NATIONAL UNIVERSITY OF SINGAPORE

PC5215 – NUMERICAL RECIPES WITH APPLICATIONS

(Semester I: AY 2022-23)

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 FIVE questions and comprises THREE printed pages (including this cover page).
3. Students are required to answer ALL questions; each question carry 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. Answer briefly the following questions without derivations.
    a. Define computational (time) complexity of an algorithm; state the computational complexity of the LU decomposition for matrix of size $N \times N$.
    b. Give the Simpson rule for numerical integration with three equally spaced points with spacings h; express the truncation error of the integration formula in terms of h.
    c. For a Gaussian quadrature formula with three points, what are the polynomials of the form x^n that can be integrated exactly, for integer n?
    d. Define the symplectic algorithms for Hamiltonian dynamics.

    a) The number of basic steps it takes to finish a calculation according to a given algorithm. For LU decomposition of matrix of N x N, the computational complexity is O(N³). b) $\int_0^{2h} f(x)dx = h(f_0 + 4f_1 + f_2)/3 + O(h^5)$. *Here* $f_i \equiv f(ih)$. *c) exact for polynomials 1, x, x², …, x⁵, and their linear combinations. d) a symplectic algorithm is such that from step n to step n+1 is a canonical transform, i.e.,* $\sum_j dq_j^{(n+1)} \wedge dp_j^{(n+1)} = \sum_j dq_j^{(n)} \wedge dp_j^{(n)}$ *. The algorithm preserves the symplectic structure of the Hamiltonian dynamics exactly.*

2. Although the Gaussian random numbers in one dimension with a variance σ^2 and mean value a can be generated more efficiently with the Box-Muller transformation method, here we use Monte Carlo method.
    a. The probability density $p(x)$ of the Gaussian random number is proportional to $e^{-\frac{(x-a)^2}{2\sigma^2}}$. State the Metropolis algorithm using this probability.
    b. Implement the Metropolis algorithm as a workable Python code to generate Gaussian random numbers.

    a) Pick an arbitrary point x to start with. Attempt to move x to x' = x + (2ξ-1)Δx, here ξ is a uniform random number from 0 to 1, and Δx is some fixed interval, say 0.1σ. compute the ratio r = p(x')/p(x) and compare with another random number ξ_2. *If* $\xi_2 < r$, *use x' as the next sample. If otherwise, use x as the next sample. [In any case, we get one additional sample]. Repeat this N times. b) omitted.*

3. Given a one variable function $f(x)$, to determine its minimum numerically, we can first fit it to a quadratic polynomial based on the values at three points, $a < c < b$, with function values $f(a)$, $f(c)$, $f(b)$.
    a. Work out an explicit expression for this polynomial of degree 2.
    b. Derive a formula for the minimum location x based on the result of part a.

a) We can use the Lagrange formula or Neville's algorithm, but it is simpler if we start from "first principles". Let $f(x) = \alpha x^2 + \beta x + \gamma$*, Since when x = a, b, c, the function need to be f(a), f(b), f(c), respectively, we need to solve the 3 by 3 matrix equation as* $\begin{pmatrix} a^2 & a & 1 \\ b^2 & b & 1 \\ c^2 & c & 1 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} f(a) \\ f(b) \\ f(c) \end{pmatrix}$*.The solution can be expressed via Cramer's rule as ratio of two determinants. [Details omitted]. b) The minimum of x is* $f'(x) = \frac{df}{dx} = 2\alpha x + \beta = 0$*. Or* $x_{min} = -\frac{\beta}{2\alpha} = -\frac{1}{2}\frac{\begin{vmatrix} a^2 & f(a) & 1 \\ b^2 & f(b) & 1 \\ c^2 & f(c) & 1 \end{vmatrix}}{\begin{vmatrix} f(a) & a & 1 \\ f(b) & b & 1 \\ f(c) & c & 1 \end{vmatrix}} = \frac{1}{2}\frac{(b^2-c^2)f(a)+(c^2-a^2)f(b)+(a^2-b^2)f(c)}{(b-c)f(a)+(c-a)f(b)+(a-b)f(c)}$.

4. A standard method to solve the time evolution of the Schrödinger equation is the Crank-Nicolson method, which is based on the approximation $\left(I + i\frac{\Delta t}{2\hbar}\,H\right)\Psi(t+\Delta t) = \left(I - i\frac{\Delta t}{2\hbar}\,H\right)\Psi(t)$, here Ψ is a column vector of complex numbers representing the wavefunction at space points, I is the identity matrix, and H is the Hamiltonian in a discretized representation as a matrix, Δt is the time step.
    a. We say an integration scheme is order n if it is accurate to Δt^n. Determine the order of the algorithm in time step Δt, i.e., what is the integer n, such that error to the solution is $O(\Delta t^{n+1})$ in one time step?
    b. Show that the solution scheme is unitary exactly to machine precision, that is the norm $\langle\Psi|\Psi\rangle = \Psi(t+\Delta t)^\dagger\Psi(t+\Delta t) = \Psi(t)^\dagger\Psi(t)$ is invariant with respect to time steps. The dagger † here denotes the Hermitian conjugate. This means that the total probability is preserved by the algorithm.

a) Let define $\alpha \equiv \frac{i\Delta t H}{\hbar}$*, the Crank-Nicolson is* $\left(1+\frac{\alpha}{2}\right)\Psi(t+\Delta t) = \left(1-\frac{\alpha}{2}\right)\Psi(t)$*, or solve for* $\Psi(t+\Delta t) = \left(1+\frac{\alpha}{2}\right)^{-1}\left(1-\frac{\alpha}{2}\right)\Psi(t)$*. Taylor expand the inverse using* $(1+x)^{-1} = 1 - x + x^2 - x^3 + \cdots$*, and multiply, we find* $\Psi(t+\Delta t) = \left(1 - \alpha + \frac{\alpha^2}{2} - \frac{\alpha^3}{4} + \cdots\right)\Psi$*. Compare this with the exact expansion of* $\Psi(t+\Delta t) = e^{-\alpha}\Psi(t)$*, we find the error is at* α^3 *(expanding the exponential gives 1/6 instead ¼ at the 3rd order). So, the algorithm is second order in* Δt*, error is* $O(\Delta t^3)$*. b) Take the Hermitian conjugate of* $\Psi(t+\Delta t) = \left(1+\frac{\alpha}{2}\right)^{-1}\left(1-\frac{\alpha}{2}\right)\Psi(t)$*, and noting that* $\alpha^\dagger = -\alpha$*, we find* $\langle\Psi(t+\Delta t)|\Psi(t+\Delta t\rangle = \langle\Psi(t)\,|\,\hat{O}|\Psi(t)\rangle$*, the operator is* $\hat{O} = \left(1+\frac{\alpha}{2}\right)\left(1-\frac{\alpha}{2}\right)^{-1}\left(1+\frac{\alpha}{2}\right)^{-1}(1-\frac{\alpha}{2})$*. These four factors commute because they are the function of a single Hamiltonian H, so we can move around and cancel them to 1.*

5. To find an energy minimum in a molecular system with the Hamiltonian $H = T + V$, one can use steepest descent, here $V = V(x)$ is the potential energy, and x is a vector of all the positions of the molecules.
    a. If the current position is at $x^{(n)}$ at n-th iteration, give the iteration equation that represents a single steepest descent to the next step at $n + 1$.
    b. However, in part a, we have not used the kinetic energy term $T = p^2/(2m)$, thus the steepest descent results in a slow method. How to incorporate the equations of motion of the Hamiltonian dynamics to give a better algorithm? In neural network, the resulting method is known as stochastic gradient with momentum.

    a) We go in the direction of negative gradient (i.e. force) as $x^{(n+1)} = x^{(n)} - \eta \nabla V|_{x^{(n)}}$. *If we perform the standard steepest descent, we do a line search to find minimum with respect to* η*, but in application in neural network, we just use a fixed small value of* $\eta > 0$. *b) we can use a method inspired by the Euler algorithm, so we update as* $x^{(n+1)} = x^{(n)} + \epsilon\, v^{(n)}$, $v^{(n+1)} = v^{(n)} - \eta \nabla V(x^{(n)})$. *This is known as stochastic gradient with momentum.*

--- End --- [JSW]