AMSC/CMSC 666

NUMERICAL ANALYSIS I

HOMEWORK # 3 (Pbs 1-3 due Tue Oct 22, Pbs 4-7 due Thu Oct 31)

1 (20 pts). *Conjugate gradient.* Consider the two-point boundary value problem in $(0, 1)$:

$$-u'' + u = f, \qquad u(0) = u(1) = 0. \tag{1}$$

(a) Let $\{x_i\}_{i=0}^{N+1}$ be a uniform partition of $(0, 1)$ with meshsize $h = 1/(N+1)$. Derive the discrete equations resulting from applying centered differences to (1). Write the equations in matrix form $\mathbf{AU} = \mathbf{F}$, where $\mathbf{U} = (U_i)_{i=1}^{N}$ is the vector of nodal values and $\mathbf{F} = (f(x_i))_{i=1}^{N}$.

(b) Show that $\mathbf{A}$ is strictly diagonally dominant, symmetric and positive definite.

(c) Write a MATLAB function `[x,k] = cg(A,b,x0,tol)` which implements the CG method for an $n \times n$ symmetric and positive definite matrices `A`, right-hand side `b`, and starting value `x0`. The program should compute the 2-norm of the residual and stop when such a norm is less than a given tolerance `tol`, giving the current iterate vector `x` and number of iterations `k`.

(d) Write a MATLAB function `[x,k] = gradient(A,b,x0,tol)` which implements the steepest descent (or gradient) method. The arguments have the same meaning as in (c).

(e) Let f be the right-hand side of (1) corresponding to the exact solution $u(x) = \sin(\pi x) - \sin(3\pi x)$. Run the functions `cg` and `gradient` for $N = 10, 20, 40$, `x0=0` and `tol =` 10^{-8}. Plot the computed solutions together with $u(x)$. Plot also the log of the 2-norm of the residual in terms of the log of the number of steps, and draw conclusions.

2 (15 pts). *Singular matrices.* Let $\mathbf{A}$ be a symmetric positive semi-definite matrix. Let the linear system $\mathbf{Ax} = \mathbf{b}$ be consistent, that is, $\mathbf{b}$ belongs to the range of $\mathbf{A}$ and thus solution exists. Prove that with initial guess $\mathbf{x}^{(0)} = 0$, the conjugate gradient method (CG) is guaranteed to produce a solution without component in the kernel of $\mathbf{A}$. To this end proceed as follows.

(a) Show that the singular value decomposition of $\mathbf{A}$ can be written as $\mathbf{A} = \mathbf{U}\Sigma\mathbf{U}^T$ with Σ diagonal, and study the properties of Σ.

(b) Derive a system equivalent to $\mathbf{Ax} = \mathbf{b}$ for $\hat{\mathbf{x}} = \mathbf{Ux}$ and $\hat{\mathbf{b}} = \mathbf{Ub}$, and study the structure of $\hat{\mathbf{b}}$.

(c) Show that CG with $\mathbf{x}^{(0)} = 0$ is equivalent to CG for a symmetric positive definite matrix with zero initial guess.

3 (10 pts). *Preconditioning.* Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ be symmetric and positive definite matrices, and let $\mathbf{b} \in \mathbb{R}^n$. Consider the quadratic function $Q(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{Ax} - \mathbf{x}^T\mathbf{b}$ for $\mathbf{x} \in \mathbb{R}^n$ and a *descent* method to approximate the solution of $\mathbf{Ax} = \mathbf{b}$:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k.$$

(a) Formulate the *steepest descent* (or gradient) method and write a pseudocode which implements it.

(b) Let $\mathbf{B}^{-1}$ be a preconditioner of $\mathbf{A}$. Show how to modify the steepest descent method to work for $\mathbf{B}^{-1}\mathbf{Ax} = \mathbf{B}^{-1}\mathbf{b}$, and write a pseudocode. Note that $\mathbf{B}^{-1}A$ may not be symmetric.

4 (10 pts). *Quadratic convergence.* Let $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$ be a map of class C^2 with a zero $\mathbf{x}_* \in D$. Show that if F satisfies $F''(\mathbf{x}_*)\langle \mathbf{h}, \mathbf{h}\rangle \neq 0$ for all $\mathbf{h} \neq 0$, $\mathbf{h} \in \mathbb{R}^n$, then the convergence of Newton method is at most quadratic, namely

$$\frac{\|\mathbf{x}_{k+1} - \mathbf{x}_*\|}{\|\mathbf{x}_k - \mathbf{x}_*\|^2} \geq C \neq 0 \quad \forall k \geq 0.$$

5 (15 pts). *Shamanskii method.* Let $F : \mathbb{R}^n \to \mathbb{R}^n$ have a Lipschitz continuous derivative and suppose $F(\mathbf{x}^*) = 0$ and $F'(\mathbf{x}^*)$ is nonsingular. Consider the following modification of Newton's method in which F' need be evaluated only every other iterate:

$$\mathbf{x}_{k+1} = \begin{cases} \mathbf{x}_k - F'(\mathbf{x}_k)^{-1}F(\mathbf{x}_k), & k \text{ even}, \\ \mathbf{x}_k - F'(\mathbf{x}_{k-1})^{-1}F(\mathbf{x}_k), & k \text{ odd}. \end{cases}$$

Set $\mathbf{y}_k := \mathbf{x}_{2k}$ and prove that the combined method has order of convergence at least 3. To this end derive the expression

$$\mathbf{y}_k = G(\mathbf{y}_{k-1}) - F'(\mathbf{y}_{k-1})^{-1} F(G(\mathbf{y}_{k-1})),$$

and recall that Newton's function $G(\mathbf{x}) = \mathbf{x} - F'(\mathbf{x})^{-1}F(\mathbf{x})$ satisfies $\|G(\mathbf{x}) - \mathbf{x}^*\| \le C\|\mathbf{x} - \mathbf{x}^*\|^2$. Use this result to determine the order of convergence of the original method.

6 (15 pts). *Jacobi-Newton method.* The Jacobi iteration for solving systems of linear equations can also be applied to nonlinear systems. For the $n \times n$ system $F(\mathbf{x}) = (f_i(\mathbf{x}))_{i=1}^n = 0$ with $F \in C^2$, the iteration reads: *for $k \ge 0$ solve*

$$f_i(x_1^k, \cdots, x_{i-1}^k, x_i^{k+1}, x_{i+1}^k, \cdots, x_n^k) = 0, \qquad 0 \le i \le n.$$

Thus each iteration involves solving n scalar nonlinear equations.

(a) If we use one iteration of Newton's method to solve each of these equations we get the Jacobi-Newton method. Show that one such iteration reads

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{D}(\mathbf{x}_k)^{-1} F(\mathbf{x}_k) = G(\mathbf{x}_k), \tag{2}$$

where $\mathbf{D}(\mathbf{x})$ stands for the diagonal of $F'(\mathbf{x})$.

(b) Conversely, we can first formulate a Newton step and then perform one Jacobi iteration for the corresponding linear system. Show that the resulting Newton-Jacobi method coincides with (2).

(c) Determine under what conditions this algorithm converges, and under what conditions does it superlinearly and quadratically.

7 (15 pts). *Newton method.* This is about experimenting with the Newton's method and some of its variants.

(a) Write a simple MATLAB program `Newton(F,DF,x,tol,miter)` to solve an n-by-n nonlinear system of equations $F(\mathbf{x}) = 0$ by the Newton's Method. The input parameters are the functions F its Jacobian DF, the starting point x, an absolute error tolerance `tol`, and the maximum number of iterations `miter`. Solve the resulting linear systems using the MATLAB command "\".

(b) The following system has four zeros in the domain $(-4, 4) \times (-4, 4)$

$$f(x, y) = x^2 + xy^3 - 9 = 0, \qquad g(x, y) = 3x^2y - y^3 - 4 = 0.$$

Use the command `contour` to plot the zero level sets of both f and g in the same picture to determine reasonable initial guesses (use `help contour` to find out information about the command). Use the commands `hold on` and `hold off` to produce this picture. Use a syntax such as:

```
x = -4:0.1:4;
y = -4:0.1:4;
[X,Y] = meshgrid(x, y);
contour(x, y, X.^2+X.*Y.^3-9, [0 0])
```

(c) Use `Newton` to approximate the four zeros with `tol`$= 10^{-7}$. Determine the number of iterations and the rate of convergence. To this end show that $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|/\|\mathbf{x}_k - \mathbf{x}^*\| \to 1$ for superlinear convergent methods.

(d) Write a Matlab code `Newdiff` that replaces the Jacobian with forward differences with $h = 10^{-7}$. Repeat (b) and compare results. You can use the code `nsol`.

(e) Repeat (b) with the Broyden method, which is implemented in `brsol.m`. Both codes `nsol` and `brsol.m` are available from the SIAM website `https://archive.siam.org/books/kelley/fr16/index.php`