COL726 Assignment 4

3 – 21 March, 2022

Note: All answers should be accompanied by a rigorous justification, unless the question explicitly states that a justification is not necessary.

Updated text is highlighted in blue.

1. Suppose $\mathbf{M} \in \mathbb{C}^{3m\times 3m}$ is a "block triangular" matrix made of $m \times m$ blocks,

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{C} \\ & \mathbf{D} & \mathbf{E} \\ & & \mathbf{F} \end{bmatrix}.$$

   Don't assume that any of the blocks themselves are upper triangular.

   (a) Show that λ is an eigenvalue of $\mathbf{M}$ if and only if it is an eigenvalue of $\mathbf{A}$, $\mathbf{D}$, or $\mathbf{F}$.

   (b) Suppose λ is an eigenvalue of $\mathbf{D}$ with eigenvector $\mathbf{y}$, but it is not an eigenvalue of $\mathbf{A}$ or $\mathbf{F}$. Show how to efficiently find the corresponding eigenvector of $\mathbf{M}$. Your answer is allowed to involve solving $m \times m$ linear systems, but not $3m \times 3m$ ones.

2. Show that Rayleigh quotient iteration indeed converges cubically as claimed. That is, suppose $\mathbf{q}_j$ is an eigenvector of a symmetric matrix $\mathbf{A} \in \mathbb{R}^{m\times m}$ with eigenvalue $\lambda_j > 0$. Let's consider two consecutive iterates of Rayleigh quotient iteration, $\mathbf{v}^{(k)}$ and $\mathbf{v}^{(k+1)}$, and denote them as $\mathbf{v}$ and $\mathbf{w}$ for brevity. Show that if $\mathbf{v} = \mathbf{q}_j + \delta\mathbf{v}$, then $\mathbf{w} = \mathbf{q}_j + \delta\mathbf{w}$ with $\|\delta\mathbf{w}\| = O(\|\delta\mathbf{v}\|^3)$.

   To simplify the calculation, you may assume that $\delta\mathbf{v} \perp \mathbf{v}$, because we know that any component of $\delta\mathbf{v}$ parallel to $\mathbf{v}$ will only change its length but not affect the Rayleigh quotient.

3. Let $\mathbf{A} \in \mathbb{R}^{m\times m}$ be a symmetric tridiagonal matrix, i.e. only entries a_{ij} with $i-1 \le j \le i+1$ are nonzero.

   (a) In the QR factorization $\mathbf{A} = \mathbf{QR}$, what is the sparsity pattern of $\mathbf{Q}$ and $\mathbf{R}$? That is, which entries of $\mathbf{Q}$ and $\mathbf{R}$ are nonzero?

   (b) Show that the first iterate of the QR algorithm, $\mathbf{A}^{(1)} = \mathbf{RQ}$, is also tridiagonal. Thus, all iterates $\mathbf{A}^{(k)}$ in the QR algorithm will remain tridiagonal.

   (c) What can you say about the sparsity pattern of the orthogonal matrices $\mathbf{U}^{(k)}$ in simultaneous iteration (denoted $\hat{\mathbf{Q}}^{(k)}$ or $\underline{\mathbf{Q}}^{(k)}$ in Trefethen & Bau)?

4. Suppose I have a system of nonlinear equations

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(x_1, \dots, x_n) \\ \vdots \\ f_n(x_1, \dots, x_n) \end{bmatrix} = \mathbf{0}$$

in which each equation $f_i(\dots) = 0$ is easy to solve for x_i holding the other variables fixed. I do this in parallel for each variable, computing $x_i^{(k+1)}$ so that $f_i(\dots, x_{i-1}^{(k)}, x_i^{(k+1)}, x_{i+1}^{(k)}, \dots) = 0$.

   (a) Show that the Jacobian of this fixed point iteration at the solution $\mathbf{x}^*$ is $\mathbf{G} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{J}$, where $\mathbf{J}$ is the Jacobian of $\mathbf{f}$, and $\mathbf{D}$ is a diagonal matrix containing just the diagonal entries of $\mathbf{J}$.

   (b) Show that if $\mathbf{J}(\mathbf{x}^*)$ is diagonally dominant by rows, i.e. $|J_{ii}| > \sum_{j \neq i} |J_{ij}|$ for all i, j, then convergence is guaranteed if started close enough to the solution.

   Hint: Use an appropriate choice of induced norm on $\mathbf{G}$.

5. In some problems, we have a time-varying matrix $\mathbf{A}(t)$, and we want to find the time t^* at which it becomes singular. For example, the columns of $\mathbf{A}$ may contain coordinates of m points in $\mathbb{R}^m$, which become coplanar at time t^*. Even in the simplest case $\mathbf{A}(t) = \mathbf{A}_0 + \mathbf{A}_1 t$, this problem cannot be solved with linear algebra alone.

   (a) Give an algorithm to compute the determinant of a matrix $\mathbf{A} \in \mathbb{R}^{m\times m}$ in $O(m^3)$ flops, based on LU factorization with pivoting.

   (b) Implement Python functions `bisect(A0, A1, t0, t1, eps)` and `secant(A0, A1, t0, t1, eps)` to find the time $t^* \in [t_0, t_1]$ such that $\mathbf{A}(t^*)$ is singular. The two methods should use bisection and the secant method respectively on $\det(\mathbf{A}(t))$. You may use `scipy.linalg.det` to compute the determinant, which uses an algorithm similar to the one in part (a).

   In `bisect`, assume that $[t_0, t_1]$ is a bracket for $\det(\mathbf{A}(t))$, and return a pair (t_k, k) containing the result t_k and iteration count k such that $|t_k - t^*|/|t_1 - t_0| \leq \epsilon$. In `secant`, use t_0 and t_1 as the initial guesses, and return (t_k, k) such that $|\det(\mathbf{A}(t_k))|/|\det(\mathbf{A}(t_0))| \leq \epsilon$.

   (c) Look up Jacobi's formula to compute $\frac{\mathrm{d}}{\mathrm{d}t}\det(\mathbf{A}(t))$, and use it to implement Newton's method as a function `newton(A0, A1, t0, eps)`. Like with the secant method, return (t_k, k) when $|\det(\mathbf{A}(t_k))|/|\det(\mathbf{A}(t_0))| \leq \epsilon$.

   Test your code on a problem with a known solution: With $m = 5$, pick a random rank-4 matrix $\mathbf{X}$ (e.g. $\mathbf{X} = \mathbf{T}\mathbf{W}$ with random $\mathbf{T} \in \mathbb{R}^{5\times 4}$, $\mathbf{W} \in \mathbb{R}^{4\times 5}$) and a random full-rank matrix $\mathbf{Y}$, and define $\mathbf{A}_0 = \mathbf{X} - h\mathbf{Y}$, $\mathbf{A}_1 = 3h\mathbf{Y}$, $t_0 = 0$, $t_1 = 1$ for some small h (e.g. $h = 0.01$). Then $t^* = \frac{1}{3}$ is a solution. Make a plot of iteration count k as a function of tolerance $\epsilon = 2^{-4}, 2^{-5}, \dots, 2^{-20}$ (with a logarithmic x-axis) for all three methods and include it in your PDF.

Bonus question (no marks): Will the convergence behaviour change in the above example if $\mathbf{X}$ is a rank-2 matrix? Why or why not?

6. Given a system of equations $\mathbf{f}(\mathbf{u}) = \mathbf{v}$, we can always define an equivalent system by transforming the domain and the codomain by invertible linear transformations, say $\mathbf{u} = \mathbf{D}\mathbf{x}$ and $\mathbf{v} = \mathbf{C}\mathbf{y}$ where $\mathbf{C}, \mathbf{D}$ are nonsingular. This yields the system of equations $\mathbf{g}(\mathbf{x}) = \mathbf{y}$, where $\mathbf{g}(\mathbf{x}) := \mathbf{C}^{-1}\mathbf{f}(\mathbf{D}\mathbf{x})$ and $\mathbf{y} := \mathbf{C}^{-1}\mathbf{v}$.

   Show that Newton's method is completely unaffected by this transformation. That is, whether we apply it to the transformed system with initial guess $\mathbf{x}^{(0)}$ or to the original system with initial guess $\mathbf{u}^{(0)} = \mathbf{D}\mathbf{x}^{(0)}$, we always get equivalent iterates $\mathbf{u}^{(k)} = \mathbf{D}\mathbf{x}^{(k)}$.

   Notes: You can derive the Jacobian of $\mathbf{g}$ using the property $\mathbf{g}(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{g}(\mathbf{x}) + \mathbf{J}_{\mathbf{g}}(\mathbf{x})\Delta\mathbf{x} + O(\|\Delta\mathbf{x}\|^2)$.

7. Suppose $\mathbf{A} \in \mathbb{R}^{m\times m}$ is a square matrix whose factorization $\mathbf{A} = \mathbf{L}\mathbf{U}$ is known. You are now given two arbitrary vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$, and you wish to compute the LU factorization of $\mathbf{A}' = \mathbf{A} + \mathbf{x}\mathbf{y}^T$. (As mentioned in class, this problem arises in efficient implementations of Broyden-type methods.) Find an $O(m^2)$ algorithm to do so. Assume that no pivoting issues arise, i.e. any diagonal entries encountered in the algorithm are sufficiently large.

   Hint: Try finding the first column and row of the new $\mathbf{L}'$ and $\mathbf{U}'$. Then show that it is possible to proceed recursively on an $(m-1)\times(m-1)$ matrix.

Collaboration policy: Refer to the policy on the course webpage.

If you collaborated with others to solve any question(s) of this assignment, give their names in your submission. If you found part of a solution using some online resource, give its URL.

Submission: This assignment has two submission forms on Gradescope. In one form, you have to submit a PDF of your answers for all questions, and in the other you have to submit your code for the programming questions. Both submissions must be uploaded before the assignment deadline.

Code submissions should contain one file `a4q5.py` which contains the requested functions and any helper functions. You are permitted but not required to include the code for producing the plot. The plot itself should be included in your PDF.