Technische Universität Berlin **WS 2013/2014**
Fakultät II – Institut f. Mathematik
Instructor: Peschka Handed out: 22.01.
Tutor: Marschall Return during lecture: 29.01.

11. Assignment „Numerische Mathematik für Ingenieure II“

http://www.moses.tu-berlin.de/Mathematik/

Solving systems of linear equations - gradient descent

---

1. **Exercise:** Minimization with constraint **8 points**

Let $A \in \mathbb{R}^{n\times n}$ a symmetric, positive definite matrix (SPD) and $b \in \mathbb{R}^n$ arbitrary. Furthermore for $m \in \mathbb{N}$, $1 \le m \le n$ let $B \in \mathbb{R}^{m\times n}$ be a matrix of rank m and $c \in \mathbb{R}^m$.

(a) Show that the task 'Minimize $J(x) = \frac{1}{2}x^\top Ax - x^\top b$ subject to the linear equality constraint $Bx = c$.' leads to the system of linear equations

$$(\star) \qquad \begin{pmatrix} A & B^\top \\ B & 0 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix}.$$

Hint: You may use the method of Lagrange multipliers for multiple constraints, i.e., consider derivatives of $L(x, \lambda) = J(x) + \lambda^\top(Bx - c)$.

(b) Show that $(\star)$ is invertible.
Hint: First show $BA^{-1}B^\top\lambda = BA^{-1}b - c$. Then explain, why $BA^{-1}B^\top$ is invertible.

(c) Consider $A \in \mathbb{R}^{n\times n}$ obtained from a Galerkin approximation of the bilinear form

$$a(u, v) = \int_\Omega \nabla u \cdot \nabla v \, \mathrm{d}\Omega,$$

with the discrete subspace $V_h \subset H^1(\Omega)$ (no essential boundary conditions). You want to minimize $\frac{1}{2}a(u_h, u_h) - f(u_h)$ for $u_h \in V_h$ subject to the constraint ....

i) ... of homogeneous Dirichlet bc's $w_i(x_j) = 0$ for $x_j \in \partial\Omega$. Describe what is B in $(\star)$ that is equivalent to minimization subject to constraints?
Hint: To keep notation simple you may assume $x_j \in \partial\Omega$ for $j = 1 \ldots m$.

ii) ... of $\int u \, \mathrm{d}\Omega = 0$. Describe how you need to choose B for the discrete problem $(\star)$?
Hint: Use the mass matrix M and consider the product $(1, ..., 1)M$.

2. **Exercise:** Properties of gradient descent **8 points**

Let $A \in \mathbb{R}^{n\times n}$ a SPD matrix and $b \in \mathbb{R}^n$ given. Consider minimization problem

$$J(x) = \tfrac{1}{2}x^\top Ax - b^\top x,$$

solved iteratively by gradient descent. For given x_i define a sequence $(x_1, x_2, ...x_n)$ with $x_n \in \mathbb{R}^n$ so that

$$x_{n+1} = x_n + \gamma_n r_n$$

with $r_n = b - Ax_n \in \mathbb{R}^n$ and $\gamma_n = \frac{r_n^\top r_n}{r_n^\top A r_n} \in \mathbb{R}$.

(a) Show $J(x_{n+1}) \le J(x_n)$ with equality only if x_n is already a solution.

(b) Show that for given direction r_n that $s = \gamma_n$ makes $J(x)$ minimal along the line $x_n + sr_n$.

(c) Let $A \in \mathbb{R}^{2\times 2}$. Explicitly confirm the formula for the convergence speed

$$\|e_n\|_A^2 = \left(\frac{\rho - 1}{\rho + 1}\right)^{2n} \|e_0\|_A^2, \qquad \text{where} \quad \|e\|_A^2 = e^\top Ae,$$

where $\rho = \max \lambda_i / \min \lambda_i$ is the condition number of A derived from its eigenvalues λ_i.

(d) Let

$$A = \begin{pmatrix} 17 & 6 \\ 6 & 8 \end{pmatrix}, \qquad b = \begin{pmatrix} 3 \\ 4 \end{pmatrix}.$$

Find x_i with $\|e_0\|_A = 1$ such that gradient descent convergences i) as fast as possible or ii) as slow as possible.

3. Programming exercise: Gradient descent **8 points**

(a) Write a `function [x,iter]=gradientdescent(A,b,tol,x0)` which solves $Ax = b$ using gradient descent as described above. Here `x` $= x_n$ and `iter` is the number of required gradient descent iterations. The sparse SPD matrix `A` and `b` are provided. Iterate gradient descent while $\|r_n\| >$ `tol`. The optional argument `x0` provides a starting vector for the iteration, if it is not provided use $x_0 = b$. **Hint:** Use `nargin`.

(b) Apply gradient descent to the matrix A from (2d) with $x_0 = b$ and compare with the fast/slow initial vector from (2d) i), ii). Also compare with random initial vector with $\|e_0\|_A = 1$. Visualize the iterations by plotting lines between the iterates and the isosurfaces of J and export to `gradient.pdf`. How many iterations n does gradient descent need to converge in each case?

Hint: The MATLAB function `contour` or `contourf` might be useful.

(c) Apply gradient descent to the matrix $A = K_n$ with $n = 100$ from assignment 1, exercise 1 with $b = (1, ..., 1)^\top$, $x_0 = b$. How many iterations do need to converge? Use the exact solution to compute $\|e_0\|_A$, $\|e_n\|_A$ and estimate the maximum number of iterations that you would have expected.

Hint: Eigenvalues and -vectors are as in the lecture, e.g. $(v_i)_j = \sqrt{2}\sin(\pi ij/(n+1))$.

(d) Let $A = \mathbb{I}_n + \frac{\tau}{h^2} K_n$ with $\mathbb{I}_n$ the $n \times n$ identity matrix, $h = 1/(n+1)$ and $\tau = 1/1000$. This is the implicit Euler discretization of $\partial_t u - \partial_{xx} u = 0$, $u(t,0) = u(t,1) = 0$ using finite-differences in 1D on $(0,1)$. Use $u_h^0(x_i) = \sin(\pi x_i)$ as initial data. Compute the discrete solution $Ax = b$ for $n = 100$ at $T = 0.1$ using gradient descent with $x_0 = b$. You have $b = u_h^k$ and $x = u_h^{k+1}$. Compare with the exact solution. How many iterations does gradient descent need to converge per time-step on average?

Remark: Use `tol=1d-7` for 3b,3c,3d). After this we say a solution has *converged.*

total sum: 24 points

As usual, use sparse matrices (where this is useful).