Holistic discretisation of wave-like PDEs

Tony Roberts Meng Cao

December 1, 2011

Try to develop good numerics of wave-like PDEs using a staggered element approach. For 'small' parameter ν, the PDEs for fields $h(x,t)$ and $u(x,t)$ are among

$$\begin{aligned} \frac{\partial h}{\partial t} &= -\frac{\partial u}{\partial x}\,, \\ \frac{\partial u}{\partial t} &= -\frac{\partial h}{\partial x} - \nu u + \nu\frac{\partial^2 u}{\partial x^2} - \nu u\frac{\partial u}{\partial x}\,. \end{aligned}$$

To be solved on elements centred on X_j with coupling condition

$$(1-\tfrac{\gamma}{2})\left[u_j(X_{j+1},t)-u_j(X_{j-1},t)\right] = \tfrac{\gamma}{2}\left[u_{j+2}(X_{j+1},t)-u_{j-2}(X_{j-1},t)\right],$$

and correspondingly for h.

1 Computer algebra constructs the model

Improve printing

```
on div; off allfac; on revpri;
linelength(70)$ factor dd,df;
```

Avoid slow integration with specific operator Introduce the sign function to handle the derivative discontinuities across the centre of each element. Define the integral operator to handle polynomials with sign functions, both indefinite ($\int_0^\xi d\xi$) and definite to $\xi = \mathtt{q} = \pm 1$ ($\int_0^q d\xi$).

```
let df(sign(~x),~y)=>0;
operator intx; linear intx;
let { intx(xi^~~p,xi)=>xi^(p+1)/(p+1)
    , intx(1,xi)=>xi
    , intx(sign(xi)*xi^~~p,xi)=>sign(xi)*xi^(p+1)/(p+1)
    , intx(sign(xi),xi)=>sign(xi)*xi
    , intx(xi^~~p,xi,~q)=>q^(p+1)/(p+1)
    , intx(1,xi,~q)=>q
    , intx(sign(xi)*xi^~~p,xi,~q)=>sign(q)*q^(p+1)/(p+1)
    , intx(sign(xi),xi,~q)=>sign(q)*q
    };
```

Introduce subgrid variable Introduced above is the subgrid variable $\xi = (x - X_j)/D$, $|\xi| < 1$, in which the fields are described.

```
depend xi,x;  let df(xi,x)=>1/dd;
```

Define evolving amplitudes Amplitudes are as $U_j(t) = u_j(X_j, t)$ and $H_j(t) = h_j(X_j, t)$, although I also try the element averages $U_j(t) = \frac{1}{2D}\int_{X_{j-1}}^{X_{j+1}} u_j(x, t)\,dx$ for conservation reasons—no real difference that I can yet see. The difference here is that we take, say, even j to be the u-elements and odd j to be the h-elements. Actually it does not matter which way around, or even if you regard the modelling as being of two disjoint systems (one one way and one the other). The amplitudes depend upon time according to some approximation stored in `gh` and `gu`.

```
operator hh; operator uu;
depend hh,t; depend uu,t;
let { df(hh(~k),t)=>sub(j=k,gh)
    , df(uu(~k),t)=>sub(j=k,gu)
    };
conserve:=0; % non-zero for conservation
```

But solvability condition is coupled Now the evolution equations are coupled together. By some symmetry we decouple the equations using this operator `ginv`. However, I expect that some problems will not decouple (look for non-cancelling pollution by `ginv` operators). In which case we have to accept that the DEs for the amplitudes are *implicit* DEs using the following operator.

Let's define $\mathcal{G} = E + E^{-1}$ so that $\mathcal{G}F_j = F_{j+1} + F_{j-1}$. Take $\mathcal{G}^{-1}$ of this equation to deduce $\mathcal{G}^{-1}F_{j\pm1} = F_j - \mathcal{G}^{-1}F_{j\mp1}$, and change subscripts, $j \mapsto k \mp 1$, to deduce $\mathcal{G}^{-1}F_k = F_{k\mp1} - \mathcal{G}^{-1}F_{k\mp2}$. That is, we change an inverse of $\mathcal{G}$ to one with subscript closer to $k = j$, or otherwise if we desire. Have here coded some quadratic transformations so we can resolve quadratic terms in the model, but I guess we also might want cubic.

```
operator ginv; linear ginv;
let { df(ginv(~a,t),t)=>ginv(df(a,t),t)
    , ginv(~a(j+~k),t)=>a(j+k-1)-ginv(a(j+k-2),t) when k>1
    , ginv(~a(j+~k),t)=>a(j+k+1)-ginv(a(j+k+2),t) when k<0
    , ginv(~a(j+~k)^2,t)=>a(j+k-1)^2-ginv(a(j+k-2)^2,t) when k>1
    , ginv(~a(j+~k)^2,t)=>a(j+k+1)^2-ginv(a(j+k+2)^2,t) when k<0
    , ginv(~a(j+~~k)*~b(j+~~l),t)=> a(j+k-1)*b(j+l-1)
      -ginv(a(j+k-2)*b(j+l-2),t) when k+l>2
    , ginv(~a(j+~~k)*~b(j+~~l),t)=> a(j+k+1)*b(j+l+1)
      -ginv(a(j+k+2)*b(j+l+2),t) when k+l<-1
    };
```

Start with linear approximation Linear approximation is the usual piecewise constant fields in each element.

```
hj:=hh(j); uj:=uu(j);
gh:=gu:=0;
let gam^6=>0;
gamma:=gam;
```

Iterate to a slow manifold Iterate to seek a solution, terminating only when residuals are zero to specified order.

```
for it:=1:19 do begin
```

Use h-equation to update u-field Compute residuals for the h-equations that give the h_j-evolution and the u_j-field, but shifted so that the ξ variables are the same for the terms.

```
hr:=sub({xi=xi-1,j=j+1},(hj where sign(xi)=>-1))$
reshr:=(df(hr,t)+df(uj,x) where sign(xi)=>+1);
write lengthreshr:=length(reshr);
hl:=sub({xi=xi+1,j=j-1},(hj  where sign(xi)=>+1))$
```

```
reshl:=(df(hl,t)+df(uj,x) where sign(xi)=>-1);
write lengthreshl:=length(reshl);
resuc:=(1-gamma/2)*(sub(xi=1,uj)-sub(xi=-1,uj))
         -gamma/2*(+sub({xi=-1,j=j+2},uj)-sub({xi=+1,j=j-2},uj));
write lengthresuc:=length(resuc);
ghd:=ginv(resuc/dd-intx(reshr,xi,+1)
                  +intx(reshl,xi,-1),t);
gh:=gh+ghd;
uj:=uj-dd*intx( (1+sign(xi))/2*(reshr+sub(j=j+1,ghd))
               +(1-sign(xi))/2*(reshl+sub(j=j-1,ghd)),xi);
if conserve then uj:=uj-(intx(uj,xi,1)-intx(uj,xi,-1))/2+uu(j);
```

Use u-equation to update h-field Second, do the converse case exactly the same but opposite, by symmetry. However, now try small bed drag and/or dissipation and/or nonlinear advection in the u-equation: find that there is no change in the subgrid fields for linear dissipation, just apparently reasonable changes in the evolution.

```
let nu=>0; % factor nu;
ur:=sub({xi=xi-1,j=j+1},(uj where sign(xi)=>-1))$
resur:=(df(ur,t)+df(hj,x)+nu*df(ur,x)*ur where sign(xi)=>+1);
write lengthresur:=length(resur);
ul:=sub({xi=xi+1,j=j-1},(uj where sign(xi)=>+1))$
resul:=(df(ul,t)+df(hj,x)+nu*df(ul,x)*ul where sign(xi)=>-1);
write lengthresul:=length(resul);
reshc:=(1-gamma/2)*(sub(xi=1,hj)-sub(xi=-1,hj))
         -gamma/2*(+sub({xi=-1,j=j+2},hj)-sub({xi=+1,j=j-2},hj));
write lengthreshc:=length(reshc);
gud:=ginv(reshc/dd-intx(resur,xi,+1)
                  +intx(resul,xi,-1),t);
gu:=gu+gud;
hj:=hj-dd*intx( (1+sign(xi))/2*(resur+sub(j=j+1,gud))
               +(1-sign(xi))/2*(resul+sub(j=j-1,gud)),xi);
if conserve then hj:=hj-(intx(hj,xi,1)-intx(hj,xi,-1))/2+hh(j);
```

Terminate the loop Exit the loop if all residuals are zero.

```
  if {reshr,reshl,resuc,resur,resul,reshc}={0,0,0,0,0,0}
  then write it:=it+100000;
```

```
  showtime;
end;
```

Equivalent PDEs Finish by finding the equivalent PDE for the discretisation. Since $\mathcal{G} = E + E^{-1} = e^{dd\partial} + e^{-dd\partial} = 2\cosh(dd\partial)$ so $\mathcal{G}^{-1} = \frac{1}{2}\operatorname{sech}(dd\partial)$.

```
let dd^8=>0;
depend uu,x; depend hh,x;
rules:={uu(j)=>uu, uu(j+~p)=>uu+(for n:=1:8 sum
               df(uu,x,n)*(dd*p)^n/factorial(n))
       ,hh(j)=>hh, hh(j+~p)=>hh+(for n:=1:8 sum
               df(hh,x,n)*(dd*p)^n/factorial(n))
       ,ginv(~a,t)=>1/2*(a-1/2*dd^2*df(a,x,2)
       +5/24*dd^4*df(a,x,4) -61/720*dd^6*df(a,x,6)
       +277/8064*dd^8*df(a,x,8) )
       }$
ghde:=(gh where rules);
gude:=(gu where rules);
```

Draw graph of subgrid field The first plot call is a dummy that appears needed on my system for some unknown reason.

```
plot(sin(xi),terminal=aqua);
u0:=(uj where {nu=>0,gam=>1,uu(j)=>1,uu(j+~k)=>0 when k neq 0});
h0:=(hj where {nu=>0,gam=>1,hh(j)=>1,hh(j+~k)=>0 when k neq 0});
plot({u0,h0},xi=(-2 .. 2),terminal=aqua);
```

Finish

```
end;
```

2 Sample output

```
1: in_tex "waveRed.tex"$

hj := hh(j)

uj := uu(j)
```

```

gh := gu := 0

gamma := gam

Time: 10 ms

Time: 10 ms

Time: 20 ms

Time: 30 ms

Time: 30 ms

it := 100006

Time: 20 ms

                                       2      1         1     3
ghde :=  - df(uu,x)*gam + df(uu,x,3)*dd *( - ---*gam + ---*gam )
                                              6         6

                        4       1          1      3    3      5
         + df(uu,x,5)*dd *( - -----*gam + ----*gam  - ----*gam )
                               120         12          40

                        6       1           13      3    1      5
         + df(uu,x,7)*dd *( - ------*gam + -----*gam  - ----*gam )
                               5040         720          16

```