University of Saskatchewan
Department of Computer Science

Parallel Programming for Scientific Computing
(CMPT 851)
Instructor: Dr. Raymond J. Spiteri

ASSIGNMENT 03
Due: 1:00 p.m. Monday, April 28, 2014

1. **[0 marks]** Write the Progress Report associated with your project according to the Progress Report Guidelines posted on the course website. A synopsis of these instructions is as follows.

   Describe the problem you are proposing to solve. Indicate anticipated scope and current status. Some or all of this report should be recyclable to the final report.

2. **[25 marks]** In this question, we will essentially re-solve Problem 4 of Assignment 1 using OpenMP.

   Find the global minimum of the function

$$f(x,y) = e^{\sin(50x)} + \sin(60e^y) + \sin(70\sin x) + \sin(\sin(80y)) - \sin(10(x+y)) + \frac{x^2+y^2}{4}.$$

   This was Problem 4 of the SIAM 100-digit challenge. The goal was to compute the answer correct to 10 significant digits. A bit of straightforward analysis yields that the global minimum must lie not too far from the origin, so we restrict our attention to the domain $\{(x,y) \in [-1,1] \times [-1,1]\}$.

   (a) Write a program in OpenMP to sample $f(x,y)$ on a regular grid of the domain with $\mathtt{N} = 8^4$ points in each direction. Find the minimum value obtained and print it to standard output.

   (b) Test the performance of your code with all combinations of processors $\mathtt{P} = 1, 4, 16$ and N, 2N, and 4N sample points in each direction. Comment on the speed ups, efficiency, and scalability of your program.

   (c) What is the resolution of the timer used? Show how you obtained your answer.

3. **[25 marks]** Using the code `omp_nbody_red.c` for the shared-memory implementation of the reduced n-body solver, determine whether a cyclic schedule for the force computation outperforms a block schedule for $n = 10,000$ particles, for $N = 10,000$ time steps with $\Delta t = 0.01$ (for a final time $T = 100$). Initial conditions may be generated randomly by the code. By experimenting with the code, determine whether there is an optimal block size for `socrates`.

4. **[50 marks]** The *symplectic Euler method* (also known as the *semi-implicit Euler method* (or was that the *semi-explicit Euler method*?), the *Euler–Cromer method*, or the *Newton–Størmer–Verlet method*) is a modification of Euler's method for solving *Hamiltonian systems*, an example of which is the classical n-body problem. Such systems preserve certain quantities such as the volume in phase space. Hence a symplectic integrator yields "better" results than standard methods in such cases.

   For the classical n-body problem, the equations of motion

$$\ddot{\mathbf{q}} = \mathbf{F}(t, \mathbf{q}) = -\tilde{\mathbf{M}}\frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}), \quad \mathbf{q}(0) = \mathbf{q}^0,\ \dot{\mathbf{q}}(0) = \tilde{\mathbf{M}}\mathbf{p}^0, \tag{1}$$

   where $\tilde{\mathbf{M}} = \operatorname{diag}(1/m_i),\ 0, 1, \ldots, n-1$, is generally a constant symmetric matrix, are derived from Hamilton's equations of motion

$$\dot{\mathbf{p}}_i = -\frac{\partial H(\mathbf{q}, \mathbf{p})}{\partial \mathbf{q}_i},\ \dot{\mathbf{q}}_i = \frac{\partial H(\mathbf{q}, \mathbf{p})}{\partial \mathbf{p}_i}, \quad i = 0, 1, \ldots, n-1, \tag{2}$$

   where the Hamiltonian for this problem

$$H(\mathbf{q}, \mathbf{p}) = T(\mathbf{p}) + U(\mathbf{q}) = \sum_{i=0}^{n-1} \frac{\mathbf{p}_i^T \tilde{\mathbf{M}} \mathbf{p}_i}{2} + \sum_{i=0}^{n-1} \sum_{j>i}^{n-1} -G\frac{m_i m_j}{||\mathbf{q}_i - \mathbf{q}_j||}$$

   is *separable*. One variant of the symplectic Euler method then takes the form

$$\begin{aligned}
\mathbf{q}_i^k &= \mathbf{q}_i^{k-1} + \Delta t\,\frac{\partial T}{\partial \mathbf{p}_i}(\mathbf{p}_i^{k-1}) = \mathbf{q}_i^{k-1} + \Delta t\,\mathbf{M}_i \mathbf{p}_i^{k-1}, \\
\mathbf{p}_i^k &= \mathbf{p}_i^{k-1} - \Delta t\,\frac{\partial U}{\partial \mathbf{q}_i}(\mathbf{q}_i^k) = \mathbf{p}_i^{k-1} + \Delta t \sum_{\substack{j=0 \\ j\neq i}}^{n-1} \mathbf{f}_{i,j}(\mathbf{q}^k), \quad i = 0, 1, \ldots, n-1,
\end{aligned}$$

   where the second-order system (1) has been converted to the first-order system (2), $\mathbf{q}_i^k \approx \mathbf{q}_i(t^k)$ and $\mathbf{p}_i^k \approx \mathbf{p}_i(t^k)$, $\mathbf{M}_i = \operatorname{diag}(1/m_i) \in \mathbb{R}^d$, where $d = 2$ is the problem dimension, and

$$\mathbf{f}_{i,j}(\mathbf{q}^k) = -\operatorname{sgn}(j-i)G\frac{m_i m_j}{||\mathbf{q}_i^k - \mathbf{q}_j^k||^3}(\mathbf{q}_i^k - \mathbf{q}_j^k),$$

   where $G = 6.673 \times 10^{-11}$ m/(kg · s^2) and

$$\operatorname{sgn}(\ell) = \begin{cases} 1 & \text{if } \ell > 0, \\ -1 & \text{otherwise.} \end{cases}$$

   Another powerful approach to solving problems of the form (1) is the *Runge–Kutta–Nystrøm method*, which treats the second-order derivative directly, thus allowing for higher order with fewer $\mathbf{F}$ evaluations.

A Runge–Kutta–Nystrøm method takes the form

$$\begin{aligned}
\mathbf{Q}_i^k &= \mathbf{q}_i^{k-1} + \Delta t\,\frac{1}{2}\tilde{\mathbf{M}}_i \mathbf{p}_i^{k-1},\\
\mathbf{p}_i^k &= \mathbf{p}_i^{k-1} + \Delta t \sum_{\substack{j=0\\ j\neq i}}^{n-1} \mathbf{f}_{i,j}(\mathbf{Q}^k),\\
\mathbf{q}_i^k &= \mathbf{q}_i^{k-1} + \frac{1}{2}\Delta t\,\tilde{\mathbf{M}}_i(\mathbf{p}_i^{k-1} + \mathbf{p}_i^k).
\end{aligned}$$

Modify the code `omp_nbody_red.c` that solves the n-body problem taking advantage of symmetry to solve the n-body problem with $n = 10,000$, $\Delta t = 0.01$, $N = 10,000$. Initial conditions can be generated randomly by the code, but for the purposes of reproducibility they should be the same for each run.

Plot the evolution of $H(\mathbf{q}, \mathbf{p})$, which represents the energy, for the forward Euler, symplectic Euler, and Runge–Kutta–Nystrøm methods.

Report timings for 4, 8, and 16 threads. Comment on the relative efficiencies of the methods compared to their accuracies.