Applied Numerical Methods
(MATH 151B-Lecture 1, Spring 2016)
Assignment 4

Note:

- **Due day**: 9:00 a.m., 23th May (Monday). Assignments handed after the due date will not be counted.

1. Consider the following functional:

$$J(x) = \frac{1}{2}||Ax - b||_2^2 \tag{1}$$

where A is a $n \times n$ matrix, x is an n vector, b is an n vector, $||v||_2 = \sqrt{\sum_{i=1}^n v_i^2}$ is the 2-norm.

(a) Show the following result: for $h \in \mathbb{R}^n$, we have

$$J(x+h) = J(x) + \langle A^T Ax - A^T b, h \rangle + \frac{1}{2}||Ah||_2^2 .$$

where $\langle v, w \rangle = \sum_{i=m}^n v_i w_i$ is the standard inner product in $\mathbb{R}^m$. You can simplify your writing by writing $||v||_2 = \langle v, v \rangle$. You may also use equalties for real inner products such as $||x+y||^2 = ||x||^2 + 2\langle x, y \rangle + ||y||^2$ freely.

(b) Use Part (a), or otherwise, to show that if x^* satisfies

$$A^T Ax^* = A^T b$$

then we have

$$J(x) \geq J(x^*)$$

for all $x \in \mathbb{R}^n$. Hint: let $h = x^* - x$.

(c) Letting $h = -\alpha(A^T Ax - A^T b)$, then by directly substitution into the form $J(x+h)$, we get (Nothing is needed to be done in this part)

$$J(x+h) = J(x) - \alpha||A^T Ax - A^T b||_2^2 + \frac{1}{2}\alpha^2||AA^T Ax - AA^T b||_2^2 .$$

Now, for a given x, write $g_x(\alpha)$ as

$$g_x(\alpha) := J(x) - \alpha||A^T Ax - A^T b||_2^2 + \frac{1}{2}\alpha^2||AA^T Ax - AA^T b||_2^2 .$$

Please try to show that α which minimizes the above expression $g_x(\alpha)$ is when

$$\alpha = \frac{||A^T Ax - A^T b||_2^2}{||AA^T Ax - AA^T b||_2^2} \,.$$

(Hint: either by completing the square, or by differentiation.)

(d) Introduce the steepest descent method to minimize $J(x)$, i.e. to find x^* such that

$$J(x^*) = \min_{x \in \mathbb{R}^n} J(x) \,.$$

(You do not need to compute anything. But please use stepsize α_k that minimize $g_{x^k}(\alpha)$ as discussed in the previous question.)

(e) (It is required to use a program to finish this sub-question. Only programs written in C / C++ / Matlab / Octave is acceptable. Hand in the code via CCLE. Print your tabulated results and graphs.) Compute 2 steps of the steepest descent method for the following functional:

$$J(x) = \frac{1}{2} \left\| \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix} x - \begin{pmatrix} 3 \\ 5 \end{pmatrix} \right\|_2^2 \tag{2}$$

with initial guess $x^{(0)} = (2, 3)$ and the following 2 update rules for stepsize:

i. With stepsize α_k that minimize $g_{x^k}(\alpha)$ as described in the previous question,

ii. Replacing the previously-mentioned stepsize rule by a fixed stepsize $\alpha_k = \alpha = 1$ for all k, i.e. we do NOT perform the line-search step.

You may also notice some of the above iterations may not be converging (if you compute more iterations). This is okay if you observe this phenomenon.

2. Consider the following eigenvalue problem for an $n \times n$ matrix A:

$$Ax = \lambda x \,. \tag{3}$$

Let A be symmetric positive definite, and therefore there exists an orthonormal basis $\{v_i\}_{i=1}^n$ and their corresponding eigenvalues $\{\lambda_i\}_{i=1}^n$such that for any vector $x \in \mathbb{R}^n$, we can write:

$$x = \sum_{i=1}^n a_i v_i \,,$$

WIth this basis,

$$Ax = \sum_{i=1}^n \lambda_i a_i v_i \,.$$

Now let $\lambda_1 > \lambda_i$ for all $i \geq 2$. (all eigenvalues are positive because A is positive definite.)

(a) Show inductively that

$$A^k x = \sum_{i=1}^n \lambda_i^k a_i v_i \,.$$

(b) Let $x^{(0)}$ be given such that $\langle x^{(0)}, v_1 \rangle > 0$, and define $\{x^{(i)}\}_{i=1}^{\infty}$ be

$$x^{(i)} = Ax^{(i-1)}/||Ax^{(i-1)}||_2 \,.$$

Show that $x^i \to v_1$ as $i \to \infty$. (Therefore $Ax^i \to \lambda_1 v_1$ as $i \to \infty$. Note that you will definitely need the condition $\langle x^{(0)}, v_1 \rangle > 0$.)

(Important hint: to compute explicitly and trace back all the normalization constant $||Ax^{(i)}||$ each time in the iteration is very difficult. Instead, one can let $\mu_i = ||Ax^{(i-1)}||^{-1}$. Then one have

$$x^{(i)} = \mu_i A x^{(i-1)} \,.$$

Therefore, we have

$$x^{(n)} = \left(\prod_{i=1}^{n} \mu_i \right) A^n x^{(0)} \,,$$

where we keep in mind that the whole clumsy constant $\prod_{i=1}^{n} \mu_i$ will normalize $x^{(n)}$ to be such that $||x^{(n)}||_2 = 1$ by definiton. With this knowledge, you can express $x^{(n)}$ by λ_i and a_i (where $x^{(0)} = \sum_{i=1}^{n} a_i v_i$) without the necessity to manipulate/compute each of the coefficients μ_i.)

(c) (It is required to use a program to finish this sub-question. Only programs written in C / C++ / Matlab / Octave is acceptable. Hand in the code via CCLE. Print your tabulated results and graphs.) Write a program to use the power method as defined in Part (b) to compute the largest eigenvalue of the matrix

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \tag{4}$$

with initial guess $x^{(0)} = (1, 0)$.

3. (It is required to use a program to finish this question. Only programs written in C / C++ / Matlab / Octave is acceptable. Hand in the code via CCLE. Print your tabulated results and graphs.) Consider the following system of nonlinear equations:

$$\begin{cases} x_1 + x_2 + x_3 + x_4 = 1 \\ x_1^2 + x_2^2 + x_3^2 + x_4^2 = 1 \\ x_1^3 + x_2^3 + x_3^3 + x_4^3 = 1 \\ x_1^4 + x_2^4 + x_3^4 + x_4^4 = 1 \end{cases}$$

Now define $F(x) = \begin{bmatrix} x_1 + x_2 + x_3 + x_4 - 1 \\ x_1^2 + x_2^2 + x_3^2 + x_4^2 - 1 \\ x_1^3 + x_2^3 + x_3^3 + x_4^3 - 1 \\ x_1^4 + x_2^4 + x_3^4 + x_4^4 - 1 \end{bmatrix}$,

(a) Let $p = (\frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5})$. Write down the ODE for $x(\lambda), \lambda \in [0,1]$ given by the homotopy method to solve with initial vaue $x(0) = p$ in the form of:

$$x'(\lambda) = f(\lambda, x(\lambda))$$

where $f : [0,1] \times \mathbb{R}^4 \to \mathbb{R}^4$ is now a vector-valued function.
Reminder: Homotopy method is to find a curve $x(\lambda)$ such that $G(\lambda, x(\lambda)) = 0$ for all λ, where $G(\lambda, x)$ is defined as

$$G(\lambda, x) := \lambda F(x) + (1-\lambda)[F(x) - F(x(0))]\,.$$

(You can leave the inverse of a matrix untouched and do not need to compute "symbolicaly" the inverse of the matrix.)

(b) Compute an approximation of $x(\lambda)$ using the following fourth-order Runge-Kutta method (RK4):

$$x_{i+1} = x_i + \frac{1}{6}(F_1 + 2F_2 + 2F_3 + F_4)$$

where

$$\begin{cases} F_1 = hf(t_i, x_i) \\ F_2 = hf(t_i + \frac{1}{2}h, x_i + \frac{1}{2}F_1) \\ F_3 = hf(t_i + \frac{1}{2}h, x_i + \frac{1}{2}F_2) \\ F_4 = hf(t_i + h, x_i + F_3)\,. \end{cases}$$

with step-size $\Delta\lambda = 0.01$.