NATIONAL UNIVERSITY OF SINGAPORE

PC5215 – NUMERICAL RECIPES WITH APPLICATIONS

(Semester I: AY 2024-25)

Time Allowed: 2 Hours

INSTRUCTIONS TO STUDENTS

1. Write your student number on the answer book. Do not write your name.
2. This assessment paper contains FOUR questions and comprises THREE printed pages (including this cover page).
3. Students must answer ALL questions; each question carries equal marks.
4. Students should write the answers for each question on a new page.
5. This is a CLOSED BOOK examination.
6. Non-programable calculators are allowed.

1. This question concerns the machine representation of the float-point numbers in a computer. In Python, the default float type for real numbers is based on the IEEE double precision format, which has 64-bit in total, including a sign bit, 11-bit for the exponent, and 52-bit for the mantissa.
    a. Define the concept of machine ϵ. What is the machine ϵ in double precision? In a typical numerical calculation, how many decimal digits can you trust to be accurate?
    b. Physical constants such as the speed of light c, Boltzmann constant k, or Planck constant h, tend to be very large or very small if we use SI units. Will this cause an accuracy problem if you use SI units? Answer this question by giving the largest and smallest possible values you can have in IEEE double precision.
    c. The solution to the quadratic equation $ax^2 + bx + c = 0$ can be computed in two ways:
    $$\frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{or} \quad -\frac{2c}{b + \sqrt{b^2 - 4ac}}$$
    Assuming $b > 0$ and a small, which of the two alternative ways of computation is more accurate? State why.

a) The machine ϵ is the smallest value such that 1 + ϵ can be represented exactly in the machine. It is the next value after 1. In IEEE double precision, this number is (in binary) 1.000…01 = 1 + 2^{-52}. So $\epsilon = 2^{-52} = 2.22 \times 10^{-16}$. In a typical computation, we expect $\log_{10} \epsilon$ = 15 significant figures. So ϵ represents the relative error. b) Based on the format of 11 bits for the exponent, the largest values is 2^{2024} = 1.8 x $10^{308}$and the tiniest number is 1.8 x 10^{-308}. Based on this reasoning, SI is OK, as k, c, or h can fit into it. But the density of representable numbers is less dense away from 1. So, I still prefer to use units such that the values of computation are not too far from 1, e.g., atomic units. c) the first one has a catastrophic cancelation, and the second form does not when a is mall. So, the second form is preferred.

2. Consider $N \times N$ square matrices of A, B, C, and D forming a $2N \times 2N$ block matrix.
    a. Determine the LU decomposition as block matrix equation
    $$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} I & 0 \\ \alpha_{21} & I \end{bmatrix} \begin{bmatrix} \beta_{11} & \beta_{12} \\ 0 & \beta_{22} \end{bmatrix}$$
    Here, I is $N \times N$ identity matrix, 0 is the zero matrix, and the rest of the entries need to be determined by solving matrix equations. Since they are matrices, the order is important.

b. Give an expression for the determinant of $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$ based on the above formula in terms of the original four matrices A to D of a small block.

c. What is the computational complexity of computing the determinant?

a) Matrix multiplying the right-hand side, we get equations $A = \beta_{11}$, $B = \beta_{12}$, $C = \alpha_{21}\beta_{11}$, $D = \alpha_{21}\beta_{12} + \beta_{22}$. *Solve as matrix equation, we find* $\alpha_{21} = CA^{-1}$, $\beta_{22} = D - CA^{-1}B$. *b)* $\det \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \det(A)\det(D - CA^{-1}B)$. *c) The computational complexity is (N^3). Comparing with the original matrix of 2N x 2N, it can be 8 times faster if we use the small sub-matrices.*

3. We consider the optimization of the function with two variables $x = (x_1, x_2)$,

$$f(x_1, x_2) = \frac{3}{2}x_1^2 + 2x_1x_2 + 3x_2^2 - 2x_1 + 8x_2$$

.

a. If we use the steepest descent method, starting from the origin $x^{(0)} = (x_1, x_2) = (0,0)$, what is the search direction n? Show that the condition to stop at the minimum of $f(x^{(0)} + \lambda n)$ is when n is orthogonal to the gradient of f at the new point $x^{(1)}$.

b. Compute this new point $x^{(1)} = x^{(0)} + \lambda n$. Write down the definition of the conjugate direction n' that is conjugate with respect to the given function f and the search direction n in part a.

c. In the conjugate gradient method, we can compute the conjugate direction $n' = g' + \gamma n$ by a linear combination of the previous search direction n and a parameter $\gamma = |g'|^2/|g|^2$, here g is the negative gradient in the previous step and g' is the negative gradient in the current step. Verify that this new search direction n' is conjugate to n according to the criterion of part b. Determine the location of minimum x_{min} of f using this method.

a) The steepest descent direction is the negative gradient direction, i.e., $g = -\nabla f = (-3x_1, +2x_2 + 2, -6x_2 - 2x_1 - 8)$. *At (0,0), we get n = (2,-8). We search for the minimum with* $x^{(0)} + \lambda n$. *The minimum is obtained by taking derivative with respect to* λ, *and set to 0. This gives* $n.\,g' = 0$. *b) Since* $x^{(0)} = (0,0)$, *we consider* $x^{(1)} = \lambda n = (2\lambda, -8\lambda)$. *The orthogonal condition* $n \cdot g' = 0$ *gives* $20\lambda + 4 - 8 \times 44\lambda + 64 = 0$. *Or* $\lambda = \frac{17}{83} \approx 0.20482$, *and* $x^{(1)} = (2\lambda, -8\lambda) \approx (0.4096, -1.6386)$. *The next step conjugate direction n′ needs to satisfy* $n^T A\, n' = 0$, *where* $A = \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix}$. *c) The new negative gradient is* $g' = (10\lambda + 2, \lambda - 4) \approx (4.0482, 1.01204)$. *This gives* $\gamma = \frac{|g'|^2}{|g|^2} \approx 0.25606$. *The new search direction is then* $n' = g' + \gamma n \approx (4.56031, -1.0364)$. *One can check that* $n^T A n' = 0$. *Finally, the overall minimum is at* $x^{(2)} = x^{(1)} + \lambda n'$, *which gives* $x_{min} = (2, -2)$ *(step omitted).*

4. Consider a one-dimensional mechanical system that follows the Hamiltonian dynamics, with the Hamiltonian $H(x, p) = \frac{p^2}{2m} + \frac{1}{2}kx^2$. This is the harmonic oscillator model.
   a. Write down the Hamilton equations of motion as two coupled equations for the position x and momentum p. Also, give the Euler algorithm that updates from the current state (x_n, p_n) to the next step (x_{n+1}, p_{n+1}) after a time step h. What is the truncation error of the algorithm for one step?
   b. Do the same as in part a but using a symplectic algorithm in the same order of accuracy. This is based on the concept of a canonical transform. Verify that your proposed method is indeed a canonical transform.
   c. Since the initial data (x_n, p_n) may contain errors, we may measure this error by a small phase space area $\delta x \delta p$. How does this area change when the dynamic variables (x, p) are updated according to part a and part b, respectively? From this, comment on the stability of the two respective algorithms.

a) Hamilton's equations of motion are $\dot{p} = -\frac{\partial H}{\partial x} = -kx$, $\dot{x} = \frac{\partial H}{\partial p} = p/m$. Using the forward difference for the time derivative, we obtain the Euler method as $p_{n+1} = p_n - kx_n h$, $x_{n+1} = x_n + hp_n/m$, with a truncation error of $O(h^2)$. b) For symplectic algorithm, we update sequentially, say p first, then it is $p_{n+1} = p_n - kx_n h$ (same as in a), $x_{n+1} = x_n + hp_{n+1}/m$ (use the current p). The canonical transform is verified by computing the wedge product $dp_{n+1} \wedge dx_{n+1}$ (step omitted, see lecture slides). c) the infinitesimal areas are given precisely by the wedge product. For the Euler algorithm in a, each step the area is larger than a factor 1 + kh^2/m, while in b the area is the same (Jacobian is 1). So the symplectic algorithm is stable, and Euler method unstable.

--The end -- WJS