Linear System

吴侃

12.17

Outline

- Direct Method
  - Gaussian elimination and LU decomposition
  - Cholesky decomposition
- Iterative Method
  - Basic stationary methods
  - Smooth mulitgrid Solver
  - Krylov subspace methods

Problem

- Solve Linear Equation

$$A\mathbf{x} = \mathbf{b},$$

Gaussian elimination and LU decomposition

1. *LU decomposition*: $A = LU$;
2. *Forward substitution*: solve $L\mathbf{y} = \mathbf{b}$;
3. *Backward substitution*: solve $U\mathbf{x} = \mathbf{y}$.

Forming the LU decomposition takes $\mathcal{O}(n^3)$ floating point operations for a general dense $A \in \mathbb{R}^{n\times n}$. It takes $\mathcal{O}(n^2)$ floating point operations to perform backward or forward substitutions.

Gaussian elimination and LU decomposition

Algorithm 1 The LU decomposition of a matrix A. Upon exit, the entries of A have been overwritten with the entries of L (below the main diagonal) and the entries of U (main diagonal and above). The diagonal entries of L are all equal to 1.

---

for $k = 1, \ldots, n-1$ **do**
    for $i = k+1, \ldots, n$ **do**
        $a_{i,k} = \frac{a_{i,k}}{a_{k,k}}$
        for $j = k+1, \ldots, n$ **do**
            $a_{i,j} = a_{i,j} - a_{i,k}a_{k,j}$
        end for
    end for
end for

Cholesky decomposition

- special method for symmetric positive definiteness matrix

$$A = FF^\top,$$

Cholesky decomposition

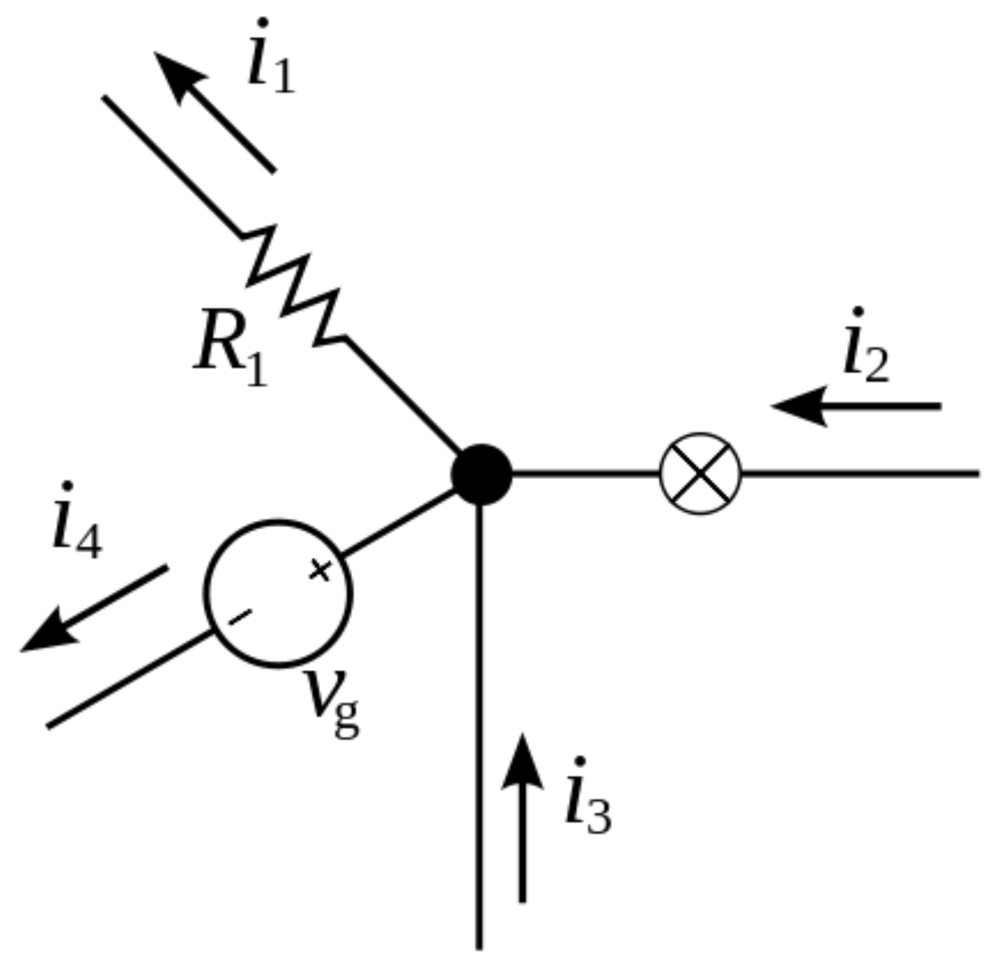
Algorithm 2 The Cholesky decomposition of a symmetric positive definite matrix A. Upon exit, the entries of A on its diagonal and below it have been overwritten with the entries of the lower triangular Cholesky factor F.

for $k = 1, \ldots, n$ **do**
  $a_{k,k} = \sqrt{a_{k,k}}$
  for $i = k+1, \ldots, n$ **do**
    $a_{i,k} = \frac{a_{i,k}}{a_{k,k}}$
  end for
  for $j = k+1, \ldots, n$ **do**
    for $i = j, \ldots, n$ **do**
      $a_{i,j} = a_{i,j} - a_{i,k} a_{k,j}$
    end for
  end for
end for

Analysis of an Electrical Network

Kirchhoff's circuit laws

$$\sum_{k=1}^{n} I_k = 0$$



1D Smooth filter

Let v be the sum of a smooth 1D signal u and IID Gaussian noise e:

$$v \; = \; u + e \; ,$$

where $u = (u_1, ..., u_N)$, $v = (v_1, ..., v_N)$, and $e = (e_1, ..., e_N)$.

- minimize the following objective

$$E(u) \; = \; \sum_{n=1}^{N} (u_n - v_n)^2 + \lambda \sum_{n=1}^{N-1} (u_{n+1} - u_n)^2$$

1D Smooth filter

- minimize the following objective

$$E(u) = \sum_{n=1}^{N} (u_n - v_n)^2 + \lambda \sum_{n=1}^{N-1} (u_{n+1} - u_n)^2$$

- calculate the gradient

$$\frac{\partial E(u)}{\partial u_n} = 2\,(u_n - v_n) + 2\lambda\,(-u_{n-1} + 2u_n - u_{n+1})$$

$$u_n + \lambda\,(-u_{n-1} + 2u_n - u_{n+1}) = v_n$$

1D Smooth filter

$$\begin{pmatrix} 1+\lambda & -\lambda & 0 & 0 & \dots & 0 \\ -\lambda & 1+2\lambda & -\lambda & 0 & \dots & 0 \\ 0 & -\lambda & 1+2\lambda & -\lambda & \dots & 0 \\ & & & & & \\ & & \ddots & \ddots & \ddots & \\ & & & & & \\ 0 & \dots & & 0 & -\lambda & 1+\lambda \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \\ \vdots \\ \\ u_N \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \\ \vdots \\ \\ v_N \end{pmatrix}$$

O(n) by using Gaussian Elimination

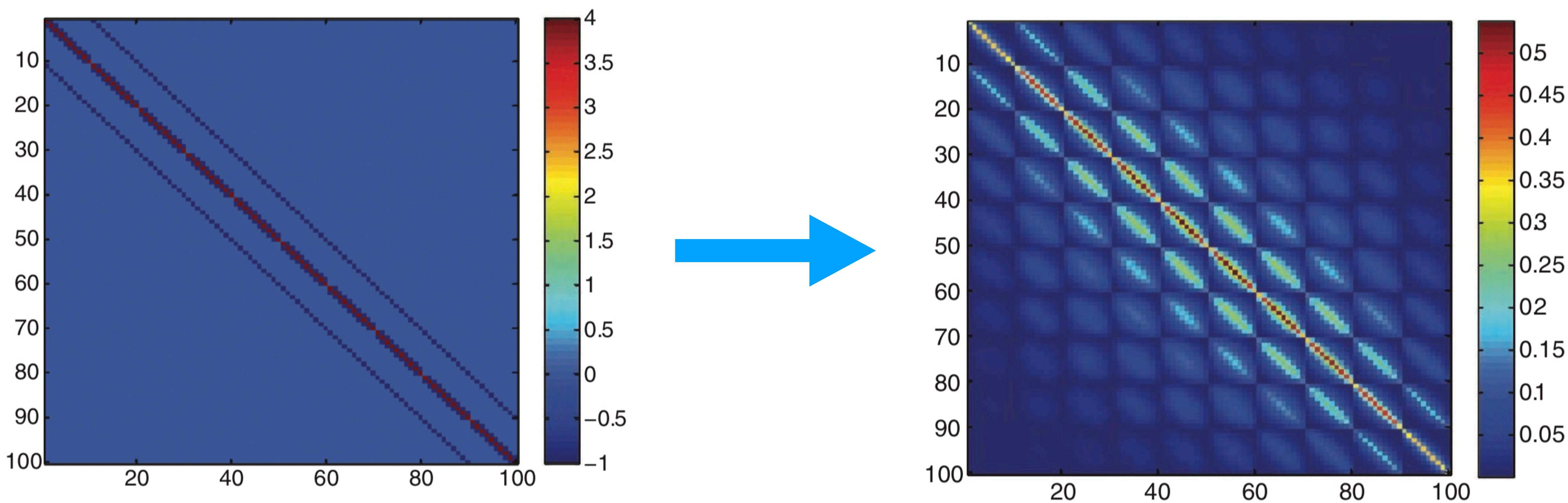
Nonzero structure and sparsity pattern

- Narrow-banded matrices
  - tridiagonal matrices O(n) for Gaussian Elimination

$$T = \begin{pmatrix} a_1 & b_1 & & & \\ c_1 & a_2 & b_2 & & \\ & c_2 & \ddots & \ddots & \\ & & \ddots & \ddots & b_{n-1} \\ & & & c_{n-1} & a_n \end{pmatrix}$$

nonzero structure and sparsity pattern

- the discrete two-dimensional Laplacian arising from discretization of the Poisson equation on a uniform mesh on a square domain
    - The matrix has approximately 5n nonzero entries
    - but the Cholesky factor contains about O(n*sqrt(n))
    - the decomposition is O(n^2) floating-point operations



Basic stationary methods

Basic stationary methods

Given $A\mathbf{x} = \mathbf{b}$, suppose $A = M - N$. We have $M\mathbf{x} = N\mathbf{x} + \mathbf{b}$, which may lead to the iteration

$$M\mathbf{x}_{k+1} = N\mathbf{x}_k + \mathbf{b}\,.$$

This is a *stationary* or a *fixed-point* iteration.

Jacobi Method

$$A = D + R \qquad \text{where} \qquad D = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} \text{ and } R = \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ a_{21} & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & 0 \end{bmatrix}$$

$$\mathbf{x}^{(k+1)} = D^{-1}(\mathbf{b} - R\mathbf{x}^{(k)}),$$

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right), \quad i = 1, 2, \ldots, n.$$

Gauss–Seidel Method

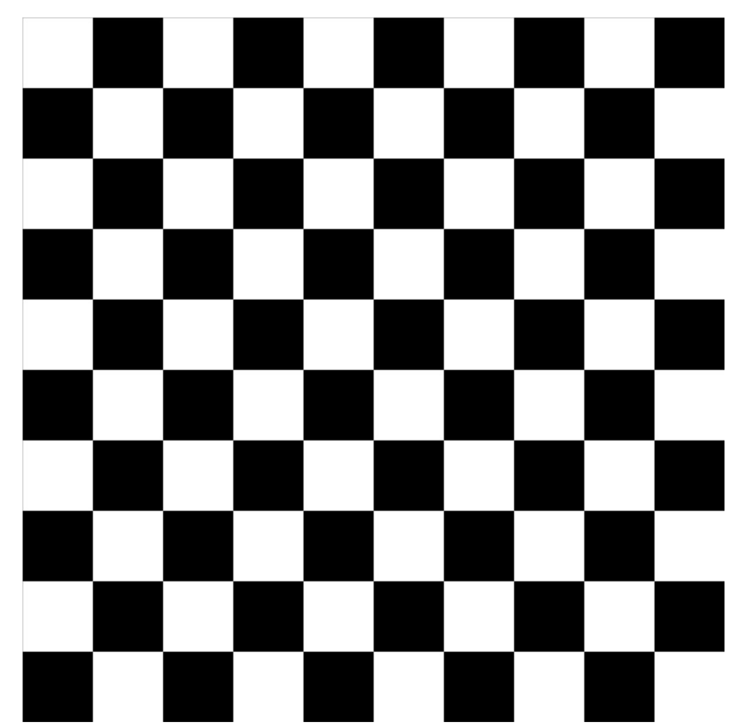
$$A = L_* + U \qquad \text{where} \qquad L_* = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad U = \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ 0 & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

$$\mathbf{x}^{(k+1)} = L_*^{-1}(\mathbf{b} - U\mathbf{x}^{(k)}).$$

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right), \quad i = 1, 2, \dots, n.$$

Basic stationary methods

- Jacobi Method: Highly Parallelizable
- Gauss–Seidel Method: Faster Convergence Speed
    - In some cases(eg. processing Images), Gauss–Seidel Method is also highly parallelizable
        - Black and white coloring



$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right), \quad i = 1, 2, \ldots, n.$$

Convergence of iterative methods

- convergence is governed by the eigenvalues of the iteration matrix

$$T = M^{-1}N = I - M^{-1}A$$

- A necessary and sufficient condition for convergence is that the eigenvalues of T are all smaller than 1

- The smaller the maximal magnitude of the eigenvalues, the faster the convergence

    - If some of the eigenvalues of T are very close to 1, then we may experience trouble. Unfortunately, this is often the case in many applications, particularly in the solution of discretized PDE.

Krylov subspace methods

- Find solution in Krylov sub space

$$\mathbf{x}_k \in \mathbf{x}_0 + \mathcal{K}^k(A;\mathbf{r}_0) \equiv \mathbf{x}_0 + \mathrm{span}\{\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, \ldots, A^{k-1}\mathbf{r}_0\}$$

- Conjugate Gradient
  - Minimize the following quadratic function

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} - \mathbf{x}^\mathsf{T}\mathbf{b}, \qquad \mathbf{x} \in \mathbf{R}^n\,.$$

$$\nabla^2 f(\mathbf{x}) = \mathbf{A}\,, \qquad \nabla f(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}\,.$$

Direct Conjugate Gradient Method

$$\mathbf{x}_* = \sum_{i=1}^{n} \alpha_i \mathbf{p}_i.$$

Based on this expansion we calculate:

$$\mathbf{A}\mathbf{x}_* = \sum_{i=1}^{n} \alpha_i \mathbf{A}\mathbf{p}_i.$$

Left-multiplying by $\mathbf{p}_k^\mathsf{T}$:

$$\mathbf{p}_k^\mathsf{T}\mathbf{A}\mathbf{x}_* = \sum_{i=1}^{n} \alpha_i \mathbf{p}_k^\mathsf{T}\mathbf{A}\mathbf{p}_i,$$

substituting $\mathbf{A}\mathbf{x}_* = \mathbf{b}$ and $\mathbf{u}^\mathsf{T}\mathbf{A}\mathbf{v} = \langle \mathbf{u}, \mathbf{v} \rangle_\mathbf{A}$:

$$\mathbf{p}_k^\mathsf{T}\mathbf{b} = \sum_{i=1}^{n} \alpha_i \langle \mathbf{p}_k, \mathbf{p}_i \rangle_\mathbf{A},$$

$$\forall i \neq k : \langle \mathbf{p}_k, \mathbf{p}_i \rangle_\mathbf{A} = 0$$

$$\langle \mathbf{p}_k, \mathbf{b} \rangle = \alpha_k \langle \mathbf{p}_k, \mathbf{p}_k \rangle_\mathbf{A}$$

$$\alpha_k = \frac{\langle \mathbf{p}_k, \mathbf{b} \rangle}{\langle \mathbf{p}_k, \mathbf{p}_k \rangle_\mathbf{A}}$$

Conjugate Gradient Method

$$\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k.$$

$$\mathbf{p}_k = \mathbf{r}_k - \sum_{i<k} \frac{\mathbf{p}_i^\top \mathbf{A}\mathbf{r}_k}{\mathbf{p}_i^\top \mathbf{A}\mathbf{p}_i} \mathbf{p}_i$$

$$f(\mathbf{x}_{k+1}) = f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) =: g(\alpha_k)$$

$$g'(\alpha_k) \stackrel{!}{=} 0 \quad \Rightarrow \quad \alpha_k = \frac{\mathbf{p}_k^\top (\mathbf{b} - \mathbf{A}\mathbf{x}_k)}{\mathbf{p}_k^\top \mathbf{A}\mathbf{p}_k}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

$$\alpha_k = \frac{\mathbf{p}_k^\top (\mathbf{b} - \mathbf{A}\mathbf{x}_k)}{\mathbf{p}_k^\top \mathbf{A}\mathbf{p}_k} = \frac{\mathbf{p}_k^\top \mathbf{r}_k}{\mathbf{p}_k^\top \mathbf{A}\mathbf{p}_k}$$

Conjugate Gradient Method

The Algorithm

$\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$

$\mathbf{p}_0 := \mathbf{r}_0$

$k := 0$

repeat

$$\alpha_k := \frac{\mathbf{r}_k^\mathsf{T}\mathbf{r}_k}{\mathbf{p}_k^\mathsf{T}\mathbf{A}\mathbf{p}_k}$$

$\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_k \mathbf{p}_k$

$\mathbf{r}_{k+1} := \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k$

if r_{k+1} is sufficiently small, then exit loop

$$\beta_k := \frac{\mathbf{r}_{k+1}^\mathsf{T}\mathbf{r}_{k+1}}{\mathbf{r}_k^\mathsf{T}\mathbf{r}_k}$$

$\mathbf{p}_{k+1} := \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k$

$k := k + 1$

end repeat

The result is $\mathbf{x}_{k+1}$

Conjugate Gradient Method

Convergence theorem

Define a subset of polynomials as

$$\Pi_k^* := \{\, p \in \Pi_k \ : \ p(0) = 1 \,\} \,,$$

where Π_k is the set of polynomials of maximal degree k.

Let $(\mathbf{x}_k)_k$ be the iterative approximations of the exact solution $\mathbf{x}_*$, and define the errors as $\mathbf{e}_k := \mathbf{x}_k - \mathbf{x}_*$. Now, the rate of convergence can be approximated as [5]

$$\begin{aligned} \|\mathbf{e}_k\|_{\mathbf{A}} &= \min_{p \in \Pi_k^*} \|p(\mathbf{A})\mathbf{e}_0\|_{\mathbf{A}} \\ &\le \min_{p \in \Pi_k^*} \max_{\lambda \in \sigma(\mathbf{A})} |p(\lambda)| \, \|\mathbf{e}_0\|_{\mathbf{A}} \\ &\le 2 \left(\frac{\sqrt{\kappa(\mathbf{A})} - 1}{\sqrt{\kappa(\mathbf{A})} + 1} \right)^k \|\mathbf{e}_0\|_{\mathbf{A}} \,, \end{aligned}$$

$$\kappa(A) = \|A\| \|A^{-1}\|$$

where $\sigma(\mathbf{A})$ denotes the spectrum, and $\kappa(\mathbf{A})$ denotes the condition number.

Note, the important limit when $\kappa(\mathbf{A})$ tends to ∞

$$\frac{\sqrt{\kappa(\mathbf{A})} - 1}{\sqrt{\kappa(\mathbf{A})} + 1} \approx 1 - \frac{2}{\sqrt{\kappa(\mathbf{A})}} \quad \text{for} \quad \kappa(\mathbf{A}) \gg 1 \,.$$

This limit shows a faster convergence rate compared to the iterative methods of Jacobi or Gauss-Seidel which scale as $\approx 1 - \frac{2}{\kappa(\mathbf{A})}$.

Krylov subspace methods

- CG proceeds by defining special search directions and minimizing

$$\|\mathbf{e}_k\|_A = \sqrt{\mathbf{e}_k^T A \mathbf{e}_k}, \text{ where } \mathbf{e}_k = \mathbf{x} - \mathbf{x}_k$$

$$f(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^{\mathsf{T}}\mathbf{A}\mathbf{x} - \mathbf{x}^{\mathsf{T}}\mathbf{b}, \qquad \mathbf{x} \in \mathbf{R}^n .$$

- We can also minimize the norm of the residual

$$\|\mathbf{b} - A\mathbf{x}_k\|_2$$

  - MINRES or GMRES

Image Operator with Laplacian Equation

$$
\begin{aligned}
\Delta f &= f_{xx} + f_{yy} \\
&= f(x+1,y) - 2f(x,y) + f(x-1,y) + f(x,y+1) \\
&\qquad\qquad\qquad\qquad\qquad\qquad - 2f(x,y) + f(x,y-1) \\
&= f(x{+}1,y) + f(x{-}1,y) + f(x,y{+}1) + f(x,y{-}1) - 4f(x,y)\,.
\end{aligned}
$$

$$
\left[\begin{array}{ccccccc}
1 & 1 & -4 & 1 & 1 & \dots & \dots \\
\dots & 1 & 1 & -4 & 1 & 1 & \dots \\
 & & & \vdots & & & \\
\dots & \dots & 1 & 1 & -4 & 1 & 1 \\
\hline
 & 0 & & | & & I_{K\times K} &
\end{array}\right]
\begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix}
=
\left[\begin{array}{c} | \\ 0 \\ | \\ \hline c_1 \\ \vdots \\ c_k \end{array}\right]
$$

Image Operator with Possion Equation

$$\left[\begin{array}{ccccccc} 1 & 1 & -4 & 1 & 1 & \dots & \dots \\ \dots & 1 & 1 & -4 & 1 & 1 & \dots \\ & & & \vdots & & & \\ \dots & \dots & 1 & 1 & -4 & 1 & 1 \\ \hline \multicolumn{3}{c}{0} & \multicolumn{1}{|c}{} & \multicolumn{3}{c}{I_{K\times K}} \end{array}\right] \left[\begin{array}{c} f_1 \\ \vdots \\ f_n \end{array}\right] = \left[\begin{array}{c} | \\ \operatorname{div}\mathbf{g} \\ | \\ \hline c_1 \\ \vdots \\ c_k \end{array}\right].$$

Poisson Image Matting

$$I = \alpha F + (1 - \alpha)B$$

Poisson Image Matting

$$\nabla I = (F - B)\nabla\alpha + \alpha\nabla F + (1 - \alpha)\nabla B$$

- Assume that F and B is very smooth

$$\nabla\alpha \approx \frac{1}{F - B}\nabla I$$

Poisson Image Matting

- We minimize the following variational problem with dirichlet boundary $\alpha|_{\partial\Omega} = \widehat{\alpha}|_{\partial\Omega}$

$$\alpha^* = \arg\min_{\alpha} \int\int_{p\in\Omega} ||\nabla\alpha_p - \frac{1}{F_p - B_p}\nabla I_p||^2 dp$$

- Use Euler-Lagrange Equation

$$J = \int_a^b F(x, f(x), f'(x))\,\mathrm{d}x \quad \longrightarrow \quad \frac{\partial F}{\partial f} - \frac{\mathrm{d}}{\mathrm{d}x}\frac{\partial F}{\partial f'} = 0$$

Poisson Image Matting

- Use Euler-Lagrange Equation

$$J = \int_a^b F(x, f(x), f'(x))\,\mathrm{d}x \quad \rightarrow \quad \frac{\partial F}{\partial f} - \frac{\mathrm{d}}{\mathrm{d}x}\frac{\partial F}{\partial f'} = 0$$

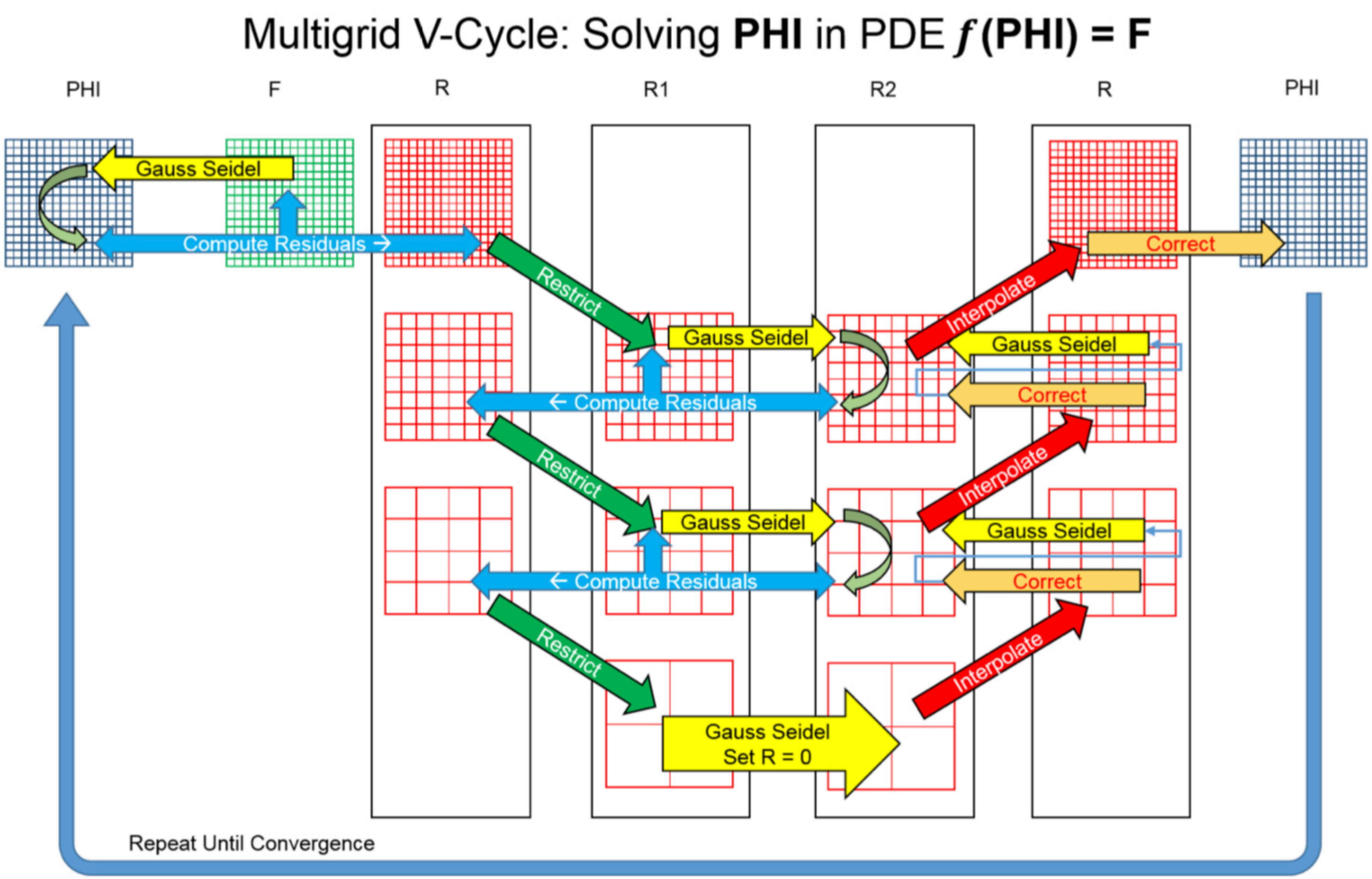
- Poisson Equation

$$\Delta\alpha = div\left(\frac{\nabla I}{F - B}\right)$$

Linear system !!!

Multigrid Method(多重网格法)

- Use Gauss–Seidel as Smoother

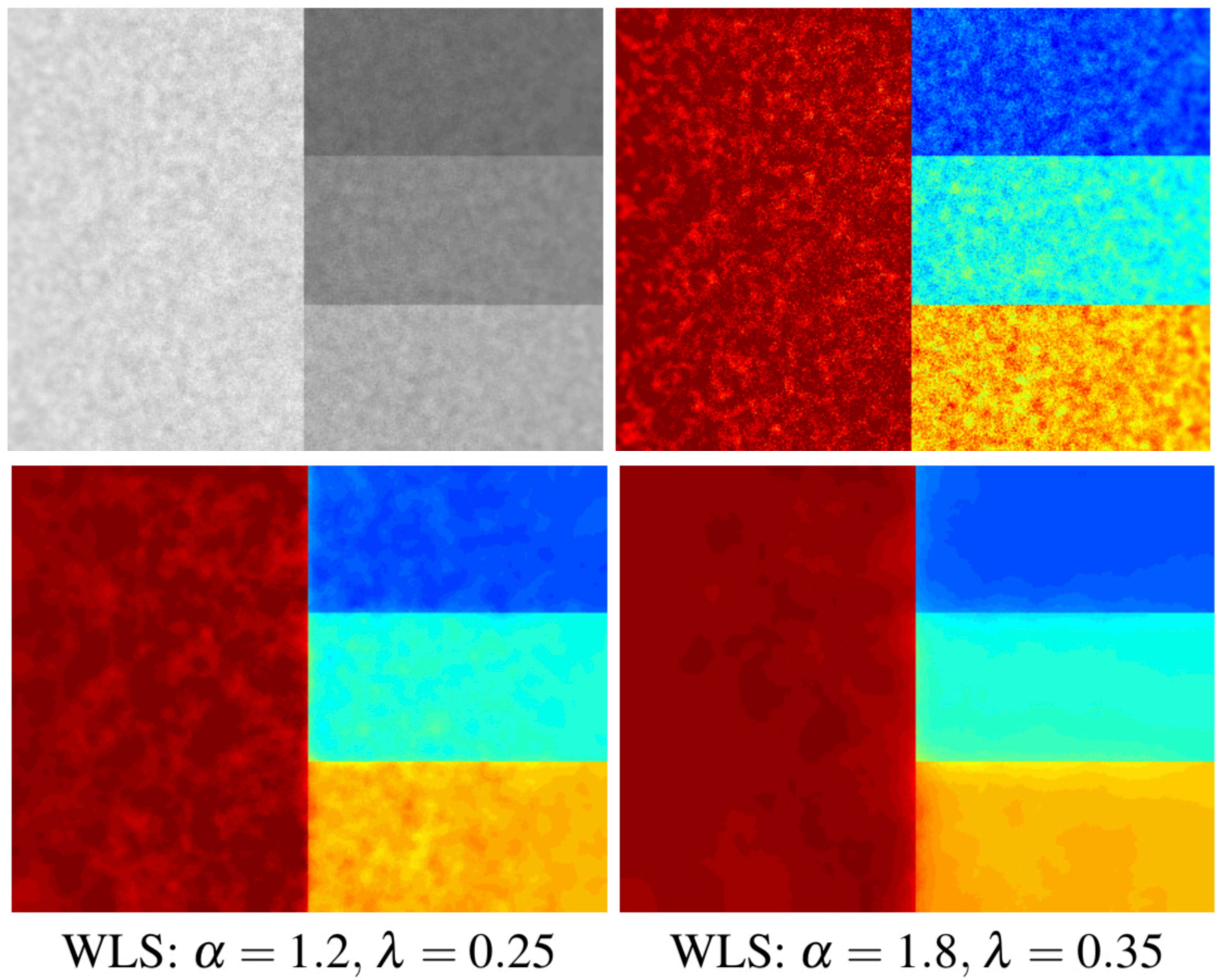


Multigrid Method(多重网格法)

- Use Gauss–Seidel as Smoother
- The Complexity of this algorithm is $O(n)$
- It is widely used in image processing and solving PDE

Weighted Least Squares Filter

- Edge preserving Smoother by WLS



Weighted Least Squares Filter

- Edge preserving Smoother by WLS

$$\sum_p \left(\left(u_p - g_p\right)^2 + \lambda \left(a_{x,p}(g) \left(\frac{\partial u}{\partial x} \right)_p^2 + a_{y,p}(g) \left(\frac{\partial u}{\partial y} \right)_p^2 \right) \right)$$

$$= \quad (u-g)^{\mathrm{T}}(u-g) + \lambda \left(u^{\mathrm{T}} D_x^{\mathrm{T}} A_x D_x u + u^{\mathrm{T}} D_y^{\mathrm{T}} A_y D_y u \right)$$

$$\left(I + \lambda L_g\right) u = g \qquad\qquad L_g = D_x^{\mathrm{T}} A_x D_x + D_y^{\mathrm{T}} A_y D_y$$

$$a_{x,p}(g) = \left(\left| \frac{\partial \ell}{\partial x}(p) \right|^{\alpha} + \varepsilon \right)^{-1} \quad a_{y,p}(g) = \left(\left| \frac{\partial \ell}{\partial y}(p) \right|^{\alpha} + \varepsilon \right)^{-1}$$

Weighted Least Squares Filter

- Edge preserving Smoother by WLS

$$(I + \lambda L_g)u = g$$ Linear system !!!

- Use Multigrid Method to solve in $O(H \times W)$ time

which method should we use?

	Dense or Small	symmetric	positive definite
Gaussian Elimination	+		
Cholesky decomposition	+	+	+
Iterative Method	-		
Conjugate gradient	-	+	+
the Bunch–Kaufman algorithm	+	+	-
MINRES	-	+	-
GMRES	-	-	

Combination of these methods

- use Multigrid method to get a good initial solution
- then use conjugate gradient method to refine the solution