Exercise Sheet No.5 Deadline: 26. Nov 2013

EXERCISE 5 PRECONDITIONED CONJUGATE GRADIENT METHOD

In the lecture we introduced the Conjugate Gradient Method for solving systems of linear equations corresponding to symmetric positive definite matrices. The effectiveness of this method may be improved by employing a preconditioner.

In this exercise you will implement both the Conjugate Gradient (CG) method and the Conjugate Gradient method with a symmetric SOR (CG SSOR) preconditioner. Therefore you will extend the interface of your `CRSMatrix` class by adding the functions

- **bool** `CRSMatrix::cgssor_solve(Vector & x, Vector & b)`
- **bool** `CRSMatrix::cg_solve(Vector & x, Vector & b)`

which solve the system $A\vec{x} = \vec{b}$.

The algorithm for the CG method *without* preconditioning is given in the following listing:

$$
\begin{aligned}
&\vec{v} = \vec{d} = \vec{b} - A\vec{x} \\
&d_0 = \vec{d}^T\vec{d}; \\
&d_k = d_0 \\
&\text{while } (d_k \geq \varepsilon^2 \cdot d_0) \\
&\{ \\
&\quad \vec{t} = A\vec{v} \\
&\quad \alpha = d_k / \left(\vec{v}^T\vec{t}\right) \\
&\quad \vec{x} = \vec{x} + \alpha\vec{v} \\
&\quad \vec{d} = \vec{d} - \alpha\vec{t} \\
&\quad d_{k_{\text{old}}} = d_k; \\
&\quad d_k = \vec{d}^T\vec{d} \\
&\quad \beta = d_k / d_{k_{\text{old}}} \\
&\quad v = \vec{d} + \beta\vec{v} \\
&\}
\end{aligned}
$$

The algorithm for the CG method *with* preconditioning reads:

$$
\begin{aligned}
&\vec{d} = \vec{b} - A\vec{x} \\
&\textsf{solve } M\vec{z} = \vec{d} \\
&\vec{v} = \vec{z} \\
&\rho_k = \rho_0 = \vec{d}^T\vec{z} \\
&\textsf{while } (\rho_k \geq \varepsilon^2 \cdot \rho_0) \\
&\{ \\
&\quad \vec{t} = A\vec{v} \\
&\quad \alpha = \rho_k / \left(\vec{v}^T\vec{t}\right) \\
&\quad \vec{x} = \vec{x} + \alpha\vec{v} \\
&\quad \vec{d} = \vec{d} - \alpha\vec{t} \\
&\quad \textsf{solve } M\vec{z} = \vec{d} \\
&\quad \rho_{k_{\text{old}}} = \rho_k \\
&\quad \rho_k = \vec{d}^T\vec{z} \\
&\quad \beta = \rho_k / \rho_{k_{\text{old}}} \\
&\quad \vec{v} = \vec{z} + \beta\vec{v} \\
&\}
\end{aligned}
$$

Here M denotes the preconditioner matrix. For the SOR method it was given by $M = L + \frac{D}{\omega}$ where L is the lower left submatrix of A and D holds only the diagonal entries of A. Within the context of the CG method we require the preconditioner to be symmetric. Hence, we employ the SSOR method corresponding to a matrix

$$M = \left(\frac{D}{\omega} + L\right) \frac{\omega}{2-\omega} D^{-1} \left(\frac{D}{\omega} + L^T\right).$$

For the SSOR-preconditioner the step $\textsf{solve } M\vec{z} = \vec{d}$ may be computed by:

$$
\begin{aligned}
&\vec{v} = 0 \\
&\textsf{for } (i = 0; i < n; ++i) \\
&\quad v_i = \omega \left(d_i - \sum_{j<i} a_{ij} v_j \right) / a_{ii} \\
&\vec{\tau} = \vec{d} - A\vec{v} \\
&\vec{\sigma} = 0 \\
&\textsf{for } (i = n-1; i >= 0; --i) \\
&\sigma_i = \omega \left(\tau_i - \sum_{j>i} a_{ij} \sigma_j \right) / a_{ii} \\
&\vec{z} = \vec{v} + \vec{\sigma}
\end{aligned}
$$

Apply both the CG method and the CG SSOR method to the test problems in *main.cc* and compare their performance. Also compare to the SSOR solver from the last exercise (set the solver through the macros at the beginning of *main.cc*). Increase the resolution of the groundwater problem until the time required for solving the resulting system with the CG SSOR method exceeds three minutes. How many degrees of freedom (in this case: grid cells) correspond to this problem?

5 Points