Exercise 10
Iterative Solvers

Janick Cardinale

29.11.2010

Outline

PRNG

Scenario: reproducible results in MC simulation for parallel systems:
For shared memory systems (dynamic schedule):

- Use a global variable (current seed) and a critical section.

For distributed memory systems:

- More difficult, application specific (depends on calculation to communcation ratio)

But, usually not a problem in MC. More problematic is the choice of the seeds.

Red Black Gauss Seidel Method

Iterative solvers:

- Initialize all unknown values with an initial guess.
- Iterate over all unknown values and update one value at a time according to the governing equation.
- Repeat the above step until all the residual r is reduced to the tolerance limit.

Update scheme: *Jacobi* vs. *Gauss* iterations:

Ex10 - Q1 - Poisson equation

We solve the 2D partial differential equation on a unit square domain:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 1; \quad u(x,y) = 0 \mid x, y \in \partial]0,1[^2 \tag{1}$$

Using finite differences:

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_y^2} = 1$$

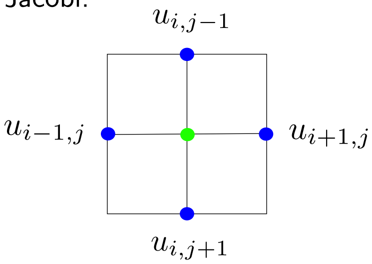
With $h_x = h_y$ solving for $u_{i,j}$:

$$u_{i,j} = \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - h^2}{4} \tag{2}$$

Iterative schemes

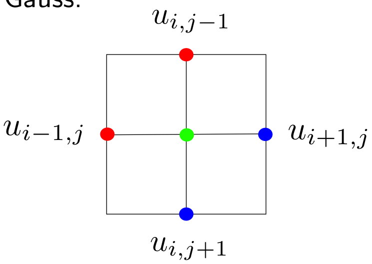
Jacobi vs Gauss iterations

Jacobi:



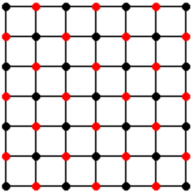
- Slower convergence
- Requires two arrays to store nodal values
- Algorithm is scalable

Gauss:



- Requires one array to store nodal values
- Convergence is fast
- Not scalable

Ex10 - Q1 - Red Black Gauss Seidel Method



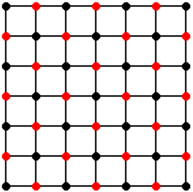
In iteration k:

- 1st pass: All red nodes are updated first using old values of black nodes.
- 2nd pass: All black nodes are updated first using updated values of red nodes

Combines advantages of both, Jacobi and GS:

- Within each pass, the nodes can be updated in parallel
- One array in memory only
- Convergence is better than Jacobi

Ex10 - Q1 - Red Black Gauss Seidel Method



In iteration k:

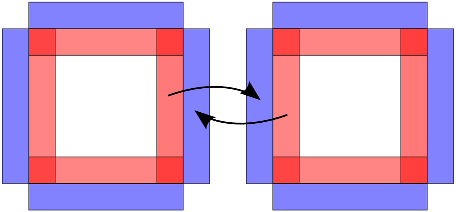
- 1st pass: All red nodes are updated first using old values of black nodes.
- 2nd pass: All black nodes are updated first using updated values of red nodes

Combines advantages of both, Jacobi and GS:

- Within each pass, the nodes can be updated in parallel
- One array in memory only
- Convergence is better than Jacobi

Ex10 - Q1 - Red Black Gauss Seidel Method

Ghost Layers



- red: Export region: ghost layers of adjacent processors.
- blue: Import region: the ghost layers. These are not calculated but obtained by adjacent processors. Use

`MPI_Send_Recv(...)` Boundary conditions:

- Set the import regions of the domain-boundary processors to boundary value (Dirichlet).
- Domain-boundary procs send and recv from `MPI_Proc_null`.

Ex10 - Q2 - Preconditioned Conjugate Gradient

- Solve the 3D Laplacian equation.
- Matrix corresponding to 3D Laplacian operator is generated using Trillinos_Util package.
- pcg can be used when the system matrix is symmetric
- LU preconditioner which is generated using the Ifpack of Trilinos
- AztecOO package solves the linear system of equations

Read the code.

Ex2 - Q2b -PCG

Optional

Implement the method:

```
int pcg(Epetra_LinearProblem &Problem, Epetra_Operator
*prec, int maxiter, double tol, int cpurank);
```

Hints:

- Argument Problem contains the matrix A, vector x, and b.
- Argument prec corresponds to the matrix M
- Solving $Mz_0 = r_0$ is equivalent $z_0 = m^{-1}r_0$
- Vecotrs r, z and q have to be generated within the method.