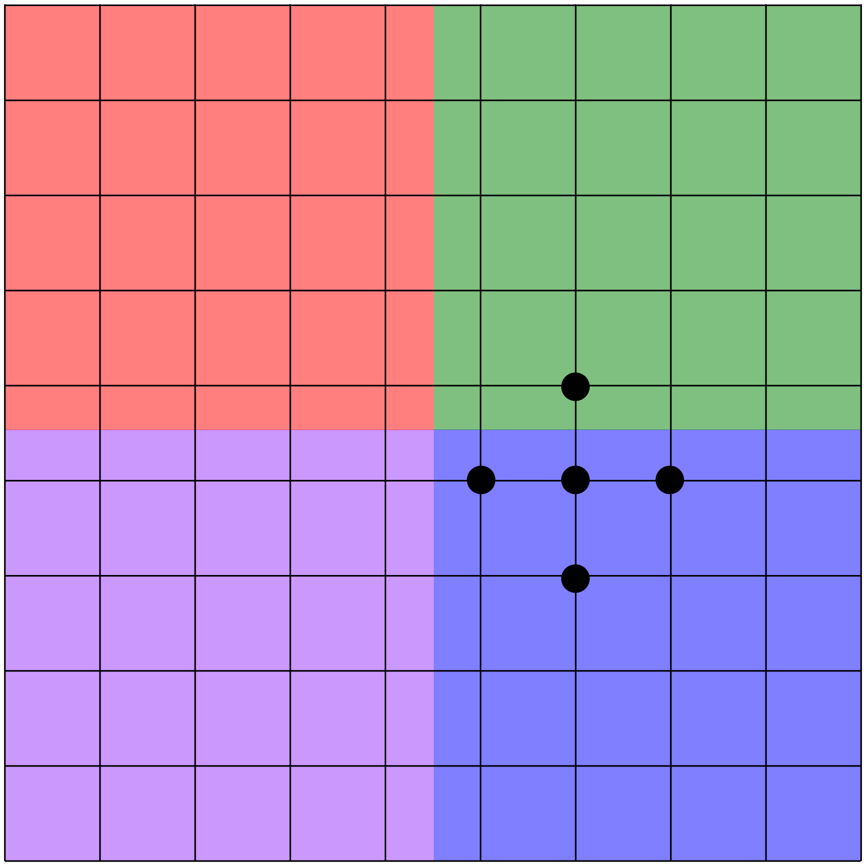
Parallelisation of CG method for 5-point stencil



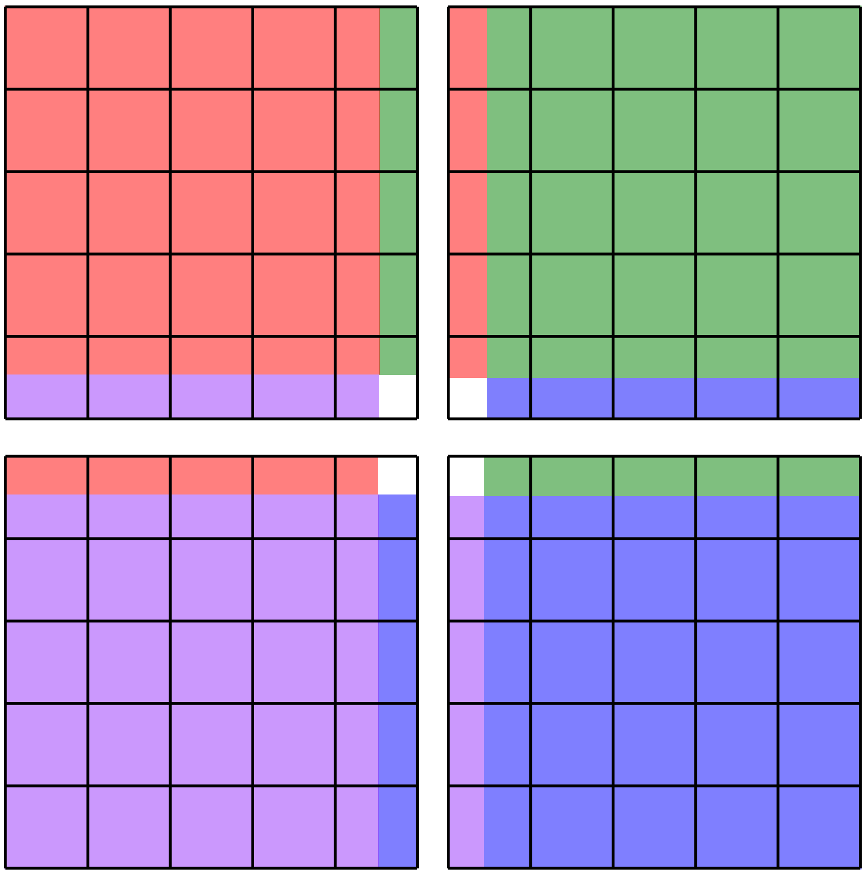
Parallelisation of CG method for 5-point stencil



Parallelisation of CG method for 5-point stencil



Ghost layer concept



Parallel implementation

- Each process needs to allocate additional memory for the ghost layer
- The ghost layers need to be updated each iteration
- Update is done by MPI send and receive of neighbouring processes
- Cartesian topology of MPI processes can be used

A glance at the CG algorithm

1. $\boldsymbol{d}_0 = \boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$
2. $\alpha_i = \frac{\boldsymbol{r}_i^T \boldsymbol{r}_i}{\boldsymbol{d}_i^T \boldsymbol{A}\boldsymbol{d}_i}$
3. $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i \boldsymbol{d}_i$
4. $\boldsymbol{r}_{i+1} = \boldsymbol{r}_i - \alpha_i \boldsymbol{A}\boldsymbol{d}_i$
5. $\beta_{i+1} = \frac{\boldsymbol{r}_{i+1}^T \boldsymbol{r}_{i+1}}{\boldsymbol{r}_i^T \boldsymbol{r}_i}$
6. $\boldsymbol{d}_{i+1} = \boldsymbol{r}_{i+1} - \beta_{i+1} \boldsymbol{d}_i$

Conclusion

- $\boldsymbol{A}$ (the stencil) is globally known
- $\boldsymbol{d}_i$ can be locally computed but requires ghost layers
- $\boldsymbol{r}_i$ can be locally computed
- α_i and β_i requires a (global) scalar product $\rightarrow$ reduction

Conclusion

- Interpreted solving a LSE as quadratic minimisation problem
- Illustrated steepest descent method and shortcomings
- Introduced A-conjugacy and conjugate Gram-Schmidt process
- Finally arrived at conjugate gradient method

- Reference and figures: J.R. Shewchuk: *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*

Conclusion

- Interpreted solving a LSE as quadratic minimisation problem
- Illustrated steepest descent method and shortcomings
- Introduced A-conjugacy and conjugate Gram-Schmidt process
- Finally arrived at conjugate gradient method

- Reference and figures: J.R. Shewchuk: *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*

Thank you very much for your attention!