Project 4

Due: November 14, 2025, at 11:59 pm (See BS for official due)

Please read the Common Projects Instructions first.

Problem 4.1: A 3D surface equation of a heart is expressed in equation,

$$\left(x^2+\left(\frac{3}{2}y\right)^2+z^2-1\right)^3-\left(x^2+\frac{1}{50}\left(\frac{3}{2}y\right)^2\right)z^3=0$$

We model the heart as a solid object with no internal cavities and uniform density. However, during the cardiac cycle, the heart actively pumps blood in and out, causing its volume, surface area, and total mass (including contained blood) to vary dynamically with time. To enrich the problem—making it both physically richer and computationally more demanding—we introduce challenging, realistic extensions that push the limits of parallel computing.

Project Tasks

1. Design and implement a parallel algorithm to compute the surface area of the heart using the provided parametric surface equation.
2. Design and implement a parallel algorithm to compute the heart's total mass at time $t = 1$, where the instantaneous density is given by

$$\rho(x,y,z;t)=\rho_0(x,y,z)\left[\frac{1+\sin(2\pi ft)}{2}\right]$$

   where f is the heart rate (in Hz) and the functional form is illustrative. You may propose a biologically plausible density variation. The baseline (required) case is $\rho_0(x,y,z)=1, f=60$.

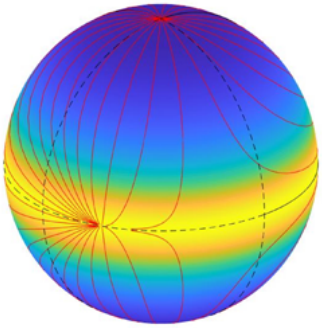
Performance Requirements

- Use $P \in \{1,4,16\}$ cores.
- Achieve at least 3 decimal digits of accuracy with minimal necessary computation.
- Report the average number of floating-point operations (FLOPs) per core for each case.

Results (floating-point operations per core) might look like,

To compute:	P=1	P=4	P=16
Surface area			
Heart mass			

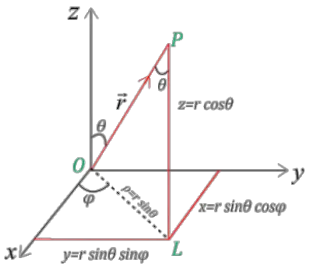
Problem 4.2: In MD simulations, bonded and non-bonded forces must be computed for each atom before integrating Newton's equations at every time step. This project focuses on the parallel computation of non-bonded Lennard-Jones forces among Nparticles.



You place N particles in a sphere of radius $R = 100$ centered spere at the origin. Particles are randomly distributed with density

$$\rho(r, \theta, \phi) = \rho_0 e^{-r/R} \cos^2 \theta$$

where ρ_0 is the normalization to ensure the total particle number is N and r, θ, ϕ are the usual coordinates in spherical coordinate system.



You generate the initial coordinates (floating-point numbers) of these particles with $\boldsymbol{x_i}\ \forall\ i = 1, 2, \ldots, N$. The particles i and j interact with the so-called Lenard-Jones "force" (ignoring constants):

$$\boldsymbol{f}_{ij} = \left(\frac{1}{r_{ij}^{12}} - \frac{2}{r_{ij}^{6}}\right)\hat{\boldsymbol{r}}_{ij}$$

where $\boldsymbol{r}_{ij} = \boldsymbol{x_i} - \boldsymbol{x_j}$ and $r_{ij} = |\boldsymbol{r}_{ij}|$, and force is a vector $\boldsymbol{f}_{ij}$ pointing to or from the partnering particle. We further assume the force will truncate at $r_c = 10$, i.e., $\boldsymbol{f}_{ij}(r_{ij} > r_c) = \boldsymbol{0}$. The total force on particle i is $\boldsymbol{F}_i = \sum_{j\neq i}^{N} \boldsymbol{f}_{ij}$.

Please complete:

(1) Design a decomposition to distribute the $N = 2^{14}$ particles across P cores.
(2) Design algorithm(s) to compute $\boldsymbol{F}_i\ \forall\ i = 1,2,\ldots, N$.
(3) Implement your algorithm(s) on a parallel system $P = 2^3, 2^4, 2^6$ cores.
(4) Report your timing for the process of force calculations.

Problem 4.3: This problem is the glamorous sequel to Problem 3.3. "Mince" it to taste the essential differences. This problem focuses on optimizing the distributions of tasks in a 3-level Strassen matrix multiplication (MM) for matrices $A_{N\times N}$ and $B_{N\times N}$, where N is a large power of 2, e.g., $N = 2^{10} = 1024$.

3-Level Strassen MM: submatrix is further decomposed recursively into a grid of sub-sub-sub-matrices (s^3-mats) which results in 64 s^3-mats per input matrix (128 total for A and B). The 3-level Strassen algorithm requires $\mathbf{7^3 = 343}$ s3-mats MMs instead of the $\mathbf{8^3 = 512}$ needed in naïve MM.

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{18} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \alpha_{81} & \alpha_{82} & \dots & \alpha_{88} \end{bmatrix}$$

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & a_{23} & b_{24} \\ b_{31} & b_{32} & a_{33} & b_{34} \\ b_{41} & b_{42} & a_{43} & b_{44} \end{bmatrix} = \begin{bmatrix} \beta_{11} & \beta_{12} & \dots & \beta_{18} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \beta_{81} & \beta_{82} & \dots & \beta_{88} \end{bmatrix}$$

Three-level Strassen → 64 s^3-matrices per input → 343 s^3-MMs (vs 512 naïve). *"Same answer, few tears."*

Tasks:

1. 1 core, unlimited RAM → Run full Strassen. Time it. Call it T_1. *"The sad, serial truth."*
2. 7 cores, 8 s^3-mats each (6 keep, 2 scratch) → Spread the 343 s^3-MMs among 7 cores and run Strassen. Time: T_7.
3. 49 cores, 6 s^3- mats each (4 keep, 2 scratch) → Spread the 343 s^3-MMs among 49 cores and run Strassen. Time: T_{49}.

Results:

Cores	Memory Budget	Time
1	∞	T_1
7	8	T_7
49	6	T_{49}

References:

1. Paper in Nature. *Discovery of fast matrix multiplication algorithms by reinforcement learning.*
2. Chou, Li, Deng, Wang: *Parallelizing Strassen's method for matrix multiplication on distributed-memory MIMD architectures*

Notes: (1) This problem, along with 4.3 and 5.3, lacks the usual generality and flexibility of mathematical problems. (2) To incentivize for selecting this problem or 4.3 or 5.3, I will boost your letter grade by one notch, e.g., from B+ to A-, based on the normalized numerical grades. You get one, and only one, such lift even if you select more than one from 3.3, 4.3, 5.3. (3) For 5.3, some students will be involuntarily assigned and will automatically receive the one-notch letter grade boost.