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.

We store the diagonals of the matrix as vectors:

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

This way we have to store only $n\,(2\,\beta + 1)$ instead of n^2 entries (which, coincindently, is the same here).

Richardson: Split $A = A - I + I$ with identity I:

$$\begin{aligned} Ax &= b \\ (A - I + I)x &= b \\ (A - I)x + x &= b \\ x &= b - (A - I)x \\ x &= (I - A)x + b \end{aligned}$$

With $M = I$ and $N = I - A$ the Richardson method is

$$x^{(n+1)} = (I - A)x^{(n)} + b$$

```
while ||A x − b|| > ε
  for i = 1 to n
    x_i = −Ã_{i,1} x_{i−2} − Ã_{i,2} x_{i−1} + (1 − Ã_{i,3}) x_i − Ã_{i,4} x_{i+1} − Ã_{i,5} x_{i+2} + b_i
```

Jacobi: Split $A = D - L - U$ where D is diagonal part of A, $-L$ is strict lower and $-U$ is strict upper triangular part, respectively:

$$\begin{aligned} (D - L - U)x &= b \\ Dx - (L + U)x &= b \\ Dx &= b + (L + U)x \\ x &= D^{-1}(L + U)x + D^{-1}b \end{aligned}$$

With $M = D$ and $N = L + U$ the Jacobi method is

$$x^{(n+1)} = D^{-1}(L + U)x^{(n)} + D^{-1}b$$

```
while ||A x − b|| > ε
  for i = 1 to n # in parallel
    x̃_i = −(1/Ã_{i,3}) (Ã_{i,1} x_{i−2} + Ã_{i,2} x_{i−1} + Ã_{i,4} x_{i+1} + Ã_{i,5} x_{i+2} + b_i)
  x = x̃
```

Gauß-Seidel: Split $A = D - L - U$:

$$\begin{aligned} (D - L - U)x &= b \\ (D - L)x - Ux &= b \\ x &= (D - L)^{-1}Ux + (D - L)^{-1}b \end{aligned}$$

With $M = D - L$ and $N = U$ the Gauß-Seidel method is

$$x^{(n+1)} = (D - L)^{-1}Ux^{(n)} + (D - L)^{-1}b$$

```
while ||A x − b|| > ε
  for i = 1 to n
    x_i = 1/Ã_{i,3} [ b_i − ( Ã_{i,1} x_{i−2} + Ã_{i,2} x_{i−1} ) − ( Ã_{i,4} x_{i+1} + Ã_{i,5} x_{i+2} ) ]
```

$$\textbf{while}\ ||A\,x - b|| > \epsilon$$
$$\quad \textbf{for}\ i = 1\ \textbf{to}\ n$$
$$\qquad x_i = \frac{1}{\tilde{A}_{i,3}}\left[b_i - \left(\tilde{A}_{i,1}\,x_{i-2} + \tilde{A}_{i,2}\,x_{i-1}\right) - \left(\tilde{A}_{i,4}\,x_{i+1} + \tilde{A}_{i,5}\,x_{i+2}\right)\right]$$

Based on the general iteration method:

$$x^{(n+1)} = M^{-1}Nx^{(n)} + M^{-1}b$$

You get the different methods using different values for M and N:

	M	N
Richardson	I	$I - A$
Jacobi	D	$L + U$
Gauss-Seidel	$D - L$	U

1.1 Residual-based notation

The residual is defined as

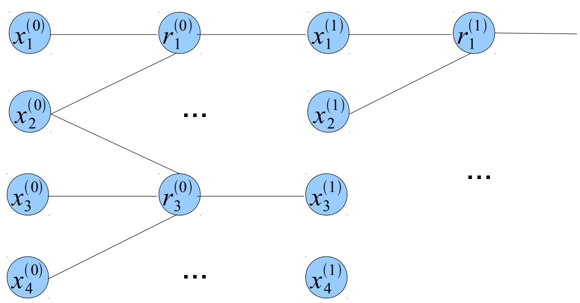
$$r = b - Ax$$

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

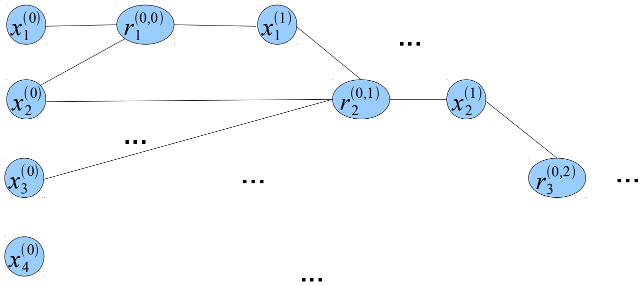
$$\begin{aligned} \text{In general:} \quad & x^{(n+1)} = x^{(n)} + M^{-1}r^{(n)} \\ \text{Richardson with residual:} \quad & x^{(n+1)} = x^{(n)} + r^{(n)} \\ \text{Jacobi with residual:} \quad & x^{(n+1)} = x^{(n)} + D^{-1}r^{(n)} \\ \text{Gauß-Seidel with residual:} \quad & x^{(n+1)} = x^{(n)} + (D-L)^{-1}r^{(n)} \end{aligned}$$

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

Sketch of data dependency between x and r for Jacobi:



Sketch of data dependency between x and r for Gauß-Seidel with residual formulation $x_j^{(n+1)} = x_j^{(n)} + (D-L)^{-1}r_j^{(n,j-1)}$ where $j-1$ updates are considered in r_j:



iii) Which parallel algorithms for matrix vector products of a tridiagonal matrix do you know (already)?
In the residual notation all solvers are reduced to matrix-vector-products. Thus, one can use e.g. blockwise decomposition, Cannon's algorithm, cyclic assignement, etc..

iv) Which operations do you find in the Gauß-Seidel algorithm that can be executed in parallel?
None

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)?
We have to solve the characteristic polynomial $(A - \lambda I)x = 0$. Therefore, we solve $\det(A - \lambda I) = 0$. We obtain the quadratic equation $\lambda^2 - 22\lambda + 40 = 0$ which has the solutions

$$\lambda_1 = 20 \quad \text{and} \quad \lambda_2 = 2.$$

All eigenvalues are positive. Hence, A is SPD.

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

$$x^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad r^{(0)} = b - Ax^{(0)} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

After the first iteration $k = 0$ we obtain

$$\alpha^{(0)} = \frac{1}{11}, \quad x^{(1)} = \begin{pmatrix} \frac{1}{11} \\ 0 \end{pmatrix}, \quad r^{(1)} = \begin{pmatrix} 0 \\ \frac{9}{11} \end{pmatrix}.$$

Putting these results in for the second iteration $k = 1$ we obtain

$$\alpha^{(1)} = \frac{1}{11}, \quad x^{(2)} = \begin{pmatrix} \frac{1}{11} \\ \frac{9}{121} \end{pmatrix}, \quad r^{(2)} = \begin{pmatrix} \frac{81}{121} \\ 0 \end{pmatrix}.$$

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.
For the residuals it holds: $r^{(k)} \perp r^{(k-1)}$ and $r^{(k-1)} \perp r^{(k-2)}$. Hence, in the field $\mathbb{R}^2$ it must hold $r^{(k)} \parallel r^{(k-2)}$ 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).
It holds

$$x^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad r^{(0)} = p^{(0)} = b - Ax^{(0)} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

The first iteration is similar to the steepest descent method. We obtain

$$\alpha^{(0)} = \frac{1}{11}, \quad x^{(1)} = \begin{pmatrix} \frac{1}{11} \\ 0 \end{pmatrix}, \quad r^{(1)} = \begin{pmatrix} 0 \\ \frac{9}{11} \end{pmatrix}, \quad p^{(1)} = \begin{pmatrix} \frac{81}{121} \\ \frac{9}{11} \end{pmatrix}.$$

Putting these results in for the second iteration $k = 1$ we obtain

$$\alpha^{(1)} = \frac{11}{40}, \quad x^{(2)} = \begin{pmatrix} \frac{11}{40} \\ \frac{9}{40} \end{pmatrix}$$

Hence, the CG method converges after $n = 2$ steps towards the solution in contrast to the steepest descent method.

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

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

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.
See sourcecode to corresponding tutorial on webpage.

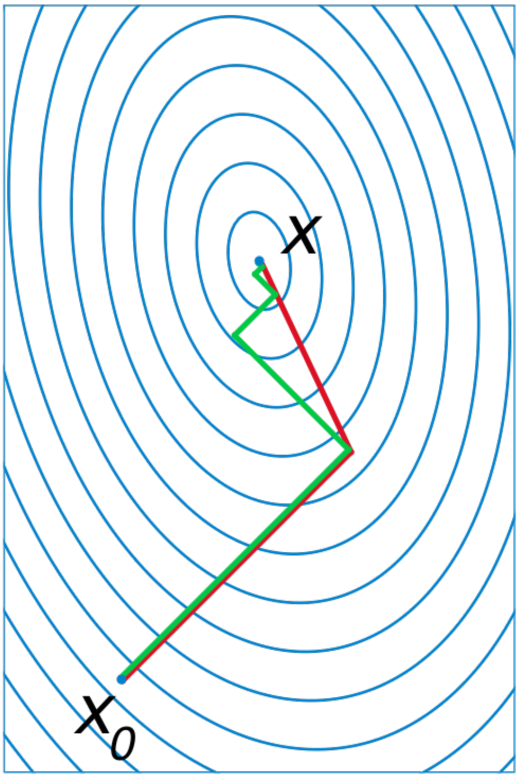


Figure 1: Comparision of Gradient (green) and CG (red)