M3D-C1 ZOOM Meeting

04/11/2022

Upcoming meetings and deadlines

CS Issues

1. LBL Update on GPU SuperLU_dist and triangular solve Yang and Nan
2. Mesh adaptation update - Morteza
3. NERSC Time
4. Changes to github master since last meeting
5. Regression tests
6. Emails from C.Liu: KPRAD on GPU, GPU speedup, 2F, Perlmutter
7. NaN on cori-haswell
8. Oom on perlmutter
9. JSOLVER and convert_polar

Physics Studies

1. ITER paper on VDE forces

Note: **meeting minutes posted on m3dc1.pppl.gov**

In attendance

Steve Jardin
Adelle Wright
Chen Zhao
Hank Strauss
Jin Chen
P Sinha
Chang Liu
Anders Kleiner
Brendan Lyons
Nate Ferraro
Dingyun Liu
Mark Shephard
Seegyoung Seol
Morteza
Nan Ding
Sherry Li
Yang Liu

Upcoming Meetings and Deadlines

- IAEA Technical Meeting on Plasma Disruptions and their Mitigation 19-22 July
  - In person at ITER HQ in France
  - Abstract submission by May 31

- APS Meeting Oct 17-21 (Spokane WA)
  - Invited talk nominations May 4

- INSITE requests for FY 23 now open for Frontier(EF), AORORA(EF), Polaris(44PF), Summit(200PF), Theta(12PF)

LBL

- Yang – progress on 3D version of triangular solve
- Nan- plan for optimizing the 2D process layout/mapping

Mesh adaptation update

March 31 email:

I managed to fix the problem that was causing the solver iterations to be much larger after adapt. In short, as Brendan suspected the zero structure of matrices was setup incorrectly due to subtlety in the way the after adapt mesh is constructed (I can provide more details in case anyone is interested).

For my initial testing, I have used the "pellet" example in the "regtest" folder. Instead of running it for only a couple of time steps, I run it for 30 timesteps and I run error estimation/mesh adapt at every 5 timesteps. Some pictures are shown in the slides here
https://docs.google.com/presentation/d/1p2gn31Ik2JiaYbXb_J8N_muu-PtZhc7faW8fDhGcSOM/edit?usp=sharing

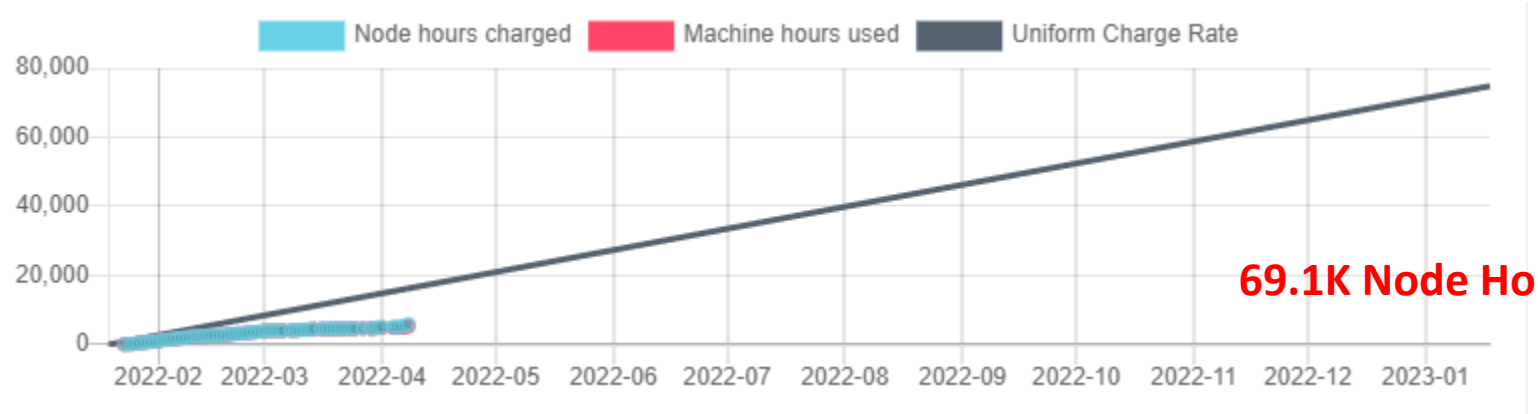
Moving forward, I can try running the above case for a longer time if that makes sense (eg 1000 timesteps with adapt every 100 timesteps, I think that is what I did for the 2D case)?

In terms of the code integration, everything is in this pull request on github (https://github.com/PrincetonUniversity/M3DC1/pull/41), with some explanation of how the new API can be used and the additional ((optional)) input variables that should be defined in the C1input. It would be good if that can be reviewed and I will be happy to implement any changes or suggestions before things can be merged into the master branch.

Thanks, Morteza

NERSC Time

mp288



69.1K Node Hours remaining!

- We are NESAP Tier 2 for Pearlmutter. . Phase-I w GPUs We have been given a repo m3984 with a small allocation. Presently we are not being charged.
- N9ES-N2 M3D-C1: J. Chen , C. Liu, S. Seol are early users

- We are under-utilizing our time on cori. If usage doesn't increase, I will contact John Mandrekas and explain that it is due to success of stellar

Changes to github master since 03/27/22

Nate Ferraro:
04/01/22: Removed iadapt_by_eta input variable which is not yet implemented

Andreas Kleinger:
03/29/22: Python routines: multiple improvements and bug fixes
04/07/22: Python routines: write fields to file, calculate injection rate

Chang Liu
03/28/22: Improve GPU matrix assembling for RMP_nonlinear
03/31/22: Improve GPU matrix assembling for NCSX

Local Systems

- PPPL centos7(04/09/22)
  - 7 jobs PASSED
- PPPL greene (04/09/22)
  - 5 jobs PASSED
- STELLAR (04/08/22)
  - 7 regression tests **PASSED** on stellar
  - NCSX failed on first attempt
- TRAVERSE_gpu(04/09/22)
  - 5 regression tests **PASSED**
  - KPRAD_2D, KPRAD_restart failed due to 0.001 fractional diff in C1ke

NERSC

- Cori-KNL (04/09/2022)
  6 regression tests **PASSED**
  NCSX **FAILED** with segmentation fault

- Cori-Haswell (04/09/2022)
  7 regression tests **PASSED**

- Perlmutter (04/09/2022)
  - 6 regression tests **PASSED**
  - NCSX failed with segmentation fault

KPRAD on GPU

March 31 from C. Liu

I have pushed my new work for GPU matrix assembling to master. With some help from Yao, now all the regtests pass except for the KPRAD ones. The result of regtest is like this

KPRAD_2D/devel_traverse_gpu:
Files differ at time 4
gamma_gr (base) = 1.7397
gamma_gr (new) = 1.73726248
Fractional difference = 0.0014011151347933385
Tolerance = 0.001
FAILED: C1ke files do not match

KPRAD_restart/devel_traverse_gpu:
Files differ at time 4
gamma_gr (base) = 1.7367
gamma_gr (new) = 1.73415818
Fractional difference = 0.001463591869637707
Tolerance = 0.001
FAILED: C1ke files do not match

GPU Speedup

March 31 From C. Liu

Regarding the speedup, here is a comparison of matrix element calculation of the pellet test (the most time consuming one) on traverse,

GPU version:

Time spent defining fields: 1.261311769485474
Time spent interpolating size field: 0.000000000000000
Time spent calculating elements: 8.884342193603516
Time spent finalizing arrays: 1.454325199127197

CPU version:

Time spent defining fields: 0.8125278949737549
Time spent interpolating size field: 0.000000000000000
Time spent calculating elements: 82.35605597496033
Time spent finalizing arrays: 5.447640180587769

2-Fluid

March 31 from C. Liu

I think the next step is to make the two fluid terms work on GPU, which will take some more time. Right now I wonder if you can think of an example of two fluid simulation and make it a new regression test, whose result depends sensitively on the correctness of the two fluid terms. I can then do the work based on that one.

I have tried some two fluid runs and always get divergent result, so I guess I missed some key points.

SJ: The 2-fluid runs require a very small timestep. Please try reducing the time-step by a factor of 10

GPU runs on Perlmutter

March 28 from C. Liu:

Given the current progress of GPU matrix assembling, I think I would try to encourage people to try their nonlinear runs on GPU machines, like perlmutter. I have tried to do my runs (AEs, fishbones), but I think I can learn from the issues others met and further improve the code.

I wonder if you are interested in running your soft-beta limit nonlinear simulations on GPUs. I think it can save some computation time. If you are interested, I will try to make the traverse_gpu code run on perlmutter and let you know the compiling instructions, about how to turn on the GPU switch, and you can give me feedback about the results.

SJ: I will try this but memory is likely a problem

cori_haswell

Trying to run a production job on cori_haswell

- Dies at different timesteps with the same initial error:

Linear solve did not converge due to DIVERGED_PC_FAILED iterations 0

From J. Chen 4/11/22:
After some testing, I found the oom crash is caused by the partition. Some integers in "IS" index set went wild so that it crashes with fixed INFOG(1)=9 but NFO(2) changes from run to run. After reducing the partition from 64 to 32, it runs for both mumps and superlu_dist using 8 ranks/node. Please copy the collapsed new mesh from /global/cscratch1/sd/jinchen/run_to_T70

SJ: This works, but uses only 8 processors per node (out of 64)

perlmutter

- I am trying to do a production run on Perlmutter
- Does with oom (out-of-memory) after 10 timesteps
- How is this possible?
- /global/cfs/cdirs/mp288/jardin/m3dnl/Perl/Run01-redo

Update 4/11/22

- Same fix as J. Chen suggested for cori_haswell?
- SJ now testing

JSOLVER

We have an application that needs the **JSOLVER** equilibrium code

- My version no longer compiles
- read_jsolver is up to date
- convert_polar is no longer in the SCOREC library

UPDATE 4/11/2022

- I could run with Jin's executable goequ.x86_64.linux
- Could we move jsolver to a GIT repo? Jin looking into this!

UPDATE 4/11/2022

- Seegyoung has updated portal readme file to point to convert_polar

ITER Paper on Disruption Forces

Nucl. Fusion **62** (2022) 056023 (11pp) https://doi.org/10

Non-axisymmetric MHD simulations of the current quench phase of ITER mitigated disruptions

F.J. Artola[1,2,*], A. Loarte[2], M. Hoelzl[1], M. Lehnen[2], N. Schwarz[1] and the JOREK Team[a]

[1] Max Planck Institute for Plasmaphysics, Boltzmannstr. 2, 85748 Garching, Germany
[2] ITER Organization, Route de Vinon sur Verdon, 13067 St Paul Lez Durance Cedex, France

- JOREK simulation of VDE in ITER
- Thru mitigation, q(a) remains above 2
- Small sideways force < 1 MN

- This is consistent with our result presented at IAEA meeting and in 2020 Theory Milestone
- John Mandrekas contacted me about this and was pleased to learn we had the same result and that we had done a 3D benchmark with JOREK
- Much different from the 40-60 MN predicted by Zakharov and others based on scaling of JET results

That’s All I have

Anything Else ?

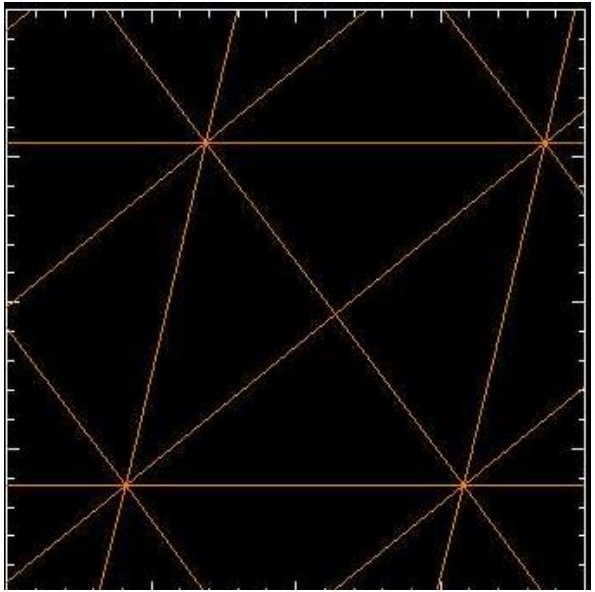
Next Meeting in 2 weeks

In attendance

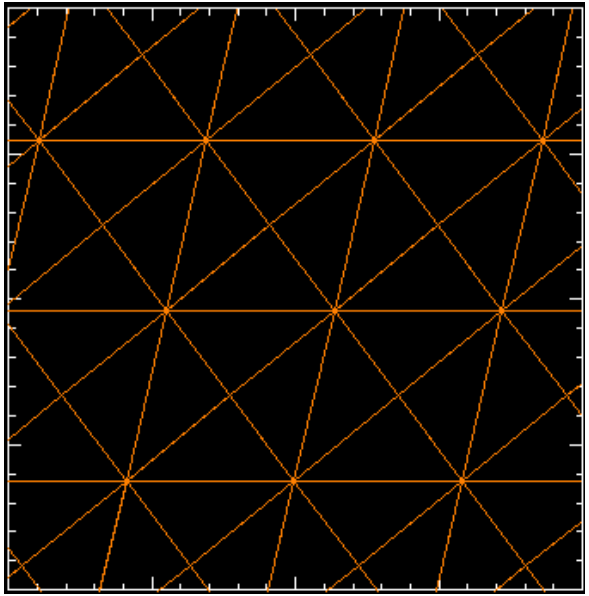
Steve Jardin
Adelle Wright
Dingyun Liu
Hank Strauss
Nate Ferraro
P Sinha
Chang Liu
Anders Kleiner
Brendan Lyons
Jin Chen
Chen Zhao
Cesar Clauser
Mark Shephard
Seegyoung Seol
Usman Riaz

Update on Soft-Beta-Limit Study

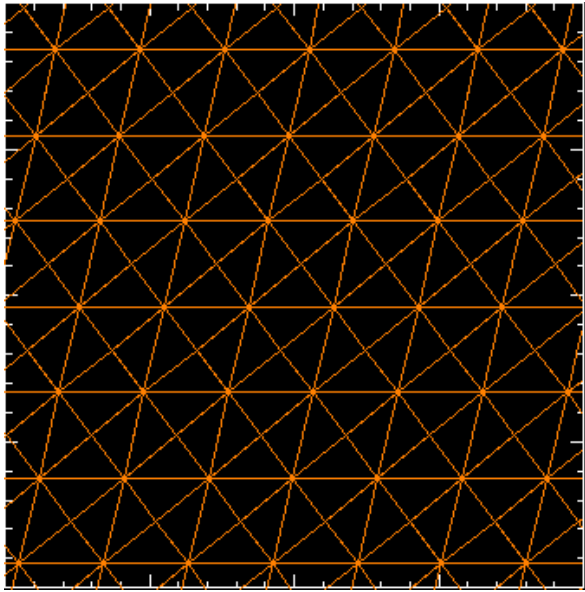
0.8 cm



0.4 cm

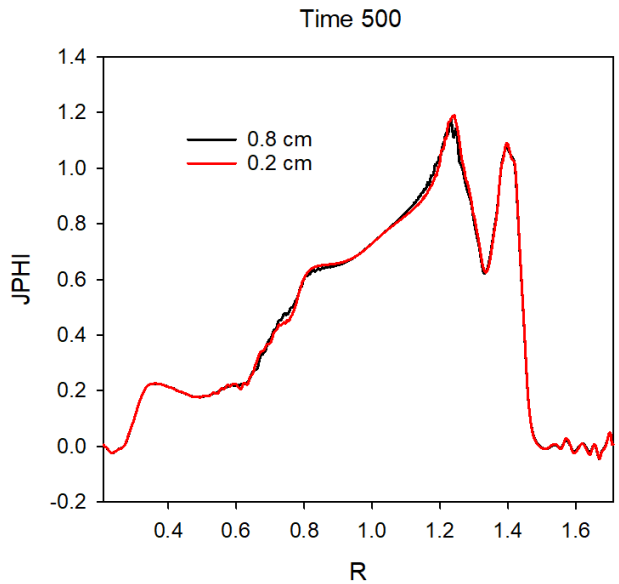


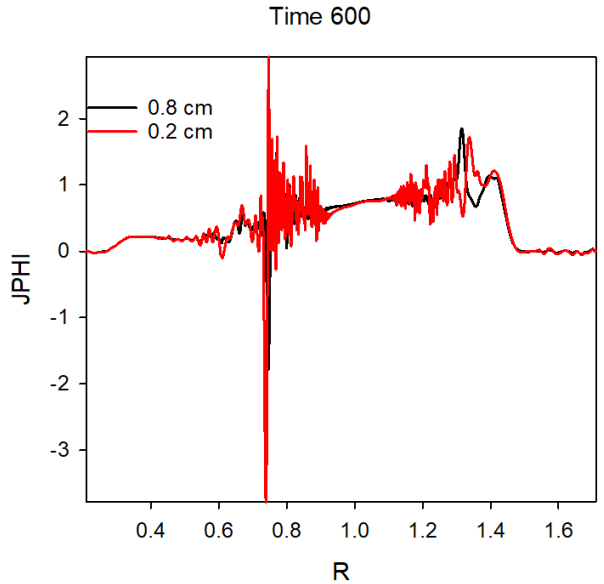
0.2 cm



These are close-ups in center of grid (near magnetic axis)

Result of Convergence Study

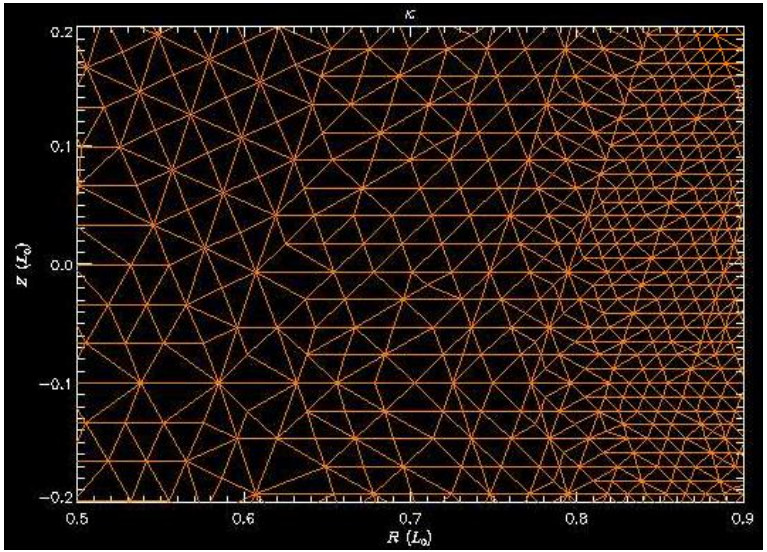




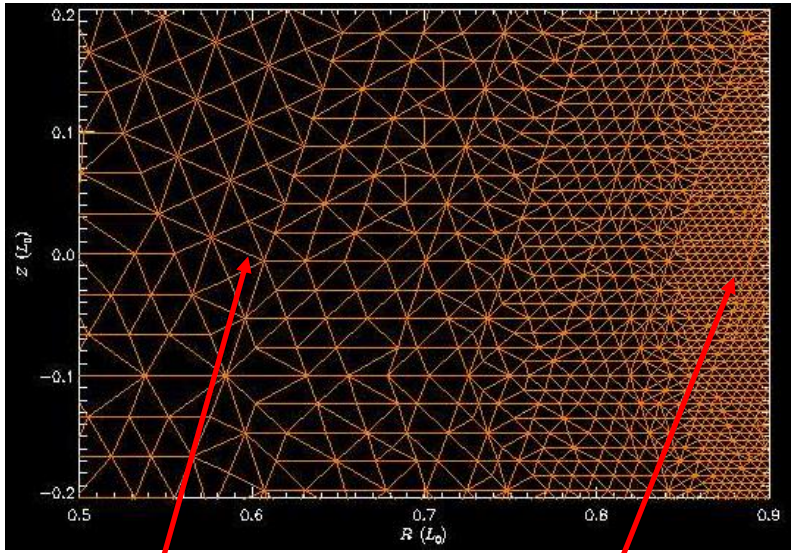
Solution for jphi still very noisy in region 0.5 < R < 0.9, even for the finest grid with 0.2 cm in center

Grid was not refined where J gets jagged

0.8 cm



0.2 cm



Resolution increased here

But not here

Now producing better grids that are refined where the current gets jagged