



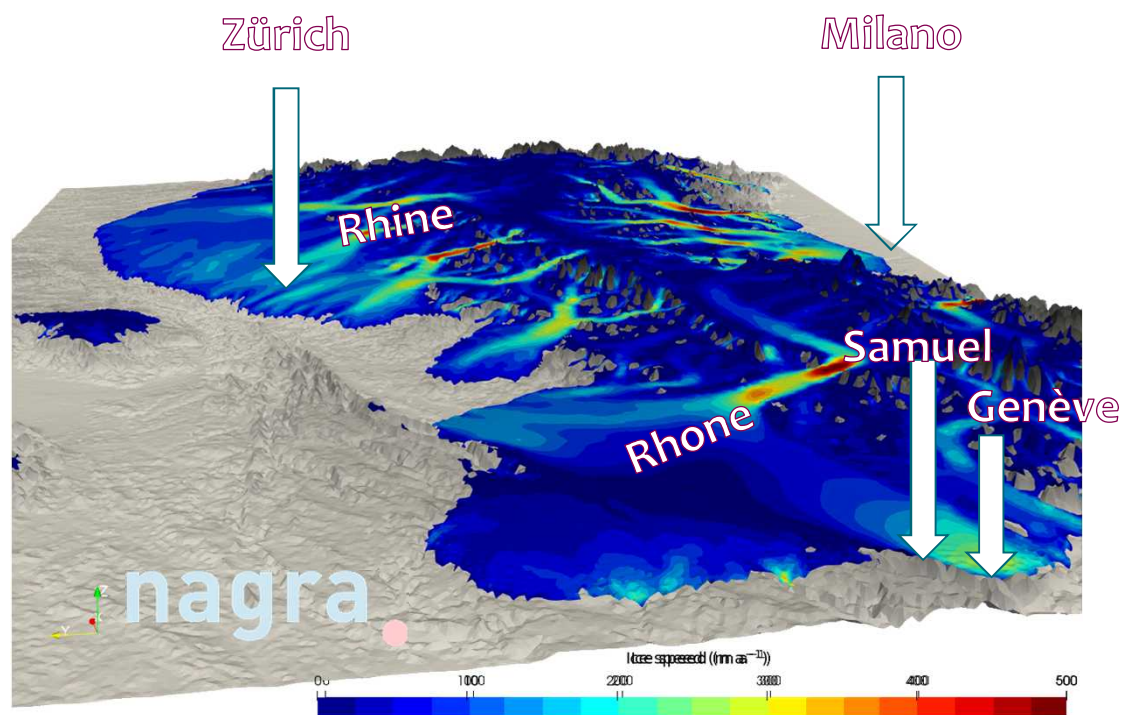
Stabilization scheme of free surface problems



André Löfgren, Josefin Ahlkrona,
Christian Helanow and Thomas Zwinger



Free surface stability on rugged terrain



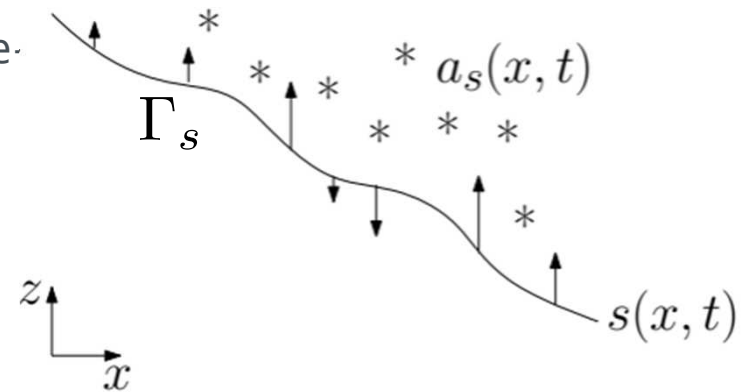
Simulation
D. Cohen
for NAGRA

The free-surface equation

- Surface evolution governed by the kinematic free-surface equation

$$\frac{\partial s}{\partial t} + u_x^s \frac{\partial s}{\partial x} + u_y^s \frac{\partial s}{\partial y} = u_z^s + a_s$$

- Mass balance between accumulation
- and the flow of ice.
- Coupled to the Stokes flow through the surface velocities.



The Stokes equation

- Bilinear form

$$a((\mathbf{u}, p), (\mathbf{v}, q)) = (2\eta(\mathbf{u})\dot{\boldsymbol{\varepsilon}}(\mathbf{u}) : \dot{\boldsymbol{\varepsilon}}(\mathbf{v}))_{\Omega} - ((\nabla \cdot \mathbf{u})q)_{\Omega} - (p\nabla \cdot \mathbf{v})_{\Omega}$$

- Linear form

$$L(\mathbf{v}) = (\mathbf{f} \cdot \mathbf{v})_{\Omega}$$

- Coupled system suffers from instabilities.



The free-surface stabilization algorithm (FSSA)

- Taylor expand force term in Stokes weak form to mimic an implicit time-stepping scheme (Kaus et al. 2010)[1]

$$(\mathbf{f} \cdot \mathbf{v})_{\Omega^{k+1}} \approx (\mathbf{f} \cdot \mathbf{v})_{\Omega^k} + \theta \Delta t ((\mathbf{u} \cdot \mathbf{n})(\mathbf{f} \cdot \mathbf{v}))_{\Gamma_s^k}$$

- Adapt to ice-sheet modeling by adding accumulation (Löfgren, Ahlkrone, Helanow 2022)[2]

$$(\mathbf{f} \cdot \mathbf{v})_{\Omega^{k+1}} \approx (\mathbf{f} \cdot \mathbf{v})_{\Omega^k} + \theta \Delta t ((\mathbf{u} \cdot \mathbf{n})(\mathbf{f} \cdot \mathbf{v}))_{\Gamma_s^k} + \theta \Delta t ((a_s \mathbf{z} \cdot \mathbf{n})(\mathbf{f} \cdot \mathbf{v}))_{\Gamma_s^k}$$

from advection

~~from
accumulation/ablation~~



Implementation in Elmer

- **IncompressibleNSVec**
- 'full' is the complete implementation from below
- 2 version originating from initial approximation (now implemented in matrix) declaring a slip coefficient
 - in normal-direction based on vertical velocity ('normal')
 - in vertical-direction (dim) based on normal velocity ('transposed')

```

SELECT CASE(FSSAFlag)
! approximation with normal pointing into z-direction
CASE ('normal')
DO p=1,nd
DO q=1,nd
DO i=dim,dim
STIFF( (p-1)*c+dim,(q-1)*c+i ) = &
STIFF( (p-1)*c+dim,(q-1)*c+i ) + &
s * FSSAcoeff * Basis(q) * Basis(p) * Normal(i)
END DO
END DO
END DO
! version 2, transposed
CASE ('transposed')
DO p=1,nd
DO q=1,nd
DO i=1,dim
STIFF( (p-1)*c+i,(q-1)*c+dim ) = &
STIFF( (p-1)*c+i,(q-1)*c+dim ) + &
s * FSSAcoeff * Basis(q) * Basis(p) * Normal(i)
END DO
END DO
END DO
CASE ('full') ! full entry matrix FSSA
DO p=1,nd
DO q=1,nd
DO i=1,dim
STIFF( (p-1)*c+dim,(q-1)*c+i ) = &
STIFF( (p-1)*c+dim,(q-1)*c+i ) + &
s * FSSAcoeff * Basis(q) * Basis(p) * Normal(i)
!PRINT *, "K(",p,q,i,")=", s * FSSAcoeff * Basis(q) * Basis(p)
& * Normal(i), STIFF( (p-1)*c+dim,(q-1)*c+i )
END DO
END DO
END DO

```

$$(\mathbf{f} \cdot \mathbf{v})_{\Omega^{k+1}} \approx (\mathbf{f} \cdot \mathbf{v})_{\Omega^k} + \theta \Delta t ((\mathbf{u} \cdot \mathbf{n})(\mathbf{f} \cdot \mathbf{v}))_{\Gamma_k}$$

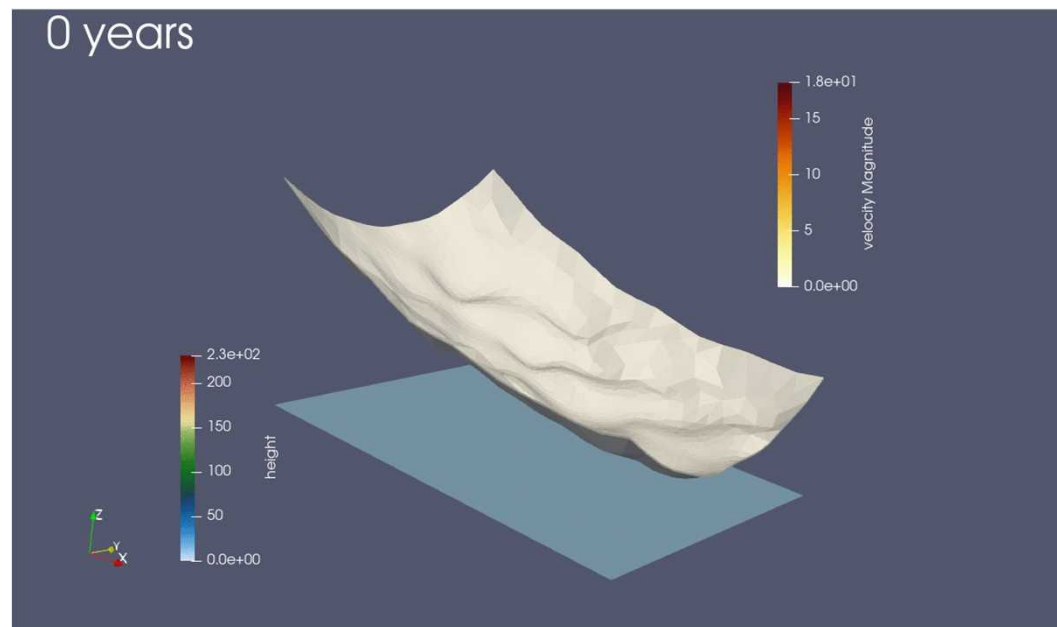
Implementation in Elmer

- In free-surface BC add 2 keywords:
- **FSSA Coefficient:**
 - $\theta \in [0,1]$
- **FSSA Flag:**
 - full
 - normal
 - transposed
 - none (=default)

```
!---LUA BEGIN
! year_s=365.25*24*3600
! pa_to_mpa=1.0E-6
! rho=910*year_s^(-2.0)*pa_to_mpa
! gravity=9.8*year_s^(2.0)
! time_step=5.0
! theta=1
! stabilization_parameter=theta*time_step*rho*gravity
!---LUA END

Boundary Condition 3
  Name = "surface"
  Top Surface = Equals "Zs"
  Body ID = 2
  FSSA Coefficient = Real #stabilization_parameter
  FSSA Flag = String "full" ! "normal" "transposed"
End
```

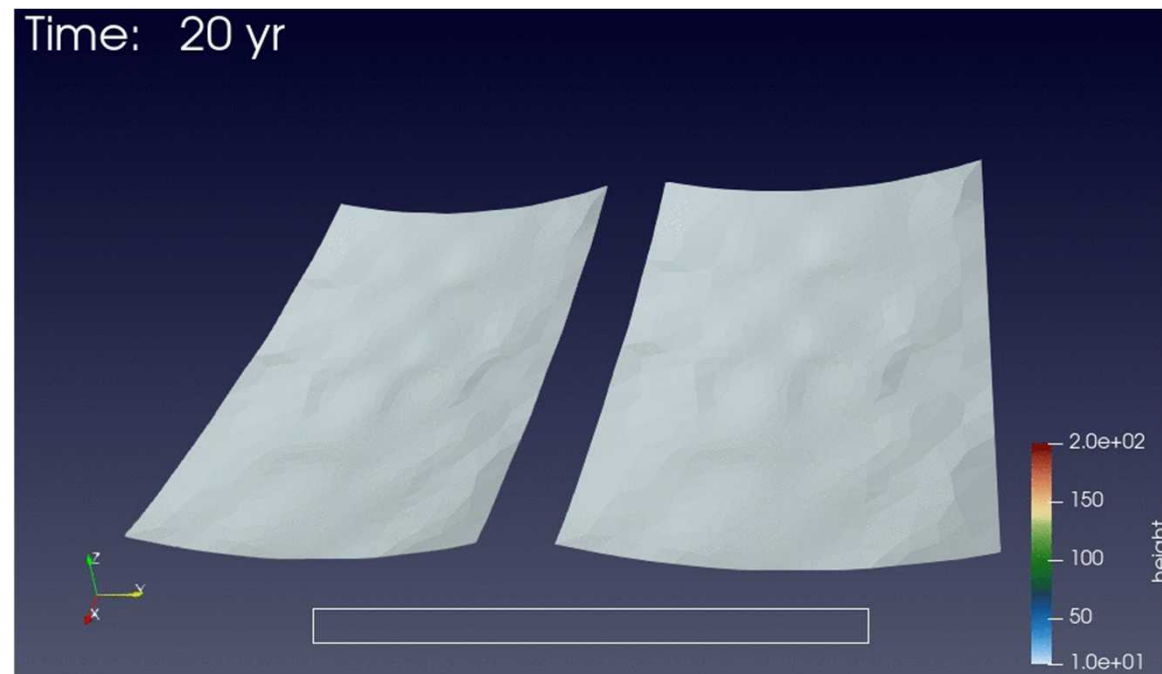
Synthetic test-case



- Synthetic landscape using algorithm also used in gaming (Perlin)
- $8 \times 5 = 40 \text{ km}^2$
- Accumulation in uppermost part, no ablation!
- Low slip in upper part, high slip in lower

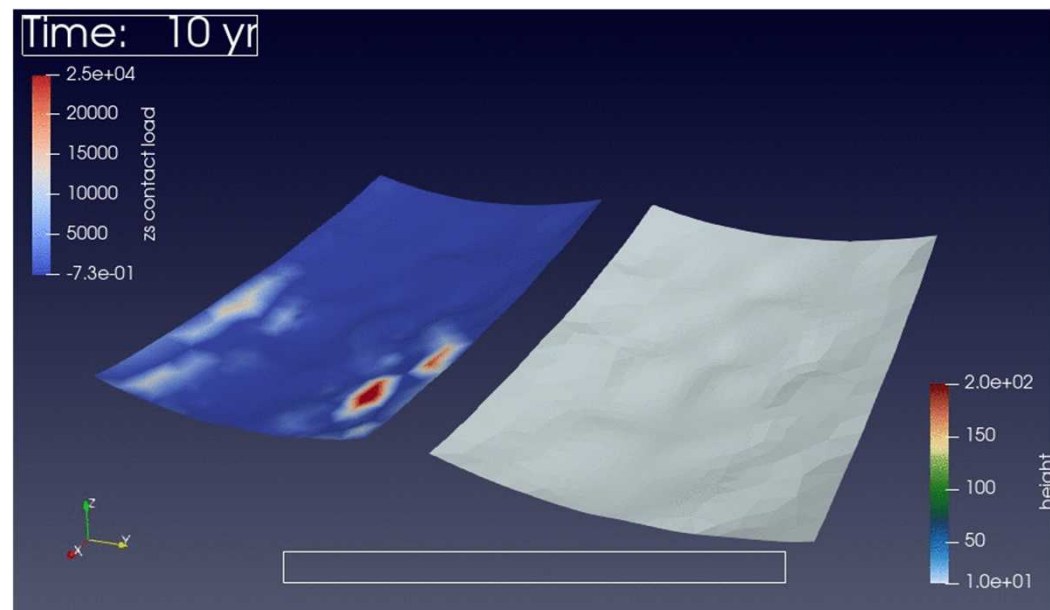
Synthetic test-case

- Comparison between run with full FSSA (left) and no surface stabilization (right)
- Timestep-size 20 years
- Minimum mesh size 100 m
- Maximum front velocities ~20 m/a



Synthetic test-case

- Minimum flow height 10 m
- Need limiters, particular in steep, non-glaciated parts
- Built-in limiters (Apply Dirichlet=True) in FreeSurfaceSolver did not work
- Suggest to use library limiters from now on



Changes to FreeSurfaceSolver

```

Meld File Edit Changes View Tabs
elmerfem-fssa FreeSurfaceS...ory, working
FreeSurfaceSolver.F90 -- repository
FreeSurfaceSolver.F90

821 ! transient simulations.
822 !-----
823 CALL DefaultFinishAssembly()
824 CALL DefaultDirichletBCs()
825
826 !-----
827 ! Manipulation of the assembled matrix due to limits
828 !-----
829 IF (ApplyDirichlet) THEN
830   OldValues = SystemMatrix % Values
831   OldRHS = ForceVector
832   !-----
833   ! manipulation of the matrix
834   !-----
835   DO i=1,Model % Mesh % NumberOfNodes
836     k = FreeSurfPerm(i)
837     IF ((ActiveNode(i,1) .AND. ActiveNode(i,2)) &
838         CALL FATAL(SolverName,"Upper as well as lower limiter active - this is a deadlock")
839     IF ((ActiveNode(i,1) .OR. ActiveNode(i,2)) .AND. (k > 0)) THEN
840       CALL ZeroRow( SystemMatrix, k )
841       CALL SetMatrixElement( SystemMatrix, k, k, 1.0_dp )
842       IF(ActiveNode(i,1)) THEN
843         SystemMatrix % RHS(k) = LowerLimit(i)
844       ELSE
845         SystemMatrix % RHS(k) = UpperLimit(i)
846       END IF
847     END IF
848   END DO
849 END IF
850
851 CALL Info( SolverName, 'Assembly done', Level=6 )
852 !-----
853 ! Solve System and check for convergence
854 !-----
855 at = CPUtime() - at
856 st = CPUtime()
857 PrevNorm = Solver % Variable % Norm
858 Norm = DefaultSolve()
859 IF ( PrevNorm + Norm /= 0.0_dp ) THEN
860   RelativeChange = 2.0_dp * ABS( PrevNorm-Norm ) / (PrevNorm + Norm)
861 ELSE
862   RelativeChange = 0.0_dp
863 END IF
864 WRITE( Message, * ) 'Result Norm : ',Norm
865 CALL Info( SolverName, Message, Level=4 )
866 WRITE( Message, * ) 'Relative Change : ',RelativeChange
867 CALL Info( SolverName, Message, Level=4 )
868
869 !-----
870 ! special treatment for periodic boundaries
871 !-----
733 ! Update global matrix and rhs vector from local matrix & vector
734 !-----
735 CALL DefaultUpdateEquations( STIFF, FORCE )
736 !-----
737 END DO ! End loop bulk elements
738
739 CALL DefaultFinishBulkAssembly()
740 !-----
741 ! Neumann & Newton boundary conditions
742 !-----
743 ! MIND: In weak formulation it is not possible to prescribe a contact angle on
744 ! a boundary in this solver. This has to be taken care of in the boundary
745 ! condition for the stress tensor in the Navier-Stokes Solver. Thus, in
746 ! generally it does not make sense to prescribe a Neumann type of
747 ! condition here.
748 !-----
749 ! FinishAssembly must be called after all other assembly steps, but before
750 ! Dirichlet boundary settings. Actually no need to call it except for
751 ! transient simulations.
752 CALL DefaultFinishAssembly()
753 CALL DefaultDirichletBCs()
754
755 CALL Info( SolverName, 'Assembly done', Level=6 )
756 !-----
757 ! Solve System and check for convergence
758 !-----
759 at = CPUtime() - at
760 st = CPUtime()
761 PrevNorm = Solver % Variable % Norm
762 Norm = DefaultSolve()
763 IF ( PrevNorm + Norm /= 0.0_dp ) THEN
764   RelativeChange = 2.0_dp * ABS( PrevNorm-Norm ) / (PrevNorm + Norm)
765 ELSE
766   RelativeChange = 0.0_dp
767 END IF
768 WRITE( Message, * ) 'Result Norm : ',Norm
769 CALL Info( SolverName, Message, Level=4 )
770 WRITE( Message, * ) 'Relative Change : ',RelativeChange
771 CALL Info( SolverName, Message, Level=4 )
772
773 !-----
774 ! special treatment for periodic boundaries
775 !-----
Ln 761, Col 39 INS

```

Changes to FreeSurfaceSolver

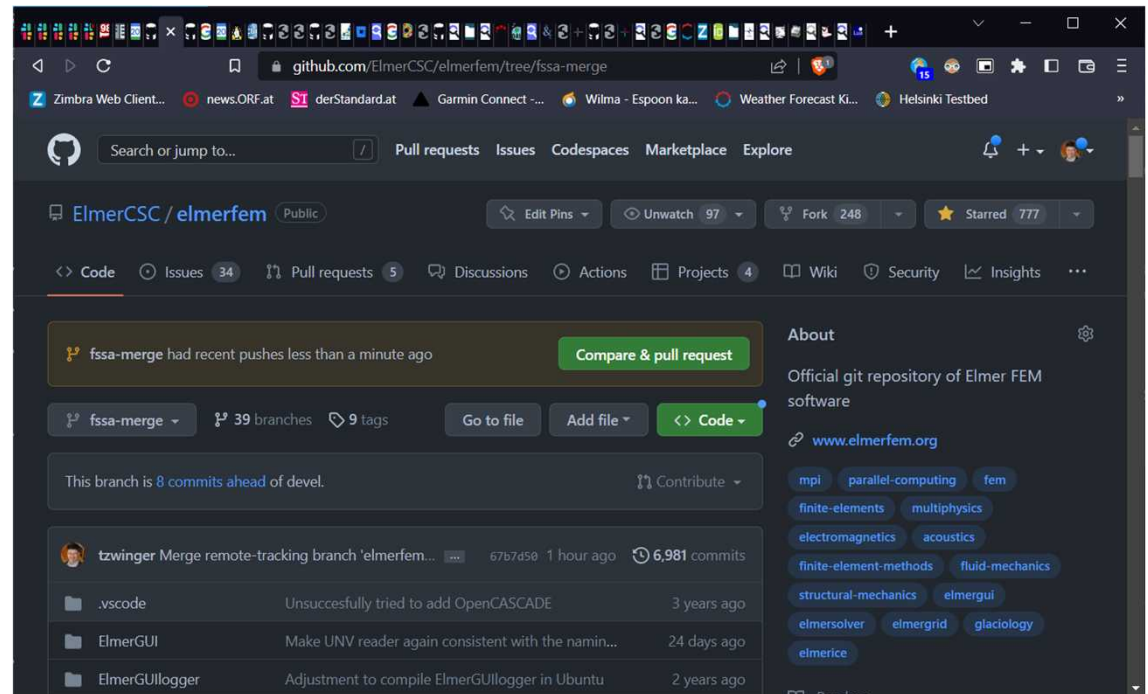
```
> gsu
On branch merge-elmercsc
Your branch is up-to-date with 'origin/merge-elmercsc'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   elmerice/Tests/AIFlowSolve/AIFlow.sif
    modified:   elmerice/Tests/Buoyancy/buoyancy.sif
    modified:   elmerice/Tests/Calving2D/calving2d.sif
    modified:   elmerice/Tests/Calving3D/calving3d.sif
    modified:   elmerice/Tests/Calving3D_lset/calvingMMG.sif
    modified:   elmerice/Tests/Contact/cavity.sif
    modified:   elmerice/Tests/Damage/damage.sif
    modified:   elmerice/Tests/GL_MISMIP/mismip.sif
    modified:   elmerice/Tests/Grounded/grounded.sif
    modified:   elmerice/Tests/MISMIP_FS-SSA/coupled.sif
    modified:   elmerice/Tests/Teterousse3a/teterousse3a.sif
    modified:   elmerice/Tests/Teterousse3a_b/test.sif
    modified:   elmerice/Tests/Teterousse_DeformHeat/test.sif
    modified:   fem/src/modules/FreeSurfaceSolver.F90
    modified:   fem/tests/freesurf_ltd/Free_ltd.sif

Untracked files not listed (use -u option to show untracked files)
> git commit -m "removed built-in Dirichlet limiters in FreeSurfaceSolver and changing
> git commit -m "removed built-in Dirichlet limiters in FreeSurfaceSolver and changing tests accord
e library-provided limiters instead"
[merge-elmercsc c4507f5] removed built-in Dirichlet limiters in FreeSurfaceSolver and changing tests
ly to use library-provided limiters instead
15 files changed, 102 insertions(+), 241 deletions(-)
```

Conclusions

- New branch under <https://github.com/ElmerCSC/elmerfem/tree/fssa-merge>
- Introduced FSSA into **IncompressibleNSVec**
 - Not in legacy Navier-Stokes!
- Significant changes to **FreeSurfaceSolver**
 - moving from built-in to library limiters
 - update of test



References

[1] Kaus, B.J.P., et al., *A stabilization algorithm for geodynamic numerical simulations with a free surface*. **Phys. Earth Planet. In.** (2010), doi:10.1016/j.pepi.2010.04.007

[2] Löfgren, A., et al., *Increasing stable time-step sizes of the free-surface problem arising in ice-sheet simulations*. **J. Comput. Phys. X** (2022), doi:10.1016/j.jcpX.2022.100114



facebook.com/CSCfi



twitter.com/CSCfi



youtube.com/CSCfi



linkedin.com/company/csc---it-center-for-science



github.com/CSCfi

<https://github.com/ElmerCSC/elmerfem>