Imperial College London

Parallelisation of the iSALE shock physics code using a hybrid OpenMP/MPI approach

Email: mohan.sun21@imperial.ac.uk
GitHub username: ms921
Repository: https://github.com/ese-msc-2021/irp-ms921

Supervisors: Prof. Gareth S. Collins and Dr. Thomas M. Davison
Student: Mohan Sun

CONTENTS

01

Introduction

Background and significance of the project

Introduction

Project description

iSALE is a multi-material, multi-rheology shock physics hydrocode[1]. This project aims to design a regridding method for iSALE3D that can coarsen the resolution of simulation domain by a factor of two, and investigate the software's performance. Programming are conducted with Fortran under Linux system.

Motivation:

1. Importance of impact modelling:
   - Effects of impact events to human's life
   - Expensive laboratory simulation
   - Development of numerical imapct models and hardware resources
2. Importance of 3D simulation:
   - The vertical impact phenomenon in the 2D simulation is rare in reality[2].
3. Importance of regridding method:
   - Expensive time and calculation costs of high resolution simulation.

Introduction

Theory

1. Governing equations:

$$Conservation\ of\ Mass:\ \frac{\partial\rho}{\partial t}+v_i\frac{\partial\rho}{\partial x_i}=-\rho\frac{\partial u_i}{\partial x_i}$$

$$Conservation\ of\ Momentum:\ \rho\frac{\partial u_i}{\partial t}=c_i\frac{\partial\sigma_{ji}}{\partial x_j}$$

$$Conservation\ of\ Energy:\ \frac{\partial E}{\partial t}=-\rho\frac{du_i}{\partial x_i}+s_{ij}\epsilon_{ij}$$

2. Variables:

Node-centered variables:

position(x, y, z), velocity(u)

Cell-centered variables:

pressure(p), density(ρ), viscosity(η), energy(E), volume(V)

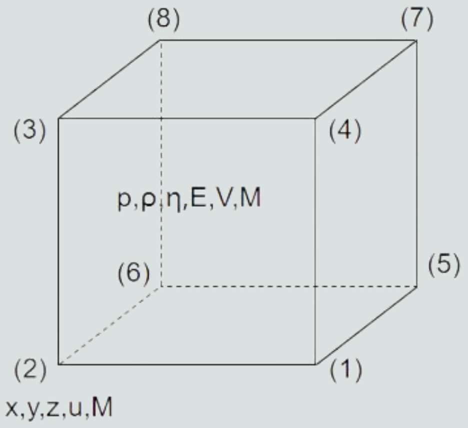

Figure 1:The assignment of variables and indexes.

3. Resolution:

The number of computational cells per projectile radius(CPPR)

Memory allocated ∝ N^M(N-number of cells, M-number of dimensions)

Time cost ∝ N^{M+1}

Different models and problems have different sensitivities to resolution.

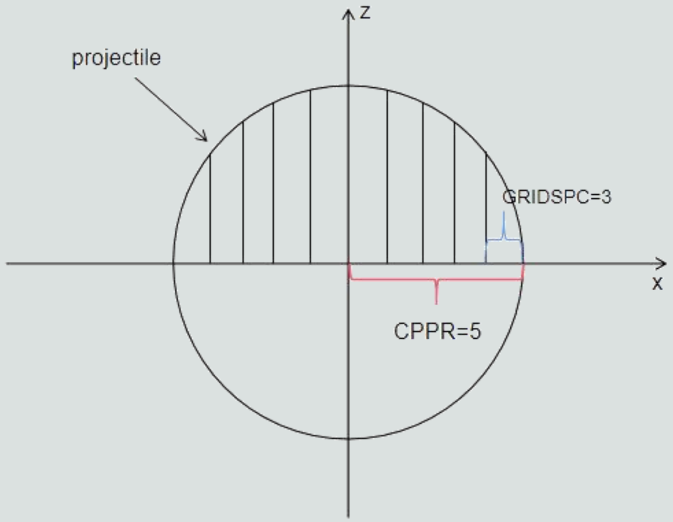

Figure 2: Cells per projectile radius(CPPR).

02

Methodology

Software description and research methods

Methodology

01

Preliminary test

Test the feasibility of the regridding method and check if coservation laws are satisfied using Python.

Two coarsening strategies are tested:

1. copy values on odd cells to new mesh, and throw values on even cells
2. involve all the values of cells on the odd mesh

The second strategy is selected in this project.

02

MPI approach

Decompose the research domain into several subdomains by slicing the mesh in x-direction.

Point-to-point communication

Ghost layers are included when creating the mesh.

cells index: is, ie

nodes index: iep

ghost layer index: igs, ige, igep

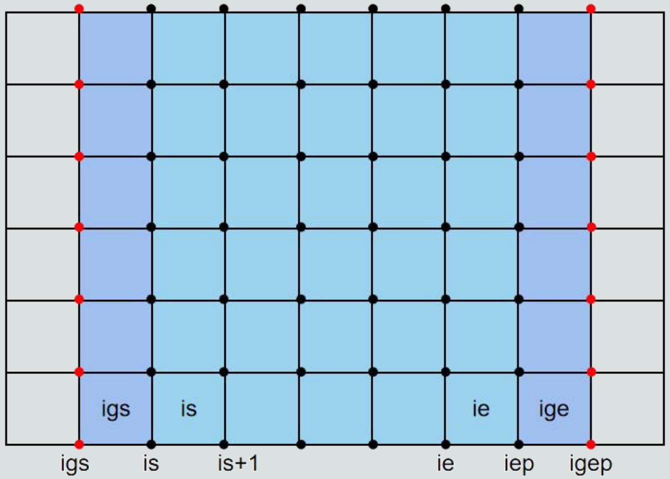

Figure 3: Ghost cells and nodes of a subdomain.

Methodology

03

Regridding method

Without coarsening: keep the resolution of original mesh

With coarsening: coarsen the resolution of original mesh

- coarsen cell-centered variables
- coarsen node-centered varibales

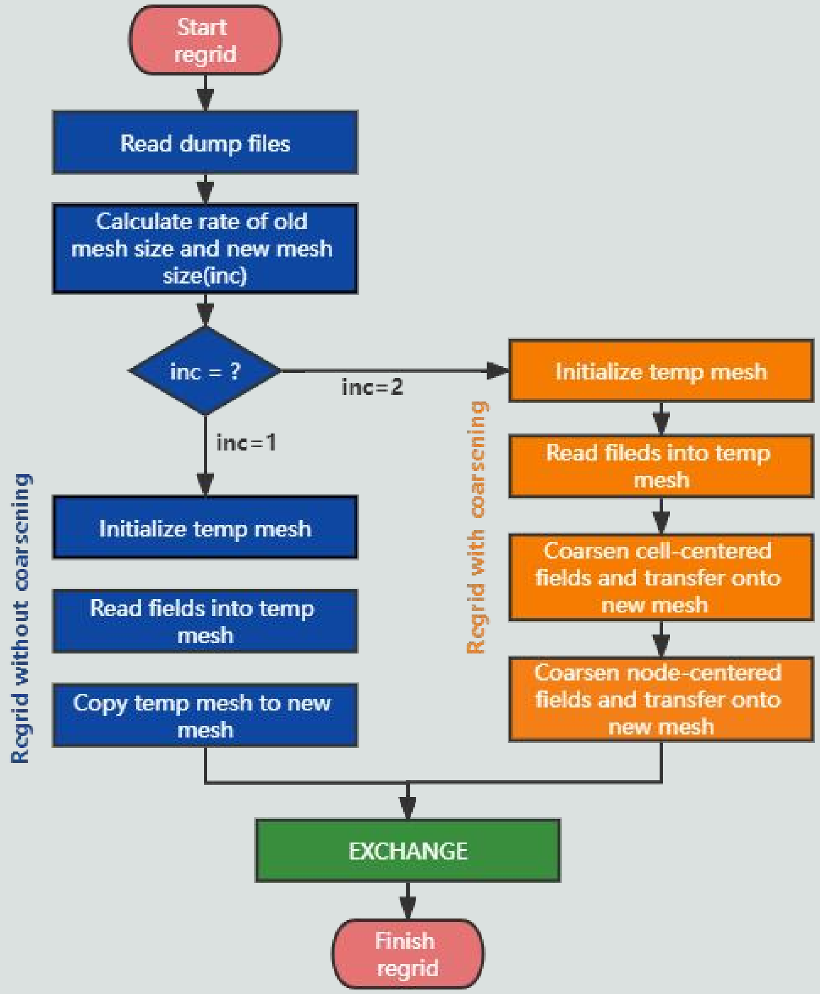

Figure 4: Regrid workflow diagram.

Methodology

Coarsening cells

Eight cells are merged into one cell

Conservations of mass and energy are applied.

Steps:

1. calculate total mass and volume of eight cells
2. calculate total values of fields on eight cells
3. divides the total values by total mass or volume

Algorithm 1: Coarsen cell

Input: $iold, jold, kold, inew, jnew, knew, nmat\_in$

Output: ro, mc, cmc, sie

$totalmass[0{:}\, nmat\_in] \leftarrow 0;$ % total mass of one material of eight cells

$totalvol[0{:}\, nmat\_in] \leftarrow 0;$ % total volume of one material of eight cells

$volu[0{:}\, nmat\_in] \leftarrow 0;$ % volume of one material in one cell

$masse[0{:}\, nmat\_in] \leftarrow 0;$ % mass of one material in one cell

$mc_{inew,jnew,knew} \leftarrow 0;$

for $k \leftarrow kold$ **to** $kold + 1$ **do**

  for $j \leftarrow jold$ **to** $jold + 1$ **do**

    for $i \leftarrow iold$ **to** $iold + 1$ **do**

      $mc_{inew,jnew,knew} = mc_{inew,jnew,knew} + mc\_tmp_{i,j,k};$

      for $m \leftarrow 0$ **to** $nmat\_in$ **do**

        $volu[m] = (C\_tmp[x]_{i+1,j,k} - C\_tmp[x]_{i,j,k}) * (C\_tmp[y]_{i,j+1,k} - C\_tmp[y]_{i,j,k}) * (C\_tmp[z]_{i,j,k+1} - C\_tmp[z]_{i,j,k}) * cmc\_tmp[m]_{i,j,k};$

        $totalvol[m] = totalvol[m] + volu[m];$

        $masse[m] = ro\_tmp[m]_{i,j,k} * volu[m];$

        $sie[m]_{inew,jnew,knew} = sie[m]_{inew,jnew,knew} + masse[m] * sie\_tmp[m]_{i,j,k};$

      end for

    end for

  end for

end for

for $m \leftarrow 0$ **to** $nmat\_in$ **do**

  $cmc[m]_{inew,jnew,knew} = totalvol[m]/sum(totalvol);$

  if $totalvol[m] > 0$ **then**

    $ro[m]_{inew,jnew,knew} = totalmass[m]/totalvol[m];$

  else

    $ro[m]_{inew,jnew,knew} = 0;$

  end if

  if $totalmass[m] > 0$ **then**

    $sie[m]_{inew,jnew,knew} = sie[m]_{inew,jnew,knew}/totalmass[m];$

  else

    $sie[m]_{inew,jnew,knew} = 0;$

  end if

end for

Methodology

Coarsening nodes

Only velocity field is considered in this section

Conservations of momentum are applied.

Steps:

1. calculate total mass and momentum of eight cells
2. divides the total momentum by total mass

Relationship between momenteum, mass, and velocity:

M = mv

Algorithm 2: Coarsen velocity

Input: *ilocs, iloce, jlocs, jloce, klocs, kloce, inew, jnew, knew*
Output: V
$cellmomenta[x:z] \leftarrow 0$;
$totalmass \leftarrow 0$;
for $k \leftarrow klocs$ **to** $kloce$ **do**
  for $j \leftarrow jlocs$ **to** $jloce$ **do**
    for $i \leftarrow ilocs$ **to** $iloce$ **do**
      $totalmass = totalmass + mc\_tmp_{i,j,k}$;
      $cellmomenta[x:z] = cellmomenta[x:z] + 0.125 * mc\_tmp_{i,j,k} * (V\_tmp[x:z]_{i,j,k} + V\_tmp[x:z]_{i+1,j,k} + V\_tmp[x:z]_{i,j+1,k} + V\_tm[x:z]p_{i,j,k+1} + V\_tmp[x:z]_{i+1,j+1,k} + V\_tmp[x:z]_{i+1,j,k+1} + V\_tmp[x:z]_{i,j+1,k+1} + V\_tmp[x:z]_{i+1,j+1,k+1})$;
    end for
  end for
end for
if $totalmass > 0$ **then**
  $V[x:z]_{inew,jnew,knew} = cellmomenta[x:z]/totalmass$;
else
  $V_{inew,jnew,knew} \leftarrow 0$;
end if

03

Results

Performance of software and relevant data

Results

Visualization of regridding

Program was run on a mini model.

Total simulation time(virtual time)= 0.8s.

Regridding at T=0.4s.

CPPR=40

Right figure shows the plots of the non-regrided and regridded pressure fields at three timesteps after T=0.4.

Table 1: Model information before and after regridding.

Model	Grid size	Gride space (m)	Object resolution
Before regridding	x:121 y: 61 z:121	dx: 100 dy: 100 dz: 100	x: 40 y: 40 z: 40
After regridding	x: 61 y: 31 z: 61	dx: 200 dy: 200 dz: 200	x: 20 y: 20 z: 20

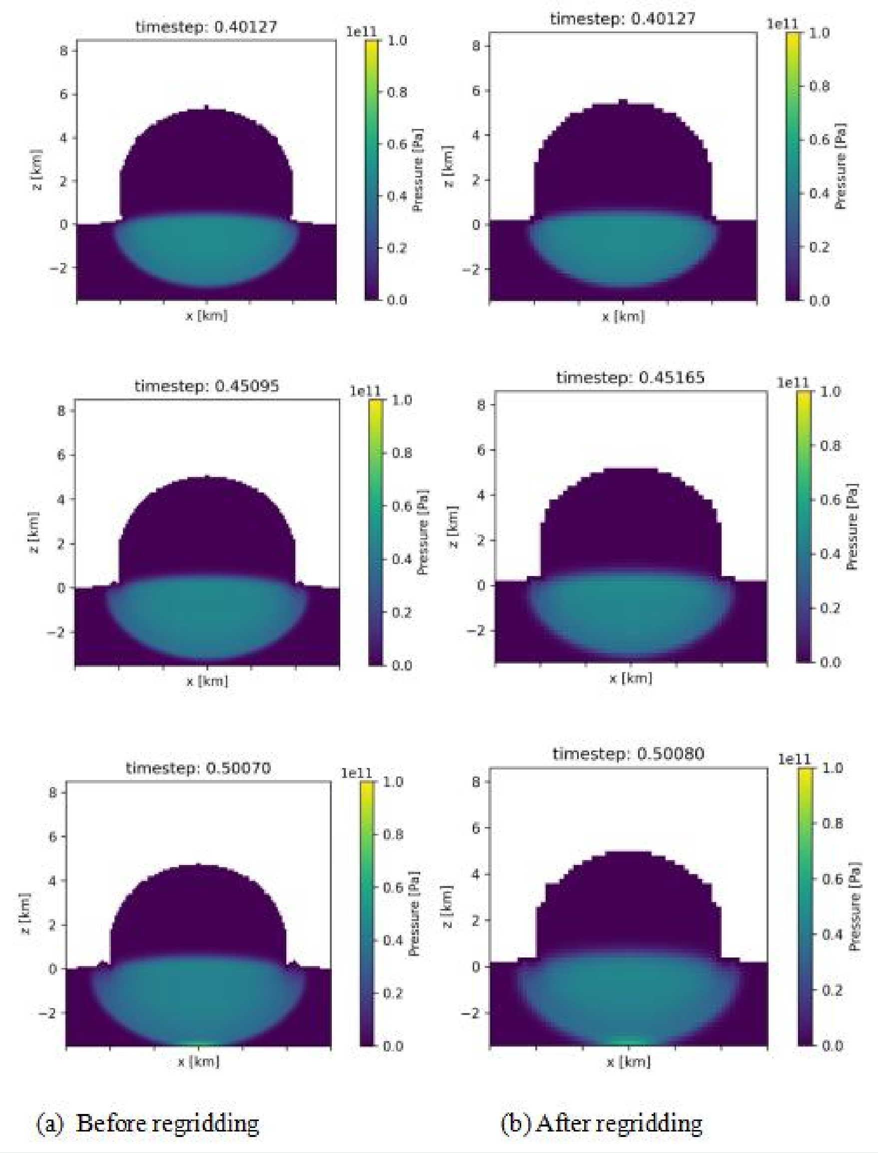

Figure 5: No-coarsened and coarsened pressure fields after T=0.4s

Results

Effects of different regridding points

Program was run on a large model.

Total simulation time(virtual time)= 3s.

CPPR=8

Five simulations are conducted:

1. No regridding
2. Regridding at T =0.4s
3. Regridding at T =0.2s
4. Regridding at T =0.1s
5. Start with half resolution

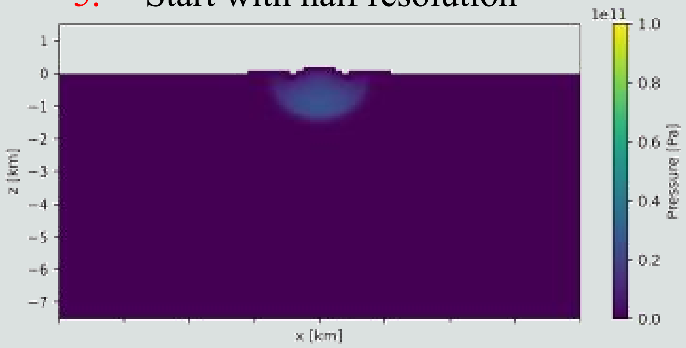

Figure 6: When the shock wave has passed out of the impactor(T=0.2s).

Table 2: Time cost and crater information of five simulations.

Simulation	Regridding point	Time cost(s)	Crater radius(m)	Crater volume(m^3)
1	None(T=3s)	4.5234×10^3	2.5513×10^3	2.8931×10^{10}
2	T=0.4s	9.1055×10^2	2.5392×10^3	2.8643×10^{10}
3	T=0.2s	6.9276×10^2	2.5298×10^3	2.8212×10^{10}
4	T=0.1s	5.1798×10^2	2.4790×10^3	2.7401×10^{10}
5	None(T=0s)	3.8983×10^2	2.3398×10^3	2.3848×10^{10}

Results

Effects on time cost

Regarding simulation 1 as regridding at T=3.0s, and simulation 5 as regridding at T=0s.

Regridding point and time cost is linearly positively correlated.

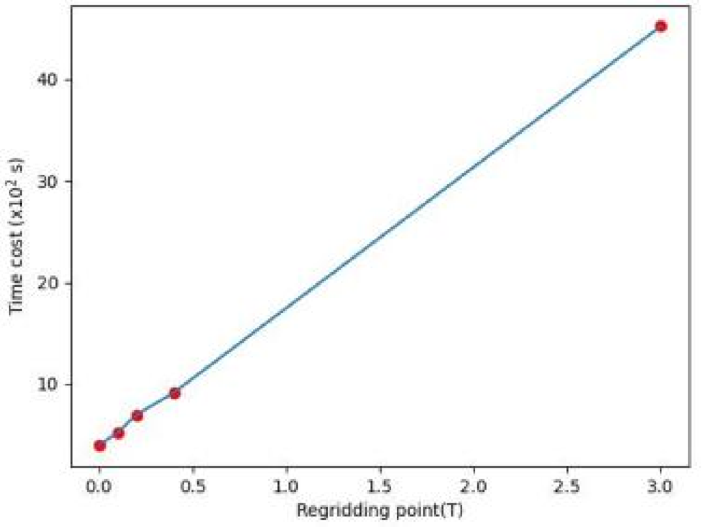

Figure 7: Time cost vs Regridding point..

Effects on simulation accuracy

Table 3: Error and accuracy at different regridding points..

Regridding point	Crater radius()		Crater volume(%)	
	Error	Accuracy improvement	Error	Accuracy improvement
T=0s	0.2115×10^3	0	0.5083×10^{10}	0
T=0.1s	0.0723×10^3	65.82%	0.1530×10^{10}	69.76%
T=0.2s	0.0215×10^3	89.93%	0.0719×10^{10}	85.85%
T=0.4s	0.0121×10^3	94.28%	0.0288×10^{10}	94.33%
T=3s	0	100%	0	100%

From regridding at T=0s to T=0.2s, there is a significant improvement on the accuracy,.

From regridding at T=0.2s to T=0.4s, there is only a small improvement on the accuracy.

Best regridding point should be T=0.2s in this simulation.

04

Discussion & Conclusion

Results analysis and summary

Discussions & Conclusion

Limitations

1. The number of cells should be even.
2. Manually edit the input file when regridding required.

Conclusion

One regridding method is completed that can coarsen the resolution by a factor of two.

Best regridding point is when the shock wave has passed out of the impactor.

Fortran programming and Linux operation are enhanced.

Further improvement

1. Be able to handle the case of odd number of cells.
2. Automatically calculate the size of the new mesh.
3. Design regridding methods for expansion of research domain:
   - Add cells with constant cells' size.
   - Increase cells's size with constant cells' number.

Reference

[1] Collins, G. S., Elbeshausen, D., Wünnemann, K., Davison, T. M., Ivanov, B., and Melosh, H. J.(2016). iSALE-Dellen manual: A multi-material, multi-rheology shock physics code for simulating impact phenomena in two and three dimensions. dx.doi.org/10.6084/m9.figshare.3473690.

[2] Pierazzo, E. and Melosh, H.J. (2000). Understanding oblique impacts from experiments, observations, and modeling. Annual Reviews in Earth and Planetary Science, 28, 141–167.

Imperial College London

Thanks for your listening!

student：Mohan Sun