Chapter 2. Parallel Computer Architecture

Section 1. Processor Architecture & Technology Trends

Moore's Law = # transistors of typical processor chip doubles every 18-24 months

Pipelining

pipeline stages = degree of parallelism

SLP (instruction-level parallelism) processors

Process/Thread

multicore processors - each core has separate flow of control

Section 2. Flynn's Taxonomy

SISD - conventional sequential computer according to von Neumann model

MISD, SIMD, MIMD

Section 3. Memory Organization

Distributed Memory Organization

point-to-point connection between nodes

DMA (direct memory access) controller

cluster - collection of complete computers with dedicated interconnection network

Shared Memory Organization

global/shared memory machines

shared variables → race conditions

thread - separate control flow sharing data through global address space

kernel thread - managed by OS

user thread - generated and controlled by parallel program

Memory Access Times

multithreading - simulation of virtual processors by each physical processor

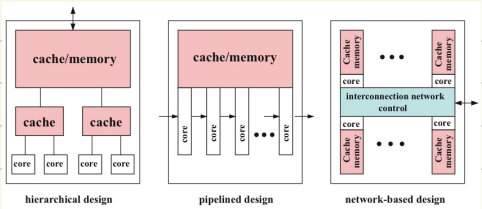
local cache

Section 4. Thread-Level

chip multiprocessing (execution cores)

timeslice, switch-on-event, simultaneous (hyperthreading)

Architecture of Multicore Processors



Section 5. Interconnection Networks

topology + routing technique

static / direct / point-to-point

↳ fixed physical link between nodes

dynamic / indirect → nodes connected through switches and links

routing technique = routing algorithm + switching strategy

Properties

diameter - max distance between 2 nodes

degree - max # direct neighbours

bisection bandwidth - min # edges removed to split system in half

node/edge connectivity - min # nodes/edges failing to disconnect system

Direct

a) complete graph

b) linear array

c) ring

d) 2D mesh

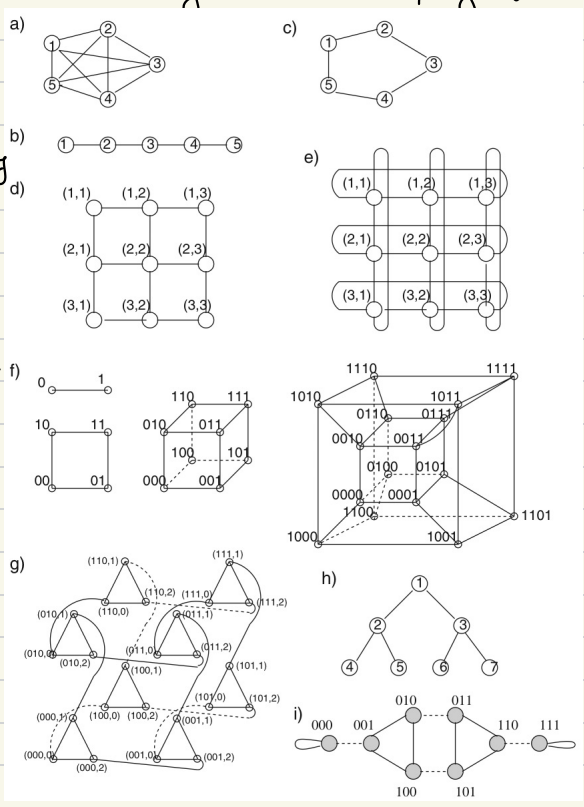
e) 2D torus

f) kD cube
k = 1, 2, 3, 4

g) 3D cube-connected-cycles

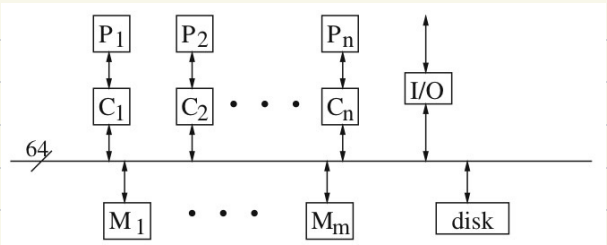
h) complete binary tree

i) shuffle-exchange



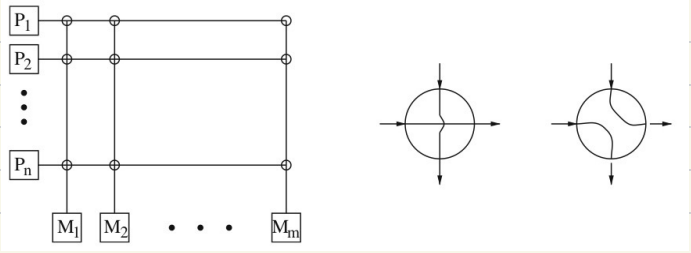
Network G with n nodes	Degree $g(G)$	Diameter $\delta(G)$	Edge- connectivity $ec(G)$	Bisection bandwidth $B(G)$
Complete graph	$n-1$	1	$n-1$	$\left(\frac{n}{2}\right)^2$
Linear array	2	$n-1$	1	1
Ring	2	$\lfloor \frac{n}{2} \rfloor$	2	2
d-Dimensional mesh ($n = r^d$)	$2d$	$d(\sqrt[d]{n}-1)$	d	$n^{\frac{d-1}{d}}$
d-Dimensional torus ($n = r^d$)	$2d$	$d\left\lfloor \frac{\sqrt[d]{n}}{2} \right\rfloor$	$2d$	$2n^{\frac{d-1}{d}}$
k-Dimensional hyper-cube ($n = 2^k$)	$\log n$	$\log n$	$\log n$	$\frac{n}{2}$
k-Dimensional CCC network ($n = k2^k$ for $k \geq 3$)	3	$2k-1+\lfloor k/2 \rfloor$	3	$\frac{n}{2k}$
Complete binary tree ($n = 2^k - 1$)	3	$2\log \frac{n+1}{2}$	1	1
k-ary d-cube ($n = k^d$)	$2d$	$d\lfloor \frac{k}{2} \rfloor$	$2d$	$2k^{d-1}$

Dynamic Bus

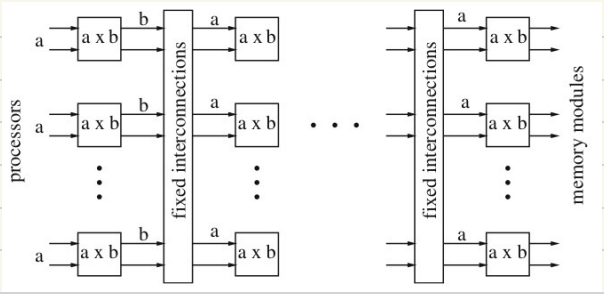


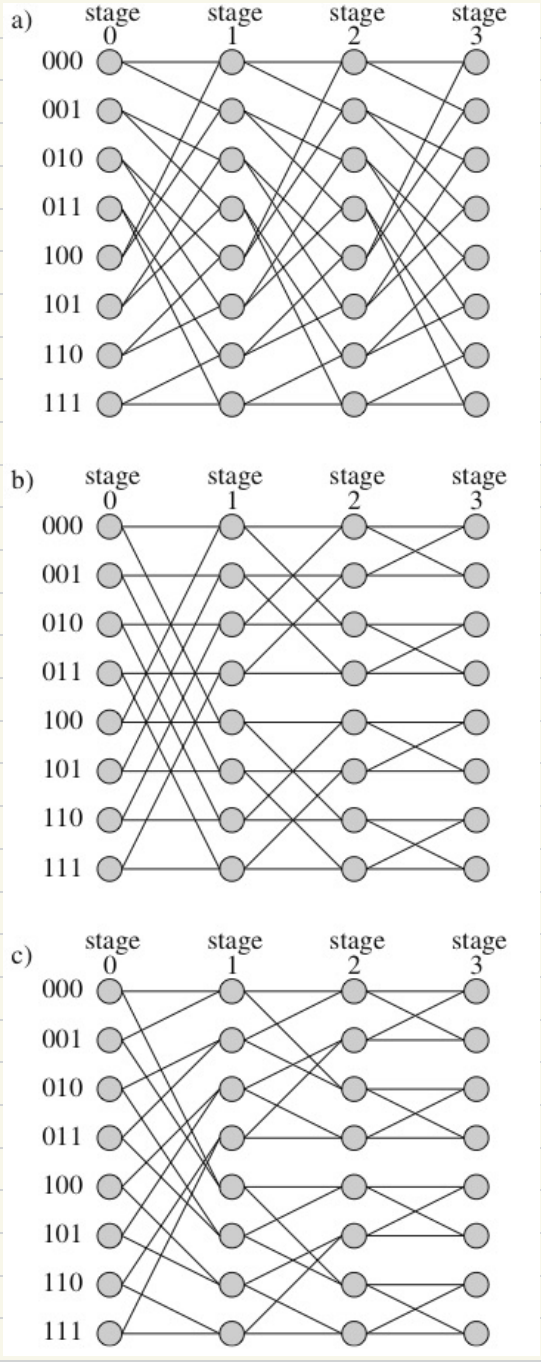
bus arbiter - coordination

Crossbar



Multistage Switching

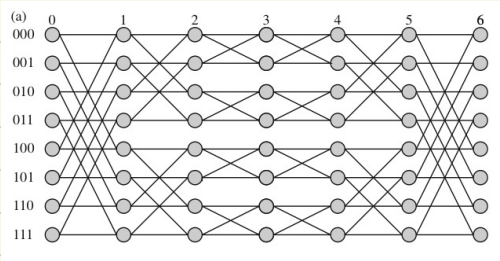




a) Omega

b) Butterfly

c) Baseline



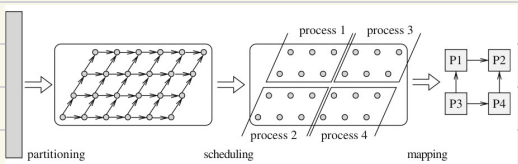
a) Beneš

b) Fat Tree

Chapter 3. Parallel Programming Models

Section 1. Models for Parallel Systems

process - different address space
thread - common address space



Section 3. Levels of Parallelism

Instruction Level

I_1: $R_1 \leftarrow R_2+R_3$	I_1: $R_1 \leftarrow R_2+R_3$	I_1: $R_1 \leftarrow R_2+R_3$
I_2: $R_5 \leftarrow R_1+R_4$	I_2: $R_2 \leftarrow R_4+R_5$	I_2: $R_1 \leftarrow R_4+R_5$
flow dependency	anti dependency	output dependency

Data Parallelism

SPMD (Single Program Multiple Data) - one program executed by all

Chapter 4. Performance Analysis of Parallel Programs

Section 1. Performance Evaluation of Computer Systems

response time - time b/w start and termination of program
throughput - average # work units executed per time unit

CPU Performance

user CPU time = CPU execution time

system CPU time = CPU execution of routines of OS

waiting time = completion of I/O + other program executions due to time sharing

clock rate = 1 / clock cycle time

$T_{u\text{-}cpu}(A) = n_{cycle}(A) \cdot t_{cycle}$

CPI (Clock cycles Per Instruction) = average # CPU cycles per instruction

$T_{u\text{-}cpu}(A) = n_{instr}(A) \cdot CPI(A) \cdot t_{cycle}$

Processor Performance (Memory Hierarchy)

$T_{u\text{-}cpu}(A) = (n_{cycles}(A) + n_{mm\text{-}cycles}(A)) \cdot t_{cycle}$

memory access ↙ $= n_{read\_cycles}(A) + n_{write\_cycles}(A)$

$n_{read\_cycles}(A) = n_{read\text{-}op}(A) \cdot r_{read\_miss}(A) \cdot n_{miss\_cycle}$

$T_{u\text{-}cpu}(A) = n_{instr.}(A) \cdot (CPI(A) + n_{rw\text{-}op}(A) \cdot r_{miss}(A) \cdot n_{miss\_cycle}) \cdot t_{cycle}$

Section 2: Performance Metrics for Parallel Programs

parallel runtime $T_p(n)$ - start-end time of all participating processors

- runtime for execution of local computations
- runtime for exchanging data
- synchronization runtime
- waiting times

Speedup & Efficiency

$C_p(n) = p \cdot T_p(n)$ - cost of parallel program (total amount of work)

cost-optimal: $C_p(n) = T^*(n)$ → fastest sequential execution time

$S_p(n) = \frac{T^*(n)}{T_p(n)}$ - speedup

$E_p(n) = \frac{T^*(n)}{C_p(n)} = \frac{S_p(n)}{p} = \frac{T^*(n)}{p \cdot T_p(n)}$ - efficiency

Amdahl's Law

$f \in [0,1]$ - constant serial fraction of code

$$\Rightarrow S_p(n) = \frac{T^*(n)}{f \cdot T^*(n) + \frac{1-f}{p} T^*(n)} = \frac{1}{f + \frac{1-f}{p}} \leq \frac{1}{f}$$

Scalability

T_f - constant ext. of sequential, $T_v(n,p)$ - ext. of parallel

$S_p(u) = \frac{\tau_f + \tau_v(u,1)}{\tau_f + \tau_v(u,p)}$ - scaled speedup

perfect serializability: $\tau_v(u,1) = T^*(1) - \tau_f$; $\tau_v(u,p) = (T^*(u) - \tau_f)/p$

$S_p(u) = \frac{\tau_f + T^*(u) - \tau_f}{\tau_f + (T^*(u) - \tau_f)/p} = \frac{\frac{\tau_f}{T^*(u) - \tau_f} + 1}{\frac{\tau_f}{T^*(u) - \tau_f} + \frac{1}{p}}$ $\Rightarrow \lim_{u \to \infty} S_p(u) = p$

Section 4. Analysis of Parallel Execution Times

Scalar Product

vectors $\vec{a}, \vec{b} \in \mathbb{R}^u$, p processors, $r = \frac{u}{p}$ values per processor

α time units for execution of arithmetic operation

β time units for communication of floating-point value

computation time: $2r\alpha$

Linear Array: $T(p,u) = 2\frac{u}{p}\alpha + \frac{p}{2}(\alpha+\beta)$ $T'(p) = -\frac{2u\alpha}{p^2} + \frac{\alpha+\beta}{2}$

Hypercube: $T(u,p) = \frac{2u\alpha}{p} + \log p(\alpha+\beta)$ $T'(p) = -\frac{2u\alpha}{p^2} + (\alpha+\beta)\frac{1}{p}\frac{1}{\ln 2}$

Matrix-Vector Product

$A \cdot \vec{b} = \vec{c}$, $A \in \mathbb{R}^{u \times u}$, $\vec{b}, \vec{c} \in \mathbb{R}^u$, $r = \frac{u}{p}$ vector elements per processor

computation time: $2\frac{u^2}{p}\alpha$

communication time: $\beta + r\cdot\gamma$

Linear Array: $T(u,p) = \frac{2u^2}{p}\alpha + p(\beta + \frac{u}{p}\gamma) = \frac{2u^2}{p}\alpha + p\beta + u\gamma$ $T'(p) = -\frac{2u^2\alpha}{p^2} + \beta$

Hypercube: $T(u,p) = \frac{2\alpha u^2}{p} + \frac{p}{\log p}(\beta + r\gamma) = \frac{2\alpha u^2}{p} + \frac{p}{\log p}\beta + \frac{\gamma u}{\log p}$ $T'(p) = -\frac{\alpha u^2}{p} + \frac{\beta}{\log p} - \frac{\beta}{\log^2 p \ln 2} - \frac{\gamma u}{p^2 \log^2 p \ln 2}$

Chapter 5. Message-Passing Programming

Section 1. Introduction to MPI

- Blocking operation - all state transitions completed before return of control
- Nonblocking operation - only starts the operation

local view / global view

- Synchronous communication - communication doesn't complete before both processes started communication
- Asynchronous communication - sender operation uncoordinated with receiver's

deadlock - processes waiting on one another when none is computing

secure - correctness doesn't depend on assumptions about specific properties

Chapter 6. Thread Programming

Section 3. OpenMP

threads running simultaneously on multiple processors or cores
fork-join - thread creation and destruction
parallel region - executed in parallel by all threads

Chapter 7. Algorithms for Systems of Linear Equations

$A\vec{x} = \vec{b}$, $A \in \mathbb{R}^{n \times n}$, $\vec{b}, \vec{x} \in \mathbb{R}^n$ (7.1)

$\exists \vec{x}$ if A - non-singular aka $\exists A^{-1}$ s.t. $A \cdot A^{-1} = I$

direct solution method - determines exact solution in fixed number of steps dependent on size n of the system (elimination and factorization methods)

iterative solution methods - determines approximation of exact solution based on acceptable precision

Section 1. Gaussian Elimination

LU decomposition

$$A\vec{x} = \vec{b} \Rightarrow \begin{vmatrix} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ \vdots \\ a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = b_i \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{vmatrix}$$

forward elimination: $Ux = b'$, U - upper triangular matrix

$n-1$ steps: $A^{(1)} := A = (a_{ij})$, $b^{(1)} := b = (b_i)$

$A^{(k)}x = b^{(k)}$,

$$A^{(k)} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1,k-1} & a_{1k} & \dots & a_{1n} \\ 0 & a_{22}^{(1)} & \dots & a_{2,k-1}^{(1)} & a_{2k}^{(1)} & \dots & a_{2n}^{(1)} \\ & & \ddots & \vdots & \vdots & & \vdots \\ & & & a_{k-1,k-1}^{(k-2)} & a_{k-1,k}^{(k-2)} & \dots & a_{k-1,n}^{(k-2)} \\ & & & 0 & a_{kk}^{(k-1)} & \dots & a_{kn}^{(k-1)} \\ & & & & \vdots & & \vdots \\ 0 & & \dots & & a_{nk}^{(k-1)} & \dots & a_{nn}^{(k-1)} \end{bmatrix}$$

elimination factor of row i: $l_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}$ (7.2)

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - l_{ik} a_{kj}^{(k)} \quad (7.3)$$

$$b_i^{(k+1)} = b_i^{(k)} - l_{ik} b_k^{(k)} \quad (7.4)$$

backward substitution: $x_k = \frac{1}{a_{kk}^{(u)}} \left(b_k^{(u)} - \sum_{j=k+1}^{u} a_{kj}^{(u)} x_j \right)$ (7.5)

$U := A^{(u)}$ L - lower triangular matrix of elimination factors

$$L = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ l_{21} & 1 & 0 & \dots & 0 \\ l_{31} & l_{32} & 1 & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ l_{n1} & l_{n2} & \dots & l_{n,n-1} & 1 \end{bmatrix}, \quad A = L \cdot U \Rightarrow A\vec{x} = LA^{(u)}\vec{x} = L\vec{y} = \vec{b}, \quad \vec{y} = A^{(u)}\vec{x} \quad (7.6)$$

1) obtain $\vec{y}$ by solving $Ly=b$ with forward substitution
2) obtain $\vec{x}$ by solving $A^{(n)}x = U_x = y$ with backward substitution
division by $a_{kk}^{(k)} \Rightarrow a_{kk}^{(k)} \neq 0$ never ensured
$\Rightarrow$ introduce pivot element to substitute $a_{kk}^{(k)}$ if $=0$
column pivoting, for column k with elements $a_{kk}^{(k)} \dots a_{nk}^{(k)}$, find $\max |a_{rk}^{(k)}|$, $k \le r \le n$
$r \neq k \Rightarrow$ rows r and k of $A^{(n)}$ and elements $b_k^{(k)}$ and $b_r^{(k)}$ of $b^{(k)}$ are exchanged
row pivoting, for row k with elements $a_{kk}^{(k)} \dots a_{kn}^{(k)}$, find $\max |a_{kr}^{(k)}|$, $k \le r \le n$
$r \neq k \Rightarrow$ columns r and k of $A^{(k)}$ and elements x_k and x_r of x are exchanged
total pivoting, element with max abs value in $\tilde{A}^{(k)} = (a_{ij}^{(k)})$, $k \le i, j \le n$, exchange both rows and columns

row-cyclic distribution: processor P_q, $1 \le q \le r$, owns rows $q, q+p, q+2p, \dots$ | $1 \le i \le n$ $q = (i-1 \bmod p)+1$
1) Determine local pivot element (per processor)
2) Determine global pivot element (communication of local pivot by each processor)
3) Exchange pivot row (local if both in same processor, otherwise through communication)
4) Distribution of pivot row (communicate to all processors)
5) Computation of elimination factors (local - 7.2)
6) Computation of matrix elements (local - 7.3 + 7.4)
backward substitution - inherently sequential (local - 7.5 + broadcast)

Section 2. Direct Methods for Linear Systems with Banded Structure

Discretization of Poisson Equation

$-\Delta u(x,y) = f(x,y), \quad \forall (x,y) \in \Omega$ (7.13)
Dirichlet BC: $u(x,y) = \varphi(x,y), \quad \forall (x,y) \in \partial\Omega$ (7.14)
Approximation: $\Delta u(x_i, y_j) = -\frac{1}{h^2}(4u_{ij} - u_{i+1,j} - u_{i-1,j} - u_{i,j+1} - u_{i,j-1})$ $\rightarrow$ five-point stencil
$\Rightarrow$ discretized Poisson / five-point formula: $\frac{1}{h^2}(4u_{ij} - u_{i+1,j} - u_{i-1,j} - u_{i,j-1} - u_{ij+1}) = f_{ij}$ (7.15) $1 \le i,j \le N$
$u_{ij} = \varphi(x_i, y_j)$ (7.16) $i = 0, N+1$ $j = 0, \dots, N+1$ and $j = 0, N+1$ $i = 0, \dots, N+1$
$z = (u_{11}, u_{21}, \dots, u_{N1}, u_{12}, u_{22}, \dots, u_{N2}, \dots, u_{1N}, u_{2N}, \dots, u_{NN})$
$z_k := u_{ij} \quad k = i + (j-1)N$
$d = (f_{11} + \frac{\varphi(0,y_1)+\varphi(x_1,0)}{h^2}, f_{21} + \frac{\varphi(x_2,0)}{h^2}, \dots, f_{N1} + \frac{\varphi(x_N,0)+\varphi(0,y_1)}{h^2}, \dots, f_{NN} + \frac{\varphi(x_N,0)+\varphi(0,y_N)}{h^2})$
$A_2 \cdot d \Rightarrow \frac{1}{h^2}(4z_{i+(j-1)N} - z_{i+1+(j-1)N} - z_{i-1+(j-1)N} - z_{i+jN} - z_{i+(j-2)N}) = d_{i+(j-1)N}$
$\Rightarrow a_{kk} = 4$
or $\frac{1}{h^2}(4z_k - z_{k+1} - z_{k-1} - z_{k+N} - z_{k-N}) = d_k$ (7.17) | $a_{k,k+1} = a_{k,k-1} = a_{k,k+N} = a_{k,k-N} = -1$

Tridiagonal Systems

A-banded matrix if $\exists r \in \mathbb{N}, r \leq n$ $a_{ij}=0$ for $|i-j|>r$

r-semi-bandwidth $r=1$-tridiagonal matrix

1) Gaussian Elimination

Each step k, only 1 elimination factor $l_{k+1} := l_{k+1,k}$, 1 row with 1 new element

$$A=\begin{bmatrix} b_1 & c_1 & & 0 \\ a_2 & b_2 & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ 0 & & a_n & b_n \end{bmatrix}$$

$u_1 = b_1$

$l_{k+1} = a_{k+1}/u_k$

$u_{k+1} = b_{k+1} - l_{k+1} \cdot c_k$

} elimination - inherently sequential

$$L=\begin{bmatrix} 1 & & 0 \\ l_2 & \ddots & \\ & \ddots & 1 \\ 0 & & l_n \; 1 \end{bmatrix} \qquad U=\begin{bmatrix} u_1 & c_1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & c_{n-1} \\ 0 & & & u_n \end{bmatrix}$$

backward substitution

$\tilde{y}_{k+1} = y_{k+1} - l_{k+1} \cdot \tilde{y}_k$

$x_n = \tilde{y}_n / u_n$, $u_i x_i + c_i x_{i+1} = \tilde{y}_i$

$x_i = \frac{\tilde{y}_i}{u_i} - \frac{c_i}{u_i} x_{i+1}$

2) Recursive Doubling

if coefficient matrix A is symmetric+positive definite or diagonally dom.

elimination $Ax=y$ $\begin{cases} b_1 x_1 + c_1 x_2 = y_1 \\ a_i x_{i-1} + b_i x_i + c_i x_{i+1} = y_i \\ a_n x_{n-1} + b_n x_n = y_n \end{cases}$

$$\left|\begin{array}{ll} a_{i-1} x_{i-2} + b_{i-1} x_{i-1} + c_{i-1} x_i & = y_{i-1} \\ a_i x_{i-1} + b_i x_i + c_i x_{i+1} & = y_i \\ a_{i+1} x_i + b_{i+1} x_{i+1} + c_{i+1} x_{i+2} & = y_{i+1} \end{array}\right.$$

$$\Rightarrow \left|\begin{array}{l} x_{i-1} = \frac{y_{i-1}}{b_{i-1}} - \frac{a_{i-1}}{b_{i-1}} x_{i-2} - \frac{c_{i-1}}{b_{i-1}} x_i \\ x_{i+1} = \frac{y_{i+1}}{b_{i+1}} - \frac{a_{i+1}}{b_{i+1}} x_i - \frac{c_{i+1}}{b_{i+1}} x_{i+2} \end{array}\right\} a_i^{(1)} x_{i-2} + b_i^{(1)} x_i + c_i^{(1)} x_{i+2} = y_i^{(1)} \quad (7.22)$$

$a_i^{(1)} = \alpha_i^{(1)} \cdot a_{i-1}$

$b_i^{(1)} = b_i + \alpha_i^{(1)} \cdot c_{i-1} + \beta_i^{(1)} \cdot a_{i+1}$

$c_i^{(1)} = \beta_i^{(1)} \cdot c_{i+1}$

$y_i^{(1)} = y_i + \alpha_i^{(1)} \cdot y_{i-1} + \beta_i^{(1)} \cdot y_{i+1}$

$\alpha_i^{(1)} := -\frac{a_i}{b_{i-1}}$

$\beta_i^{(1)} := -\frac{c_i}{b_{i+1}}$ (7.23)

$i=1,2,n-1,n$: $b_1^{(1)} = b_1 + \beta_1^{(1)} \cdot a_2$, $y_1^{(1)} = y_1 + \beta_1^{(1)} \cdot y_2$, $a_1^{(1)} = a_2^{(1)} = 0$

$b_n^{(1)} = b_n + \alpha_n^{(1)} \cdot c_{n-1}$, $y_n^{(1)} = b_n + \alpha_n^{(1)} \cdot y_{n-1}$, $c_{n-1}^{(1)} = c_n^{(1)} = 0$

$A^{(1)} x = y^{(1)}$, $A^{(1)} = \begin{bmatrix} b_1^{(1)} & 0 & c_1^{(1)} & & 0 \\ 0 & b_2^{(1)} & 0 & \ddots & \\ a_3^{(1)} & 0 & b_3^{(1)} & \ddots & c_{n-2}^{(1)} \\ & \ddots & \ddots & \ddots & 0 \\ 0 & & a_n^{(1)} & 0 & b_n^{(1)} \end{bmatrix}$ Next step $a_i^{(2)} x_{i-4} + b_i^{(2)} x_i + c_i^{(2)} x_{i+4} = y_i^{(2)}$

step k:

$$a_{i-2^{k-1}}^{(k-1)} x_{i-2^k} + b_{i-2^{k-1}}^{(k-1)} x_{i-2^{k-1}} + c_{i-2^{k-1}}^{(k-1)} x_i = y_{i-2^{k-1}}^{(k-1)}$$

$$a_i^{(k-1)} x_{i-2^{k-1}} + b_i^{(k-1)} x_i + c_i^{(k-1)} x_{i+2^{k-1}} = y_i^{(k-1)}$$

$$a_{i+2^{k-1}}^{(k-1)} x_i + b_{i+2^{k-1}}^{(k-1)} x_{i+2^{k-1}} + c_{i+2^{k-1}}^{(k-1)} x_{i+2^k} = y_{i+2^{k-1}}^{(k-1)}$$

$a_i^{(k)} = \alpha_i^{(k)} \cdot a_{i-2^{k-1}}^{(k-1)}$, $i = 2^{k+1}, \dots, u$, otherwise 0 (7.24)

$c_i^{(k)} = \beta_i^{(k)} \cdot c_{i+2^{k-1}}^{(k-1)}$, $i = 1, \dots, u-2^k$, otherwise 0

$b_i^{(k)} = \alpha_i^{(k)} \cdot c_{i-2^{k-1}}^{(k-1)} + b_i^{(k-1)} + \beta_i^{(k)} \cdot a_{i+2^{k-1}}^{(k-1)}$ $\quad \alpha_i^{(k)} := -a_i^{(k-1)} / b_{i-2^{k-1}}^{(k-1)}$, $i = 2^{k-1}+1, \dots, u$ (7.25)

$y_i^{(k)} = \alpha_i^{(k)} \cdot y_{i-2^{k-1}}^{(k-1)} + y_i^{(k-1)} + \beta_i^{(k)} \cdot y_{i+2^{k-1}}^{(k-1)}$ $\quad \beta_i^{(k)} := -c_i^{(k-1)} / b_{i+2^{k-1}}^{(k-1)}$, $i = 1, \dots, u-2^{k-1}$

$a_i^{(k)} x_{i-2^k} + b_i^{(k)} x_i + c_i^{(k)} x_{i+2^k} = y_i^{(k)}$ (7.26)

Initialization: $a_1^{(0)} = 0$ $\quad a_i^{(0)} = a_i$, $i = 2, \dots, u$

$b_i^{(0)} = b_i$, $y_i^{(0)} = y_i$, $\forall i$

$c_u^{(0)} = 0$ $\quad c_i^{(0)} = c_i$, $i = 1, \dots, u-1$

$N = \lceil \lg u \rceil$ steps, $A^{(N)} = \operatorname{diag}(b_1^{(N)}, \dots, b_u^{(N)})$, $\vec{x} = \vec{y}^{(N)} / \vec{b}^{(N)}$

3) Cyclic Reduction

elimination: for $k = 1, \dots, \lfloor \log u \rfloor$ compute $a_i^{(k)}, b_i^{(k)}, c_i^{(k)}, y_i^{(k)}$ with $i = 2^k, \dots, u$, step size 2^k

substitution: for $k = \lfloor \log u \rfloor, \dots, 0$ compute x_i for $i = 2^k, \dots, u$, step size 2^{k+1}

p processors, $n = p \cdot q$, $q \in \mathbb{N}$ and $q = 2^Q$, $Q \in \mathbb{N}$ $\Rightarrow$ block of q consecutive rows

1) parallel reduction: each processor performs $\log q$ steps + message of size 4 $(a_j^{(k)}, b_j^{(k)}, c_j^{(k)}, y_j^{(k)})$

2) parallel recursive doubling: $\lceil \log p \rceil$ steps + message of size 4 $(\tilde{a}_j^{(k)}, \tilde{b}_j^{(k)}, \tilde{c}_j^{(k)}, \tilde{y}_j^{(k)})$

3) parallel substitution: $\log q$ steps + message of size 1 $(\tilde{x}_i)$

Section 3: Iterative Methods for Linear Systems

Standard Iteration Methods

$A = M - N$, $M, N \in \mathbb{R}^{n \times n}$

M - non-singular matrix, easily computible M^{-1} (diagonal)

$Mx^* = Nx^* + b \Rightarrow Mx^{(k+1)} = Nx^{(k)} + b$

$x^{(k+1)} = Cx^{(k)} + d$ (7.36) $\quad C := M^{-1}N$, $\quad d := M^{-1}b$

convergence: $\forall x^{(0)} \in \mathbb{R}^n \to$ convergence $\iff \lim_{k \to \infty} C^k = 0$, $\rho(C) < 1$ (ρ → largest eigenvalue)

1) Jacobi

$A = D - L - R$, $D, L, R \in \mathbb{R}^{n \times n}$

D - diagonal elements of A

$-L$ - lower triangular of A w/t diagonal

$-R$ - upper triangular of A w/t diagonal

$$Dx^{(k+1)} = (L+R)x^{(k)} + b$$

$$C_{Ja} := D^{-1}(L+R) \quad \text{or} \quad c_{ij} = \begin{cases} -a_{ij}/a_{ii} & \text{for } i \neq j \\ 0 & i = j \end{cases}$$

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1, j \neq i}^{n} a_{ij} x_j^{(k)} \right), \quad i = 1, \dots, n \qquad (7.37)$$

2) Gauss-Siedel

$$(D-L)x^{(k+1)} = Rx^{(k)} + b$$

$$C_{Ga} := (D-L)^{-1}R$$

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right), \quad i = 1, \dots, n \qquad (7.38)$$

3) JOR

$A = \frac{1}{\omega} D - L - R - \frac{1-\omega}{\omega} D$, $\omega \in \mathbb{R}$ - relaxation parameter

$$x_i^{(k+1)} = \frac{\omega}{a_{ii}} \left(b_i - \sum_{j=1, j \neq i}^{n} a_{ij} x_j^{(k)} \right) + (1-\omega) x_i^{(k)}, \quad i = 1, \dots, n \qquad (7.39)$$

4) SOR

$$\hat{x}_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right), \quad i = 1, \dots, n \qquad (7.40)$$

$$x_i^{(k+1)} = x_i^{(k)} + \omega \left(\hat{x}_i^{(k+1)} - x_i^{(k)} \right), \quad i = 1, \dots, n \qquad (7.41)$$

$$x_i^{(k+1)} = \frac{\omega}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right) + (1-\omega) x_i^{(k)} \qquad (7.42)$$

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

Red-Black Ordering

$$\hat{A}\hat{x} = \begin{pmatrix} D_R & F \\ E & D_B \end{pmatrix} \cdot \begin{pmatrix} \hat{x}_R \\ \hat{x}_B \end{pmatrix} = \begin{pmatrix} \hat{b}_1 \\ \hat{b}_2 \end{pmatrix} = \hat{b} \quad (7.46)$$

D_R, D_B – independence of black and red elements

E, F – specify dependence b/w red and black elements

1) Gauss-Siedel

$$\hat{A} = \hat{D} - \hat{L} - \hat{U}$$

$$\hat{D} = \begin{pmatrix} D_R & 0 \\ 0 & D_B \end{pmatrix}, \quad \hat{L} = \begin{pmatrix} 0 & 0 \\ -E & 0 \end{pmatrix}, \quad \hat{U} = \begin{pmatrix} 0 & -F \\ 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} D_R & 0 \\ E & D_B \end{pmatrix} \cdot \begin{pmatrix} x_R^{(k+1)} \\ x_B^{(k+1)} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} - \begin{pmatrix} 0 & F \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_R^{(k)} \\ x_B^{(k)} \end{pmatrix} \quad (7.47)$$

$$D_R \cdot x_R^{(k+1)} = b_1 - F \cdot x_B^{(k)} \quad k = 1, 2, \dots \quad (7.48)$$

$$D_B \cdot x_B^{(k+1)} = b_2 - E \cdot x_R^{(k)} \quad k = 1, 2, \dots \quad (7.49)$$

$$\left(x_R^{(k+1)}\right)_i = \frac{1}{\hat{a}_{ii}} \left(\hat{b}_i - \sum_{j \in N(i)} \hat{a}_{ij} \cdot \left(x_B^{(k)}\right)_j \right), \quad i = 1, \dots, n_R$$

$$\left(x_B^{(k+1)}\right)_i = \frac{1}{\hat{a}_{i+n_R, i+n_R}} \left(\hat{b}_{i+n_R} - \sum_{j \in N(i)} \hat{a}_{i+n_R, j} \cdot \left(x_R^{(k+1)}\right)_j \right), \quad i = 1, \dots, n_B$$

2) SOR

$$\begin{aligned} \tilde{x}_B^{(k+1)} &= D_R^{-1} \cdot b_1 - D_R^{-1} \cdot F \cdot x_B^{(k)} \\ \tilde{x}_B^{(k+1)} &= D_B^{-1} \cdot b_2 - D_B^{-1} \cdot E \cdot x_R^{(k+1)} \\ x_B^{(k+1)} &= x_R^{(k)} + \omega \left(\tilde{x}_R^{(k+1)} - x_B^{(k)} \right) \\ \tilde{x}_D^{(k+1)} &= x_B^{(k)} + \omega \left(\tilde{x}_B^{(k+1)} - x_B^{(k)} \right) \end{aligned} \quad (7.50) \quad k = 1, 2, \dots$$

$$\hat{A} = \frac{1}{\omega} \hat{D} - \hat{L} - \hat{U} - \frac{1-\omega}{\omega} \hat{D}$$

$$\begin{pmatrix} D_R & 0 \\ \omega E & D_B \end{pmatrix} \cdot \begin{pmatrix} x_R^{(k+1)} \\ x_B^{(k+1)} \end{pmatrix} = (1-\omega) \begin{pmatrix} D_R & 0 \\ 0 & D_B \end{pmatrix} \cdot \begin{pmatrix} x_R^{(k)} \\ x_B^{(k)} \end{pmatrix} - \omega \begin{pmatrix} 0 & F \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_R^{(k)} \\ x_B^{(k)} \end{pmatrix} + \omega \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad (7.51)$$

$$S_\omega = \left(\frac{1}{\omega} \hat{D} - \hat{L} \right) \left(\frac{1-\omega}{\omega} \hat{D} + \hat{U} \right)$$