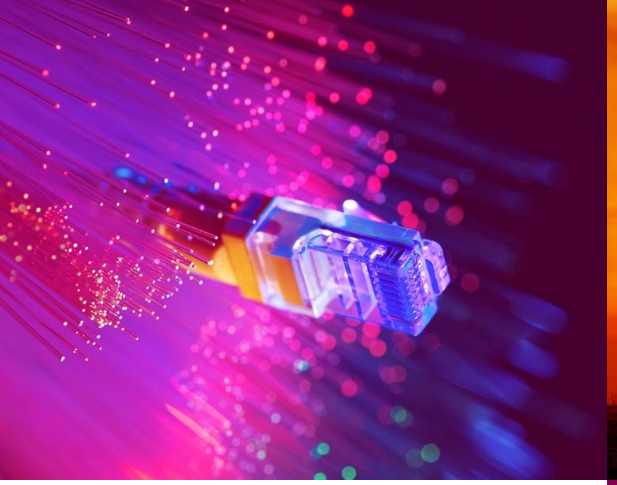


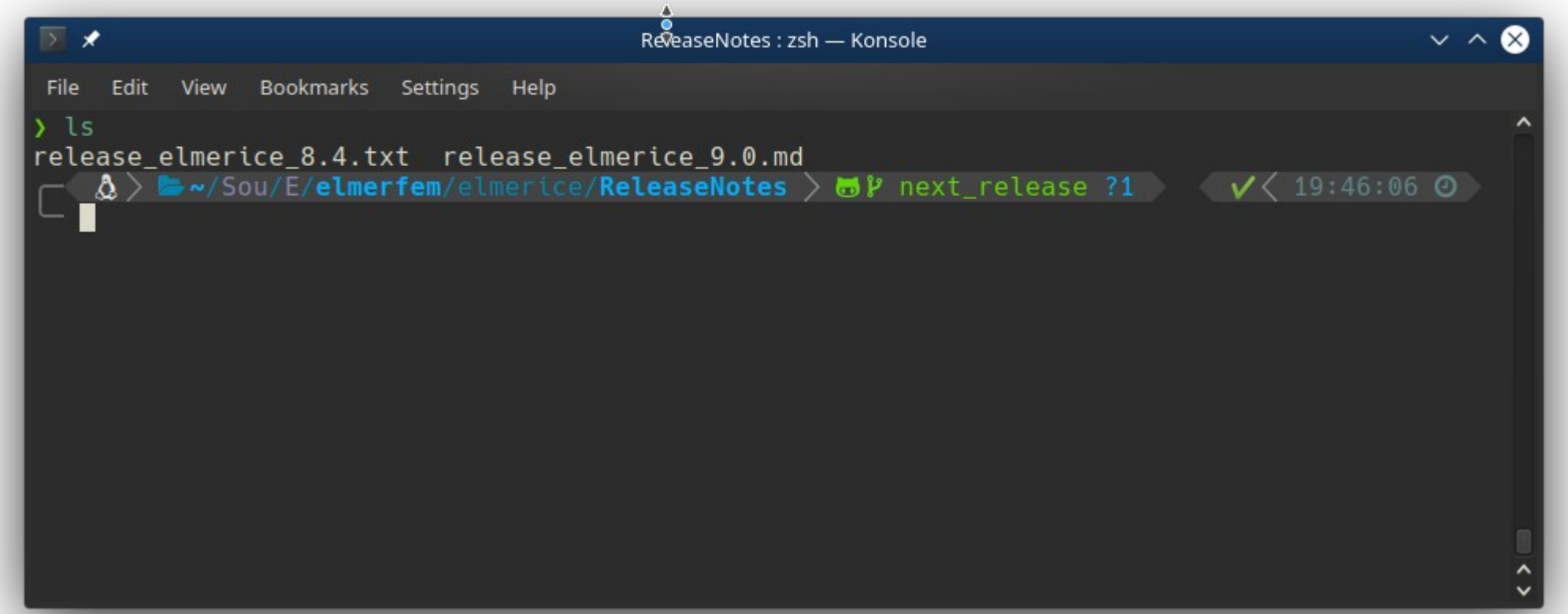
New Elmer Release 9.0 from Elmer/Ice point of view

Thomas Zwinger, Peter Råback

CSC – Suomalainen tutkimuksen, koulutuksen, kulttuurin ja julkishallinnon ICT-osaamiskeskus

Elmer/Ice release notes (up to 4.10.2020)

- Current release notes to be found under branch origin/next_release
- Release notes found under elmerfem/elmerice/ReleaseNotes:



Release notes in MarkDown

Elmer/Ice Release Notes for version 9.0

Previous release: **8.4**

Period covered: **18 Dec 2018 - 30 Aug 2020**

Number of commits: ~**110** (excluding merges of other branches)

These release notes provide information on most essential changes in Elmer/Ice functionalities. Starting from the uppermost directory of the source tree, you can inquire changes inside the elmerice-directory using

```
git log --since="2018-12-18" -- elmerice
```

Overview of changes/enhancements

- Improvements to inversion methods
- Coupling of the GlaDS solvers with the calving solvers in a two-mesh, 3D simulation. Includes a new plume solver that currently relies on the external ODEPack library (not included in the Elmer distribution).
- New thermodynamically consistent model for permafrost with saturated aquifers

New Solver/Userfunction Modules

- `Calving3D_lset.F90` : Return calving as a level set function (work in progress).
- `CalvingRemeshMMG.F90` : Cut a calving event directly out of a 3D mesh without external gmsh or mesh extrusion. Initial work on allowing calving margins to migrate.
- `PlumeSolver.F90` - associated ODEPack library files: `opkda1.F`, `opkda2.F`, `opkdmain.F` (not included in Elmer reopsitory): Provides plume melt rates across the calving front of a glacier. Fed by output from GlaDS solvers. Simulates a continuous sheet-style plume across entire front, split up into segments defined by frontal nodes and mesh resolution.
- `CalvingHydroInterp.F90` : Interpolates required variables between 3D ice mesh and 2D hydrology mesh, if using a multi-mesh approach. This is more complicated than it sounds.
- `HydroRestart.F90` : Allows separate 2D hydrology mesh to be restarted in a multi-mesh simulation.
- `USF_SourceCalcCalving.F90` : User function that calculates the source term for GlaDS as a combination of surface melt (provided in some user-specified variable or input file) and basal melt (worked out automatically from the residual of the TemperateIce solver)
- `BasalMelt3D.F90` : Solver that works out basal melt on ungrounded portions of a glacier.
- `GMValid.F90` : Solver that discriminates between ungrounded areas that are connected to the fjord and isolated ungrounded patches inland.
- `Permafrost.F90` : Collection of solvers for permafrost simiations of a saturated aquifer including heat transfer and phase change as well as solute transport and mechanical deformation (the latter involving the linear elasticity solver)
- `PermafrostMaterials.F90` : Module for material functions given by either a thermodynamically consistent model by Hartikainen or a simplified model by Andersson. Reads default values from `permafrostsolutedb.dat` and `permafrostmaterialdb.dat`.

Enhanced Solver/Userfunction Modules

- `GlaDSCoupledSolver.F90` : Modified to work on a secondary hydrology mesh (as opposed to the primary ice mesh) and to discriminate properly between fjord-connected ungrounded areas and isolated ungrounded patches inland. Also should work on the basal boundary of an internally extruded 3D mesh.
- `GlaDSchannelSolver.F90` : Changes to achieve the same outcome as above.
- `CalvingRemesh.F90` and `Calving3D.F90` : Changed to avoid interpolating hydrology-specific solvers to the ice mesh after calving. Also changed to allow ice solvers and calving to run at different timestep to hydrology.
- `GroundedSolver.F90` : Minor tweak to allow frontal grounded basal nodes to be listed as grounding-line nodes, so that the plume solver knows where to stick plumes

ElmerSolver library functionality

- Added Zoltan repartitioning capabilities to permit continuous runtime load balancing and to assist with calving remeshing.
- Added support for MMG3D remeshing/mesh adaptation.

Overview of changes/enhancements

- Improvements to inversion methods
- Coupling of the GlaDS solvers with the calving solvers in a two-mesh, 3D simulation. Includes a new plume solver that currently relies on the external ODEPack library (not included in the Elmer distribution).
- New thermodynamically consistent model for permafrost with saturated aquifers

New Solver/Userfunction Modules

- **Calving3D_lset.F90**: Return calving as a level set function (work in progress).
- **CalvingRemeshMMG.F90**: Cut a calving event directly out of a 3D mesh without external gmsh or mesh extrusion. Initial work on allowing calving margins to migrate.
- **PlumeSolver.F90** - associated ODEPack library files: opkda1.F, opkda2.F, opkdmain.F(not included in Elmer reopsitory): Provides plume melt rates across the calving front of a glacier. Fed by output from GlaDS solvers. Simulates a continuous sheet-style plume across entire front, split up into segments defined by frontal nodes and mesh resolution.
- **CalvingHydroInterp.F90**: Interpolates required variables between 3D ice mesh and 2D hydrology mesh, if using a multi-mesh approach. This is more complicated than it sounds.
- **HydroRestart.F90**: Allows separate 2D hydrology mesh to be restarted in a multi-mesh simulation.
- **USF_SourceCalcCalving.F90**: User function that calculates the source term for GlaDS as a combination of surface melt (provided in some user-specified variable or input file) and basal melt (worked out automatically from the residual of the TemperateIce solver)
- **BasalMelt3D.F90**: Solver that works out basal melt on ungrounded portions of a glacier.
- **GMValid.F90**: Solver that discriminates between ungrounded areas that are connected to the fjord and isolated ungrounded patches inland

New Solver/Userfunction Modules

- **Permafrost.F90**: Collection of solvers for permafrost simlations of a saturated aquifer including heat transfer and phase change as well as solute transport and mechanical deformation (the latter involving the linear elasticity solver)
- **PermafrostMaterials.F90**: Module for material functions given by either a thermodynamically consistent model by Hartikainen or a simplified model by Andersson. Reads default values from permafrostsolutedb.datand permafrostmaterialdb.dat.

Enhanced Solver/Userfunction Modules

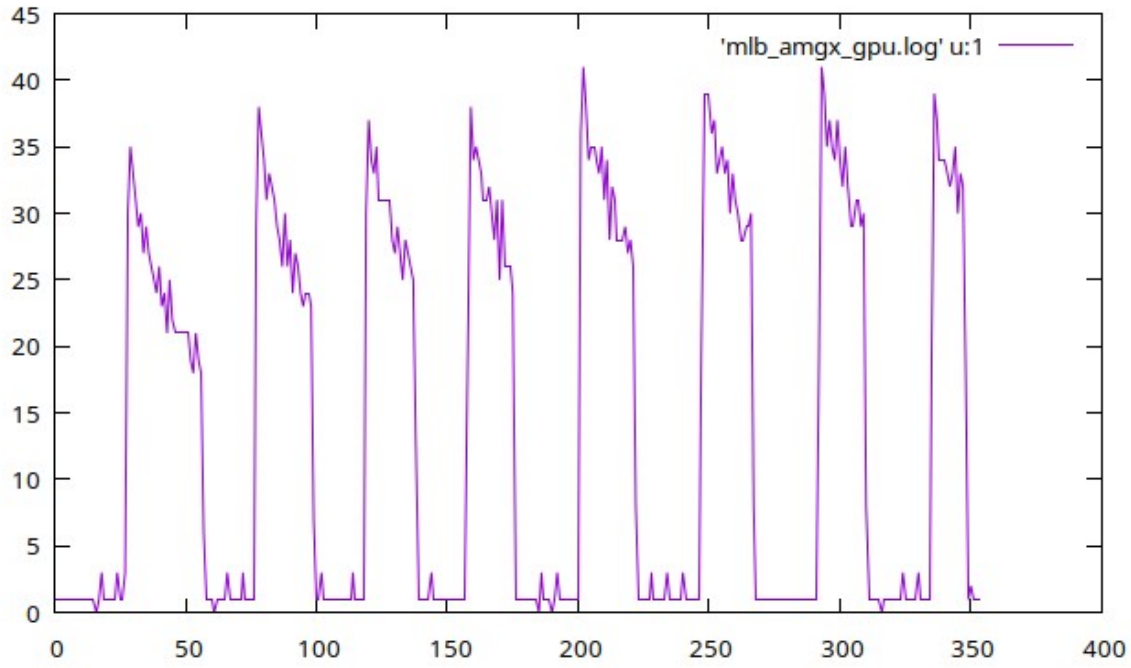
- **GlaDSCoupledSolver.F90**: Modified to work on a secondary hydrology mesh (as opposed to the primary ice mesh) and to discriminate properly between fjord-connected ungrounded areas and isolated ungrounded patches inland. Also should work on the basal boundary of an internally extruded 3D mesh.
- **GlaDSchannelSolver.F90**: Changes to achieve the same outcome as above.
- **CalvingRemesh.F90** and **Calving3D.F90**: Changed to avoid interpolating hydrology-specific solvers to the ice mesh after calving. Also changed to allow ice solvers and calving to run at different timestep to hydrology.
- **GroundedSolver.F90**: Minor tweak to allow frontal grounded basal nodes to be listed as grounding-line nodes, so that the plume solver knows where to stick plumes

ElmerSolver library functionality

- Added Zoltan repartitioning capabilities to permit continuous runtime load balancing and to assist with calving remeshing.
- Added support for MMG3D remeshing/mesh adaptation.

AMGX interface in Elmer

- New branch origin/amgx_interface
- Interface to GPU enabled linear solver library AMGX (https://developer.nvidia.com/amgx)
- Catch:
  - Needs a NVIDIA GPU
  - Currently only serial (CPU) and single GPU
  - Only treats the solution part – assembly needs to be done on CPU (in serial, at the moment)
- Test on gaming PC shows similar performance of the new serial GPU version compared to a multi-core CPU-only run



AMGX

- Test-runs on gaming PC (6 physical cores latest Gen i5 + NVidia RTX 2060) comparing the MLB example between a 6-core MPI run using block precoditioner with GCR + IDRS and a single-core + GPU run using AMGX

```
==> mlb_linsys_25.log <==
ComputeChange: NS (ITER=4) (NRM,RELC): (  1.0783757     0.69187330E-01 ) :: stokes-vec
ComputeChange: NS (ITER=5) (NRM,RELC): ( 0.93999831     0.13711768     ) :: stokes-vec
ComputeChange: NS (ITER=6) (NRM,RELC): ( 0.94859671     0.91056054E-02 ) :: stokes-vec
ComputeChange: NS (ITER=7) (NRM,RELC): ( 0.94869944     0.10828618E-03 ) :: stokes-vec
ComputeChange: NS (ITER=8) (NRM,RELC): ( 0.94869973     0.30732043E-06 ) :: stokes-vec
ComputeChange: SS (ITER=1) (NRM,RELC): ( 0.94869973      2.0000000     ) :: stokes-vec
ElmerSolver: *** Elmer Solver: ALL DONE ***
ElmerSolver: The end
SOLVER TOTAL TIME(CPU,REAL):       315.01      319.54
ELMER SOLVER FINISHED AT: 2020/06/03 13:33:29
```

```
==> mlb_amgx_ser_25.log <==
Total Time: 0.00515078
    setup: 2.464e-06 s
    solve: 0.00514832 s
    solve(per iteration): 0.000572036 s
ComputeChange: NS (ITER=8) (NRM,RELC): ( 0.93789503     0.30825326E-06 ) :: stokes-vec
ComputeChange: SS (ITER=1) (NRM,RELC): ( 0.93789503      2.0000000     ) :: stokes-vec
ElmerSolver: *** Elmer Solver: ALL DONE ***
ElmerSolver: The end
SOLVER TOTAL TIME(CPU,REAL):       345.84      347.24
ELMER SOLVER FINISHED AT: 2020/06/03 13:25:51
```