Finite and Infinite Dimensional Matrices as Operators

- We have discussed matrices. These are "operators" in finite (n) dimensional linear space.
- You can generalize to infinite dimensional space.

  — e.g., space of "suitable" functions $f(x)$
- $\frac{d}{dx}$ is an operator!
- $\frac{d}{dx}f(x) = g(x)$ is another function.

  — e.g., another operator

$\frac{d}{dx}$ as an Operator and Its Scalar Products

- Consider the Integral

$$I = \int_a^b g(x) \frac{d}{dx} f(x) dx = \langle g | D | f \rangle$$

where D is operator $\frac{d}{dx}$.

- Integrate by parts.

$$\begin{aligned} I = & - \int_a^b f(x) \frac{d}{dx} g(x) \, dx \\ & + [f(x) g(x)]_a^b \\ & = f(b) \, g(b) - f(a) \, g(a) \end{aligned}$$

- The last term

$$[f(x) g(x)]_a^b = f(b) \, g(b) - f(a) \, g(a)$$

- We need to get rid of this surface (endpoint) term.

$i\frac{d}{dx}$ as a Hermitean Operator

- Suppose we consider periodic functions

$$\begin{aligned} f(b) &= f(a) \\ g(b) &= g(a) \end{aligned}$$

— i.e., impose particular boundary conditions

- Then surface term vanishes and

$$\int_a^b g(x)\frac{d}{dx}f(x)\,dx = -\int_a^b f(x)\frac{d}{dx}g(x)\,dx$$

- The operator $L = i\frac{d}{dx}$ with <u>boundary conditions</u> $f(a) = f(b)$ is Hermitean. The i in L cancels minus sign.

- Must include boundary conditions when discussing operators in infinite dimensional space

The Laplacian as a (Matrix) Operator

- $\nabla^2 = \frac{d^2}{dx^2} = -L\,L$
- $\left(\nabla^2\right)^\dagger = -L^\dagger L^\dagger = -L\,L = \nabla^2$
- Note: $(A\,B)^\dagger = B^\dagger A^\dagger$ in general
- Thus, ∇^2 ($\frac{d^2}{dx^2}$ in one dimension) is Hermitean (symmetric) operator

Mapping of Function Spaces to a Finite Number of Dimensions

- In solving differential equations numerically, we take infinite dimensional operators on space of functions and convert them to finite matrix operators in space of n dimensional vectors.
- The different numerical approaches correspond to different ways of mapping the infinite dimensional space to a finite space—there is no unique way and some are much better than others.

Mathematical and Pseudo Code Form of Gauss Seidel Iteration Method

Choose an initial guess $x^{(0)}$ to the solution x.
for $k = 1,2,\ldots$
    for $i = 1,2,\ldots,n$
        $\sigma = 0$
        for $j = 1,2,\ldots,i-1$
            $\sigma = \sigma + a_{i,j}\, x_j^{(k)}$
        end
        for $j = i+1,\ldots,n$
            $\sigma = \sigma + a_{i,j}\, x_j^{(k-1)}$
        end
        $x_i^{(k)} = (b_i - \sigma)\,/a_{i,i}$
    end
    check convergence; continue if necessary
end

Pseudo code for Gauss-Seidel Method

Mathematical (Matrix) Form of Gauss Seidel

- Mathematically, Gauss Seidel is

$$x_i^{(k)} = \frac{\left[b_i - \sum_{j<i} a_{i,j}\, x_j^{(k)} - \sum_{j>i} a_{i,j}\, x_j^{(k-1)} \right]}{a_{i,i}}$$

- In matrix language

$$\underline{x}^{(k)} = (D - L)^{-1} \left(U \underline{x}^{(k-1)} + \underline{b} \right)$$

- where D=Diagonal, L=lower, and U=Upper parts of matrix **A**
- **A**$= D - L - U$

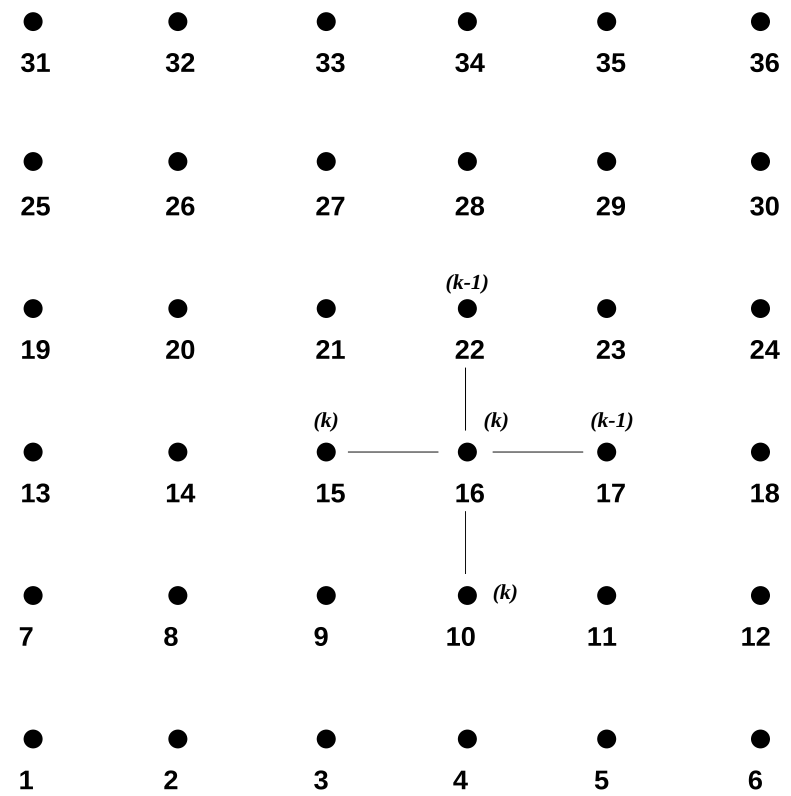
Parallelism in Gauss-Seidel Iteration

- Conventional Gauss-Seidel canNOT be easily parallelized because it has sequential update order

$$
\begin{aligned}
x_n^{(k)} &\text{ requires } x_{n-1}^{(k)} \quad \ldots \\
x_{n-1}^{(k)} &\text{ requires } x_{n-2}^{(k)} \quad \ldots \\
&\vdots \\
x_2^{(k)} &\text{ requires } x_1^{(k)}, \text{etc.}
\end{aligned}
$$

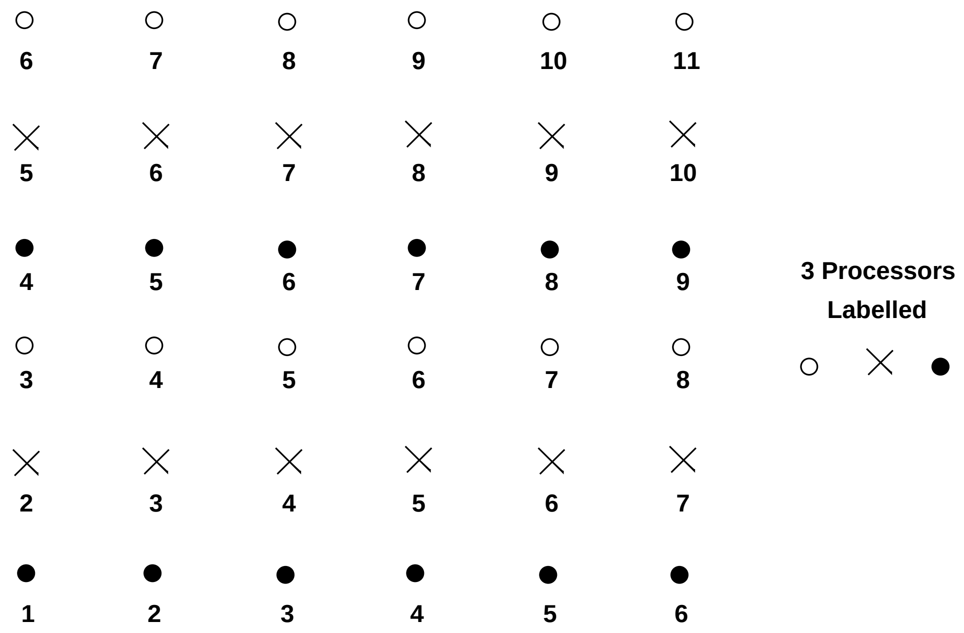
- There are two parallelization strategies
  1. Keep same algorithm—pipelined wavefront method
  2. Change algorithm—red/black two-phase

6×6 Matrix Example Stencil



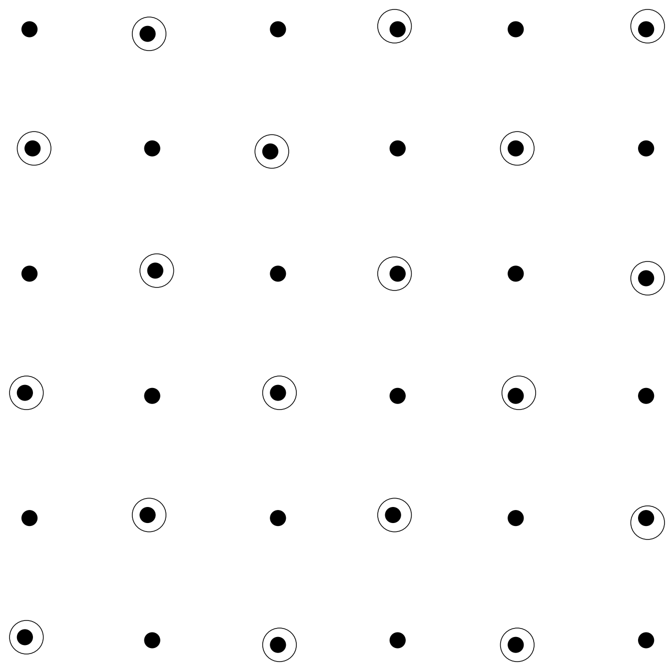
- We can exploit special structure of matrix of **A** (e.g., $x_{16}^{(k)}$ requires $x_{15}^{(k)}$, $x_{17}^{(k-1)}$, $x_{10}^{(k)}$, $x_{22}^{(k-1)}$, etc.).

6×6 Matrix—Wavefront Parallelism for Gauss Seidel



- Here number l labels $l = 1 \ldots 11$ phases
- Each phase l ONLY depends on $\underline{x}^{(k)}$ components from $(l-1)$ th phase
- Can execute each update in a given phase concurrently
- Some Load Imbalance
- Speedup of 2.5 in example of three processors

The Red-Black Two Phase Parallel Gauss Seidel Iteration



Red $\odot$ Black $\bullet$

- Note calculation of $\odot$ only depends on $\bullet$ points and vice versa
- So two phase algorithm.

  Calculate all $\odot$—Phase I.

  Then all $\bullet$—Phase II.

Analysis of Parallel Red Black Gauss Seidel

- This can be parallelized as easily as Jacobi!
- Phase I follows by Phase II.
- Red black is a different ordering from original Gauss Seidel.
- For (some) elliptic problems, can show red black as good as and maybe better than original ordering.
  - i.e., red black does not need more iterations than pure Gauss Seidel.
- For related hyperbolic problems—see NAS benchmark SSOR discussion in CPS713—not so clear.
- For higher order differencing, need more phases (more colors) but principle the same.

Eigenvalues of Gauss Seidel Iteration Matrix

- It was relatively easy to find the eigenvalues and eigenfunctions of Jacobi's method for a simple geometry.
- Gauss-Seidel it is not so easy as matrix $M^{-1}N$ is not symmetric. There is a nice discussion in

  "Numerical Computation of Internal and External Flows," C. Hirsch, Wiley, 1988.
- Eigenvalues λ_{GS} of Gauss-Seidel Iteration Matrix are related to those of Jacobi Iteration λ_J by $\lambda_{GS} = \lambda_J^2$
- We showed that

$$1 - \epsilon \lesssim \lambda_J \lesssim -1 + \epsilon \qquad \epsilon \sim \frac{\pi^2}{2N^2}$$

- Therefore,

$$|\lambda_{GS}|_{\max} = (|\lambda_J|_{\max})^2 \sim 1 - 2\epsilon$$

Comparison of Convergence of Gauss-Seidel and Jacobi Iteration

- Convergence can be characterized by a given fractional error (say .001).
- This correponds to

$$|\lambda|^k_{\max} \sim .001$$

  where k number of required iterations
- $\exp\left[-k\left(1 - |\lambda|_{\max}\right)\right] = .001$
  i.e., value of $k \propto \frac{1}{(1-|\lambda|_{\max})}$
- We showed Gauss Seidel doubles value of

$$1/(1 - |\lambda|_{\max}),$$

  i.e., it converges in half as many iterations as Jacobi

Successive Overrelaxation Iteration Method (SOR)

- This is relaxation (acceleration) method applied to Gauss-Seidel.
- In Jacobi, the optimal case was $\omega = 1$ (i.e., no gain in using relaxation).
- However, for Gauss-Seidel, we will find $\omega \sim 2$ is best and does improve the method.
- The spectral radius ($|\lambda|_{\max}$) of Jacobi is given by

  $$\rho_J = 1 - \frac{\pi^2}{2N^2}$$

  and for conventional (unaccelerated) Gauss Seidel by

  $$\rho_{GS} = 1 - \frac{\pi^2}{N^2}$$

- For choice

  $$\omega_{opt} \sim 2\left(1 - \frac{\pi}{N}\right)$$

  one can show that

  $$\rho_{SOR} \sim 1 - \frac{\pi}{N}$$

Convergence of SOR Compared to Jacobi and Gauss Seidel

- Remember number of iterations is proportional to

$$
\begin{aligned}
\frac{1}{1-\rho} &= \frac{2N^2}{\pi^2} \quad \text{Jacobi} \\
&= \frac{N^2}{\pi^2} \quad \text{Gauss Seidel} \\
&= \frac{N}{\pi} \quad \text{SOR}
\end{aligned}
$$

— i.e., number of iterations decreases in SOR by a factor N/π, which is large!

- Unfortunately, the above analysis is only valid in simple cases
  - simple geometry
  - some orderings in Gauss-Seidel

Estimate of Over Relaxation Parameter ω

- Thus, in general for practice problems, one cannot precalculate optimal overrelaxation parameter.
- Find ω by numerical experimentation.
- However, note that one can expect a choice $1 < \omega < 2$ to be better than natural $\omega = 1$.
- One can use the estimate

$$\begin{aligned} \omega_{opt} &= 2 - \frac{\text{constant}}{N} \\ &= 2 - \text{constant} \cdot h \end{aligned}$$

(h is grid spacing)

Pseudo Code for SOR—Successive Over Relaxation

Choose an initial guess $x^{(0)}$ to the solution x.
for $k = 1,2,\ldots$
    for $i = 1,2,\ldots,n$
        $\sigma = 0$
        for $j = 1,2,\ldots,i-1$
            $\sigma = \sigma + a_{i,j}\, x_j^{(k)}$
        end
        for $j = i+1,\ldots,n$
            $\sigma = \sigma + a_{i,j}\, x_j^{(k-1)}$
        end
        $\sigma = \left(b_i - \sigma\right)/a_{i,i}$
        $x_i^{(k)} = x_i^{(k-1)} + \omega\left(\sigma - x_i^{(k-1)}\right)$
    end
    check convergence; continue if necessary
end

The SOR Method