1. Consider 16 particles in 2D interacting according to the pair potential

\[ V_{honey}(r) = \frac{5}{r^{12}} - \frac{6.50}{r^{10}} + 18.19e^{-2.21r} - 0.4e^{-40(r-1.755)^2}. \]  

This potential was specially designed to favor the self-assembly of the honeycomb lattice [1]. The function to be minimized is the total potential of interaction of 16 particles

\[ f(x_1, y_1, x_2, y_2, \ldots, x_{16}, y_{16}) = \sum_{i=1}^{15} \sum_{j=i+1}^{16} V_{honey}(r_{ij}), \]

where \( r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \) Pick a suitable optimization method and set up a reasonable initial configuration to find the potential minimum shown in figure below.

Submit your code via ELMS.

**Hint 1:** A matlab routine \([f, g] = \text{Honey}(z)\) that computes the potential \( f(z) \) and its gradient \( g(z) \) is provided. \( z \) is a column vector with 32 entries, the first 16 entries are \( x_1, \ldots, x_{16} \), the last 16 entries are \( y_1, \ldots, y_{16} \). A matlab routine \text{draw_configuration}(z)\ is also provided.

**Hint 2:** In order to set up a good initial configuration, plot the graph of \( V_{honey}(r) \) versus \( r \) and find its minima. (To find the minima, you can use matlab’s \text{fminsearch}). The radii of the circles in the figure above are equal to one half of the first local minimizer \( r_1 \approx 1 \).
2. Integrate the canonical equations for the simple harmonic oscillator

\[
\frac{dp}{dt} = -q, \quad \frac{dq}{dt} = p, \quad p(0) = 1, \quad q(0) = 0,
\]

for the time interval \(0 \leq t \leq T_{\text{max}} = 32\pi\) using the Hammer-Hollingsworth method with the Butcher array given in Table 1. Do this using time steps \(h = h_0 2^{-k}\), for \(k = 1, 2, \ldots, 10\), \(h_0 = \pi/2\). For each time step, compare your numerical solution at \(T_{\text{max}}\) with the exact one (find it analytically). Plot the graph of the error at \(T_{\text{max}}\) versus \(h\) in the log-log scale. Estimate the constant \(C\) and the power \(\rho\) in the error formula

\[\text{Error} \approx Ch^\rho.\]

Submit your code via ELMS.

*Hint: You can use matlab’s command polyfit.*


(b) Write a matlab code to compute the mean magnetization \(m\) in the Ising model in 2 dimensions by the Metropolis algorithm (the Markov chain Monte Carlo algorithm presented in [2]), on a 30 × 30 lattice as a function of \(\beta\). Make the boundary conditions periodic, i.e., the nearest neighbors of site \((i,j)\) where \(0 \leq i, j \leq 29\) are \((i \pm 1 \mod 30, j \pm 1 \mod 30)\) (see the matlab help for the command mod). Note that the analytic expression for the mean magnetization is

\[
m(\beta) = \begin{cases} 
(1 - \sinh(2\beta))^{-1/8}, & \beta > 1/T_c = 0.4408, \\
0, & \beta < 1/T_c = 0.4408.
\end{cases}
\]

(c) Calculate \(m\) for the set of values of \(\beta = 0.2:0.01:1\). For each Monte Carlo run, make your program to plot the running mean of the magnetization, the
running variance of the magnetization, and the running variance of the mean magnetization:

$$\bar{m}_k = \frac{1}{k} \sum_{i=1}^{k} m_i, \quad \text{Var}(m)_k = \frac{1}{k - 1} \sum_{i=1}^{k} (m_i - \bar{m}_i)^2,$$

$$\text{Var}(\bar{m})_k = \frac{1}{k - 1} \sum_{i=1}^{k} \left( \bar{m}_i - \frac{1}{i} \sum_{j=1}^{i} \bar{m}_j \right)^2.$$

Stop iterations as the running variance of the mean magnetization becomes less than some reasonable threshold. **Plot 1:** Plot the graph of the computed mean magnetization as a function of $\beta$ and superimpose it with the graph of $m(\beta)$ given by Eq. (3). Plot the graph of the number of Monte Carlo iterations necessary to reach your stopping criterion vs $\beta$. For the case if there is some value of $\beta$ for which convergence is not achieved in a reasonable time, restrict the maximal number of Monte Carlo steps by some $N_{max}$. If for some values of $\beta$ convergence is not achieved in $N_{max}$ steps, report about it in the comments to your program.

Submit your code via ELMS. Also submit a single pdf file with **Plot 1**.

**References**
