holistic discretisation that ensures continuity between adjacent elements

AJR

Feb 2002 – December 6, 2013

Execute in Reduce with `in_tex "ctyop.tex"$`

Seeks to model the 1D advection-diffusion PDE

$$\frac{\partial u}{\partial t} = -c\frac{\partial u}{\partial x} + \frac{\partial^2 u}{\partial x^2}$$

on a macroscale grid, for 'small' advection speed c. The jth element is $X_{j-1} \le x \le X_j$.

Improve printing.

```
on div; off allfac; on revpri;
factor hh,uu,c,d;
```

Define shift right/left operators `ep` and `em`: use that in terms of centred mean and difference operators, μ and δ, they are $1 \pm \mu\delta + \frac{1}{2}\delta^2$ (National Physical Laboratory 1961, p.65). Also encode the identity that $\mu^2 = 1 + \delta^2/4$. Define the 'spline' operator `ss` $= S := (1 + \delta^2/6)^{-1}$.

```
ep:=1+mu*del+del^2/2;
```

```
em:=1-mu*del+del^2/2;
let { mu^2=>1+del^2/4
    , ss*del^2=>6-6*ss };
```

Write the solution in terms of the microscale variable $\xi := (x - X_{j-1})/H$.

```
depend xi,x;
let df(~a,x)=>df(a,xi)/hh;
```

To find corrections, linear operator `linv` solves DEs of the form $\partial^2\hat{u}/\partial\xi^2 = \text{Res}$ such that $\hat{u} = 0$ at $\xi = 0, 1$.

```
operator linv; linear linv;
let { linv(xi^~~p,xi)=>(xi^(p+2)-xi)/(p+1)/(p+2)
    , linv(1,xi)=>(xi^2-xi)/2 };
```

Write the slow manifold in terms of amplitudes $U_j(t) := u(X_j, t)$. These depend upon time according to $dU_j/dt = g_j$. We let all the j dependence be in the operators.

```
depend uu,t;
let df(uu,t)=>g;
```

The linear solution are equilibria, $g = 0$, of piecewise linear field between U_{j-1} at $\xi = 0$ and U_j at $\xi = 1$.

```
g:=0;
u:=xi*uu+(1-xi)*em*uu;
```

Iterate until the slow manifold model is found to the following specified order of accuracy. Resolving to errors $\mathcal{O}\big(c^3\big)$ in the advection speed c allows us to explore any stabilising effect of our analysis in the presence of otherwise destabilising advection.

```
let { gamma^4=>0, c=>0 };
for it:=1:19 do begin
```

Compute residuals of governing equations.

```
    write pde:= -df(u,t)+df(u,x,x)-c*df(u,x);
    write amp:=sub(xi=1,u)-uu;
    write cty:=sub(xi=0,ep*u)-sub(xi=1,u);
    hux:=hh*df(u,x)$
    write jmp:=-sub(xi=0,ep*hux)+sub(xi=1,hux)
        +(1-gamma)*sub(xi=1,ep*u-2*u+em*u);
```

Correct approximations based upon the residuals. These ad hoc corrections are not optimal, but they do work after enough iterations.

```
    write g:=g+(gd:=-ss*jmp/hh^2);
    u:=u-linv(pde-(xi+(1-xi)*em)*gd,xi)*hh^2;
```

Exit the loop when all residuals are zero to the order specified.

```
    showtime;
    if {pde,amp,cty,jmp}={0,0,0,0} then write it:=it+10000;
end;
```

Get equivalent PDE, but need to improve to be able to analyse to any order.

```
ssd:=1-hh^2*d^2/6+hh^4*d^4/72-hh^6*d^6/2160;
let d^7=>0;
gde:=sub(ss=ssd,g);
```

This appears to simplify the form of the evolution: introducing `gamdel2` $:= \gamma\delta^2$.

```
factor gamdel2;
g:=(g where ss*gamma=>gamma-ss/6*gamdel2);
u:=(u where { ss*gamma=>gamma-ss/6*gamdel2
            , gamma*del^2=>gamdel2})$
```

Fin.

```
end;
```

References

National Physical Laboratory (1961), *Modern Computing Methods*, Vol. 16 of *Notes on Applied Science*, 2nd edn, Her Majesty's Stationery Office, London.