Parallel Scientific Computing

Course AMS301 — Fall 2023 — Lecture 3

Iterative methods for linear systems (1)

Stationary methods & Application to finite differences

Axel Modave

Find $\mathbf{x} \in \mathbb{R}^n$ such that $\boxed{\mathbf{Ax} = \mathbf{b}}$ with $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{b} \in \mathbb{R}^n$.

Motivation

Many computational procedures require the solution of linear systems:

- for linear physical problems,
- for non-linear physical problems,
- for optimization procedures,
- ...

for many applications:

- electromagnetic compatibility, aeroacoustic studies, CFD,
- medical imaging, geophysical imaging,
- ...

From a mathematical point of view:

- discretized elliptic problem
  $\Rightarrow$ linear system to solve!
- discretized hyperbolic problem with an implicit time stepping scheme
  $\Rightarrow$ linear system to solve at each time step!

Find $\mathbf{x} \in \mathbb{R}^n$ such that $\boxed{\mathbf{A}\mathbf{x} = \mathbf{b}}$ with $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{b} \in \mathbb{R}^n$.

Solution procedures

- **Direct methods:** Factorization of $\mathbf{A}$ into triangular and diagonal matrices (ex. $\mathbf{A} = \mathbf{L}\mathbf{U}$) and solution of simpler problems.

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad \Leftrightarrow \quad \mathbf{L}\mathbf{U}\mathbf{x} = \mathbf{b} \quad \Leftrightarrow \quad \left| \begin{aligned} \mathbf{L}\mathbf{y} &= \mathbf{b} \\ \mathbf{U}\mathbf{x} &= \mathbf{y} \end{aligned} \right.$$

  Advantages: exact solution known after a given number of operations
  Difficulties: heavy computational cost *(operations/memory)*, hard to parallelize

- **Iterative methods:** Iterative procedure to minimizing an error $\|\mathbf{x}^{(\ell)} - \mathbf{x}_{\text{ref}}\|$ and/or a residual $\|\mathbf{A}\mathbf{x}^{(\ell)} - \mathbf{b}\|$.

$$\left| \begin{aligned} \mathbf{x}^{(0)} &= \mathrm{Iter}_{(0)}\left(\mathbf{A}, \mathbf{b}\right) \\ \mathbf{x}^{(\ell+1)} &= \mathrm{Iter}^{(\ell+1)}\left(\mathbf{x}^{(\ell)}, \mathbf{x}^{(\ell-1)}, \ldots, \mathbf{A}, \mathbf{b}\right), \quad \text{pour } \ell \geq 0 \end{aligned} \right.$$

  Advantages: limited cost per iteration *(operations/memory)*, easy to parallelize
  Difficulties: approximate solution, control of the convergence of the process

Find $\mathbf{x} \in \mathbb{R}^n$ such that $\boxed{\mathbf{A}\mathbf{x} = \mathbf{b}}$ with $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{b} \in \mathbb{R}^n$.

- **Iterative methods:**

$$\left| \begin{aligned} \mathbf{x}^{(0)} &= \mathrm{Iter}_{(0)}\left(\mathbf{A}, \mathbf{b}\right) \\ \mathbf{x}^{(\ell+1)} &= \mathrm{Iter}^{(\ell+1)}\left(\mathbf{x}^{(\ell)}, \mathbf{x}^{(\ell-1)}, \ldots, \mathbf{A}, \mathbf{b}\right), \quad \text{pour } \ell \geq 0 \end{aligned} \right.$$

The order of the method is the numb. of steps which the current iter. depends on.
Stationary method if the functions $\mathrm{Iter}^{(\ell)}$ are indep. of ℓ, otherwise nonstationary
Linear method if the functions $\mathrm{Iter}^{(\ell)}$ are linear, otherwise nonlinear

Today, we consider stationary linear iterative schemes of first order:

$$\left| \begin{aligned} &\mathbf{x}^{(0)} \text{ given} \\ &\mathbf{x}^{(\ell+1)} = \mathbf{B}\mathbf{x}^{(\ell)} + \mathbf{f}, \quad \ell \geq 0 \end{aligned} \right.$$

where $\mathbf{B} \in \mathbb{R}^{n \times n}$ is the iteration matrix and $\mathbf{f} \in \mathbb{R}^n$ depends on $\mathbf{b}$.

Iterative methods for linear systems

Stationary methods

System arising from a finite difference discretization

Find $\mathbf{x} \in \mathbb{R}^n$ such that $\boxed{\mathbf{A}\mathbf{x} = \mathbf{b}}$ with $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{b} \in \mathbb{R}^n$.

We consider a general procedure:

$$\left|\begin{array}{l} \mathbf{x}^{(0)} \text{ given} \\ \mathbf{x}^{(\ell+1)} = \mathbf{B}\mathbf{x}^{(\ell)} + \mathbf{f}, \quad \ell \geq 0 \end{array}\right.$$

where $\mathbf{B} \in \mathbb{R}^{n\times n}$ is the iteration matrix and $\mathbf{f} \in \mathbb{R}^n$ depends on $\mathbf{b}$.

Definitions and properties

- Consistent method if the solution is a fixed point of the scheme (*i.e.* $\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{f}$).

  $\boxed{\text{OK}}$ if and only if $\mathbf{f} = (\mathbf{I} - \mathbf{B})\mathbf{A}^{-1}\mathbf{b}$

- Convergent method if $\lim_{\ell\to\infty} \mathbf{x}^{(\ell)} = \mathbf{x}$ for all $\mathbf{x}^{(0)}$.

  For a consistent method, $\boxed{\text{OK}}$ if and only if $\boxed{\rho(\mathbf{B}) < 1}$

  The spectral radius $\rho(\mathbf{B})$ is the max. of the absolute values of the eigenval. of $\mathbf{B}$.

Find $\mathbf{x} \in \mathbb{R}^n$ such that $\boxed{\mathbf{A}\mathbf{x} = \mathbf{b}}$ with $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{b} \in \mathbb{R}^n$.

We consider a general procedure:

$$\left|\begin{array}{l} \mathbf{x}^{(0)} \text{ given} \\ \mathbf{x}^{(\ell+1)} = \mathbf{B}\mathbf{x}^{(\ell)} + \mathbf{f}, \quad \ell \geq 0 \end{array}\right.$$

where $\mathbf{B} \in \mathbb{R}^{n \times n}$ is the iteration matrix and $\mathbf{f} \in \mathbb{R}^n$ depends on $\mathbf{b}$.

Definitions and properties

- We use a stopping criteria on the number of iterations and the norm of the residual:

$$\ell \leq \ell_{\text{tol}} \qquad \text{and} \qquad \left\|\mathbf{r}^{(\ell)}\right\| / \left\|\mathbf{r}^{(0)}\right\| \leq \varepsilon_{\text{tol}}$$

  with the residual vector $\boxed{\mathbf{r}^{(\ell)} := \mathbf{b} - \mathbf{A}\mathbf{x}^{(\ell)}}$.

  For stationary methods, one has:

$$\mathbf{A}\mathbf{e}^{(\ell)} = \mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{x}^{(\ell)} = \mathbf{r}^{(\ell)}$$

$$\Rightarrow \quad \|\mathbf{r}^{(\ell)}\| \leq |||\mathbf{A}|||\, \|\mathbf{e}^{(\ell)}\| \quad \text{et} \quad \|\mathbf{e}^{(\ell)}\| \leq |||\mathbf{A}^{-1}|||\, \|\mathbf{r}^{(\ell)}\|$$

$\longrightarrow$ *Choice of $\mathbf{B}$ for fast convergence and efficient computation?*

Stationary methods — Standard methods [1/3]

We consider a regular decomposition: $\boxed{\mathbf{A} = \mathbf{M} - \mathbf{N}}$ where $\mathbf{M} \in \mathbb{R}^{n \times n}$ is inversible.

Stationary method
$\mathbf{x}^{(0)} \in \mathbb{C}^n$ **for** $\ell = 0, 1, \ldots$ **do** $\quad \mathbf{M}\mathbf{x}^{(\ell+1)} = \mathbf{N}\mathbf{x}^{(\ell)} + \mathbf{b}$ **end**

Choices

	By points	By blocks
Jacobi	$\mathbf{M} = \mathbf{D}$	$\mathbf{M} = \mathbf{D}^{\text{blk}}$
Gauss-Seidel	$\mathbf{M} = \mathbf{D} + \mathbf{L}$	$\mathbf{M} = \mathbf{D}^{\text{blk}} + \mathbf{L}^{\text{blk}}$

$$\underbrace{\begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix}}_{\mathbf{A}} = \underbrace{\begin{bmatrix} \times & & \\ & \times & \\ & & \times \end{bmatrix}}_{\mathbf{D}} + \underbrace{\begin{bmatrix} & & \\ \times & & \\ \times & \times & \end{bmatrix}}_{\mathbf{L}} + \underbrace{\begin{bmatrix} & \times & \times \\ & & \times \\ & & \end{bmatrix}}_{\mathbf{U}}$$

We consider a regular decomposition: $\boxed{\mathbf{A} = \mathbf{M} - \mathbf{N}}$ where $\mathbf{M} \in \mathbb{R}^{n \times n}$ is inversible.

Stationary method *with relaxation*

$\mathbf{x}^{(0)} \in \mathbb{C}^n$

for $\ell = 0, 1, \ldots$ **do**

  $\mathbf{M}\tilde{\mathbf{x}} = \mathbf{N}\mathbf{x}^{(\ell)} + \mathbf{b}$

  $\mathbf{x}^{(\ell+1)} = \omega\tilde{\mathbf{x}} + (1-\omega)\mathbf{x}^{(\ell)}$ (ω is a real parameter)

end

Choices

	By points	By blocks
Jacobi over relaxation (JOR)	$\mathbf{M} = \mathbf{D}$	$\mathbf{M} = \mathbf{D}^{\text{blk}}$
Successive over relaxation (SOR)	$\mathbf{M} = \mathbf{D} + \mathbf{L}$	$\mathbf{M} = \mathbf{D}^{\text{blk}} + \mathbf{L}^{\text{blk}}$

$$\underbrace{\begin{bmatrix} [\times] & [\times] & [\times] \\ [\times] & [\times] & [\times] \\ [\times] & [\times] & [\times] \end{bmatrix}}_{\mathbf{A}^{\text{blk}}} = \underbrace{\begin{bmatrix} [\times] & & \\ & [\times] & \\ & & [\times] \end{bmatrix}}_{\mathbf{D}^{\text{blk}}} + \underbrace{\begin{bmatrix} & & \\ [\times] & & \\ [\times] & [\times] & \end{bmatrix}}_{\mathbf{L}^{\text{blk}}} + \underbrace{\begin{bmatrix} & [\times] & [\times] \\ & & [\times] \\ & & \end{bmatrix}}_{\mathbf{U}^{\text{blk}}}$$

Convergence of stationary methods

- Convergence of and only if $\rho(\mathbf{B}) < 1$ with $\mathbf{B} = \mathbf{M}^{-1}\mathbf{N}$.
- If $\mathbf{A}$ is a strict diagonal dominant matrix (*i.e.* $|a_{ii}| > \sum_{j \neq i} |a_{ij}|, \ \forall i$)
  - Jacobi converges
  - GS converges
  - SOR converges if $0 < \omega \leq 1$
- Si $\mathbf{A}$ is a symmetric positive definite matrix (*i.e.* $\mathbf{A} = \mathbf{A}^*$ et $\left(\mathbf{Ax}, \mathbf{x}\right) > 0, \ \forall \mathbf{x} \neq 0$)
  - Jacobi converges if $(2\mathbf{D} - \mathbf{A})$ is a symmetric positive definite matrix
  - GS converges
  - SOR converges if and only if $0 < \omega < 2$

What we generally expect for the convergence rates:

Jacobi < Gauss-Seidel < SOR

By points < By blocks

Jacobi method $(\mathbf{M} = \mathbf{D}$ and $\mathbf{N} = -(\mathbf{L}+\mathbf{U}))$
$\mathbf{x}^{(0)} \in \mathbb{R}^n$ **for** $\ell = 0, 1, \ldots$ **do** $\mathbf{D}\mathbf{x}^{(\ell+1)} = \mathbf{b} - (\mathbf{L}+\mathbf{U})\mathbf{x}^{(\ell)}$ **end**

Jacobi method *(rewriting)*

$x_i^{(0)} \in \mathbb{R}$ for $i = 1 \ldots n$
for $\ell = 0, 1, \ldots$ **do**
    for $i = 1 \ldots n$ **do**
        $x_i^{(\ell+1)} = a_{ii}^{-1}\left(b_i - \sum_{i \neq j} a_{ij} x_j^{(\ell)}\right)$
    end
end

Discussion

- Matrix-vector product with dense matrice $(\mathbf{L}+\mathbf{U})$ (BLAS in //)
- Linear combinations because $\mathbf{D}$ is diagonal (Lin. combi. in //)
- The iterations of the interior loop are independent.

Gauss-Seidel method ($\mathbf{M} = \mathbf{L} + \mathbf{D}$ and $\mathbf{N} = -\mathbf{U}$)
$\mathbf{x}^{(0)} \in \mathbb{R}^n$ **for** $\ell = 0, 1, \ldots$ **do** $(\mathbf{D} + \mathbf{L})\mathbf{x}^{(\ell+1)} = \mathbf{b} - \mathbf{U}\mathbf{x}^{(\ell)} \quad \Leftrightarrow \mathbf{D}\mathbf{x}^{(\ell+1)} = \mathbf{b} - \mathbf{L}\mathbf{x}^{(\ell+1)} - \mathbf{U}\mathbf{x}^{(\ell)}$ **end**

Gauss-Seidel method *(rewritting)*

$x_i^{(0)} \in \mathbb{R}$ for $i = 1 \ldots n$
 for $\ell = 0, 1, \ldots$ **do**
 for $i = 1 \ldots n$ **do**
 $x_i^{(\ell+1)} = a_{ii}^{-1} \left(b_i - \sum_{j<i} a_{ij} x_j^{(\ell+1)} - \sum_{i<j} a_{ij} x_j^{(\ell)} \right)$
 end
 end

Discussion

- For each ℓ, solution of a inferior triangular system (Descent in //)
- The interations of the interior loop are dependant:
  For each i, the solution is updated by using the last available values.

Block Jacobi/Gauss-Seidel methods *(rewritting)*

$\mathbf{x}_I^{(0)} \in \mathbb{R}^{n_I}$ for $I = 1 \dots n^{\mathrm{blk}}$

for $\ell = 0, 1, \dots$ **do**

  for $I = 1 \dots n^{\mathrm{blk}}$ **do**

$$\mathbf{A}_{II}\mathbf{x}_I^{(\ell+1)} = \mathbf{b}_I - \textstyle\sum_{I \neq J} \mathbf{A}_{IJ}\mathbf{x}_J^{(\ell)} \qquad \textit{if block Jacobi}$$

$$\mathbf{A}_{II}\mathbf{x}_I^{(\ell+1)} = \mathbf{b}_I - \textstyle\sum_{J<I} \mathbf{A}_{IJ}\mathbf{x}_J^{(\ell+1)} - \textstyle\sum_{I<J} \mathbf{A}_{IJ}\mathbf{x}_J^{(\ell)} \qquad \textit{if block G.-S.}$$

  end

end

Discussion

- Interior loop over n^{blk} blocks of $\mathbf{x}^{(\ell)}$, with $n^{\mathrm{blk}} \leq n$.
- Jacobi: Matrix-vector product with dense matrix $(\mathbf{L}^{\mathrm{blk}} + \mathbf{U}^{\mathrm{blk}})$ (BLAS in //)
- Jacobi: Solution of a block diagonal system (Blocks solved in //)
- G.-S.: Solution of a block inferior triangular system (Descent in //)

Iterative methods for linear systems

Stationary methods

System arising from a finite difference discretization

Definition of the problem

The field $u(x, y)$ is governed by the Poisson equation on a square domain:

$$\begin{cases} \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = f(x, y), & \text{for } (x, y) \in \Omega =]a, b[\times]a, b[, \\ u = 0, & \text{for } (x, y) \in \partial\Omega. \end{cases}$$

Discretization and numerical scheme

The problem is discretized on a regular grid:

- Discretization points: $(x_i, y_j) = (a + ih, a + jh)$ $(i, j = 0, \ldots, n + 1)$
- Spatial step: $h = (b - a)/(n + 1)$
- Approximate field: $u_{i,j} \approx u(x_i, y_j)$

We consider a standard finite difference scheme with a **5-point stencil**:

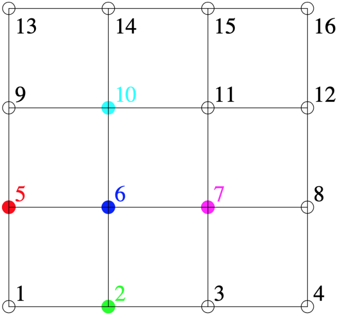
$$\boxed{\frac{1}{h^2}\left(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}\right) = f_{i,j}} \quad (i, j = 1, \ldots, n)$$

with $u_{i,j} = 0$ $(i \text{ and/or } j \in \{0, n + 1\})$ and $f_{i,j} = f(x_i, y_j)$.

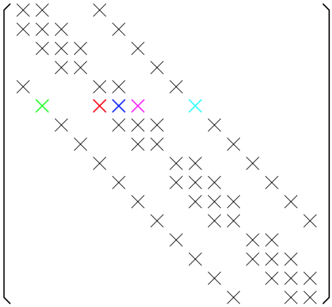
Asymptotic accuracy: $\mathcal{O}(h^2)$

$$\begin{cases} \frac{1}{h^2}\left(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}\right) = f_{i,j} & (i,j = 1,\ldots,n) \\ u_{i,j} = 0 & (i \text{ and/or } j \in \{0, n+1\}) \end{cases}$$

Matrix representation of the problem: $\boxed{\mathbf{A}\mathbf{u} = \mathbf{f}}$ $\quad \mathbf{A} \in \mathbb{R}^{n^2 \times n^2} \quad \mathbf{u}, \mathbf{f} \in \mathbb{R}^{n^2}$



Finite difference grid



Matrix of the system

Finite difference scheme — Description [3/3]

$$\begin{cases} \frac{1}{h^2}\left(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}\right) = f_{i,j} & (i, j = 1, \ldots, n) \\ u_{i,j} = 0 & (i \text{ and/or } j \in \{0, n+1\}) \end{cases}$$

Matrix representation of the problem: $\boxed{\mathbf{Au} = \mathbf{f}}$

$$\mathbf{A} = \frac{1}{h^2}\begin{bmatrix} -4 & 1 & & & 1 & & \\ 1 & -4 & 1 & & & \ddots & \\ & \ddots & \ddots & \ddots & & & 1 \\ & & & & & & \\ 1 & & & \ddots & \ddots & \ddots & \\ & \ddots & & & 1 & -4 & 1 \\ & & 1 & & & 1 & -4 \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} u_{1,1} \\ u_{1,2} \\ \vdots \\ u_{i,j} \\ \vdots \\ u_{n,n-1} \\ u_{n,n} \end{bmatrix} \qquad \mathbf{f} = \begin{bmatrix} f_{1,1} \\ f_{1,2} \\ \vdots \\ f_{i,j} \\ \vdots \\ f_{n,n-1} \\ f_{n,n} \end{bmatrix}$$

Properties of $\mathbf{A}$: Pentadiagonal matrix + Symmetric positive definite matrix

Finite difference scheme — Solution with Jacobi [1/3]

$$\boxed{\mathbf{M}\mathbf{x}^{(\ell+1)} = \mathbf{N}\mathbf{x}^{(\ell)} + \mathbf{b}}$$

with $$\mathbf{A} = (\mathbf{M} - \mathbf{N}) = -\frac{4}{h^2}\mathbf{I} - \frac{1}{h^2}\begin{bmatrix} 0 & -1 & & & -1 & & \\ -1 & 0 & -1 & & & \ddots & \\ & \ddots & \ddots & \ddots & & & -1 \\ -1 & & & \ddots & \ddots & \ddots & \\ & \ddots & & & -1 & 0 & -1 \\ & & -1 & & & -1 & 0 \end{bmatrix}$$

Sequential algorithm with Jacobi

$u_{i,j}^{(0)} \in \mathbb{R}$ for $i, j = 1 \dots n$

for $\ell = 0, 1, \dots$ **do**

  for $i = 1, \dots, n$ **do**

    for $j = 1, \dots, n$ **do**

$$-\frac{4}{h^2} u_{i,j}^{(\ell+1)} = -\frac{1}{h^2}\left(u_{i+1,j}^{(\ell)} + u_{i-1,j}^{(\ell)} + u_{i,j+1}^{(\ell)} + u_{i,j-1}^{(\ell)}\right) + f_{i,j}$$

    end

  end

end

Finite difference scheme — Solution with Jacobi [2/3]

$$\boxed{\mathbf{M}\mathbf{x}^{(\ell+1)} = \mathbf{N}\mathbf{x}^{(\ell)} + \mathbf{b}}$$

$$\text{avec } \mathbf{A} = (\mathbf{M} - \mathbf{N}) = -\frac{4}{h^2}\mathbf{I} - \frac{1}{h^2}\begin{bmatrix} 0 & -1 & & & -1 & & \\ -1 & 0 & -1 & & & \ddots & \\ & \ddots & \ddots & \ddots & & & -1 \\ -1 & & & \ddots & \ddots & \ddots & \\ & \ddots & & & -1 & 0 & -1 \\ & & -1 & & & -1 & 0 \end{bmatrix}$$

Sequential algorithm with Jacobi *(rewritting)*

$u_{i,j}^{(0)} \in \mathbb{R}$ for $i, j = 1 \ldots n$

for $\ell = 0, 1, \ldots$ **do**

  for $i = 1, \ldots, n$ **do**

    for $j = 1, \ldots, n$ **do**

$$u_{i,j}^{(\ell+1)} = \frac{1}{4}\left(u_{i+1,j}^{(\ell)} + u_{i-1,j}^{(\ell)} + u_{i,j+1}^{(\ell)} + u_{i,j-1}^{(\ell)}\right) - \frac{h^2}{4} f_{i,j}$$

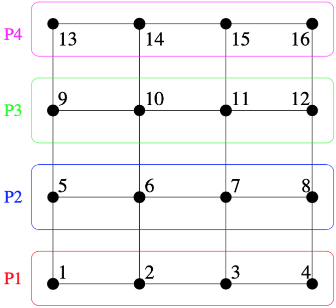
    end

  end

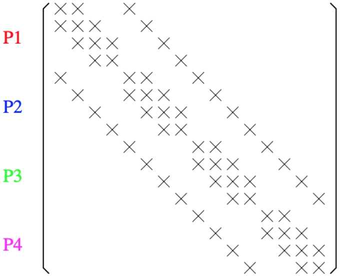
end

Parallelization

- Domain partitioning and matrix partitioning:



Finite difference grid



Matrix of the system

- Analysis of communications:
  - Each process has to communicate with both neighbors.
  - Sending/reception of n updated unknowns with each neighbor.
  - Only (local) point-to-point communications.

Parallel algorithm with Jacobi *(1D partition)*

On each process p:

$u_{i,j}^{(0)} \in \mathbb{R}$ for $i = i_{\text{start},p}, \dots, i_{\text{end},p}$ and $j = 1, \dots, n$

for $\ell = 0, 1, \dots$ **do**

  Communication phase:

  - If $p > 0$: send $u_{i_{\text{start},p},\star}$ to process $p-1$
  - If $p > 0$: receive $u_{i_{\text{start},p}-1,\star}$ from process $p-1$
  - If $p < (P-1)$: send $u_{i_{\text{end},p},\star}$ to process $p+1$
  - If $p < (P-1)$: receive $u_{i_{\text{end},p}+1,\star}$ from process $p+1$

  \\ Update of unknowns

  for $i = i_{\text{start},p}, \dots, i_{\text{end},p}$ **do**

    for $j = 1, \dots, n$ **do**

$$u_{i,j}^{(\ell+1)} = \frac{1}{4}\left(u_{i+1,j}^{(\ell)} + u_{i-1,j}^{(\ell)} + u_{i,j+1}^{(\ell)} + u_{i,j-1}^{(\ell)}\right) - \frac{h^2}{4} f_{i,j}$$

    end

  end

end

(In the communications, $\star$ indicates that the whole line is sent.)

$$\boxed{\mathbf{M}\mathbf{x}^{(\ell+1)} = \mathbf{N}\mathbf{x}^{(\ell)} + \mathbf{b}}$$

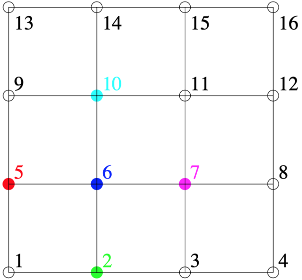
with $\mathbf{A} = (\mathbf{M} - \mathbf{N}) = \frac{1}{h^2}\begin{bmatrix} -4 & & & & & & \\ 1 & -4 & & & & & \\ & \ddots & \ddots & & & & \\ 1 & & & \ddots & \ddots & & \\ & \ddots & & & 1 & -4 & \\ & & 1 & & & 1 & -4 \end{bmatrix} - \frac{1}{h^2}\begin{bmatrix} 0 & -1 & & & -1 & & \\ & 0 & -1 & & & \ddots & \\ & & \ddots & \ddots & & & -1 \\ & & & & \ddots & \ddots & \\ & & & & & 0 & -1 \\ & & & & & & 0 \end{bmatrix}$

Sequential algorithm with Gauss-Seidel
$u^{(0)}_{i,j} \in \mathbb{R}$ for $i, j = 1 \dots n$
for $\ell = 0, 1, \dots$ **do**
for $i = 1, \dots, n$ **do**
for $j = 1, \dots, n$ **do**
$u^{(\ell+1)}_{i,j} = \frac{1}{4}\left(u^{(\ell)}_{i+1,j} + u^{(\ell+1)}_{i-1,j} + u^{(\ell)}_{i,j+1} + u^{(\ell+1)}_{i,j-1}\right) - \frac{h^2}{4} f_{i,j}$
end
end
end

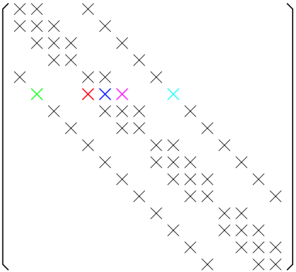
Finite difference scheme — Solution with Gauss-Seidel [2/6]

Parallelization of the scheme

The Gauss-Seidel method uses the last available values for the update.
$\Rightarrow$ This procedure is *(a priori)* sequential



Finite difference grid

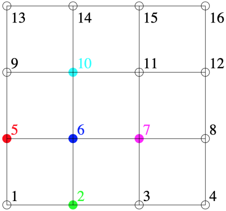


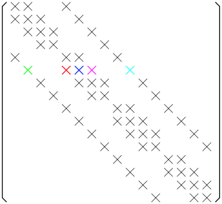
Matrix of the system

Idea: change the order of evaluation of the unknowns *(i.e. permutation of lines)*
to make this procedure parallelizable

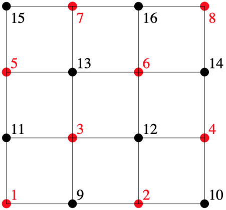
Parallelization of the scheme *(with coloring)*

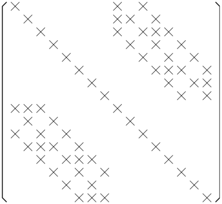
Natural numbering





Numbering with *“red-black”* coloring





The unknowns associated to a given color can be updated in parallel.

Sequential algorithm with Gauss-Seidel *(with red-black coloring)*

$u_{i,j}^{(0)} \in \mathbb{R}$ for $i, j = 1 \dots n$

for $\ell = 0, 1, \dots$ **do**

  \\\\ Update of red unknowns

  for $i = 1, \dots, n$ **do**

    for $j = 1, \dots, n$ **do**

      If (i, j) red: $u_{i,j}^{(\ell+1)} = \frac{1}{4}\left(u_{i+1,j}^{(\ell)} + u_{i-1,j}^{(\ell)} + u_{i,j+1}^{(\ell)} + u_{i,j-1}^{(\ell)}\right) - \frac{h^2}{4} f_{i,j}$

    end

  end

  \\\\ Update of black unknowns

  for $i = 1, \dots, n$ **do**

    for $j = 1, \dots, n$ **do**

      If (i, j) black:

      $u_{i,j}^{(\ell+1)} = \frac{1}{4}\left(u_{i+1,j}^{(\ell+1)} + u_{i-1,j}^{(\ell+1)} + u_{i,j+1}^{(\ell+1)} + u_{i,j-1}^{(\ell+1)}\right) - \frac{h^2}{4} f_{i,j}$

    end

  end

end

Parallel algorithm with Gauss-Seidel *(with red-black coloring)*

For each process p:

$u_{i,j}^{(0)} \in \mathbb{R}$ for $i = i_{\text{start},p}, \ldots, i_{\text{end},p}$ and $j = 1, \ldots, n$

for $\ell = 0, 1, \ldots$ **do**

  Communication phase *(as for Jacobi)*

  \\ Update of red unknowns

  for $i = i_{\text{start},p}, \ldots, i_{\text{end},p}$ **do**

    for $j = 1, \ldots, n$ **do**

      If (i, j) red: $u_{i,j}^{(\ell+1)} = \frac{1}{4}\left(u_{i+1,j}^{(\ell)} + u_{i-1,j}^{(\ell)} + u_{i,j+1}^{(\ell)} + u_{i,j-1}^{(\ell)}\right) - \frac{h^2}{4} f_{i,j}$

    end

  end

  Communication phase *(as for Jacobi)*

  \\ Update of black unknowns

  for $i = i_{\text{start},p}, \ldots, i_{\text{end},p}$ **do**

    for $j = 1, \ldots, n$ **do**

      If (i, j) black: $u_{i,j}^{(\ell+1)} = \frac{1}{4}\left(u_{i+1,j}^{(\ell+1)} + u_{i-1,j}^{(\ell+1)} + u_{i,j+1}^{(\ell+1)} + u_{i,j-1}^{(\ell+1)}\right) - \frac{h^2}{4} f_{i,j}$

    end

  end

end

Comments on parallelization strategies with coloring

- Basic idea:
  - Each color = Unknowns updated in parallel
  - Communication phase between each color
- Different numbering, so . . .
  - Different algorithm, but still Gauss-Seidel
  - Different numerical solution, but scheme with the same properties
- Some extensions:
  - If larger stencil $\rightarrow$ Coloring with more colors
  - If unstructured mesh $\rightarrow$ Algorithms for automatic coloring

Ressources

- *Méthodes Numériques : Algorithmes, analyse et applications*
  A. Quarteroni, R. Sacco, F. Saleri (2007), Springer
- *Calcul scientifique parallèle*
  F. Magoulès et F.-X. Roux (2017), Dunod
- *Calcul scientifique parallèle*
  P. Ciarlet and E. Jamelot, polycopié de cours