ulm university universität ulm

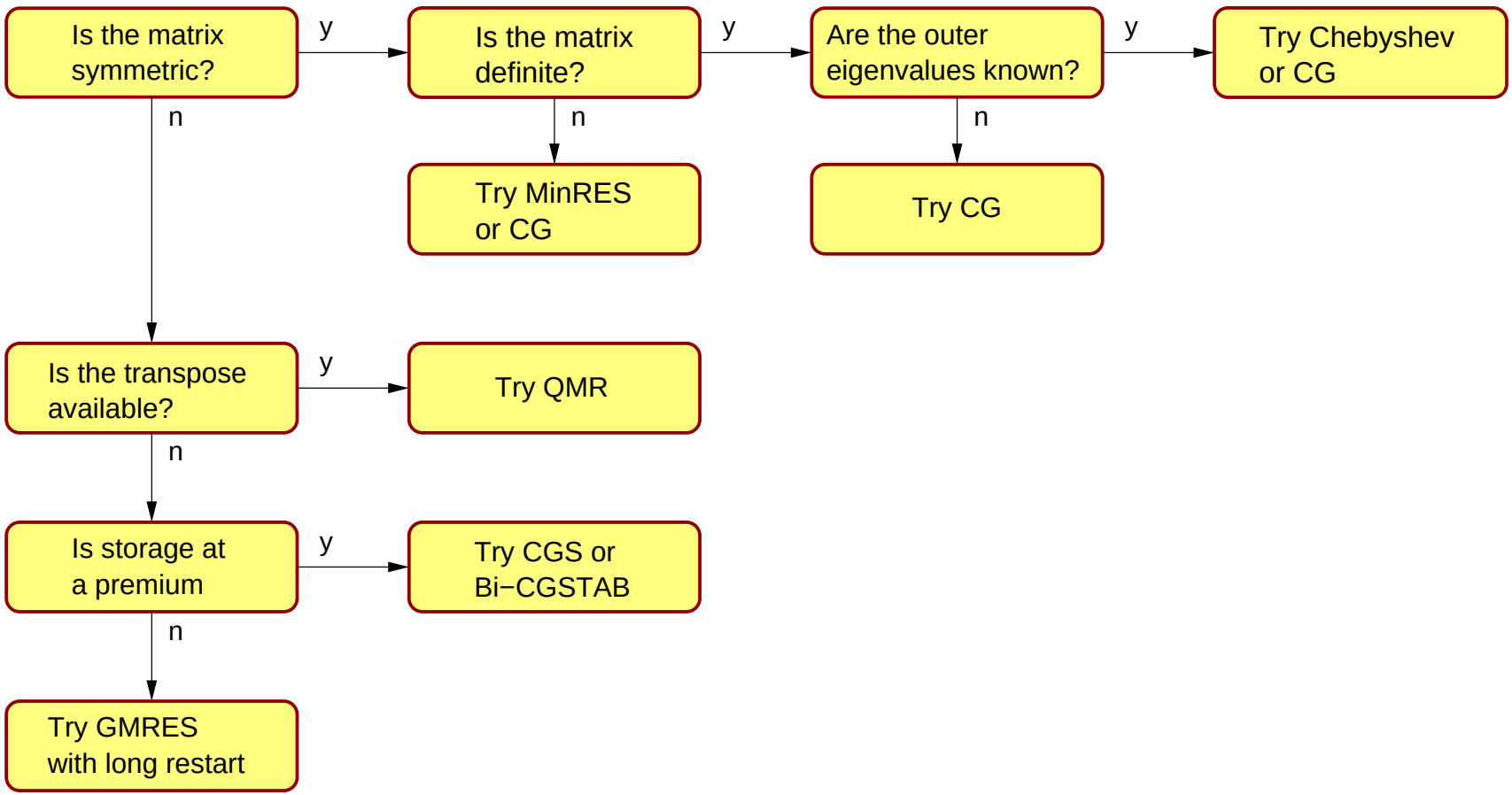
Scientific Computing

Parallele Algorithmen

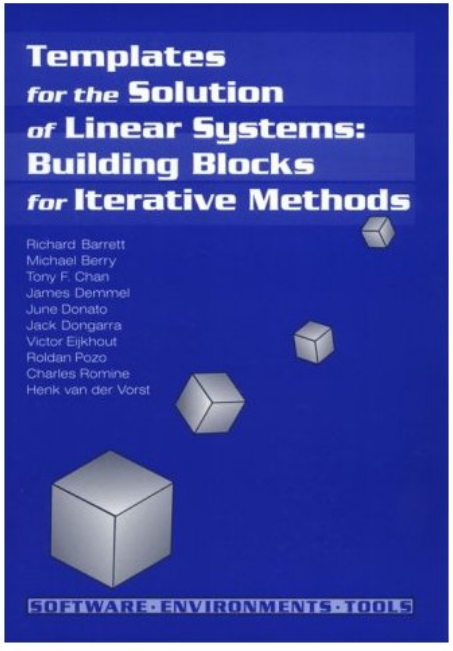
Prof. Dr. Stefan Funken, Prof. Dr. Alexander Keller,
Prof. Dr. Karsten Urban | 11. Januar 2007

Iterative Methods

Flowchart with suggestions for the selection of iterative methods.



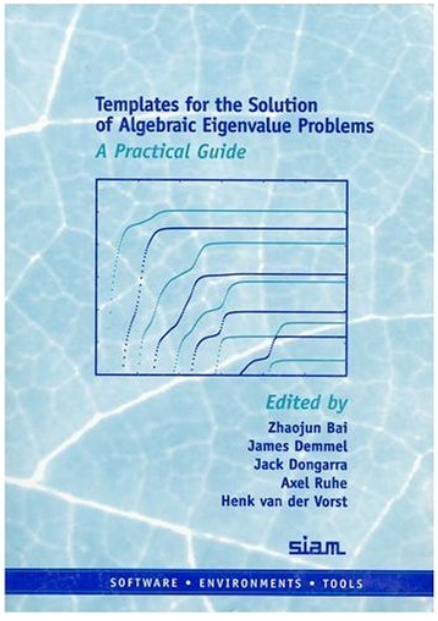
Iterative Methods



This book is also available in Postscript from

ftp.netlib.org/templates/templates.ps.

Iterative Methods for Algebraic Eigenvalue Problems



There exists also a similiar book for algebraic eigenvalue problems.

This is also available as online document at

www.cs.utk.edu/~dongarra/etemplates/book.html.

Preconditioner

Convergence rate of iterative methods depends on spectral properties of the coefficient matrix.

Example: CG-method

$$\|x - x^{(k)}\|_A \leq 2\rho^k \|x - x^{(k)}\|_A \quad \text{with } \rho^2 := \frac{\kappa_2(A) - 1}{\kappa_2(A) - 1}$$

and $\|x - x^{(k)}\|_A^2 := \langle x - x^{(k)}, A(x - x^{(k)})\rangle$.

Note: The number of iterations to reach a relative reduction of ϵ in the error is proportional to $\sqrt{\kappa_2}$.

Preconditioner

Convergence rate of iterative methods depends on spectral properties of the coefficient matrix.

Hence, one may attempt to transform the linear system into one that is equivalent in the sense that it has the same solution, but that has more favorable spectral properties.

A preconditioner is a matrix that effects such a transformation.

For instance, if a matrix W approximates the coefficient matrix A in some way, the transformed system

$$W^{-1}Ax = W^{-1}b$$

has the same solution as the original system $Ax = b$, but the spectral properties of its coefficient matrix $W^{-1}A$ may be more favorable.

Preconditioner

In devising a preconditioner, we are faced with a choice between

finding a matrix W that approximates A,

and for which solving a system is easier than solving one with A,

Preconditioner

In devising a preconditioner, we are faced with a choice between

finding a matrix W that approximates A,
and for which solving a system is easier than solving one with A,

or finding a matrix W that approximates A^{-1},
so that only multiplication by W is needed.

Preconditioner

In devising a preconditioner, we are faced with a choice between

finding a matrix W that approximates A,
and for which solving a system is easier than solving one with A,

or finding a matrix W that approximates A^{-1},
so that only multiplication by W is needed.

The majority of preconditioners falls in the first category.

Preconditioner

In devising a preconditioner, we are faced with a choice between

finding a matrix W that approximates A,
and for which solving a system is easier than solving one with A,

or finding a matrix W that approximates A^{-1},
so that only multiplication by W is needed.

The majority of preconditioners falls in the first category.

On parallel machines there is a further trade-off between the efficancy of a preconditioner in the classical sense, and its parallel efficiency.

Preconditioner

In devising a preconditioner, we are faced with a choice between

finding a matrix W that approximates A,
and for which solving a system is easier than solving one with A,

or finding a matrix W that approximates A^{-1},
so that only multiplication by W is needed.

The majority of preconditioners falls in the first category.

On parallel machines there is a further trade-off between the efficancy of a preconditioner in the classical sense, and its parallel efficiency.

Many of the traditional preconditioners have a large sequential component.

Preconditioner

We consider the following parallel preconditioners

Preconditioner

We consider the following parallel preconditioners

- Richardson method,

Preconditioner

We consider the following parallel preconditioners

- Richardson method,
- Jacobi method,

Preconditioner

We consider the following parallel preconditioners

- Richardson method,
- Jacobi method,

Preconditioner

We consider the following parallel preconditioners

- Richardson method,
- Jacobi method,
- non-overlapping domain decomposition,

Preconditioner

We consider the following parallel preconditioners

- Richardson method,
- Jacobi method,
- non-overlapping domain decomposition,

and the parallelization of the Gauß-Seidel and SOR method with

- wavefront numbering,

Preconditioner

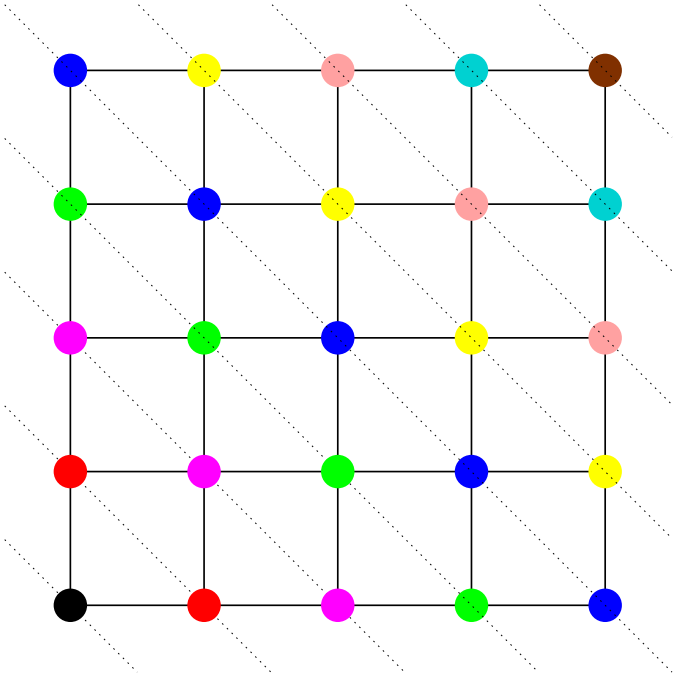
We consider the following parallel preconditioners

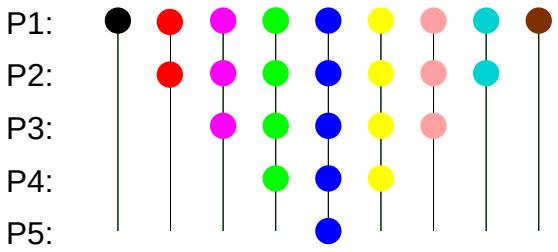
- Richardson method,
- Jacobi method,
- non-overlapping domain decomposition,

and the parallelization of the Gauß-Seidel and SOR method with

- wavefront numbering,
- red-black numbering.

Wavefront Numbering

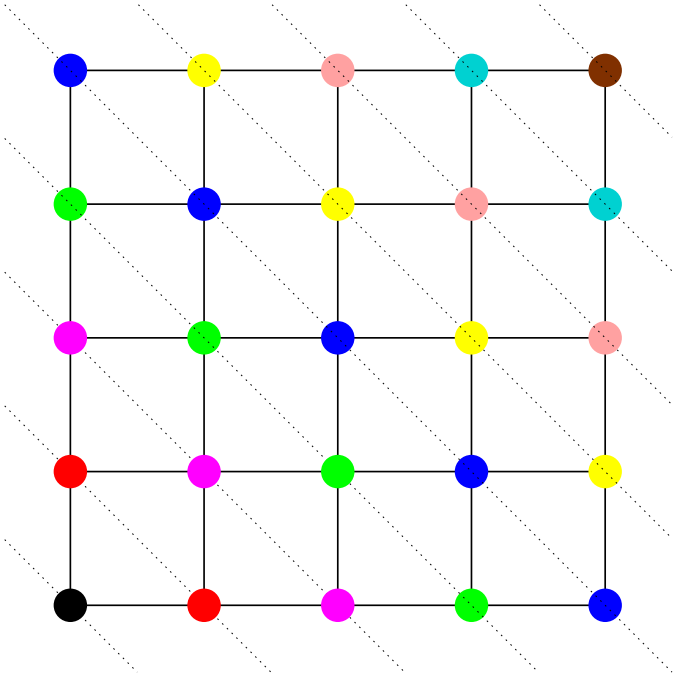


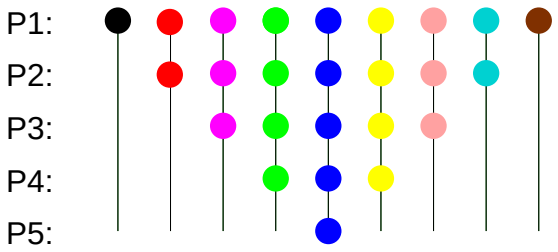


Algorithm

1. on each diagonale, each component can be computed seperatly

Wavefront Numbering

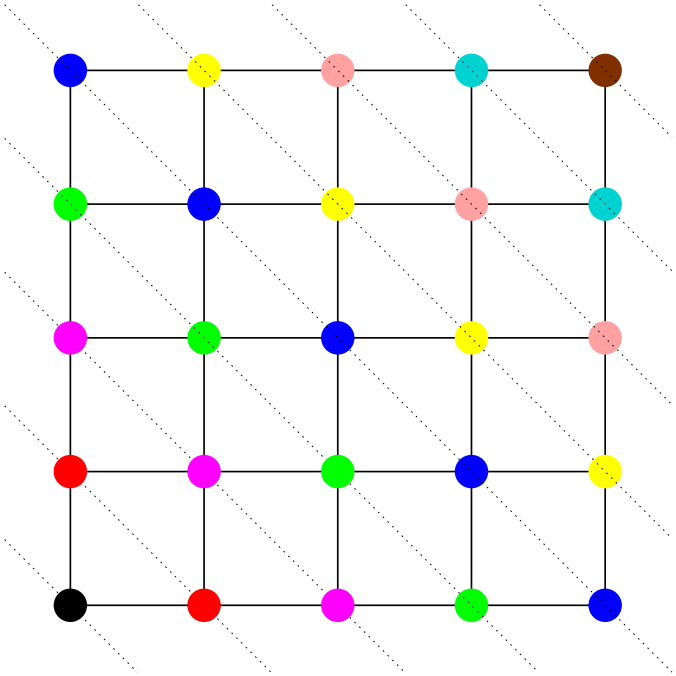


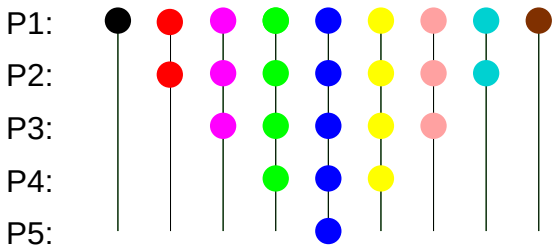


Algorithm

1. on each diagonale, each component can be computed seperatly
2. work load unbalanced

Wavefront Numbering

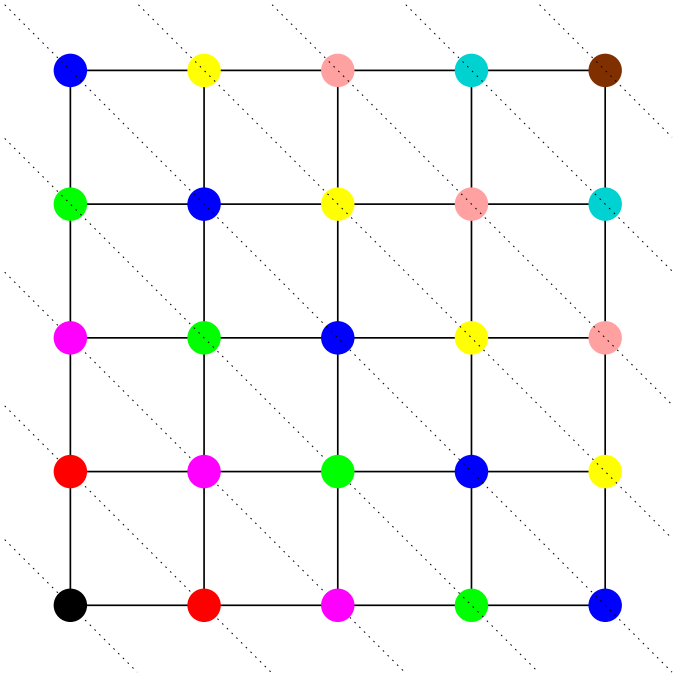


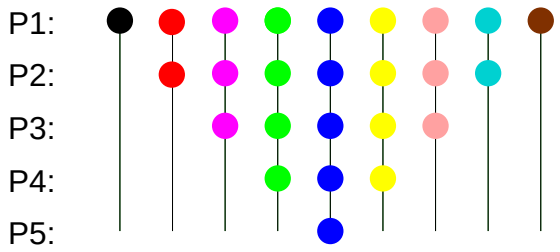


Algorithm

1. on each diagonale, each component can be computed seperatly
2. work load unbalanced
3. maximal possible speed-up in a $P \times P$-mesh is $\approx P/2$

Wavefront Numbering

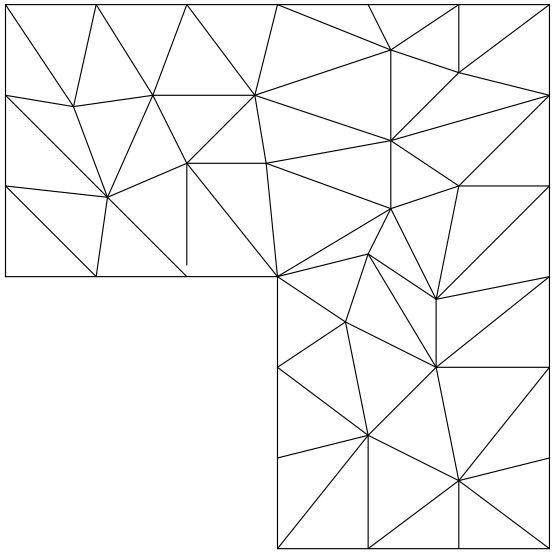




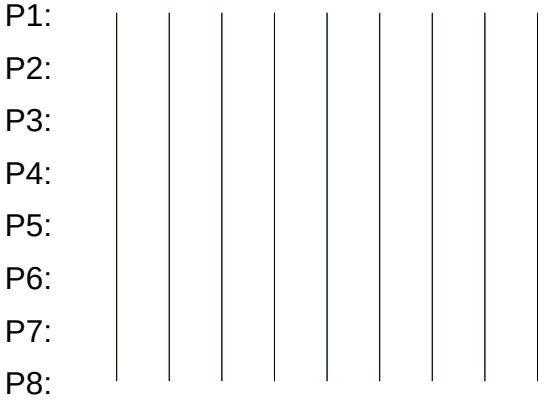
Algorithm

1. on each diagonale, each component can be computed seperatly
2. work load unbalanced
3. maximal possible speed-up in a $P \times P$-mesh is $\approx P/2$
4. what about more general meshes (no 'quadratic' mesh)?

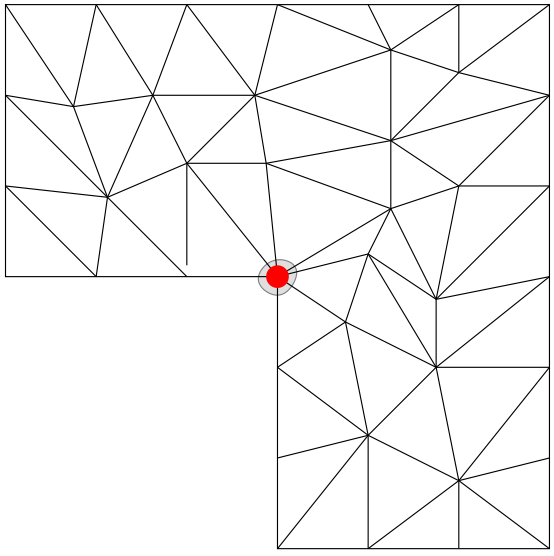
Wavefront Numbering



Algorithm

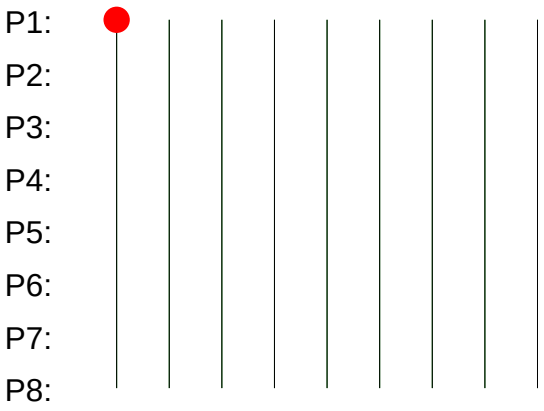


Wavefront Numbering

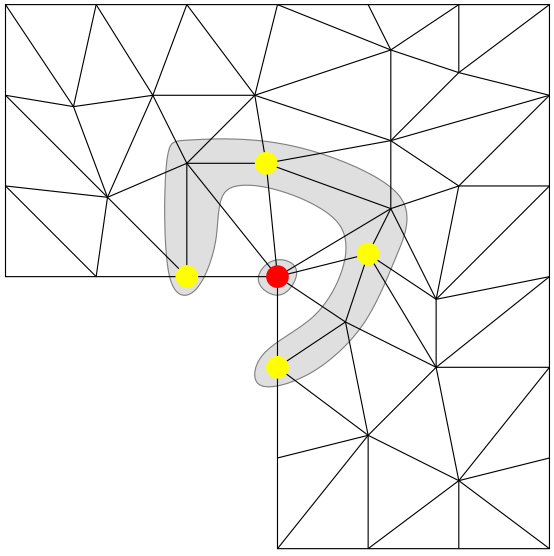


Algorithm

1. start at a node s.t. number of layers is 'minimal'

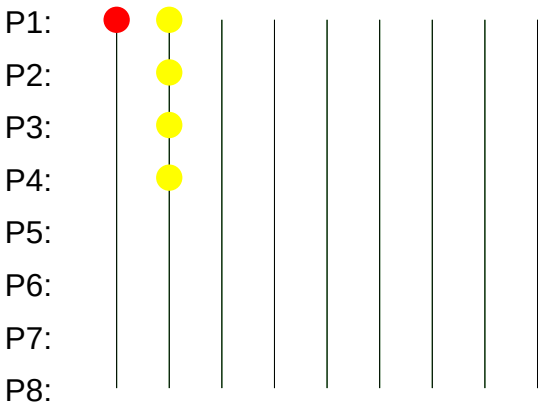


Wavefront Numbering

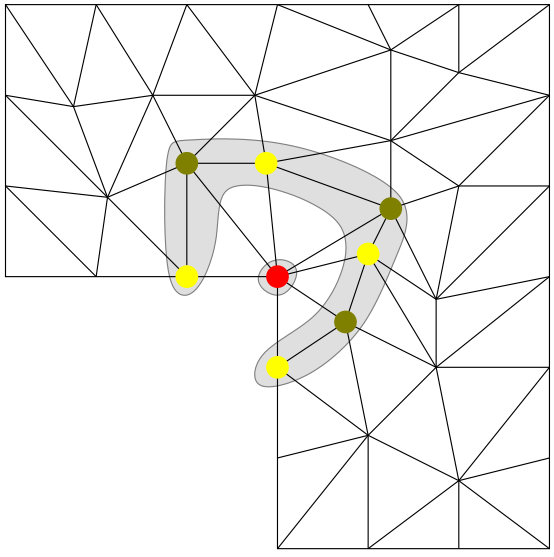


Algorithm

1. start at a node s.t. number of layers is 'minimal'
2. mark next layer and update as much nodes as possible

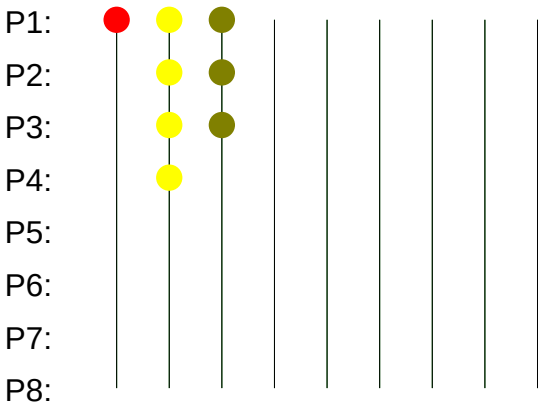


Wavefront Numbering

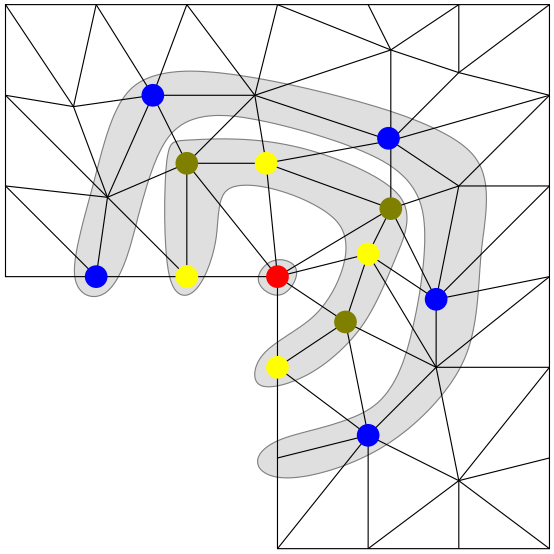


Algorithm

1. start at a node s.t. number of layers is 'minimal'
2. mark next layer and update as much nodes as possible
3. update remainding nodes before marking next layer

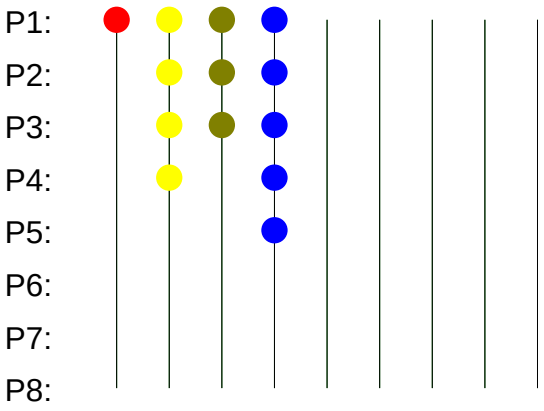


Wavefront Numbering

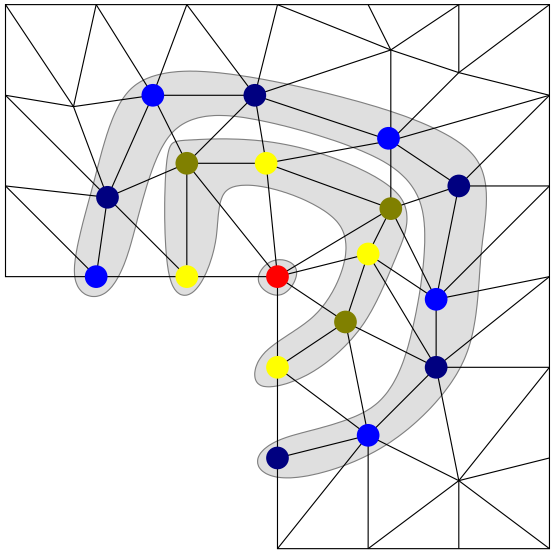


Algorithm

1. start at a node s.t. number of layers is 'minimal'
2. mark next layer and update as much nodes as possible
3. update remainding nodes before marking next layer
4. continue with 2.

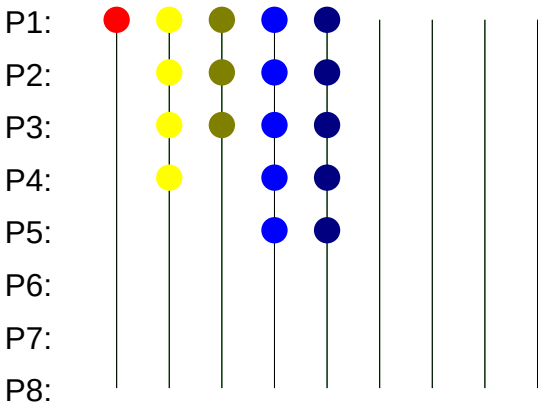


Wavefront Numbering

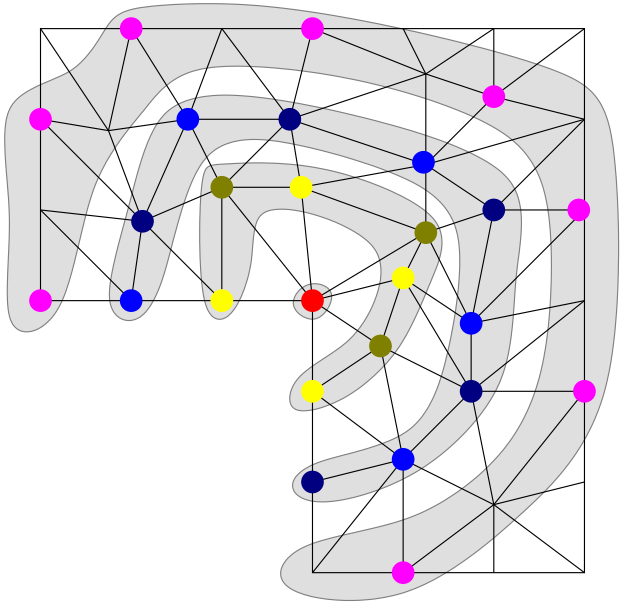


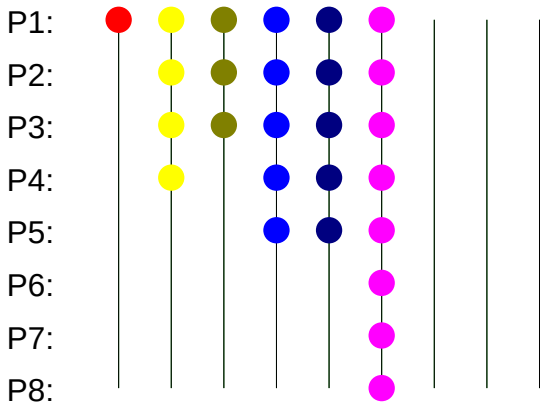
Algorithm

1. start at a node s.t. number of layers is 'minimal'
2. mark next layer and update as much nodes as possible
3. update remainding nodes before marking next layer
4. continue with 2.



Wavefront Numbering

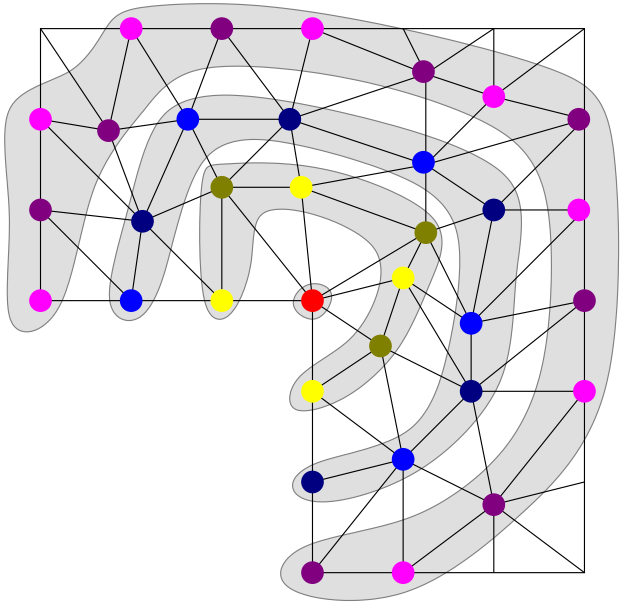




Algorithm

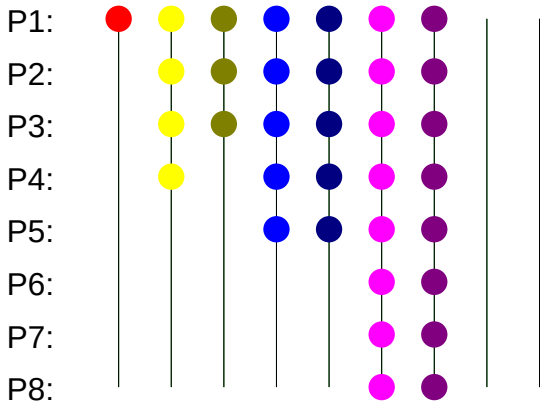
1. start at a node s.t. number of layers is 'minimal'
2. mark next layer and update as much nodes as possible
3. update remainding nodes before marking next layer
4. continue with 2.

Wavefront Numbering

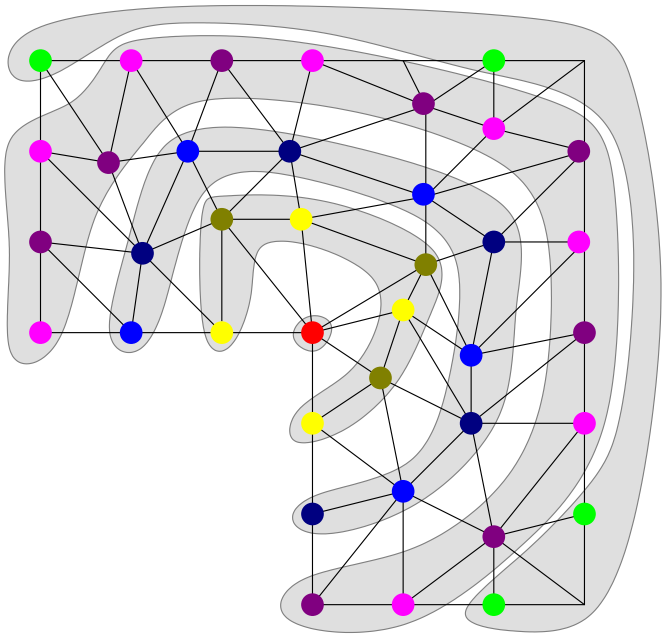


Algorithm

1. start at a node s.t. number of layers is 'minimal'
2. mark next layer and update as much nodes as possible
3. update remainding nodes before marking next layer
4. continue with 2.

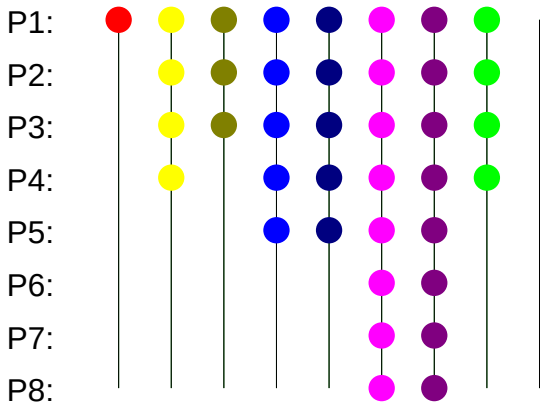


Wavefront Numbering

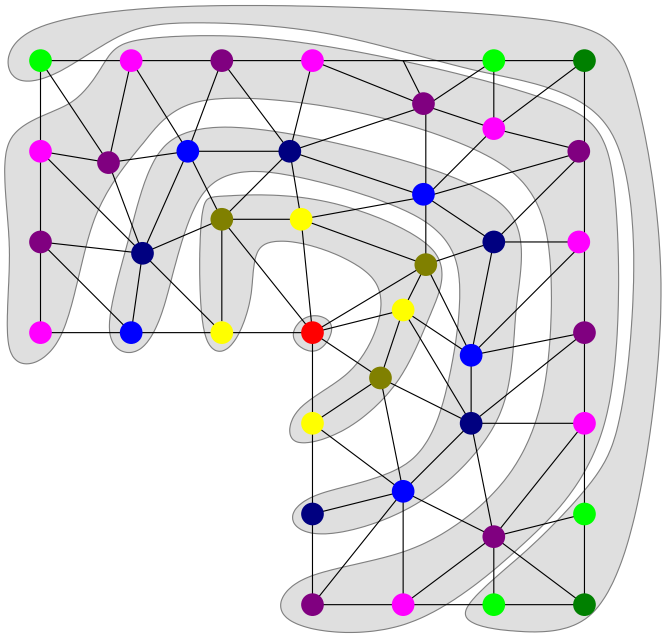


Algorithm

1. start at a node s.t. number of layers is 'minimal'
2. mark next layer and update as much nodes as possible
3. update remainding nodes before marking next layer
4. continue with 2.

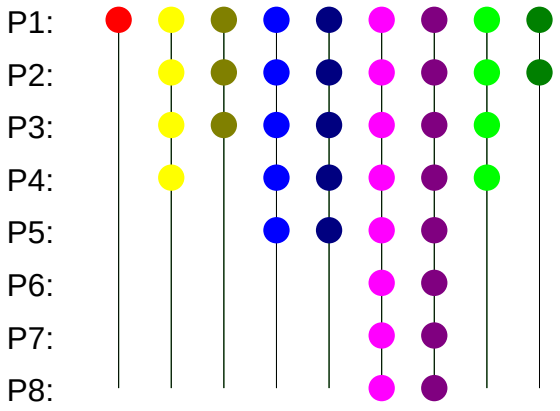


Wavefront Numbering

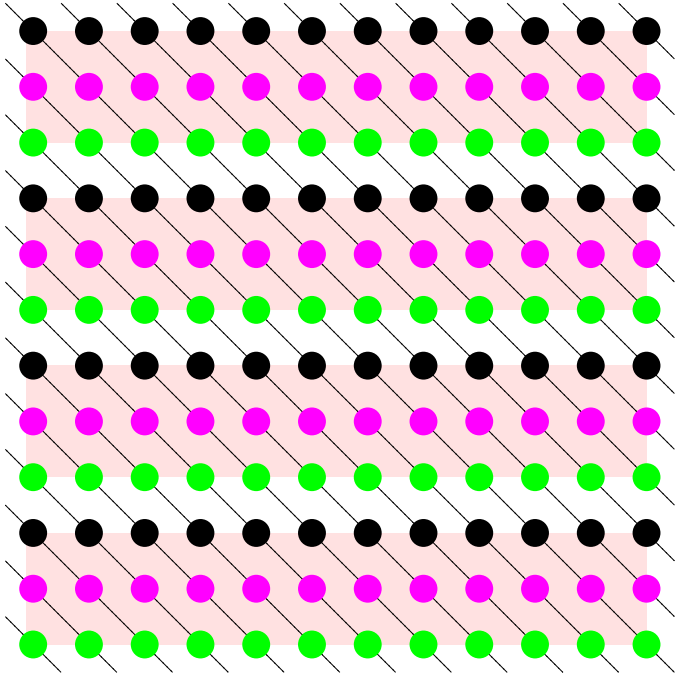


Algorithm

1. start at a node s.t. number of layers is 'minimal'
2. mark next layer and update as much nodes as possible
3. update remainding nodes before marking next layer
4. continue with 2.



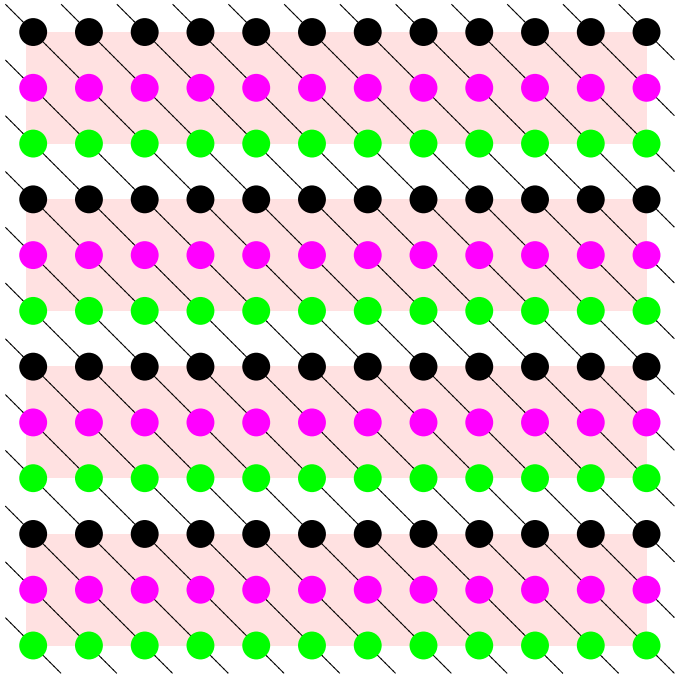
Block-Strips



Algorithm

1. each block strip will be computed one after another

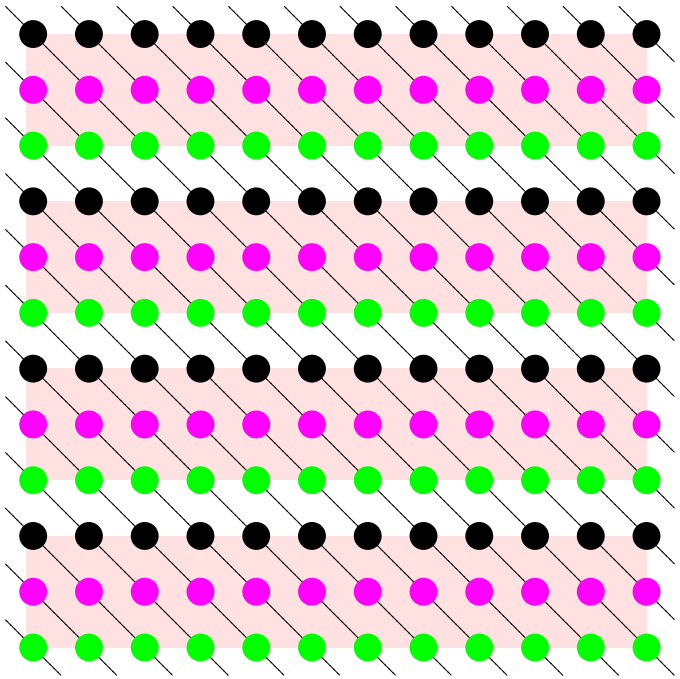
Block-Strips



Algorithm

1. each block strip will be computed one after another
2. work load balanced
   (optimal for $kP \times kP$-meshes)

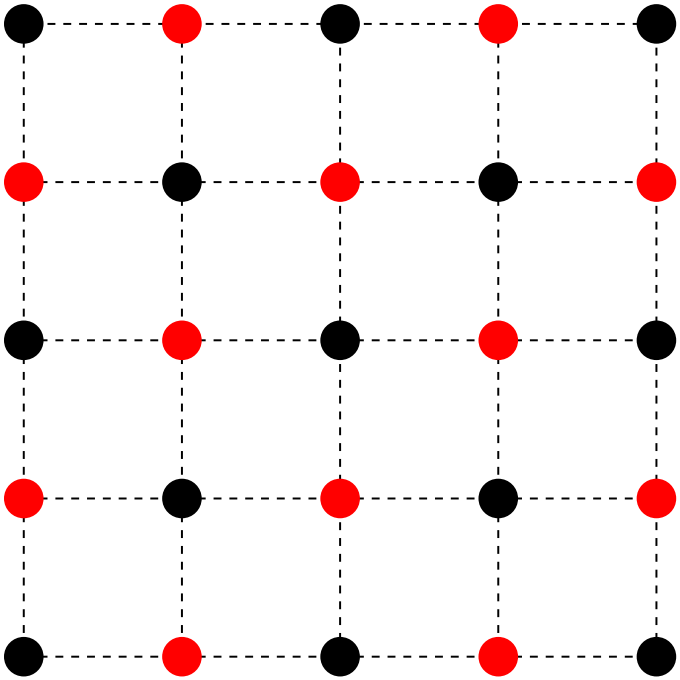
Block-Strips



Algorithm

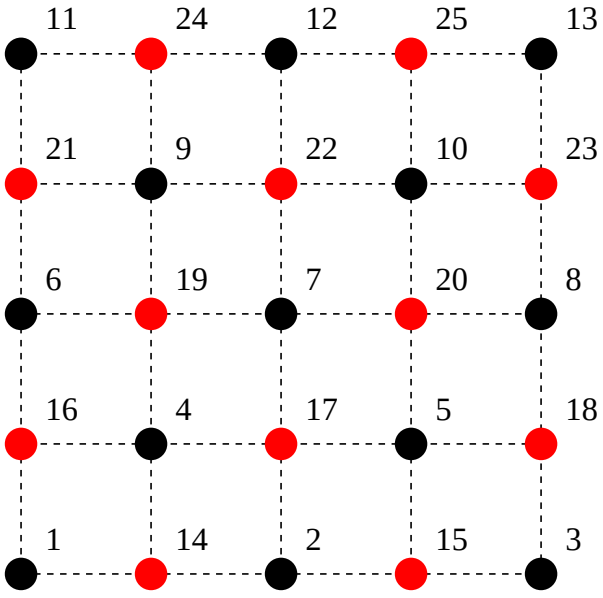
1. each block strip will be computed one after another
2. work load balanced
   (optimal for $kP \times kP$-meshes)
3. maximal possible speed-up is
   $kP/(k+1)$

Red-Black Numbering



What happens, if we number all red nodes first?

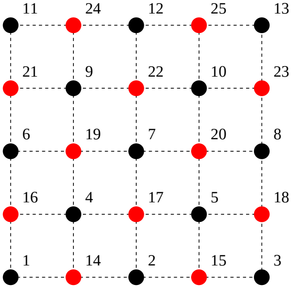
Red-Black Numbering



What happens, if we number all red nodes first?

$$\begin{pmatrix} * & & & & & * & & * & \\ & * & & & & * & * & & * \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \end{pmatrix}$$

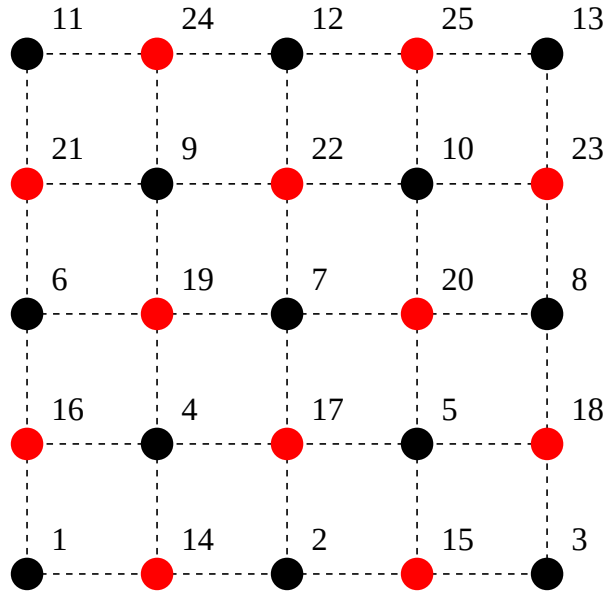
Red-Black Numbering



What happens, if we number all red nodes first?

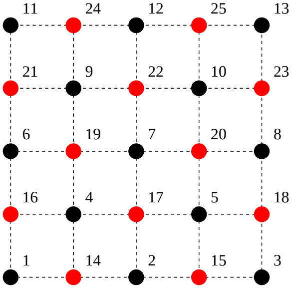
$$\left(\begin{array}{ccccccccccccccccccccccccc}
* &&&&&&&&&&&&& * && * &&&&&&&&& \\
& * &&&&&&&&&&&& * & * && * &&&&&&&& \\
&& * &&&&&&&&&&&& * &&& * &&&&&&& \\
&&&&&&&&&&&&&&&&&&&&&&&& \\
\end{array}\right)$$

Red-Black Numbering



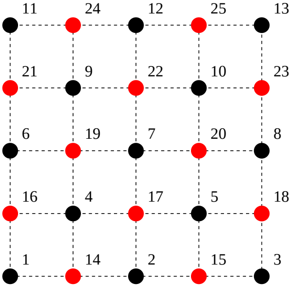
What happens, if we number all red nodes first?

Red-Black Numbering



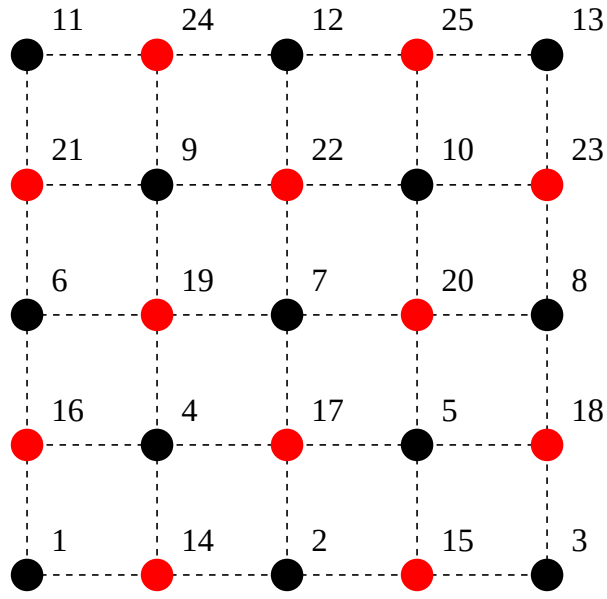
What happens, if we number all red nodes first?

Red-Black Numbering



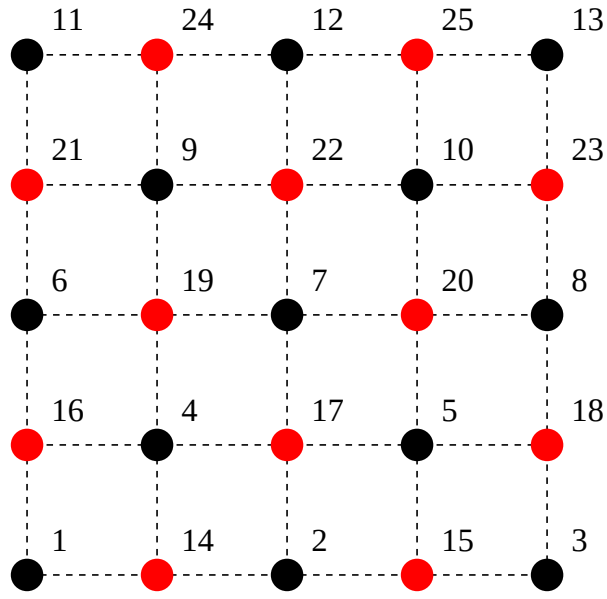
What happens, if we number all red nodes first?

Red-Black Numbering

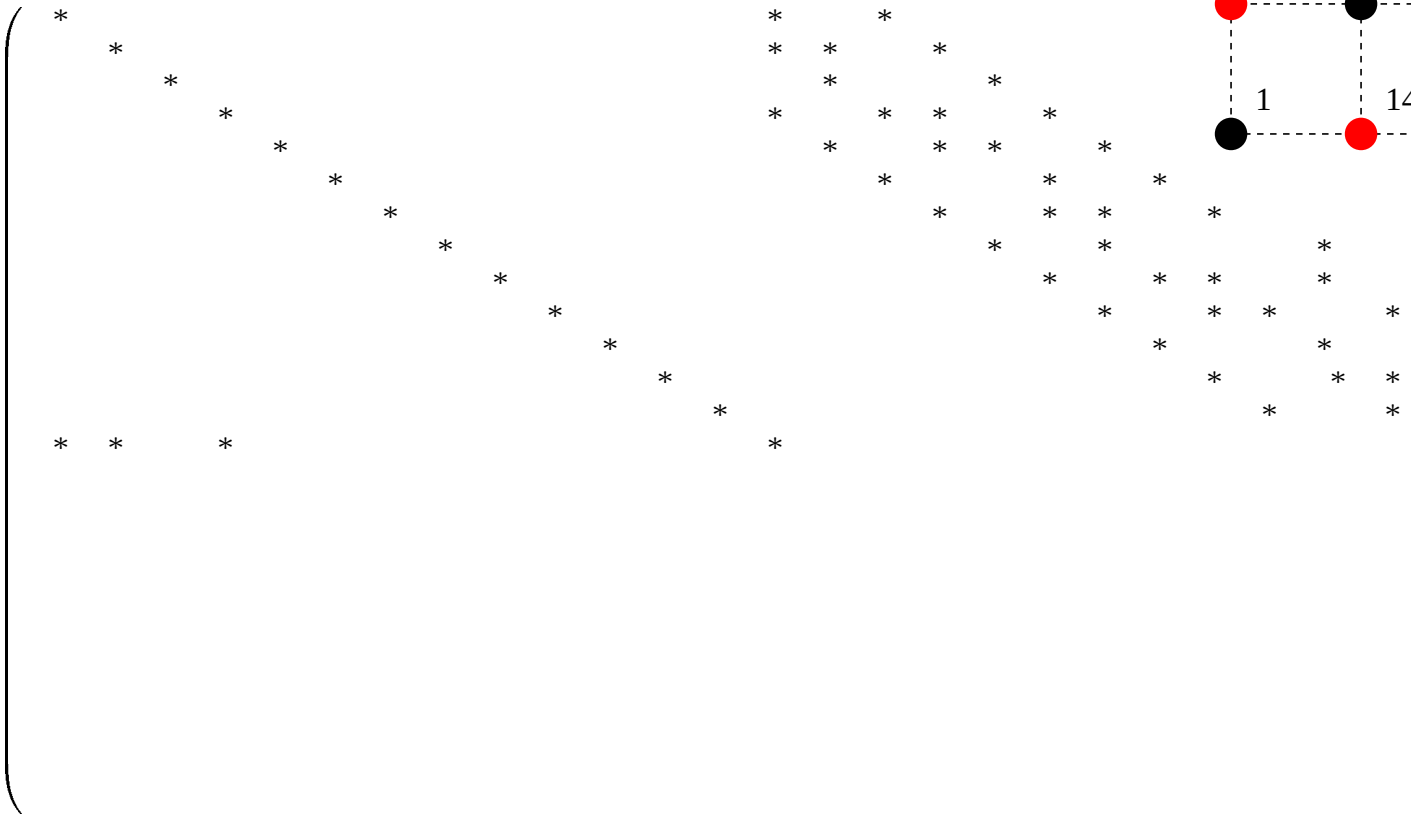


What happens, if we number all red nodes first?

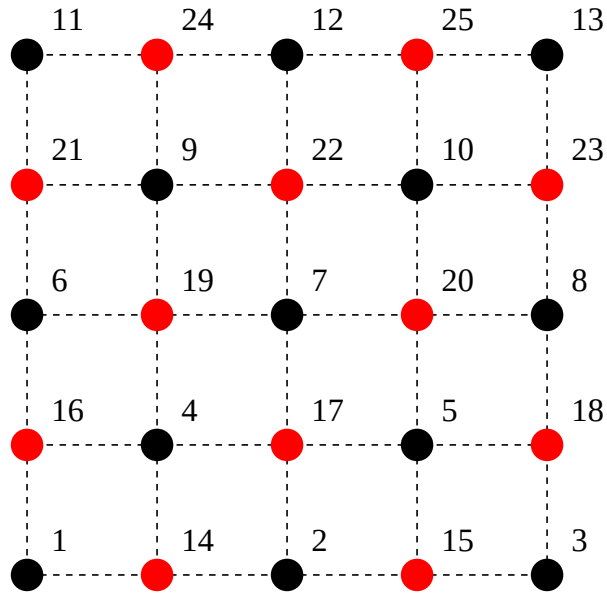
Red-Black Numbering



What happens, if we number all red nodes first?

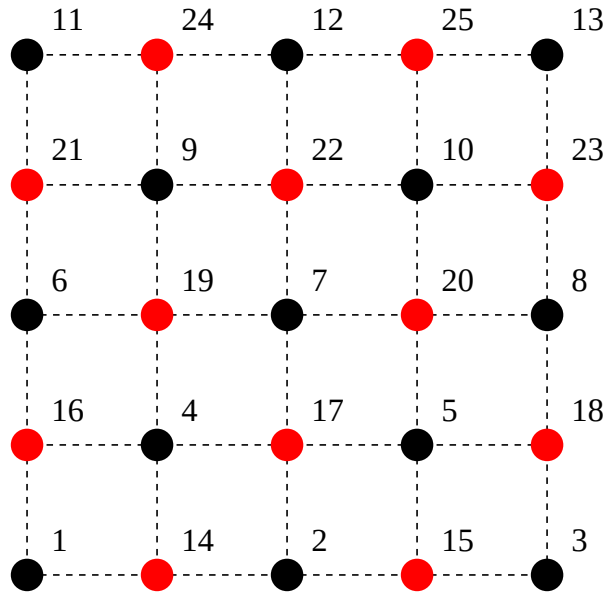


Red-Black Numbering

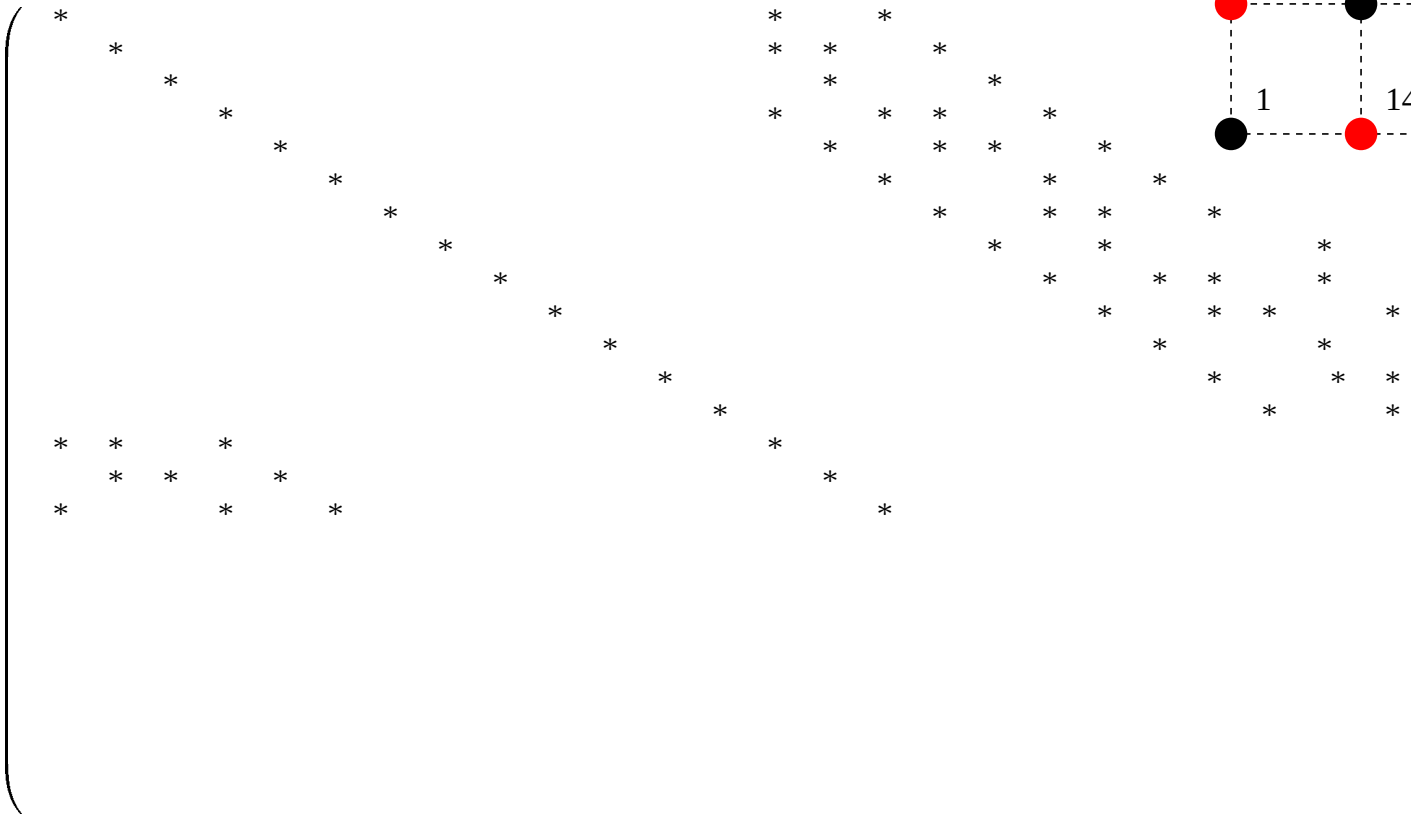


What happens, if we number all red nodes first?

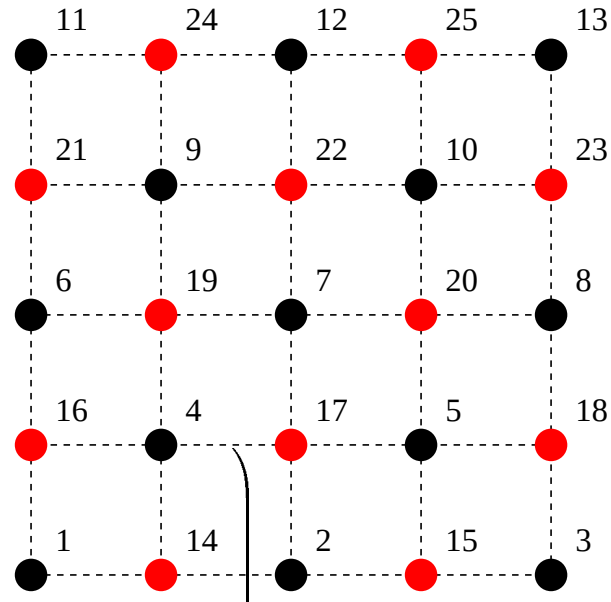
Red-Black Numbering



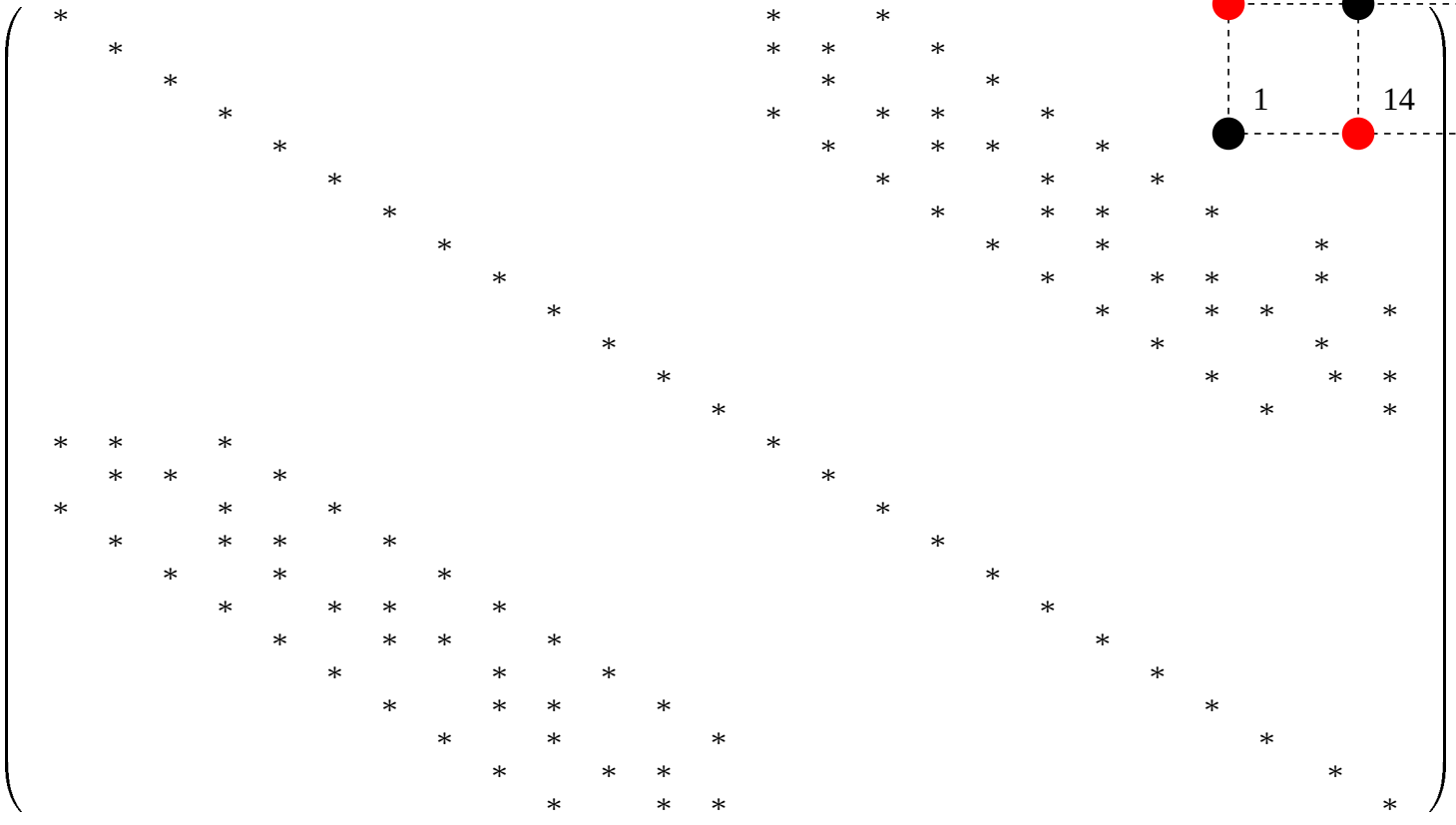
What happens, if we number all red nodes first?



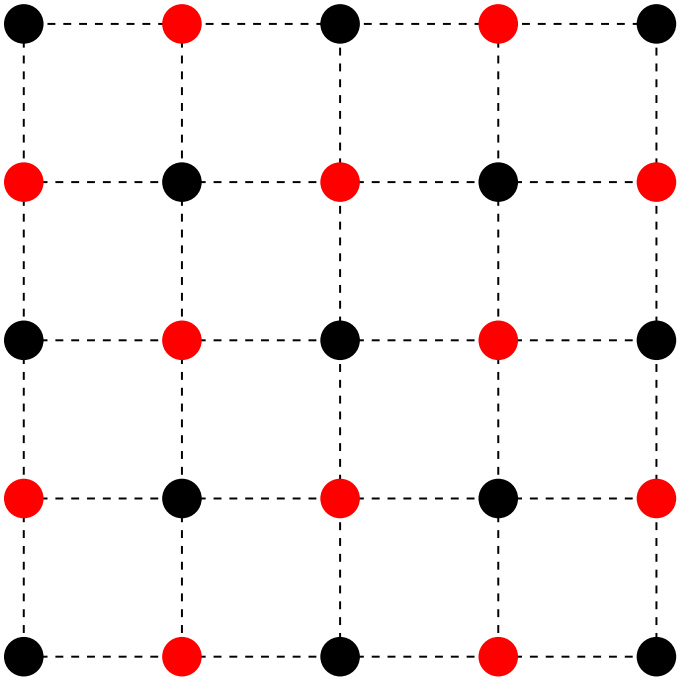
Red-Black Numbering



What happens, if we number all red nodes first?



Red-Black Numbering

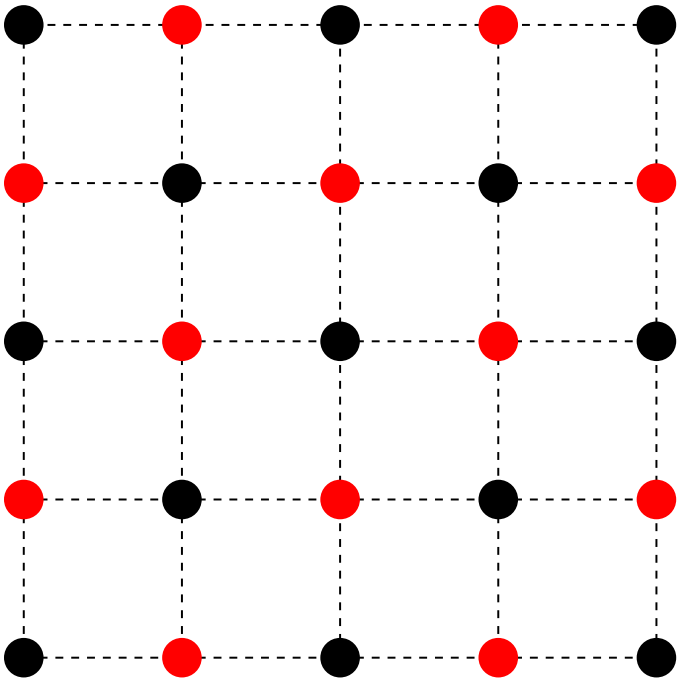


What happens, if we number all red nodes first?

Properties

1. FEM-matrix with
   swaped rows and columns

Red-Black Numbering

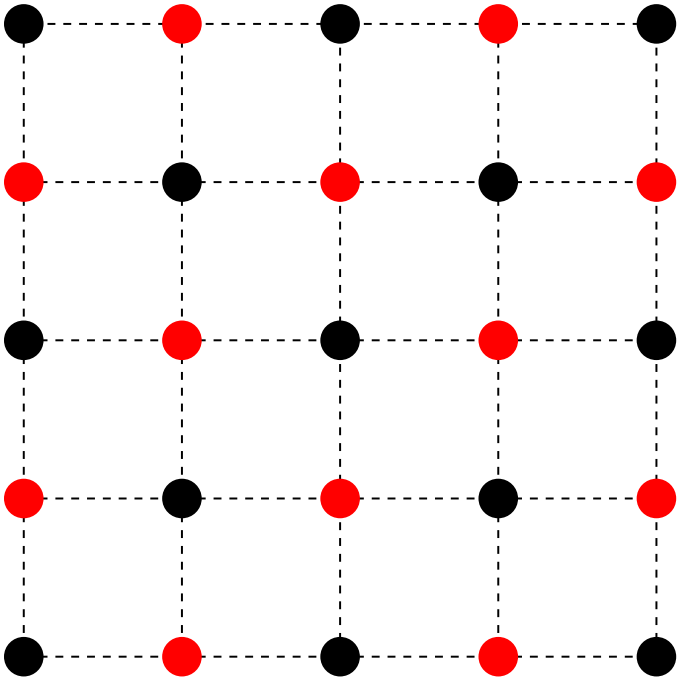


What happens, if we number all red nodes first?

Properties

1. FEM-matrix with swaped rows and columns
2. 2×2 block matrix

Red-Black Numbering



What happens, if we number all red nodes first?

Properties

1. FEM-matrix with swaped rows and columns
2. 2×2 block matrix
3. diagonal blocks are diagonal matrices

Jacobi / Gauß-Seidel Iteration

Consider the system $Ax = b$ and the decomposition $A = L + D + U$.

Sequential version of Jacobi iteration.

$$x^{(k+1)} := D^{-1}(b - Lx^{(k)} - Ux^{(k)})$$

If D^{-1} is available on each processor, only communication is necessary to exchange parts of $x^{(k+1)}$ after updating.

Sequential version of Gauß-Seidel iteration.

$$x^{(k+1)} := D^{-1}(b - Lx^{(k+1)} - Ux^{(k)})$$

or

$$x^{(k+1)} := D^{-1}(b - Lx^{(k)} - Ux^{(k+1)})$$

Parallel Gauß-Seidel Iteration (Red-Black-Numbering)

Let $A = (a_{ij}) \in \mathbb{R}^{n \times n}$.
Assume, we have at least two disjoint index sets I_{red} and I_{black},
s.t. $a_{ij} \equiv 0$ for all $i \neq j \in I_{red}$ resp. I_{black}.

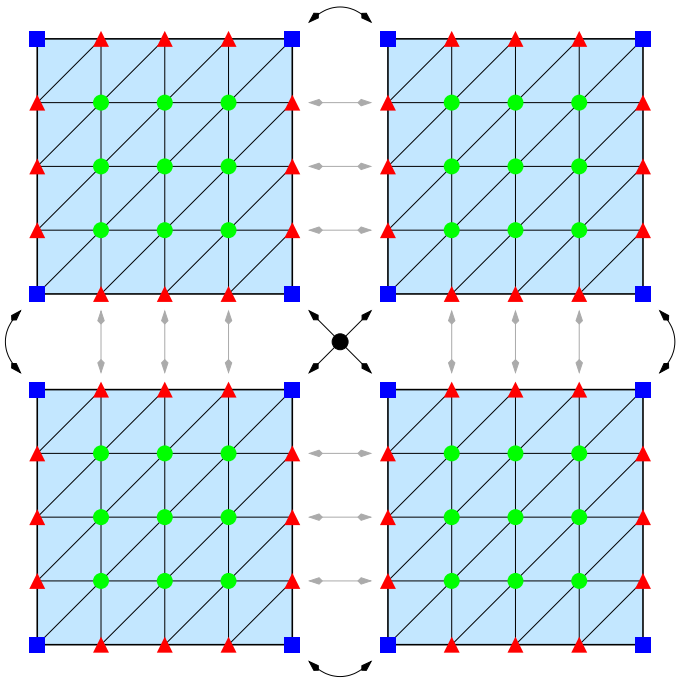
Parallel version of Gauß-Seidel iteration.

$$x_{red}^{(k+1)} := D_{red}^{-1}(b_{red} - (L_{rb} + U_{rb})x_{black}^{(k)})$$

$$x_{black}^{(k+1)} := D_{black}^{-1}(b_{black} - (L_{br} + U_{br})x_{red}^{(k)})$$

If $P >= 2$ it is recommended to use a block version,
s.t. blocks of the same color need no communication.

Non-overlapping Subdomains



Different Indizes

1. **I** nodes in interior of subdomains $[N_I = \sum_{j=1}^{p} N_{I,j}]$.
2. **E** nodes in interior of subdomains-edges $[N_E = \sum_{j=1}^{n_e} N_{E,j}]$. ($n_e$ number of subdomain-edges)
3. **V** crosspoints, i.e. endpoints of subdomain-edges $[N_V]$

Types of Vectors

Two types of vectors, depending on the storage type:

type I: $\overline{u}$ is stored on P_k as restriction $\overline{u}_k = C_k \overline{u}$.
'Complete' value accessable on P_k.

type II: $\underline{r}$ is stored on P_k as $\underline{r}_k$, s.t.
$\underline{r} = \sum_{k=1}^{p} C_k^T \underline{r}_k$.
Nodes on the interface have only a part of the full value.

How should we parallelize the Gauß-Seidel iteration if we have non-overlapping subdomains?

$$x^{(k+1)} := D^{-1}(b - Lx^{(k+1)} - Ux^{(k)})$$

resp.

$$\underline{x}_i^{(k+1)} := \left(C_i \sum_{k=1}^{p} C_k^T \mathrm{diag}(D) \right)^{\cdot -1} \sum_{\ell=1}^{p} C_\ell^T (\underline{b} - L\overline{x}^{(k+1)} - U\overline{x}^{(k)})$$

Parallel Gauß-Seidel (Non-Overlapping Domains

Consider the following ordering of global index set: (V, E, I)

$$\begin{pmatrix} A_{VV} & A_{VE} & A_{VI} \\ A_{EV} & A_{EE} & A_{EI} \\ A_{IV} & A_{IE} & A_{II} \end{pmatrix} \begin{pmatrix} x_V \\ x_E \\ x_I \end{pmatrix} = \begin{pmatrix} b_V \\ b_E \\ b_I \end{pmatrix}$$

Parallel Gauß-Seidel (Non-Overlapping Domains, Draft)

Let $\overline{d} := \{1/d_{ii}\}_{i=1,\ldots,n}$, $\circledast$ componentwise multiplication.

$$
\begin{aligned}
\underline{r}_V &:= \underline{b}_V - A_{VV}\overline{x}_V^k - A_{VE}\overline{x}_E^k - A_{VI}\overline{x}_I^k \\
\overline{w}_V &:= \sum_{\ell=1}^{p} C_{V,\ell}^T \underline{r}_{V,\ell} \quad \text{communication} \\
\overline{x}_V^{k+1} &:= \overline{x}_V^k + \overline{d}_V \circledast \overline{w}_V
\end{aligned}
$$

Parallel Gauß-Seidel (Non-Overlapping Domains, Draft)

Let $\overline{d} := \{1/d_{ii}\}_{i=1,\ldots,n}$, $\circledast$ componentwise multiplication.

$$
\begin{aligned}
\underline{r}_V &:= \underline{b}_V - A_{VV}\overline{x}_V^k - A_{VE}\overline{x}_E^k - A_{VI}\overline{x}_I^k \\
\overline{w}_V &:= \sum_{\ell=1}^{p} C_{V,\ell}^T \underline{r}_{V,\ell} \quad \text{communication} \\
\overline{x}_V^{k+1} &:= \overline{x}_V^k + \overline{d}_V \circledast \overline{w}_V \\
\underline{r}_E &:= \underline{b}_E - A_{EV}\overline{x}_V^{k+1} - A_{EE}\overline{x}_E^k - A_{EI}\overline{x}_I^k \\
\overline{w}_E &:= \sum_{\ell=1}^{p} C_{E,\ell}^T \underline{r}_{E,\ell} \quad \text{communication, real Gauß-Seidel???} \\
\overline{x}_E^{k+1} &:= \overline{x}_E^k + \overline{d}_E \circledast \overline{w}_E
\end{aligned}
$$

Parallel Gauß-Seidel (Non-Overlapping Domains, Draft)

Let $\overline{d} := \{1/d_{ii}\}_{i=1,\ldots,n}$, $\circledast$ componentwise multiplication.

$$
\begin{aligned}
\underline{r}_V &:= \underline{b}_V - A_{VV}\overline{x}_V^k - A_{VE}\overline{x}_E^k - A_{VI}\overline{x}_I^k \\
\overline{w}_V &:= \sum_{\ell=1}^{p} C_{V,\ell}^T \underline{r}_{V,\ell} \quad \text{communication} \\
\overline{x}_V^{k+1} &:= \overline{x}_V^k + \overline{d}_V \circledast \overline{w}_V \\
\underline{r}_E &:= \underline{b}_E - A_{EV}\overline{x}_V^{k+1} - A_{EE}\overline{x}_E^k - A_{EI}\overline{x}_I^k \\
\overline{w}_E &:= \sum_{\ell=1}^{p} C_{E,\ell}^T \underline{r}_{E,\ell} \quad \text{communication, real Gauß-Seidel???} \\
\overline{x}_E^{k+1} &:= \overline{x}_E^k + \overline{d}_E \circledast \overline{w}_E \\
\underline{r}_I &:= \underline{b}_I - A_{IV}\overline{x}_V^{k+1} - A_{IE}\overline{x}_E^{k+1} - A_{II}\overline{x}_I^k \\
\overline{w}_I &:= \sum_{\ell=1}^{p} C_{I,\ell}^T \underline{r}_{I,\ell} \quad \text{no communication!!!} \\
\overline{x}_I^{k+1} &:= \overline{x}_I^k + \overline{d}_I \circledast \overline{w}_I
\end{aligned}
$$

Parallel Gauß-Seidel (Non-Overlapping Domains, modified)

Assume at least one node on each coupling edge and no connection between different edges.

$$
\begin{aligned}
\underline{r}_V &:= \underline{b}_V - A_{VV}\overline{x}_V^k - A_{VE}\overline{x}_E^k - A_{VI}\overline{x}_I^k \\
\overline{w}_V &:= \sum_{\ell=1}^{p} C_{V,\ell}^T \underline{r}_{V,\ell} \quad \text{communication} \\
\overline{x}_V^{k+1} &:= \overline{x}_V^k + \overline{d}_V \circledast \overline{w}_V
\end{aligned}
$$

Parallel Gauß-Seidel (Non-Overlapping Domains, modified)

Assume at least one node on each coupling edge and no connection between different edges.

$$
\begin{aligned}
\underline{r}_V &:= \underline{b}_V - A_{VV}\overline{x}_V^k - A_{VE}\overline{x}_E^k - A_{VI}\overline{x}_I^k \\
\overline{w}_V &:= \sum_{\ell=1}^{p} C_{V,\ell}^T \underline{r}_{V,\ell} \quad \text{communication} \\
\overline{x}_V^{k+1} &:= \overline{x}_V^k + \overline{d}_V \circledast \overline{w}_V \\
\underline{r}_E &:= \underline{b}_E - A_{EV}\overline{x}_V^{k+1} - A_{EE}\overline{x}_E^k - A_{EI}\overline{x}_I^k \\
\overline{w}_E &:= \sum_{\ell=1}^{p} C_{E,\ell}^T \underline{r}_{E,\ell} \\
\overline{x}_E^{k+1} &:= \overline{x}_E^k + A_{EE}^{-1}\overline{w}_E \quad \text{block diagonal matrix, each block tridiagonal}
\end{aligned}
$$

Parallel Gauß-Seidel (Non-Overlapping Domains, modified)

Assume at least one node on each coupling edge and no connection between different edges.

$$
\begin{aligned}
\underline{r}_V &:= \underline{b}_V - A_{VV}\overline{x}_V^k - A_{VE}\overline{x}_E^k - A_{VI}\overline{x}_I^k \\
\overline{w}_V &:= \sum_{\ell=1}^{p} C_{V,\ell}^T \underline{r}_{V,\ell} \quad \text{communication} \\
\overline{x}_V^{k+1} &:= \overline{x}_V^k + \overline{d}_V \circledast \overline{w}_V \\
\underline{r}_E &:= \underline{b}_E - A_{EV}\overline{x}_V^{k+1} - A_{EE}\overline{x}_E^k - A_{EI}\overline{x}_I^k \\
\overline{w}_E &:= \sum_{\ell=1}^{p} C_{E,\ell}^T \underline{r}_{E,\ell} \\
\overline{x}_E^{k+1} &:= \overline{x}_E^k + A_{EE}^{-1}\overline{w}_E \quad \text{block diagonal matrix, each block tridiagonal} \\
\underline{r}_I &:= \underline{b}_I - A_{IV}\overline{x}_V^{k+1} - A_{IE}\overline{x}_E^{k+1} - A_{II}\overline{x}_I^k \\
\overline{w}_I &:= \sum_{\ell=1}^{p} C_{I,\ell}^T \underline{r}_{I,\ell} \quad \text{no communication!!!} \\
\overline{x}_I^{k+1} &:= \overline{x}_I^k + A_{II}^{-1}\overline{w}_I
\end{aligned}
$$

Gauß-Seidel via Jacobi

Definition: A Matrix $A \in \mathbb{R}^{m \times n}$ is called non-negativ, if all coefficients a_{ij} of A are non-negativ.

Satz: [Stein and Rosenberg] Let the iteration matrix $C_J \in \mathbb{R}^{n \times n}$ of the Jacobi-iteration be non-negativ. Then there hold the following properties

i) $\varrho(C_J) = \varrho(C_G) = 0$

ii) $\varrho(C_J) = \varrho(C_G) = 1$

iii) $0 < \varrho(C_G) < \varrho(C_J) < 1$

iv) $1 < \varrho(C_J) < \varrho(C_G)$

Example:

$$A = \begin{pmatrix} 2 & -1 & & 0 \\ -1 & 2 & -1 & \\ & -1 & 2 & -1 \\ 0 & & -1 & 2 \end{pmatrix}, C_J = \begin{pmatrix} 0 & \frac{1}{2} & & \\ \frac{1}{2} & 0 & \frac{1}{2} & \\ & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & & \frac{1}{2} & 0 \end{pmatrix}$$

$\Rightarrow$ Gauß-Seidel is faster than Jacobi (for FEM matrices, also in 2D/3D)!