CBEE 102 (3 Cr.) – Engineering Problem Solving and Computations
Winter Quarter 2015
School of Chemical, Biological, and Environmental Engineering
Oregon State University
2015.02.27
PROJECT DESCRIPTION: Devil's Tower (Monte Carlo Simulation)

Project Description: Devil's Tower

Due Date: 2015.03.02 1700
(No late projects will be accepted without pre-arranged consent of the instructors.)

Additional Information: Please refer to the syllabus for further information.

Team Project Grading: *20% of course grade*

MATLAB Code	60% of the project grade
Project Report	30% of the project grade
Administrative Tasks	10% of the project grade

If you determine that a regrade is necessary, the entire assignment will be regraded.

Timeline

- Lec 01: during the first lecture of the course (2015.01.05), a brief overview of the project will be given during the introduction of the course.
- Lab 01A: prior to the end of the first laboratory section, you will be required to complete a homework assignment surveying your interests and skills. This survey will help the instructors organize the students into teams.
- Lab 01B: during the second laboratory section, you will be organized into teams based on the results of the survey.
- Lec 04: during the fourth lecture of the course (2015.01.26), a theoretical background of the Monte Carlo method and a theoretical background of the diffusion problem will be given.
- Lab 04B: during the fourth week of laboratory exercises, you will complete a brief experiment with your team to collect brute force data.
- Lec 05: at 1700 on 2015.02.02, the minutes of your required meeting with acknowledgement of the assigned tasks will be due.
- Lec 06: at 1700 on 2015.02.09, a penultimate flow chart of your project code will be due.
- Lec 09: at 1700 on 2015.03.02, your team project will be due.

MATLAB Code: 60%

The MATLAB Code will be the major portion of your project. This MATLAB Code will produce a simple Monte Carlo simulation where a number of 2D random walkers (or "particles") will diffuse from a center origin radially outward. The random walk of these particles will be tracked, and a number of statistical values will be compared to the theory presented including the mean number of steps necessary for a particle to walk a set distance from the origin and the average final concentration of particles after a set number of steps.

The following list gives requirements for the code. Note that the *length scale* is give as one "stride" or motion of the particle, the *time scale* is give as one "step" or time between individual motions, and the *mass scale* is give as one "part" or an individual particle. This list should provide you a clear path toward creating a flow chart. Also, please realize that the following statement includes all of the parts of the code that were used to make the figures show in Lec 04; however, you may be able to write a working code that does not use the same set of steps. Further, in writing the solution used in Lec 04, the following pieces were not written in the order that they are presented here.

- Create one *m-file* script called "devil.m" that will be the only submission.
    - Explicitly required: make a header comment describing the program and giving overview of use/syntax to run it.
    - Explicitly required: use in-line comments describing the purpose of any non-trivial code.
    - Explicitly required: define a comment block containing description of each variable parameter and its physical significance.
- Create a set of non-dimensional values defining the number of particles (M), the number of steps (N), and the length of the stride (δr). Also, a (pseudo-)bound (B) is defined to calculate the average number of steps that are needed to reach a certain distance.
- Calculate the diffusivity $(\mathscr{D})$ from the information given, such that

$$\mathscr{D} = \frac{\delta r^2}{4}. \tag{1}$$

- Calculate the estimate for the number of steps necessary to cross the bound, such that

$$\tilde{N} \approx \left[\frac{B}{\delta r}\right]^2. \tag{2}$$

- Initialize the arrays that will be used to store the location of the particles and the number of steps necessary to cross the predefined bound.
- Use a *random number generation* using `rand` to define the angle at which the particle will take its next step.
- Use a `for`-loop to track each particle that will be introduced.
- Use a `for`-loop to make each step that a particle will take.
- Calculate the new position of the particle based on the previous position, the step size, and the angle, using trigonometric functions.

- Use an `if`-statement to conditionally test for the number of steps that the particle takes to cross the boundary.
- Calculate the position of the particles in radial distance from the origin, the maximum radial position of all particles, and the standard deviation of the final position.
- Calculate the maximum ring size that is needed to calculate the average concentration per concentric ring. Note that these rings will be integer scalers of the initial bound.
- Initialize an array having a length equal to the number of rings for storing the average concentration.
- Use a `for`-loop to calculate the final concentration as a function of radial distance (i.e., number of particles per area), based on the number of particles found in each ring.
- Remove all particles that did not cross the boundary from the Figure 4 statistics.
- Find the mean number of steps and the standard deviation necessary for the particles to cross the bound.
- Create the following figures.
    - Figure 1: create a *plot* of the solution (concentration versus radial position) to the 2D time-dependent diffusion equation as a function of time, such that

      $$\hat{C} = \frac{M}{\pi\left(4\mathscr{D}t\right)}\exp\left[-\frac{r^2}{4\mathscr{D}t}\right], \tag{3}$$

      where $\hat{C}$ is a two-dimensional concentration (i.e., $\hat{C} = HC$, where H is the height, and C is the 3D concentration), r is the coordinate system in strides, and t is time in number of steps.
    - Figure 2: create a *scatter plot* representation of the ability of the function `rand` to provide an unbiased input using the data associated with the angle of the randon walk. This plot should show the angle divided by 2π as a function of the step for a given particle.
    - Figure 3: create a *cartoon* representation (MATLAB-generated plot) of the path of each particle past a ring at the desired bound distance. This plot should represent the particle paths as lines of different colors, and the plot should designate the end of the each particle's path with a red 'x'.
    - Figure 4: create a *histogram* (using the function `hist`) binning the total steps necessary for each particle to initially pass the desired bound, quoting the mean and the standard deviation of the set of trials in the legend or in a text label. Also, note the expected number of steps $\left(\tilde{N}\right)$.

      $$\tilde{N} \approx \left[\frac{B}{\delta r}\right]^2 \tag{4}$$
    - Figure 5: create a *cartoon* representation (MATLAB-generated plot) of the final position of each particle with rings at incremental distances.
    - Figure 6: create a *scatter plot* representation of the concentration of particles in each annulus (s) after a certain amount of time, and compare this Monte Carlo prediction to the solution to Fick's second law of diffusion, where (M_s) is the number of particles in the s-annulus, and r_s is the radius of the outer edge of the s-annulus, respectively:

$$\hat{C} = \frac{M_s}{\pi\left(r_s^2 - r_{s-1}^2\right)}. \tag{5}$$

- Print figures 1, 2, 4, 5, and 6 to PDF.
- Print figure 3 to JPEG.

In an effort to relieve confusion, please check back to this portion of the problem statement to find useful hints.

- The 2014W CBEE102 Buffon's needle project description and solution is located at the following URL:
  `http://web.engr.oregonstate.edu/~walkert2/pdfs/coursework/2014w_cbee102_project.pdf`.
- Equations from the Lec 04 presentation have been added in appropriate parts above.
- You are required to identify the first time step when they cross the bound.
- The random walkers are allowed to continue beyond the bound.
- Note that $r_0 = 0$ in Equation (5) (i.e., the area of the first "annulus" is just the area of the circle with radius r_1).
- The abscissa of Figure 4 should say "measured number of steps to bound [steps]." This change is reflected in an updated Lec04.
- An m-file called `circle.m` has been added to Canvas at `<Files/project>`. You will be able to use this file to draw your circles on your plots.

Project Report: 30%

In addition to an electronic submission of your project code, a Project Report will be required. The Project Report should be no longer than five (5) pages for sections 1-7 listed below, including figures. A printout of the final flow chart will be added as an appendix. A printout of the MATLAB Code will be added as an appendix. The printed code is to be monospaced font with a minimum size of 72 columns, formatted by MATLAB "Print" output, NotePad++, or another syntax-highlighting program. An optional one-page appendix comparing and contrasting the use of `rand` versus `randn` can be added for extra credit with direct inference to the use of `randn` with the angle of the step and an accompanying speculation on its effect on the angle. No further content should be included. Please see the handout entitled "`CBEE_Technical_Writing_Guidelines_2013.pdf`" as a guide in writing your report and the handout entitled "`CBEE414ReportTemplate.pdf`" as an example of the expected professionalism. The project grade will include neatness and attention to formatting as given in the Technical Writing Guidelines. Also, note that the report must be typed, while any color figures must be printed in color. We understand that this report may be your first opportunity to experience technical writing, and we will take this generalization into consideration during the grading.

1. Abstract

2. Background
3. Experimental Materials and Methods
4. Experimental Results and Discussion
5. Computational Materials and Methods
6. Computational Results and Discussion
7. Conclusions

I. Appendix 1: Flow Chart
II. Appendix 2: MATLAB Code
III. Appendix 3 (optional): Random Number Generators

Administrative Tasks: 10%

- Survey
- Team meeting outside of class with assigned tasks
- Penultimate flow chart
- Team evaluations

FORM FOR PROJECT TEAM ASSESSMENTS

A form entitled `2015w_cbee102_team_assessment.pdf` has been uploaded to the Project module. Every person must submit an assessment to receive credit for completing the project. The assessments are due Friday, 2015.03.06, to Charlotte Williams in the CBEE Main Office (Gleeson 103). We will go over this assessment during class on Monday (2015.03.02).

Project Submission

A physical version of your report is due at the start of class on Monday, 2015.03.02, at 1700. A digital version of your one MATLAB function file (devil.m) will be submitted through T.E.A.C.H. by Monday, 2015.03.02, at 1700. Information regarding how to submit filed through T.E.A.C.H. is provided below.

Submission to T.E.A.C.H.

Please submit your main MATLAB function file (one file) using the assignment submission system of the OSU Engineering T.E.A.C.H. website, which is located at the following website.

`https://secure.engr.oregonstate.edu:8000/teach.php`

Once you have accessed this web page, please do the following steps to submit your files (successful completion of these steps is part of this assignment):

1. Log into the TEACH system using your Engineering username and password.

2. Select “Submit Assignment” (under the “Class Tools” heading) in the menu on the right side of the screen.
3. Choose the “`project_matlab_script`” assignment from the pull-down menu that is displayed; then, select “Submit Query” to continue.
4. Select the “Browse...” buttons to upload one (1) files. Please upload your MATLAB function file (devil.m).
5. Select the “Submit Query” button to submit your files.

Note that, if you use MATLAB through Citrix, you will need to upload the files from your engineering home directory (i.e., Z: drive). Directions for accessing your ENGR home directory from your personal computer are available at the following website.

`http://engineering.oregonstate.edu/computing/fileaccess/windows_file_sharing/#map_network_drive`

If you need help submitting your MATLAB files, please see one of the TAs or the instructor.

Submit your completed project MATLAB files by 1700 on Monday, 2015.03.02.

Important Dates:

Add/Drop Deadline	2015.01.16
Midterm	2015.02.09 1100-1250
Withdraw Deadline	2015.02.20
Project Deadline	**2015.03.02 1700**
Final Exam	2015.03.16 2000-2150