M3D-C1 ZOOM Meeting

09/11/2023

Upcoming Meetings

CS Issues

1. LBL Update
2. Adaptation update --RPI
3. Reduced precision SuperLU …Jin Chen
4. NERSC Time
5. Changes to github master since last meeting
6. Regression tests
7. MIT SPARC cluster Jin Chen

Physics Studies

1. 2D VDE on SPARC cluster
2. Anything else

In attendance

Steve Jardin
Saurabn Saxena
Chang Liu
Jin Chen
Brendan Lyons
Cesar Clauser
Priyanjana Sinha
Adelle Wright
Hank Strauss
Andreas Kleiner

Usman Riaz
Seegyoung Seol
Mark Shephard

Sam Williams
Sherry Li
Nan Ding
Hans Johansen

Upcoming Meetings

ITPA(MHD)	Sept 19-22	General Atomics
IAEA	Oct 16-21	London, UK
MHD Stability	Oct 26-28	Boulder, CO (US-Japan)
APS	Oct 30 – Nov 1	Denver, CO
AAPPS-DPP	Nov 12-17	Nagoya, JP

LBL Report

LBL?

Adaption Update

RPI?

Reduce tolerance for adapt?
Or, run the adapt reg. test only on a single processor.

Saved to this PC

Reduced Precision SuperLU

```
-hard_sub_pc_factor_mat_solver_type superlu_dist
-hard_sub_mat_superlu_dist_rowperm NOROWPERM
-hard_sub_mat_superlu_dist_parsymbfact
-hard_sub_mat_superlu_dist_colperm PARMETIS
```

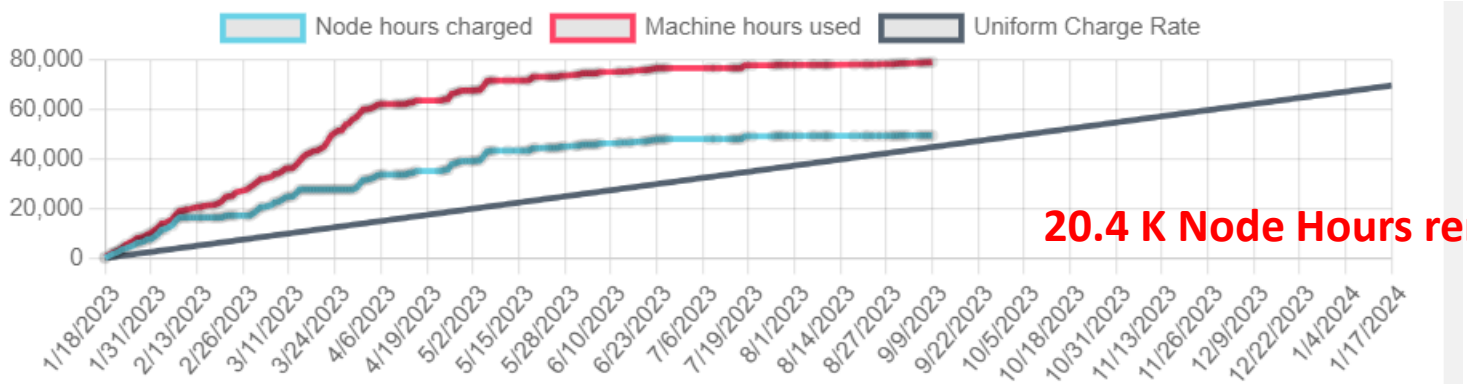
velocity, nplane=4, 32partitions, memory in MB, time in second

	V_mesh	A_mxn	A_nnz	Total memory		Mem_superlu		Solve time		mumps	
										mem	time
16F	5255	756720	566961120	234330 208600	11%	4771 2710	43%	23 16	31%	11687	19
32B	9614										
32H	19218	2767392	2080528416	867930 773170	11%	24057 13248	45%	59 39	34%	41036	49
32J	21955	3161520	2374774848	995700 881080	11%	28578 15680	45%	72 46	34%	50296	61
32K	49494	7127136	5365160064			OOM				122398	181

Jin Chen

NERSC Time 2023

mp288



20.4 K Node Hours remaining!

- MP288 usage rate is very close to linear
- Also, 5.8K k GPU node hours remaining.

Changes to github master --after 2023-07-10 (1 of 2)

Nate Ferraro:
08/31/23: Added m3dc1_mfmgen to “make bin”

Jin Chen
08/12/23: upgrade MIT to intel compiler, all regtests passed except a small error in NCSX
08/18/23: regtests batch job script using test queuemk
08/21/23: lift optimization level to O2 in mit_intel.

Seegyoung Seol
08/06/23: adding mesh verification program
08/06/23: minor updates for SCOREC s/w in stellar, traverse, and centos7
08/09/23: fixing crash in regtest/adapt on perlmutter_cpu
08/14/23: Revising mesh-gen.tex
08/26/23: adding iadapt_snap to support snapping in adaptation (default: true)
09/04/23: removing compilation flag ADAPT
09/06/23: fixing m3dc1_mfmgen model file bug
09/08/23: modifying m3dc1_model_load to load new format of m3dc1_mfmgen
09/08/23 reverting changes re. snapping

Usman Riaz
08/18/23: Few additions to support snapping operation in mesh adaptation

Local Systems

- PPPL centos7(09/10/23)
  - 7 jobs PASSED
- PPPL greene (09/10/23)
  - 5 jobs PASSED
- STELLAR (09/10/23)
  - 7 regression tests **PASSED** on stellar
- TRAVERSE-nvhpc (09/10/23)
  - All jobs **PASSED**

NERSC

- Perlmutter_cpu (09/11/23) …Jin Chen
  - 6 jobs **PASSED**
  - adapt **FAILED** with a Segmentation violation

- Perlmutter_gpu (09/11/2023)
  - 4 jobs **PASSED**
  - NCSX, KPRAD_2D **FAILED** with very small differences
  - adapt **FAILED** with a Segmentation violatio

	Stellar intel	MIT intel Sparc FOPT=0	MIT intel Sparc FOPT=O2
M3DC1 32H	283s	534s	324s

		Stellar intel	MIT intel	
				Pure Do loop is faster on Stellar Communication intensive solve is faster on MIT intel Reason: Intel is built on gnu on MIT, so the pure DO loop is slower.
M3DC1 32H	ludef	283s	324s	
	solve	138s	30s	
Skeleton pure do loops		40s	470s	
Petsc bjacobi+mumps		44s	40s	

Stellar intel

> which mpif90
/opt/intel/oneapi/mpi/2021.7.0/bin/mpif90

> mpif90 --version
ifort (IFORT) 2021.7.0 20220726

MIT intel
> which mpif90
/orcd/nese/psfc/001/software/spack/2023-07-01-physics-rpp/spack/opt/spack/linux-rocky8-x86_64/gcc-12.2.0/intel-oneapi-mpi-2021.9.0-xxdsl35zf6swjol2pxado64fdirbldbs/mpi/2021.9.0/bin/mpif90

> mpif90 --version
GNU Fortran (GCC) 8.5.0 20210514 (Red Hat 8.5.0-16)

	Stellar intel FOPT=O2	MIT intel FOPT=O0	MIT intel FOPT=O2	MIT intel FOPT=O3
M3DC1 32H	283s	534s	324s	254s
	O3: seg fault		unroll	vectorization

		Stellar intel	MIT intel	
				-O2 Do Loop is faster with Stellar intel, but –O3 Do Loop with MIT intel came catch up. Communication intensive solve is faster on MIT intel Reason: MIT Intel is built with gnu. So –O3 does the vectorization.
M3DC1 32H	ludef	283s	324s	
	solve	138s	30s	
Skeleton pure do loops		40s (O3)	470s (O3)	
Petsc bjacobi+mumps		44s	40s	

Stellar intel

> which mpif90
/opt/intel/oneapi/mpi/2021.7.0/bin/mpif90

> mpif90 --version
ifort (IFORT) 2021.7.0 20220726

MIT intel
> which mpif90
/orcd/nese/psfc/001/software/spack/2023-07-01-physics-rpp/spack/opt/spack/linux-rocky8-x86_64/gcc-12.2.0/intel-oneapi-mpi-2021.9.0-xxdsl35zf6swjol2pxado64fdirbldbs/mpi/2021.9.0/bin/mpif90

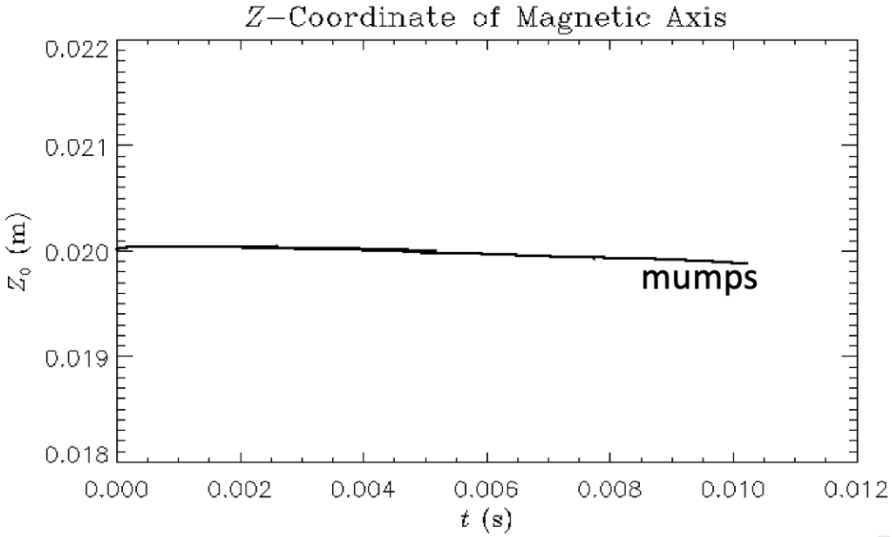
> mpif90 --version
GNU Fortran (GCC) 8.5.0 20210514 (Red Hat 8.5.0-16)

2D VDE on SPARC cluster

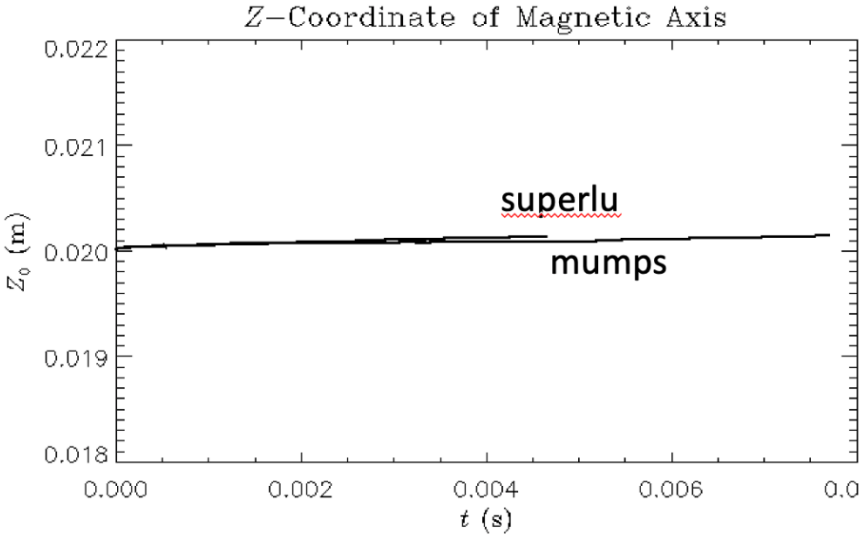
Email from Cesar on Sept 7: give

- 2D VDE on MIT cluster with mumps goes wrong direction
- On Stellar, superlu and mumps give different results (but correct direction)

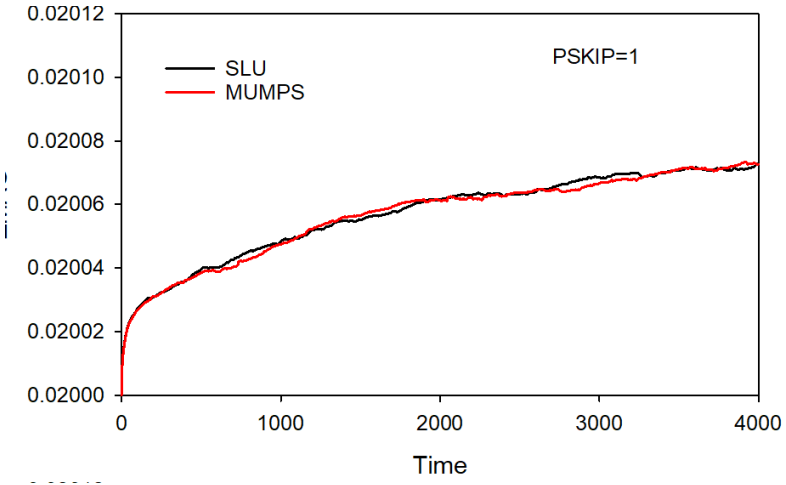
MIT – intel



PPPL-Stellar

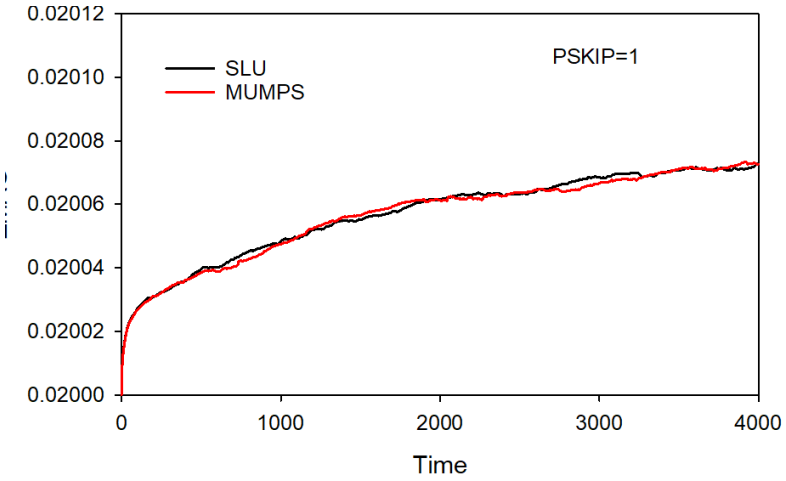


2D VDE on stellar

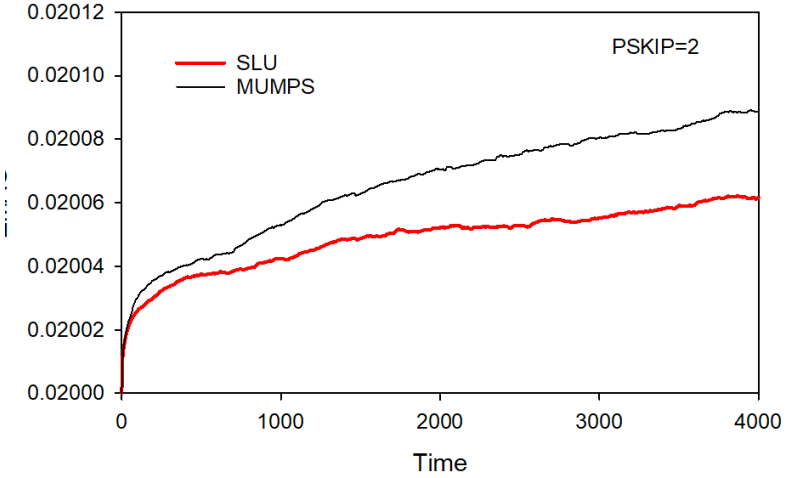


Fairly good agreement between Super_LU and MUMPS initially for PSKIP=1

2D VDE on stellar

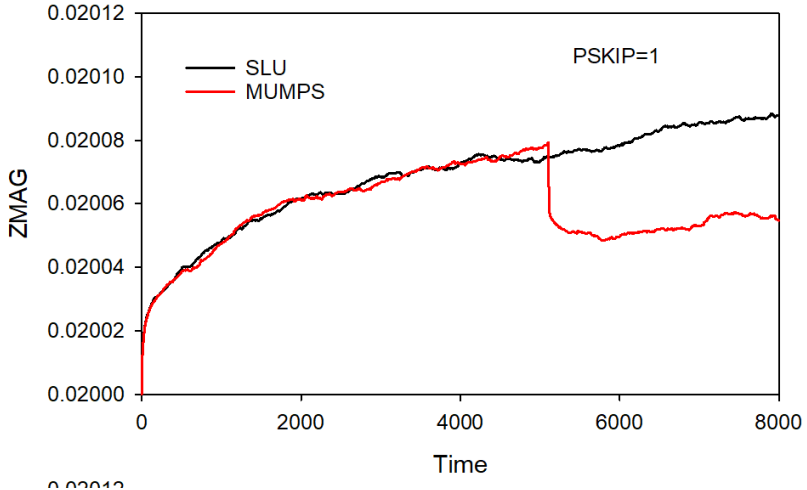


Fairly good agreement between Super_LU and MUMPS initially for PSKIP=1



Not as good agreement for PSKIP=2

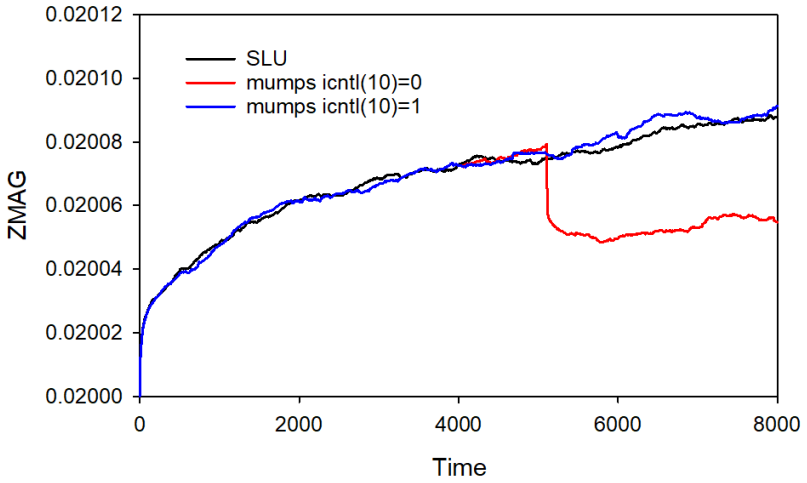
2D VDE on STELLAR



Agreement worsens over time(after restart). MUMPS looks to be in error.

Now trying iterative refinement in mumps ICNTL(10)=1

2D VDE on STELLAR



- Much better agreement between superlu & mumps w icntl(10) = 1
  - Should also try icntl(10) = 2
- also , set pskip=1, idouble_out = 1
- Might look at dependence on time step size also

That’s All I have

Anything Else ?