Illustration: John Maggard for Hayward Baker

ICETCI Hackathon

Discrete Element Method
GPU Tech

Problem statement

ICETCI Hackathon

Discrete Element Method on NVIDIA GPUs
Problem statement

Introduction

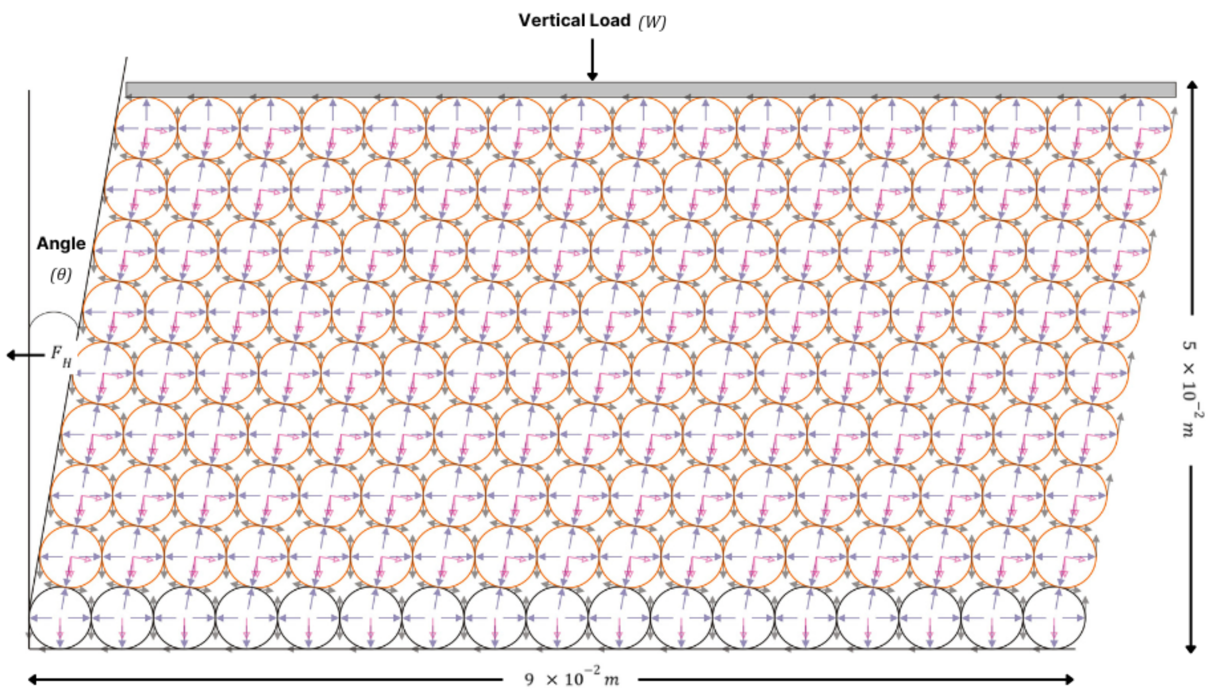
Discrete Element Method is a group of numerical methods that deals with the motion and effect of a large number of small particles. This is a simpler alternative to the Finite Element Method, which has largely been used to model the problem to follow. Discrete Element Method is preferred over Finite Element Method due to its lower computational requirements.

Despite this simplification, as particle numbers increase, so does computation, which leads to increasing times on consumer processors. The objective of this problem statement is to reduce or speed-up some of that computation and effectively model the given scenario.

This kind of problem finds its place wherever there is a need to model granular flow, like soil mechanics (Civil Engineering), powder mechanics (Mining/Manufacturing), heat transfer (Physics), chemical reactions (Chemistry) etc. The given scenario is novel and the computation has not been parallelized yet.[1]

Problem statement

The main challenge is to **accurately model particle flow** in the fastest possible time using GPU-accelerated CUDA parallel code. The setup is as follows:



This model contains 153 rollers, of which the ones outlined in orange need to be modeled. There is a load applied from top, a uniform vertical load. The task is to find the resultant horizontal force required to displace the model (F_H)[2] as the rollers slide over each other at a certain angle (θ).

This can be calculated from the given strain rate (horizontal displacement per unit time)[3] while taking horizontal distance at a certain time and model height as follows:

$$\theta = tan^{-1}(horizontal\ displacement/model\ height)$$

Bottom row of the model is fixed (assumed to be undisturbed during the test). Rest of the rows will slide due to horizontal displacement. This results in the horizontal force mentioned.

Inputs will be the following:

1. 153 rollers distributed across 9 rows and 17 columns.
2. Vertical load applied (per sphere):
   $W = 8.7$ N
3. Initial dimensions of the model:
   Height = $5 \times 10^{-2}\, m$
   Length = $9 \times 10^{-2}\, m$
4. Roller properties:
   Radius (r) = $2.6 \times 10^{-3}\, m$
   Weight $(W_r) = 3.5 \times 10^{-2}$ N
5. Strain rate: $1.25 \times 10^{-3}\, m/min$
6. Horizontal displacement ΔH (in mm) to be calculated from Strain rate.

Expected **output**:

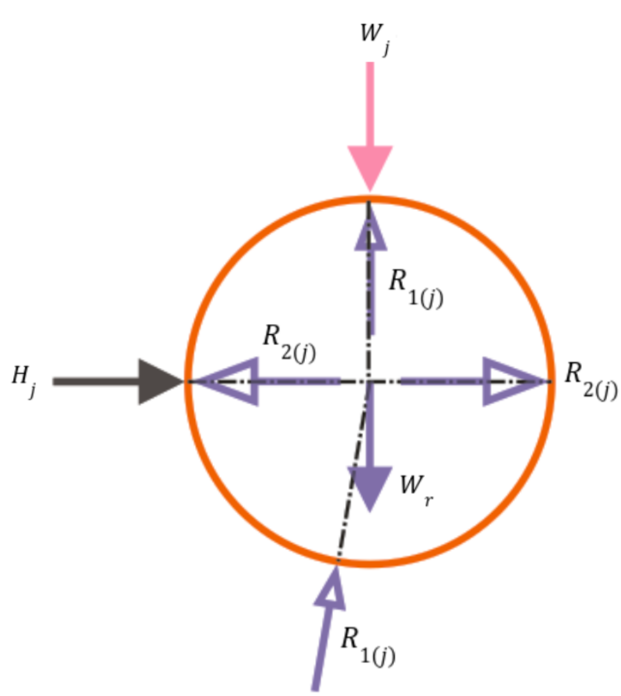
1. Horizontal force (F_H).

Evaluation:

1. Accuracy of output against experimental observations of volumetric strain and horizontal force.

Mathematical formulation

1. **For all top layer, just below the vertical load (which is movable)**



Iterating over columns, denoted by j (from 1 to 17).

$$W_r - R_{1(j)}(1 + cos\theta) + W = 0 \quad \textbf{[1]}$$

$$H_j - 2R_{2(j)} + R_{1(j)}sin\theta = 0 \quad \textbf{[2]}$$

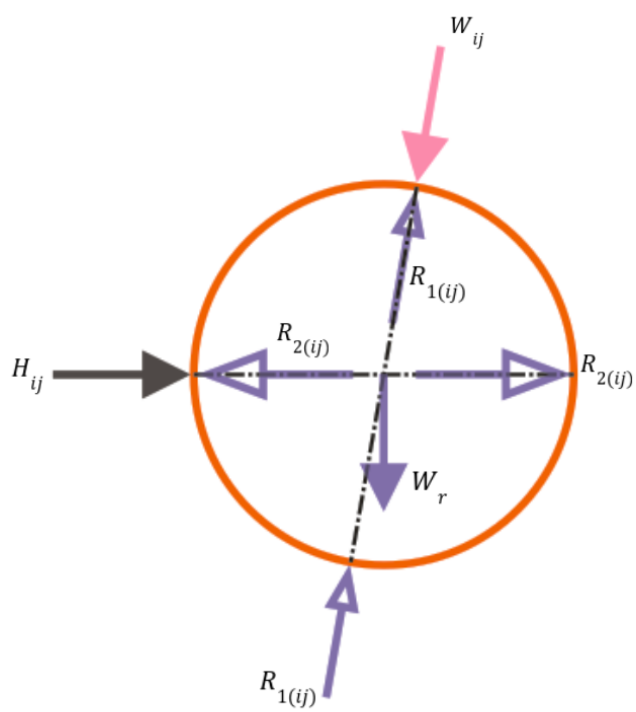
$$17r(H_j - 2R_{2(j)}) + (W - R_{1(j)})[(2j - 1)r + 16rsin\theta] + W_r[(2j - 1)r + 16rsin\theta] - R_{1(j)}cos\theta[(2j - 1)r + 15rsin\theta] + R_{1(j)}sin\theta \times 16r = 0 \quad \textbf{[3]}$$

where

$R_{1(j)}$, $R_{2(j)}$: Reaction forces as shown in figure

H_j: Unbalanced horizontal force of the roller in the j^{th} column

2. **For all intermediate layers (which are movable)**



Iterating over rows, denoted by i (from 1 to 6). Column meanings (denoted by j) remain as usual.
Note: Rows starting at 1 ignores the bottom, fixed row for the generalized formula.

$$W_r - 2R_{1(ij)}cos\theta + W_{ij}\,cos\theta \;=\; 0 \quad \textbf{[4]}$$

$$H_{ij} + 2R_{1(ij)}sin\theta - 2R_2 - W_{ij}sin\theta \;=\; 0 \quad \textbf{[5]}$$

$$\begin{gathered} H_{ij}(2i + 1)r \;-\; 2R_{2(ij)}(2i + 1)r \;+\; 4irR_{1(ij)}sin\theta + R_{1(ij)}sin\theta r \\ -\; 2R_{1(ij)}cos\theta[(2j - 1)r + 2irsin\theta] \;+\; R_{1(ij)}cos\theta \times rsin\theta \\ -\; 2(i + 1)rW_{ij}sin\theta + W_{ij}cos\theta[(2j - 1)r + (2i + 1)rsin\theta] \\ + W_r[(2j - 1)r + 2irsin\theta] \;=\; 0 \quad \textbf{[6]} \end{gathered}$$

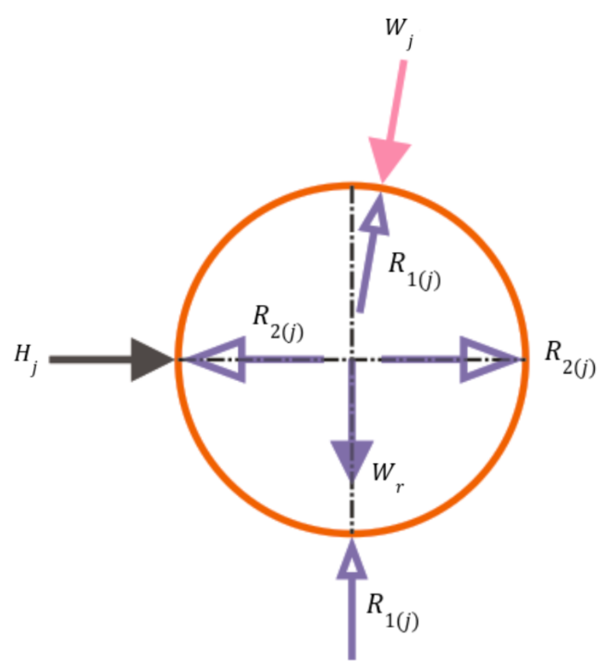
where

$W_{ij} = W + \sum_{i=1}^{7}(8 - i)W_r$

$R_{1(ij)}$, $R_{2(ij)}$: Reaction forces as shown in figure

H_{ij}: Unbalanced horizontal force of roller in the i^{th} row and j^{th} column

3. **For bottom layer which is fixed (undisturbed during the test)**



Iterating over columns, denoted by j (from 1 to 17), we have the following equations the individual rollers in the bottom layer:

$$W_r - R_{1(j)}(1 + cos\theta) + W_j cos\theta = 0 \qquad \textbf{[7]}$$

$$H_j - 2R_{2(j)} + (R_{1(j)} - W_j)sin\theta = 0 \qquad \textbf{[8]}$$

$$H_j r - 2R_{2(j)} r - W_j sin\theta r + W_j cos\theta[(2j - 1)r + rsin\theta] - R_{1(j)}(2j - 1)r[1 + cos\theta] + R_{1(j)} sin\theta r + W_r r = 0 \qquad \textbf{[9]}$$

where

$W_j = W + 8W_r$

$R_{1(j)}$, $R_{2(j)}$: Reaction forces as shown in figure

H_j: Unbalanced horizontal force of the roller in the j^{th} column

4. **To calculate the Horizontal force (F_H)**

$$F_H = \sum_{j=1}^{17} H_j \text{ (top layer)} + \sum_{i=1}^{6}\sum_{j=1}^{17} H_{ij} \text{ (intermediate layers)} + \sum_{j=1}^{17} H_j \text{ (bottom layer)} \qquad \textbf{[10]}$$

Steps to apply

Notes:

1. Sequential code to be parallelized is made available in this GitHub repository: [ICECTI-Hackathon-GPU by dipyamanroy | GitHub](#).
2. Questions and queries will be addressed via [insert channel name] on the ICETCI Hackathon Slack group [insert link].

Level 1

1. Given problem/sequential program is to be parallelized on a GPU accelerator using CUDA C.
2. Program must be executed using the GPU accelerator on Google Colab (Free version). Accessed via `Runtime → Change Runtime Type → Hardware accelerator → GPU`.
3. Accuracy versus experimental results will be tested.
4. Parallelism demonstrated will be evaluated using established routines.

5. Links to Google Colab notebooks containing the source code and a report mentioning the approach must be privately shared via Google Forms. Submission timelines will be announced shortly.

Level 2

1. Access to 1×NVIDIA A100 Tensor Core GPU will be provided by Mahindra University to shortlisted entrants.
2. Speed-ups over sequential code have to be demonstrated.
3. Links to Google Colab notebooks containing the source code and a final report mentioning speed-ups must be privately shared via Google Forms. Submission timelines will be announced shortly.