Semi-Lagrangain Advection in Underworld

Dave Lee

February 24, 2015

The semi-Lagrangian scheme is applied as different components for different geometries, as listed below. In each case, when using periodic boundary conditions, the parallel decomposition must be applied as a breadslice decomposition in the aperiodic dimension. This is because the mesh does not incorporate periodic boundaries as shadow zones and so these cannot be used to interpolate departure points. This is done by including the following in the XML input file:

```
<include>StgFEM/BreadSliceDecomposition.<dim>.xml</include>
```

Where `<dim>=x,y,z` is the aperiodic dimension.

1 Cartesian Geometry

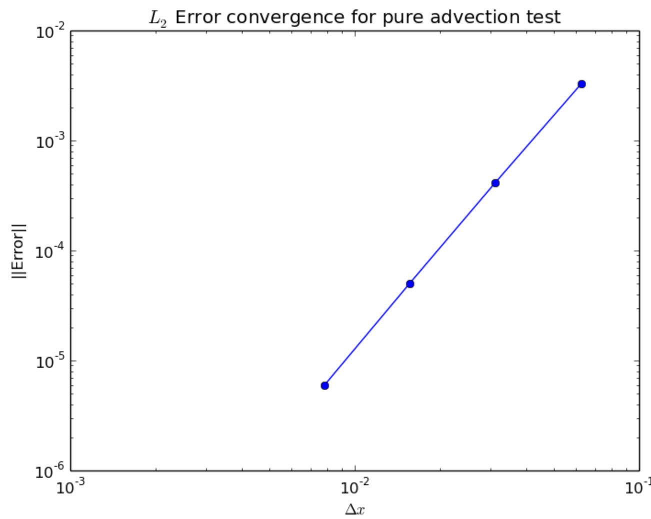
The Castesian version of the semi-Lagrangian integrator is specified in the component `SemiLagrangianIntegrator` which is instatiated in the input file `StgFEM/Apps/StgFEM_Components/SemiLagrangianIntegrator.xml`. The component itself in instatiated as

```
<struct name="integrator">
    <param name="Type">SemiLagrangianIntegrator</param>
    <param name="VelocityField">VelocityField</param>
    <param name="Context">context</param>
    <list name="fields">
        <param>TemperatureField</param>
        <param>temperatureStarField</param>
        <param>false</param>
    </list>
</struct>
```

Here the `fields` list may contain a list of entries where each entry has three parts, the field to be advection (`TemperatureField`), the temporary field generated by the integrator (`temperatureStarField`) and a boolean value which determines if the field is purely advected (`true`), or is advected as part of a SLE and so is updated as part of the solve (`false`). In this way the advector may be applied to multiple fields.

1.1 Test 1: Pure Advection

A test for a solitary wave field advected in a shear cell is located at `StgFEM/Apps/SemiLagrangian`. The test may be run from the python script `testSL.py` located there, which lauches a set of pure advection models at varying resolutions and generates an error convergence plot as shown below.



These errors have a convergence rate of 3.03, which is expected for a scheme with 3rd order interpolants run with a static time step as done here.

1.2 Test 2: Advection Diffusion

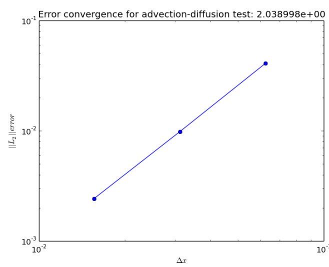
A test for the semi-Lagrangian scheme as part of an advection diffusion equation is found in `Underworld/InputFiles/AdvDiffSL` and may be run with the script `testSL.py` located there. This generates an error convergence plot for an advection diffusion equation of the form

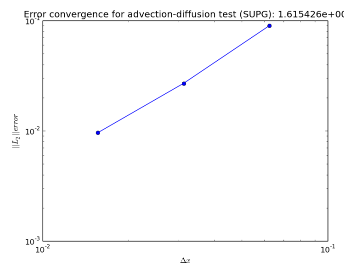
$$\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = \nu\nabla^2 T \tag{1}$$

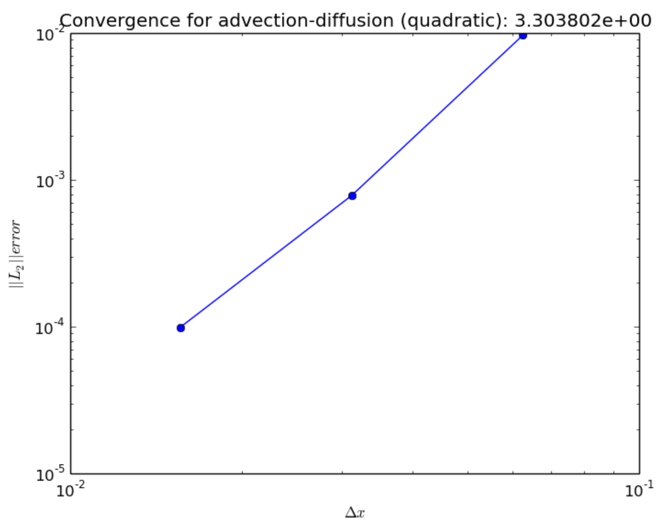
with the solution $T(x, y, t) = e^{kx+ly-\nu(k^2+l^2)t)} \cos(kx + ly - (ku + lv)t)$. The problem is discretised in time using the Crank-Nicholson formulation for second order accuracy as

$$[1 + 0.5\Delta t\nabla^2]T^{n+1} = (1 - 0.5\Delta t\nabla^2)T^d \tag{2}$$

where T^d is the temperature at the semi-Lagrangian departure points. These have a convergence rate of 2.04. The lower convergence rate than the pure advection test reflests the fact that a diffusion problem must also be solved


Error convergence for advection-diffusion test: 2.038998e+00
$||L_2||error$
Δx


Error convergence for advection-diffusion test (SUPG): 1.615426e+00
$||L_2||error$
Δx

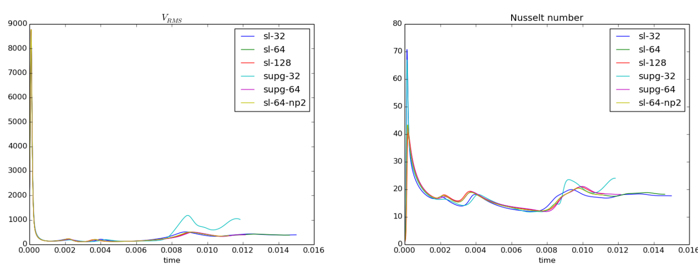

Convergence for advection-diffusion (quadratic): 3.303802e+00
10^{-2}
10^{-3}
10^{-4}
10^{-5}
$||L_2||error$
10^{-2}
10^{-1}
Δx

for each fixed time step, and this has lower spatial accuracy than the semi-Lagrangian scheme. This convergence rate is superior to that of the SUPG scheme however. The Cartesian semi-Lagrangian integrator is also implemented for quadratic elements. As shown here the error convergence rate for advection diffusion problems is 3.3, such that the accuracy of the interpolation in the semi-Lagrangian scheme is the limiting factor.

Also located in this folder is the file `AdvDiffSolverCN.xml`. This sets up the advection-diffusion problem with the semi-Lagrangian integrator using a second order accurate Crank-Nicholson time discretisation. It may be included in other models as well. The semi-Lagrangian advection scheme may be substituted for the SUPG scheme by including the xml file `Underworld/InputFiles/BaseApps/SemiLagrangianADESolverHijack.xml`.

1.3 Test 3: Thermal Convection

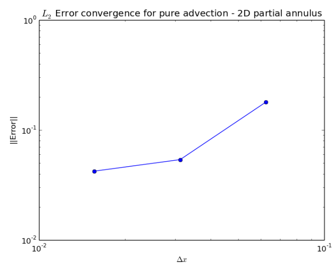
Within the same directory is a setup file for a thermo-convection model for comparison against existing results, `FrankKamenetskii.xml`.

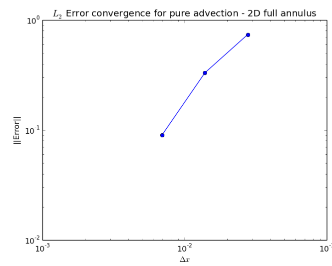


2 2D Polar Geometry

The semi-Lagrangian scheme for 2D polar geometry is specified in the component `SLIntegrator_Polar`. This has the same XML interface as the Cartesian integrator, and may be used by including the file `SLIntegrator_Polar.xml` located in the directory `Spherical/InputFiles/SemiLagrangian`. This directory also includes the setup files for pure advection for a partial and full annulus, `PureAdvection_2DPartialAnulus.xml` and `PureAdvection_2DAnulus.xml` respectively.

Note that the for all the 2D and 3D spherical geometry models the outer wall boundary condition must be Dirichlet. This is because the element geometry is not curved and so departure points may end up outside any element but still inside the bounding sphere. For a fixed time step with varying spatial resolution, the convergence rate is also sub-optimal as characteristics can overshoot the outer boundary elements at high resolution, as can be seen for the convergence rates below, which are 1.04 and 1.52 for the partial and full annuli respectively.



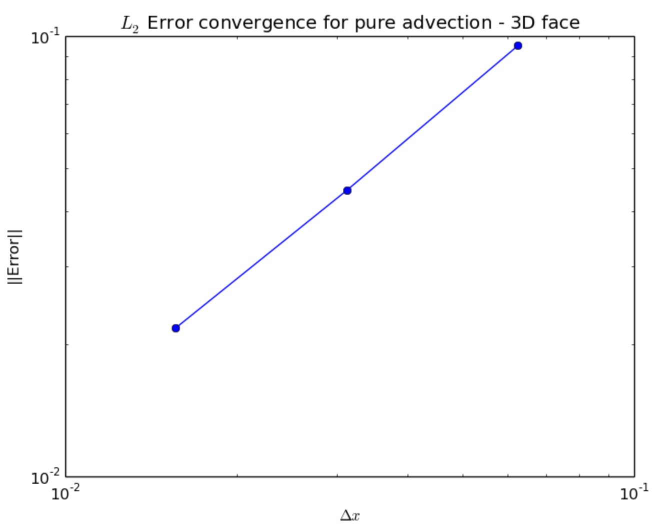


3 3D Cubed Sphere (Single Face)

The 3D implementation for a single face is given in the component `SLIntegrator_Spherical` and may be used via the XML file `SLIntegrator_Spherical.xml` within the directory `Spherical/InputFiles/SemiLagrangian`.

3.1 Test 1: Pure Advection

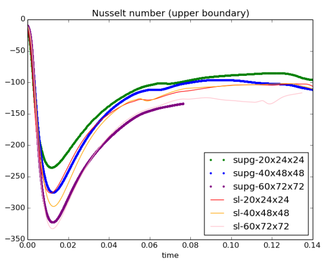
A pure advection test for a velocity field in a parametric circle is given in the file `PureAdvection_3DFace.xml`.

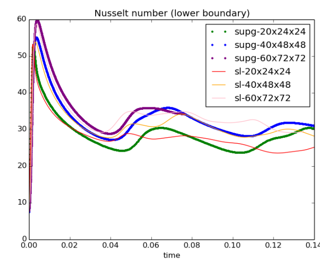


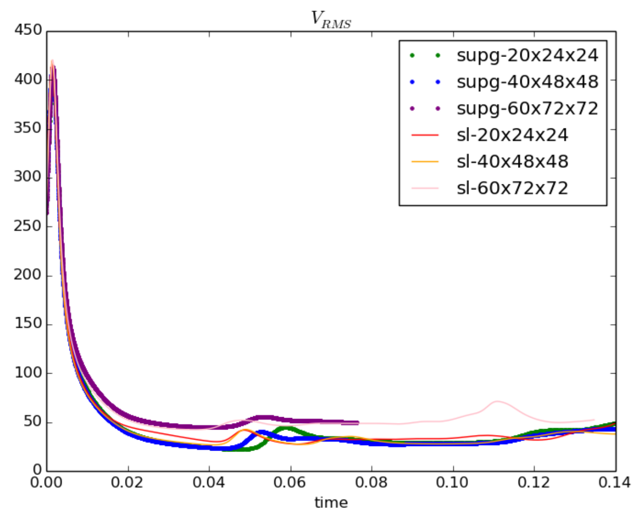
These errors have a convergence rate of 1.06. This is significantly lower than the theoretical convergence rate of 3.0, which requires further investigation.

3.2 Test 2: Thermal Convection

A set of convergence tests have been performed for a thermal convection benchmark in the 3D face geometry, and compared against the SUPG implementation for the energy equation. This test is driven by the `Benchmark2_sixth.py` script. The comparisons for the Nusselt number at the upper and lower boundaries and the root mean square velocity for the semi-Lagrangian and SUPG schemes are given below.



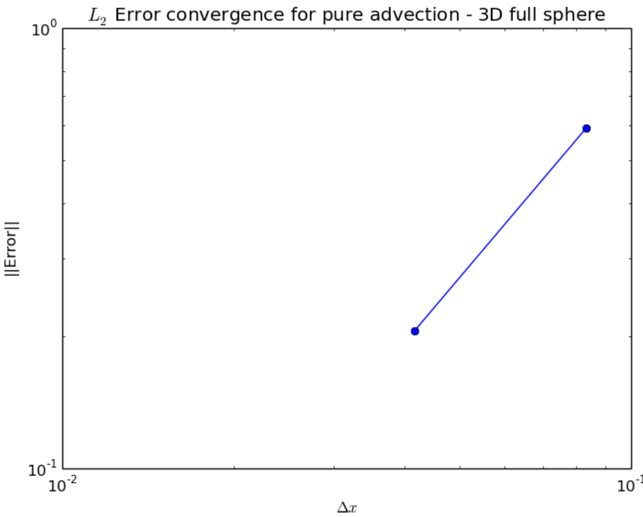




4 3D Cubed Sphere (Full Sphere)

The 3D implementation for the semi-Lagrangian scheme in the full sphere is given in the component `SLIntegrator_FullSphere`. Within the directory `Spherical/InputFiles/SemiLagrangian` a test for pure advection with a parametric circle velocity field is given in `PureAdvection_3DSphere.xml`.

The errors for pure advection on the full sphere converge at a rate of 1.52, again significantly below the anticipated rate of 3.0. Note that currently the mesh generator for the full 3D sphere `SphericalMesh` is not working in parallel, and so the `SLIntegrator_FullSphere` has not been tested in parallel.



5 Timings

Model time / CPU time

Model	SUPG	SL
Cartesian	0.001195	0.001026
Polar	3.45e-5	1.56e-5
Spherical	6.279e-6	3.94e-6