Universität Stuttgart
Institut für parallele und verteilte Systeme
Prof. Dr. rer. nat. Miriam Mehl
Dipl.-Ing. Florian Lindner

SS 2016

# Parallel Numerics

Exercise 4: Iterative Methods

## 1 Stationary Methods

To solve the equation system $Ax = b$ stationary methods split up the matrix $A$ into $A = M - N$:

$$\begin{aligned} Ax &= b \\ (M-N)x &= b \\ Mx &= Nx + b \\ Mx^{(n+1)} &= Nx^{(n)} + b \end{aligned}$$

Given a matrix $A$:

$$A = \begin{pmatrix} a_1 & b_1 & c_1 & 0 & 0 \\ d_1 & a_2 & b_2 & c_2 & 0 \\ e_1 & d_2 & a_3 & b_3 & c_3 \\ 0 & e_2 & d_3 & a_4 & b_4 \\ 0 & 0 & e_3 & d_4 & a_5 \end{pmatrix}$$

i.e. a banded matrix with five diagonals ($\beta = 2$).

i) Give the Richardson, Jacobi and Gauß-Seidel method using matrix notation. Give an implementation using pseudo code. Choose an appropriate sparse format for $A$ and exploit its banded form.

## 1.1 Residual-based notation

The residual is defined as

$$r = b - Ax$$

i) Give the Richardson, Jacobi and Gauß-Seidel method using the residual.

ii) Give a sketch of the data dependendy graph for both computing the residual and updating the solution according to the Jacobi and the GS scheme. (To simplify matters: Assume that $A$ is tridiagonal

iii) Which parallel algorithms for matrix vector products of a tridiagonal matrix do you know (already)?

iv) Which operations do you find in the Gauß-Seidel algorithm that can be executed in parallel?

## 2 Steepest Descent

Consider the linear system $Ax = b$ where

$$A = \begin{pmatrix} 11 & -9 \\ -9 & 11 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{1}$$

a) Is the matrix $A$ symmetric positive definite (SPD)?

b) Apply the first two iterations of the steepest descent method. Use the initial vector $x^{(0)} = (0, 0)^T$.

c) Show that the residuals $r^{(k)}$ and $r^{(k-2)}, k \geq 2$, are parallel in $\mathbb{R}^2$ for the steepest descent method.

d) Solve (1) with the CG method for the initial solution $x^{(0)} = (0, 0)^T$. Compare your results to part ii). (see algorithm snippet below for the CG algorithm).

e) Consider the Conjugate Gradient method that computes the solution $x$ iteratively as a series $\{x^{(k)}\}$:

$$\begin{aligned}
p^{(0)} &= r^{(0)} = b - Ax^{(0)} \\
\alpha^{(k)} &= -\frac{\langle r^{(k)}, r^{(k)} \rangle}{\langle p^{(k)}, Ap^{(k)} \rangle} \\
x^{(k+1)} &= x^{(k)} - \alpha^{(k)} p^{(k)} \\
r^{(k+1)} &= r^{(k)} + \alpha^{(k)} Ap^{(k)} \\
\textbf{if } \|r^{(k+1)}\|_2^2 &\leq \epsilon \textbf{ then break} \\
\beta^{(k)} &= \frac{\langle r^{(k+1)}, r^{(k+1)} \rangle}{\langle r^{(k)}, r^{(k)} \rangle} \\
p^{(k+1)} &= r^{(k+1)} + \beta^{(k)} p^{(k)}
\end{aligned}$$

Implement this algorithm. Try to implement with just one matrix-vector product. Think about parallelizability of the operations and their computational complexity given $p$ processors and matrix size $n$.

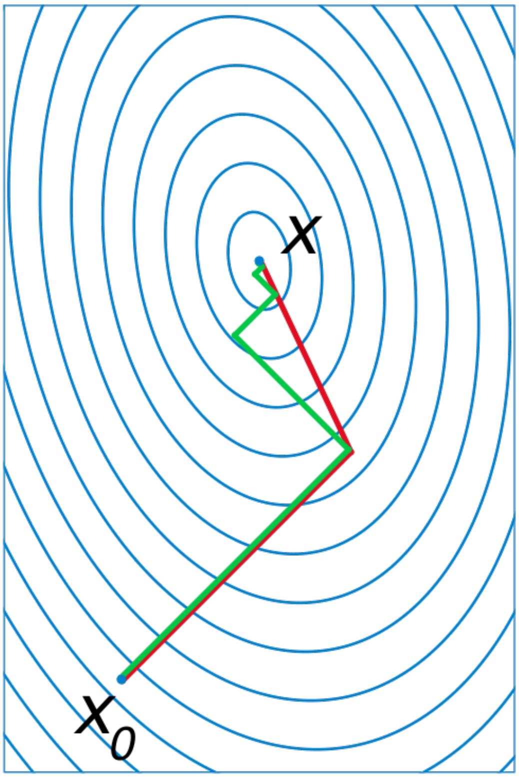

Figure 1: Comparision of Gradient (green) and CG (red)