Report 19/08/2020

1 Testing AD Exact/ dissecting domains/ etc – Forward

Testing the exact solution for AD on an infinite domain. It works well on domain in Figure 1. Errors are small and decrease with increasing number of points.

$$N = 20 \quad Abs = 0.00015915 \quad Rel = 1.2555 \times 10^{-7}$$
$$N = 30 \quad Abs = 1.1279 \times 10^{-6} \quad Rel = 1.1311 \times 10^{-9}$$
$$N = 40 \quad Abs = 1.1141 \times 10^{-6} \quad Rel = 1.1173 \times 10^{-9}.$$

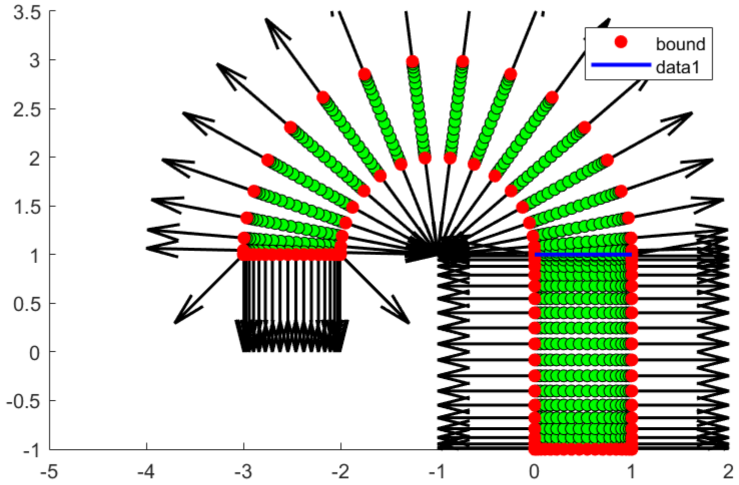


Figure 1: Domain1

When dissecting the two shapes into two sub-shapes each, the domain is like in Figure 2. Running the problem with $N = 20$, the solution looks the same to the above and gives the errors:

$$N = 20 \quad Abs = 1.6234 \times 10^{-6} \quad Rel = 1.1511 \times 10^{-9}.$$

Running the 'CompareAll' function on the dissected domain gives errors as in screenshot Figure 3. The only potential point of concern is the interpolation error, see Figure 4. We get errors

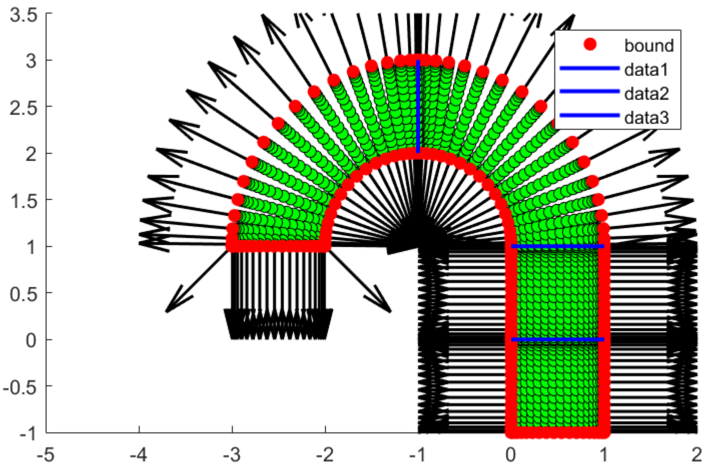


Figure 2: Domain1 dissected

of up to 10^{-4}. However, I think overall it is ok. I think it may be that 20 points on only two shapes are not enough but on four shapes they are. When rerunning interpolation with $N = 30$, the error decreases to 10^{-6}. Out of interest, I ran the exact solution on a larger domain, see Figure 5. The errors are essentially the same as the ones without the extra shape. However, most of the particle mass is in the first two shapes anyway, so I am not quite sure if this is the best test.

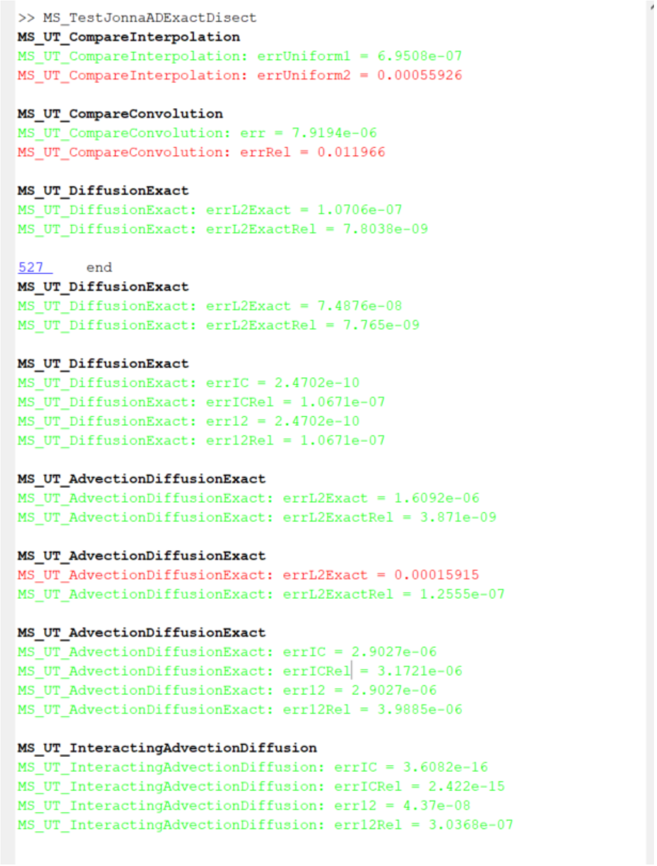
2 AD Not Exact

Tested two different problems with an advection-diffusion no-flux problem. Example 1 is made of 4 shapes and the problem is solved without interactions present. The advection vector is just constant in the 'direction of the shape' and of strength 10. The initial condition is:$exp(-2((y1-0.5)^2+(y2+1)^2))$. The dynamics is in Figure 6. In the second problem we add interaction and a no-flux wall. We have three shapes. The initial condition is: $exp(-2((y1-0.8)^2+(y2-0.2)^2))$. The result for $\kappa = 1$ is in Figure 7. The result for $\kappa = -1$ is in Figure 8. The differences are not very visible, however the two figures do show slightly different profiles for ρ.

3 Migrating to PDECO

Started migrating the code to PDECO framework. A few problems arise:

- No 'ComputeAll' function for MS. Therefore rewrote Computation but probably not the best



```
>> MS_TestJonnaADExactDisect
MS_UT_CompareInterpolation
MS_UT_CompareInterpolation: errUniform1 = 6.9508e-07
MS_UT_CompareInterpolation: errUniform2 = 0.00055926

MS_UT_CompareConvolution
MS_UT_CompareConvolution: err = 7.9194e-06
MS_UT_CompareConvolution: errRel = 0.011966

MS_UT_DiffusionExact
MS_UT_DiffusionExact: errL2Exact = 1.0706e-07
MS_UT_DiffusionExact: errL2ExactRel = 7.8038e-09

527     end
MS_UT_DiffusionExact
MS_UT_DiffusionExact: errL2Exact = 7.4876e-08
MS_UT_DiffusionExact: errL2ExactRel = 7.765e-09

MS_UT_DiffusionExact
MS_UT_DiffusionExact: errIC = 2.4702e-10
MS_UT_DiffusionExact: errICRel = 1.0671e-07
MS_UT_DiffusionExact: err12 = 2.4702e-10
MS_UT_DiffusionExact: err12Rel = 1.0671e-07

MS_UT_AdvectionDiffusionExact
MS_UT_AdvectionDiffusionExact: errL2Exact = 1.6092e-06
MS_UT_AdvectionDiffusionExact: errL2ExactRel = 3.871e-09

MS_UT_AdvectionDiffusionExact
MS_UT_AdvectionDiffusionExact: errL2Exact = 0.00015915
MS_UT_AdvectionDiffusionExact: errL2ExactRel = 1.2555e-07

MS_UT_AdvectionDiffusionExact
MS_UT_AdvectionDiffusionExact: errIC = 2.9027e-06
MS_UT_AdvectionDiffusionExact: errICRel = 3.1721e-06
MS_UT_AdvectionDiffusionExact: err12 = 2.9027e-06
MS_UT_AdvectionDiffusionExact: err12Rel = 3.9885e-06

MS_UT_InteractingAdvectionDiffusion
MS_UT_InteractingAdvectionDiffusion: errIC = 3.6082e-16
MS_UT_InteractingAdvectionDiffusion: errICRel = 2.422e-15
MS_UT_InteractingAdvectionDiffusion: err12 = 4.37e-08
MS_UT_InteractingAdvectionDiffusion: err12Rel = 3.0368e-07
```

Figure 3: Errors dissected

way of fixing this.
- Preprocessing in PDECO is not compartible with MS because inputs don't match. Changed it in Multishape but that probably isn't working for directly making the shape.
- There's a problem with DataStorage and multiple shapes. It can take one rectangular quadrilateral though.

Tested with a box made by MS code and Box code. Tested this with both exact AD on infinite domain and on exact solution for Neumann Flow.
Tested two boxes that make our standard domain with ADInf. Tested box and wedge with ADInf.
Compared the solution to normal box by plotting. It's the same
All errors with respect to the exact solution are of order 10^{-9} or 10^{-8}. Errors on the same domain for the two different shape classes are 10^{-13} for $rhoNum$.
can run a simple AD problem like above. with constant velocity profile (See figure 6 for example). Checked no flux boundary between two shapes too.

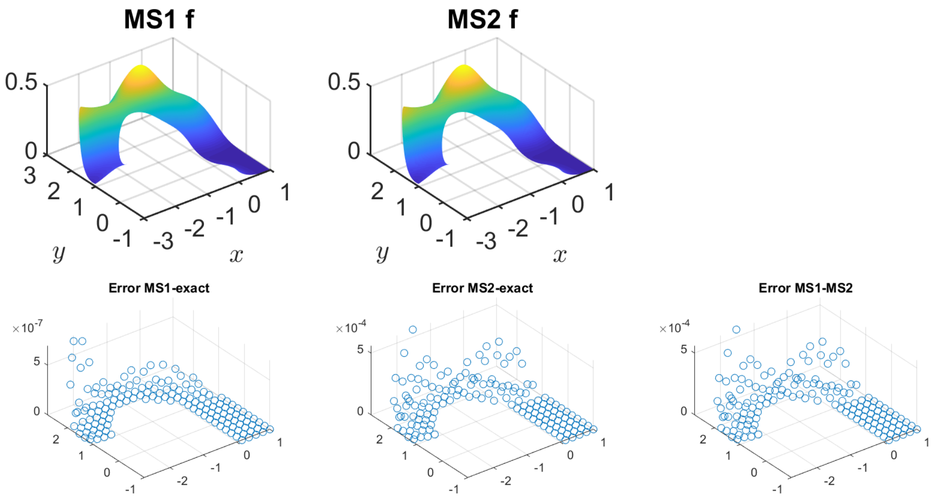


Figure 4: Interpolation Errors

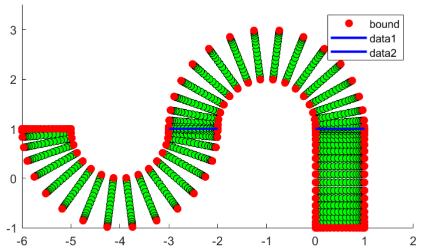


Figure 5: Domain3

Started implementing the optimization aspect. Works for box shaped MS code and Neumann Flow Exact problem.

Tried to split box and still use Neumann Flow Exact. Doesn't work because the error in the adjoint is too large – why? Plotting it, the two adjoints look very similar.

Started a random problem with constant flow on square+wedge, with constant flow. It runs but is slow obviously, so don't know result.

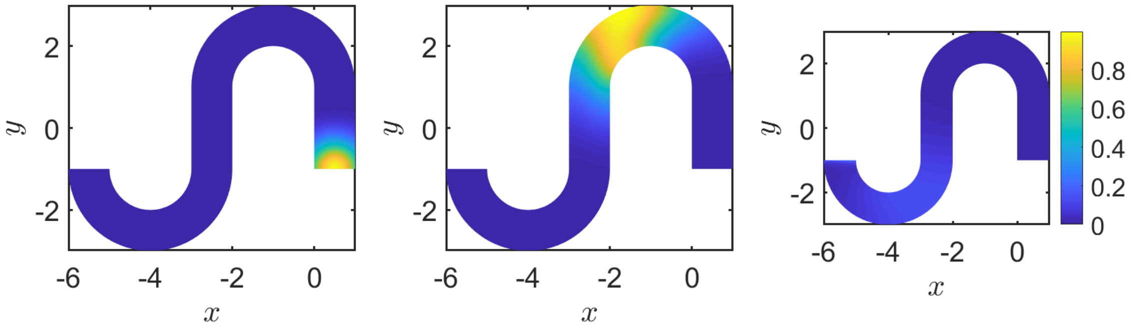


Figure 6: Dynamics 1

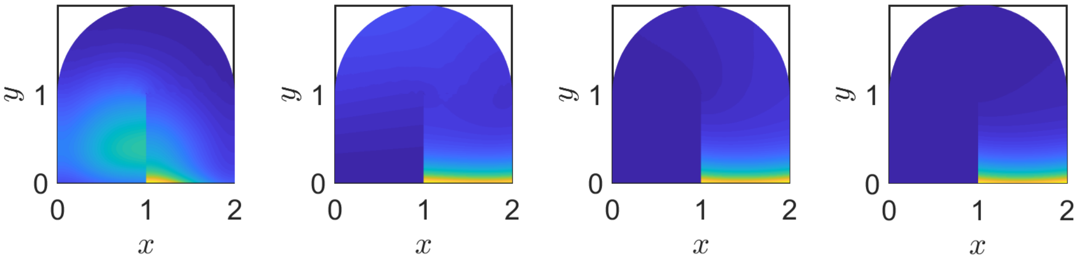


Figure 7: Dynamics 2 $\kappa = 1$, over time

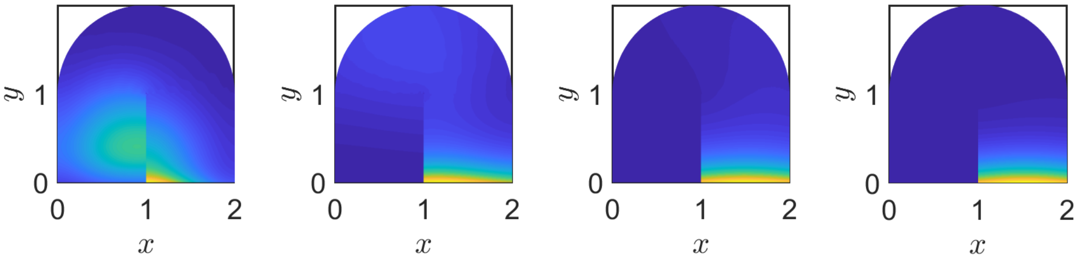


Figure 8: Dynamics 2 $\kappa = -1$, over time