Computer algebra derives discretisations via self-adjoint multiscale modelling

A. J. Roberts[*]

July 17, 2008

Abstract

The computer algebra routines[1] documented here empower you to reproduce and check details described in a partner article [14]. We consider a region of a spatial domain far from any boundaries, and derive a discrete model for the dynamics on the slow manifold on a coarse scale lattice. The approach automatically accounts for fine-grid scale interactions within and between coarse-grid elements to form a systematic approximation of the accurate closure on the coarse grid. You may straightforwardly adapt these routines to model many similar multiscale dynamical systems.

Contents

---

[*]Department of Mathematics & Computing, University of Southern Queensland, Toowoomba, Queensland 4352, Australia. mailto:aroberts@usq.edu.au

[1]The computer algebra routines are written in the package REDUCE. At the time of writing, information about REDUCE was available via http://www.reduce-algebra.com/. Free demonstration versions of REDUCE were available and will execute these routines.

1   Introduction

Holistic discretisation uses centre manifold theory to rationally model on a coarse scale the multiscale dynamics that arise in spatio-temporal continuum dynamics [12, 10, 16, e.g.] or that arise on a multigrid hierarchy

of lattices [15]. However, previous approaches do not necessarily preserve, at each stage of approximation, the self-adjoint symmetries of the underlying microscale system. Here we develop computer algebra to implement interelement coupling conditions that do preserve self-adjointness. These new coupling conditions then empower us to model and connect self-adjoint models across wide ranges of space-time scales.

For the first of two examples, building upon the routine for modelling linear continuum diffusion in Section 2, the computer algebra of Section 3 recommends the following self-adjoint discrete model of the example nonlinear continuum diffusion

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left[u\frac{\partial u}{\partial x}\right] \quad \mapsto \quad \frac{dU_j}{dt} = \frac{1}{2h^2}\left(U_{j+1}^2 - 2U_j^2 + U_{j+1}^2\right). \tag{1}$$

As a second example, building upon the routine for modelling linear discrete diffusion in Section 4, the homogenisation of Section 6 commends the following self-adjoint coarser grid lattice model of the fine grid lattice dynamics

$$\frac{du_i}{dt} = \kappa_{i-1/2}u_{i-1} - (\kappa_{i-1/2} + \kappa_{i+1/2})u_i + \kappa_{i+1/2}u_{i+1}$$

$$\mapsto \quad \frac{dU_j}{dt} = \frac{1}{4}\left[\mathcal{K}_{j-1}U_{j-2} - (\mathcal{K}_{j-1} + \mathcal{K}_{j+1})U_j + \mathcal{K}_{j+1}U_{j+2}\right], \tag{2}$$

where the coarser scale diffusivity $\mathcal{K}_j \approx \frac{1}{4}(\kappa_{2j-2} + \kappa_{2j-1} + \kappa_{2j+1} + \kappa_{2j+2})$ is an average over the local fine grid diffusivities. Such transformations from fine lattice dynamics to coarser lattice dynamics may be iterated to cover step-by-step the wide range of space-time scales on a multigrid hierarchy.

The above transformations only show nearest neighbour interactions. The use of centre manifold theory empowers us to systematically incorporate interactions in the dynamics over wider stencils in space. As in earlier work, we use the parameter γ to control such interactions.

Although the specific computer algebra routines coded here are for specific microscale dynamics, straightforward modifications allow one to model a wide range of microscale dynamics. For two other examples: Section 5 builds upon the modelling of linear discrete diffusion in Section 4 to model nonlinear reaction-diffusion lattice dynamics; similarly Section 7 uses the same basis to develop the envelope equation for the modulation of nonlinear zigzag modes on a lattice.

Throughout we only address deterministic dynamics in one spatial dimension. Analogous approaches for stochastic dynamics [13, e.g.] or higher dimensions in space [9, e.g.] is left for further research.

2 Self-adjoint holistic modelling of diffusion

The simplest macroscale modelling in continuum mechanics is the discretisation of linear diffusion as governed by the PDE

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}\,. \tag{3}$$

As described in the companion article [14, §2], embed the spatio-temporal dynamics into dynamics on overlapping spatial elements: let $u_j(x,t)$ denote the field in the jth element.

The following computer algebra readily constructs approximations to the slow manifold of the evolution of the grid values. Let the coarse scale grid be uniform, $\Delta X_j = h$, and define the grid values $U_j(t) = u_j(X_j, t)$. Computer algebra finds the subgrid, intraelement field is

$$u_j = \left[1 + \gamma(\xi\mu\delta + \tfrac{1}{2}|\xi|\delta^2) + \gamma^2(\tfrac{1}{2}\xi^2\delta^2 - \tfrac{1}{2}|\xi|\delta^2)\right]U_j + \mathcal{O}(\gamma^3)\,. \tag{4}$$

in terms of the subgrid variable $\xi = (x - X_j)/h$, and centred mean μ and difference δ operators. The corresponding evolution on the slow manifold is the discretisation

$$\dot{U}_j = \tfrac{1}{h^2}\gamma^2\delta^2 U_j - \tfrac{6-5\gamma}{12h^2}\gamma^3\delta^4 U_j + \tfrac{45-75\gamma+32\gamma^2}{180h^2}\gamma^4\delta^6 U_j + \mathcal{O}(\gamma^7, \delta^8)\,. \tag{5}$$

Evaluated at full physical coupling $\gamma = 1$, this model recovers the classic centred finite difference formula for the discretisation. Computing to higher orders in coupling γ gives more and more terms in the classic formula.

Parameters   Compute the asymptotic series to this order of asymptotic error in the small coupling parameter γ.

$$\triangleright\triangleright \text{ sahmd } \triangleleft\triangleleft$$

```
let gam^7=>0;
```

Overview of the algorithm Initialise, then iterate to improve the approximation of the slow manifold until the residuals are zero to the above specified order of error in coupling parameter γ.

```
                    ▷▷ sahmd ◁◁+
% see caddsamm.pdf for documentation
◁◁ sahmd preamble ▷▷
◁◁ sahmd initialisation ▷▷
repeat begin
    ◁◁ sahmd pde ▷▷
    ◁◁ sahmd coupling ▷▷
    ◁◁ sahmd update ▷▷
    showtime;
end until {res,rcc,lcc,ctx}={0,0,0,0};
◁◁ sahmd write ▷▷
◁◁ sahmd transform ▷▷
end;
```

2.1 Preamble

Trivially improve printing.

```
                ▷▷ sahmd preamble ◁◁
on div; off allfac; on revpri;
factor h,gam;
linelength 70;
```

Use a homotopy that is convex combination from isolated to fully coupled elements.

```
                ▷▷ sahmd preamble ◁◁+
gamm:=gam; gamd:=1-gam;
```

Parametrise solutions by evolving amplitudes $U_j(t)$ (or 'order parameters'). Generalising the particular example (5), their evolution is $dU_j/dt = \dot{U}_j = g_j(\mathbf{U}, \gamma)$.

<div align="center">▷▷ sahmd preamble ◁◁+</div>

```
operator uu; depend uu,t;
let df(uu(~k),t)=>sub(j=k,gj);
```

Make the subgrid, intraelement structure a function of $\xi = (x - x_j)/h$, noting that the `sign` function copes with two different parts of the subgrid structure, one for either side of the mid-point of the element.

<div align="center">▷▷ sahmd preamble ◁◁+</div>

```
depend xi,x;
let df(xi,x)=>1/h;
let { df(sign(xi),x)=>0, sign(xi)^2=>1 };
```

2.2 Initialise

The initial approximation to the slow manifold is an arbitrary equilibrium point, $g_j = 0$, on the slow subspace of piecewise constant solutions u_j.

<div align="center">▷▷ sahmd initialisation ◁◁</div>

```
uj:=uu(j)$
gj:=0$
```

2.3 Compute residuals

The partial differential equation to solve is the simple diffusion equation (3). The 'flux' derivative is later used in one of the coupling conditions.

<div align="center">▷▷ sahmd pde ◁◁</div>

```
flux:=-df(uj,x);
```

```
res:=df(uj,t)+df(flux,x);
```

Couple the diffusion PDE (3) in each overlapping element by a condition which asserts that on the one hand when $\gamma = 0$, each element is isolated, but on the other hand when $\gamma = 1$, the field in the jth element when extrapolated to the surrounding elements, 'hits' the surrounding grid values precisely:

$$u_j(X_{j\pm1}) = \gamma' u_j(X_j) + \gamma u_{j\pm1}(X_{j\pm1}). \tag{6}$$

The residuals of these two conditions are the following.

$$\triangleright\triangleright \text{ sahmd coupling } \triangleleft\triangleleft$$

```
rcc:=sub(xi=+1,uj)-gamd*sub(xi=0,uj)
                -gamm*sub({j=j+1,xi=0},uj);
lcc:=sub(xi=-1,uj)-gamd*sub(xi=0,uj)
                -gamm*sub({j=j-1,xi=0},uj);
```

To make the above part of a self-adjoint system of coupling conditions, we then have to couple the gradients according to

$$f_j(X_j^+) - f_j(X_j^-) + \gamma\big[f_{j+1}(X_j) - f_{j-1}(X_j)\big] - \gamma'\big[f_j(X_{j+1}) - f_j(X_{j-1})\big] = 0\,, \tag{7}$$

where here the flux $f_j = u_{jx}$, the derivative. In the following code, note the use of `sign` to get evaluation at ±0.

$$\triangleright\triangleright \text{ sahmd coupling } \triangleleft\triangleleft+$$

```
ctx:=sub(xi=0,(flux where sign(xi)=>+1)
              -(flux where sign(xi)=>-1))
      -gamd*(sub(xi=1,flux)-sub(xi=-1,flux))
      +gamm*(sub({j=j+1,xi=-1},flux)
            -sub({j=j-1,xi=+1},flux));
```

2.4   Update slow manifold

Update the evolution on the slow manifold using a solvability condition. This particular solvability condition may not be optimal, but it does work.

▷▷ sahmd update ◁◁

```
gj:=gj+(gd:=-intx(res,xi)/4-(rcc+lcc)/h^2-ctx/2/h);
```

For execution speed, the above integration over an element is done via the `intx` linear operator.

▷▷ sahmd preamble ◁◁+

```
operator intx; linear intx;
let { intx(xi^~p,xi)=>(1+(-1)^p)/(p+1)
    , intx(xi,xi)=>0, intx(1,xi)=>2
    , intx(sign(xi)*xi^~p,xi)=>(1-(-1)^p)/(p+1)
    , intx(sign(xi)*xi,xi)=>1, intx(sign(xi),xi)=>0
    };
```

Given the above update to the evolution, we next update the subgrid field via a linear operator `solv`, here defined for the preamble, and the coupling conditions.

▷▷ sahmd update ◁◁+

```
uj:=uj+h^2*solv(gd+res,xi)
    +ctx*sign(xi)*xi*h/4+(lcc-rcc)*xi/2;
```

▷▷ sahmd preamble ◁◁+

```
operator solv; linear solv;
let { solv(xi^~p,xi)=>xi^(p+2)/(p+1)/(p+2)
    , solv(xi,xi)=>xi^3/2/3, solv(1,xi)=>xi^2/2
    , solv(sign(xi)*~a,xi)=>sign(xi)*solv(a,xi)
    , solv(sign(xi),xi)=>sign(xi)*solv(1,xi)
    };
```

2.5   Write some of the slow manifold

Check that the continuity at the middle of each element, $u_j(X_j^-) = u_j(X_j^+)$, and that the amplitude is the grid value at the middle of each element,

$u_j(X_j) = U_j$.

$$\triangleright\triangleright \texttt{ sahmd write } \triangleleft\triangleleft$$

```
res_cty:=sub(xi=0,(uj where sign(xi)=>+1)
                   -(uj where sign(xi)=>-1));
res_amp:=sub(xi=0,uj)-uu(j);
```

Print out the evolution on the slow manifold.

$$\triangleright\triangleright \texttt{ sahmd write } \triangleleft\triangleleft\texttt{+}$$

```
gj:=gj;
```

2.6   Transform evolution

Rewrite the model evolution in terms of centred differences on the grid to generate more appealing versions of the evolution such as (5). These rules use the identities that the spatial shifts $\varepsilon_\pm = (\mu \pm \delta/2)^2$ and that $\mu^2 = 1 + \delta^2/4$.

$$\triangleright\triangleright \texttt{ sahmd transform } \triangleleft\triangleleft$$

```
gg:=(gj where {uu(j)=>1,mu^2=>1+del^2/4
     ,uu(j+~p)=>(mu-del/2)^(-2*p) when p<0
     ,uu(j+~p)=>(mu+del/2)^(+2*p) when p>0});
dels:=coeff(gg,del)$
for p:=0:deg((1+gam)^99,gam) do
    write "h^(-2)*del^",p,factorize(h^2*part(dels,p+1));
```

Also write as an equivalent partial differential equation. The operator `du(n)` denotes the nth derivative of the grid value U_j.

$$\triangleright\triangleright \texttt{ sahmd transform } \triangleleft\triangleleft\texttt{+}$$

```
operator du; factor du;
ord:=8$
let h^~p=>0 when p>ord;
write gepde:=(gj where uu(~k)=>(du(0)+for n:=1:ord+3
```

```
            sum h^n*(k-j)^n*du(n)/factorial(n)));
  for p:=0:ord do write "h^",p,factorize(coeffn(gepde,h,p));
  write gepde_hexpansion:=coeff(sub(gam=1,gepde),h);
```

3 Self-adjoint modelling of nonlinear diffusion

As a specific example application to nonlinear dynamics [8, 19, e.g.][14, §4], consider nonlinear diffusion governed by the PDE

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(u\frac{\partial u}{\partial x}\right).\tag{8}$$

To generate the slow manifold model of this nonlinear diffusion we reuse much of the previous code for linear diffusion: simply change the residual of the PDE and the expression for the updates.

The computer algebra readily constructs the subgrid field of the slow manifold:

$$\begin{aligned}
u_j &= \left[1 + \gamma\xi\mu\delta + \tfrac{1}{2}\gamma(1-\gamma)|\xi|\delta^2) + \gamma^2\tfrac{1}{2}\xi^2\delta^2\right]U_j \\
&\quad + \gamma^2\big[\xi(1-|\xi|)(\mu\delta U_j + \tfrac{1}{2U_j}\mu\delta U_j^2) + (|\xi|-\xi^2)(\tfrac{1}{2}\delta^2 U_j - \tfrac{1}{4U_j}\delta^2 U_j^2)\big] \\
&\quad + \mathcal{O}(\gamma^3).
\end{aligned}\tag{9}$$

The evolution on this slow manifold gives the discrete model

$$\dot{U}_j = \tfrac{1}{2h^2}\gamma^2\delta^2 U_j^2 - \tfrac{6-5\gamma}{24h^2}\gamma^3\delta^4 U_j^2 + \tfrac{45}{360h^2}\gamma^4\delta^6 U_j^2 + \mathcal{O}(\gamma^5).\tag{10}$$

Parameters Compute the asymptotic series to this order of error in the small coupling parameter γ.

<div align="center">▷▷ samnd ◁◁</div>

```
  let gam^5=>0;
```

Overview of the algorithm   Initialise, then iterate to improve the approximation of the slow manifold until the residual is zero to the required error.

```
                        ▷▷ samnd ◁◁+
% see caddsamm.pdf for documentation
◁◁ sahmd preamble ▷▷
◁◁ sahmd initialisation ▷▷
repeat begin
    ◁◁ samnd pde ▷▷
    ◁◁ sahmd coupling ▷▷
    ◁◁ samnd update ▷▷
    showtime;
end until {res,rcc,lcc,ctx}={0,0,0,0};
◁◁ sahmd write ▷▷
◁◁ sahmd transform ▷▷
end;
```

3.1   Compute residuals of nonlinear diffusion

The partial differential equation to solve is the nonlinear diffusion equation (8).

```
                      ▷▷ samnd pde ◁◁
flux:=-uj*df(uj,x);
res:=df(uj,t)+df(flux,x);
```

3.2   Update the nonlinear slow manifold

Update the evolution on the slow manifold using a solvability condition. These need modification by factors of `uu(j)` to cater for the nonlinear diffusion.

▷▷ samnd update ◁◁
```
gj:=gj+(gd:=-intx(res,xi)/4-uu(j)*(rcc+lcc)/h^2-ctx/2/h);
```

Given the update to the evolution, we next update the subgrid field via a linear operator `solv` and the coupling conditions.

▷▷ samnd update ◁◁+
```
uj:=uj+h^2*solv(gd+res,xi)/uu(j)
    +ctx*sign(xi)*xi*h/4/uu(j)+(lcc-rcc)*xi/2;
```

4 Derive coarse grid model of discrete diffusion

Dynamics on a discrete lattice is often used in spatio-temporal research [4, 5, 11, 18, 7, e.g.]. The remaining sections explore how to transform the discrete dynamics of variables $u_i(t)$ on a fine grid of spacing h, into discrete dynamics of variables $U_j(t)$ on a coarser grid of spacing $H = 2h$. The technique is to embed the dynamics into overlapping local elements and construct the slow manifold by controlling the coupling between neighbouring elements [14, §6].

The most basic example of self-adjoint dynamics on a lattice is governed by the discrete diffusion equation nondimensionalised to

$$\dot{u}_i = u_{i-1} - 2u_i + u_{i+1}, \tag{11}$$

on equi-spaced grid points $x_i = ih$. To form overlapping elements [14, §6], identify the variables $v_{j,i}$, $i = 0, \ldots, 7$, of the jth element with the fine grid variables $u_{2j+i-\frac{7}{2}-\mathrm{sign}(i-\frac{7}{2})}$. Then the fine scale evolution rule within each element and coupling between elements is

$$D\dot{\mathbf{v}}_j = L\mathbf{v}_j, \tag{12}$$

where $\boldsymbol{v}_j = (v_{j,0}, \ldots, v_{j,7})$, $D = \mathrm{diag}(0,1,1,0,0,1,1,0)$ and

$$
L = \begin{bmatrix}
-1 & 1 & 0 & -\gamma'-\gamma\bar{\varepsilon}_- & \gamma'+\gamma\bar{\varepsilon}_- & 0 & 0 & 0 \\
1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 \\
-\gamma'-\gamma\bar{\varepsilon}_+ & 0 & 1 & -1 & 0 & 0 & 0 & \gamma'+\gamma\bar{\varepsilon}_- \\
\gamma'+\gamma\bar{\varepsilon}_+ & 0 & 0 & 0 & -1 & 1 & 0 & -\gamma'-\gamma\bar{\varepsilon}_- \\
0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 \\
0 & 0 & 0 & \gamma'+\gamma\bar{\varepsilon}_+ & -\gamma'-\gamma\bar{\varepsilon}_+ & 0 & 1 & -1
\end{bmatrix}.
$$
(13)

Then each element is isolated when $\gamma = 0$ and $\gamma' = 1$ to form a basis for theoretical support, whereas full coupling occurs with $\gamma = 1$ and $\gamma' = 0$ to model the discrete diffusion (11).

Executing the computer algebra deduces that the fine grid, intraelement structure is

$$
\begin{aligned}
\boldsymbol{v}_j \;=\; \Big[\;\; & (1,1,1,1,1,1,1,1) \\
& + \frac{\gamma}{4}(-5,-3,-1,1,-1,1,3,5)\bar{\mu}\bar{\delta} \\
& + \frac{\gamma^2}{4}(3,1,0,0,0,0,1,3)(\bar{\delta}^2 + \tfrac{1}{4}\bar{\delta}^4) \\
& + \frac{\gamma^2}{16}(13,7,1,-5,5,-1,-7,-13)\bar{\mu}\bar{\delta}^3 \Big] U_j + \mathcal{O}(\gamma^3),
\end{aligned}
$$

in terms of the centred mean $\bar{\mu}$ and difference $\bar{\delta}$ operators on the coarse grid. The corresponding evolution of the coarse grid 'order parameters' U_j, is

$$
\dot{U}_j = \gamma^2\left(\tfrac{1}{4}\bar{\delta}^2 + \tfrac{1}{16}\bar{\delta}^4\right) + \gamma^3\left(-\tfrac{5}{16}\bar{\delta}^4 - \tfrac{5}{64}\bar{\delta}^6\right) + \mathcal{O}(\gamma^4).
$$
(14)

Obtain the lowest accuracy model from the first line, by neglecting terms $\mathcal{O}(\gamma^3)$, and then evaluating at the physical fully coupled case $\gamma = 1$: the model is $\dot{U}_j = \frac{1}{16}(U_{j-2} - 2U_j + U_{j+2})$ which is appropriate although surprisingly only involves every second coarse grid value.

Parameters   Compute the asymptotic series to this order of error in the small interelement coupling parameter γ.

▷▷ cgmdd ◁◁

```
let gam^5=>0;
```

Overview of the algorithm Initialise, then iterate to improve the approximation of the slow manifold until the residual is zero to the required error. In REDUCE, `tp` denotes the transpose of a matrix.

▷▷ cgmdd ◁◁+

```
% see caddsamm.pdf for documentation
◁◁ cgmdd preamble ▷▷
◁◁ cgmdd initialisation ▷▷
zero:=tp mat((0,0,0,0,0,0,0,0))$
repeat begin
  ◁◁ cgmdd residual ▷▷
  ◁◁ cgmdd update ▷▷
  showtime;
end until res=zero;
◁◁ cgmdd write ▷▷
◁◁ cgmdd transform ▷▷
end;
```

4.1 Preamble

Trivially improve printing.

▷▷ cgmdd preamble ◁◁

```
on div; off allfac; on revpri;
linelength 70;
factor gam;
```

Use homotopy that is convex combination from isolated, $gam = 0$, to fully coupled, $gam = 1$.

▷▷ cgmdd preamble ◁◁+

```
gamm:=gam; gamd:=1-gam;
```

Parametrise solutions by evolving amplitudes $U_j(t)$ (or 'order parameters') chosen to be $U_j = \frac{1}{4}(v_{j,2} + v_{j,3} + v_{j,4} + v_{j,5})$. Their evolution is $dU_j/dt = \dot{U}_j = g_j(\mathbf{U}, \gamma)$.

▷▷ cgmdd preamble ◁◁+

```
operator u; depend u,t;
let df(u(~k),t)=>sub(j=k,gj);
```

4.2   Initialise

The initial approximation is an arbitrary equilibrium point, $g_j = 0$, on the slow subspace of piecewise constant solutions $\mathbf{v}_j$.

▷▷ cgmdd initialisation ◁◁

```
gj:=0$
vj:=u(j)*tp mat((1,1,1,1,1,1,1,1))$
```

4.3   Compute residual

The evolution on the fine grid is the isolated diffusion modified by the coupling with neighbouring elements.

▷▷ cgmdd residual ◁◁

```
res:=-tp mat((0,df(vj(2,1),t),df(vj(3,1),t),0
              ,0,df(vj(6,1),t),df(vj(7,1),t),0))
    +ll*vj
    +gamm*tp mat((
        sub(j=j-1,-vj(4,1)+vj(5,1)),
        0,0,
        sub(j=j-1,vj(8,1))-sub(j=j+1,vj(1,1)),
```

```
        sub(j=j+1,vj(1,1))-sub(j=j-1,vj(8,1)),
        0,0,
        sub(j=j+1,+vj(4,1)-vj(5,1))
        ));
```

Where in the above the linear operator of diffusion on an element, decoupled from the other elements is the following.

▷▷ cgmdd preamble ◁◁+

```
ll:=mat((-1,1,0,-gamd,+gamd,0,0,0)
        ,(1,-2,1,0,0,0,0,0)
        ,(0,1,-2,1,0,0,0,0)
        ,(-gamd,0,1,-1,0,0,0,+gamd)
        ,(+gamd,0,0,0,-1,1,0,-gamd)
        ,(0,0,0,0,1,-2,1,0)
        ,(0,0,0,0,0,1,-2,1)
        ,(0,0,0,+gamd,-gamd,0,1,-1))$
```

4.4 Update approximation to the slow manifold

Update the coarse grid evolution from the vector of residuals using the solvability condition.

▷▷ cgmdd update ◁◁

```
gd:=(1/4)*mat((1,1,1,1,1,1,1,1))*res;
gj:=gj+gd(1,1);
```

Update the fine grid, intraelement structure from the vector of residuals using the generalised inverse `gil`.

▷▷ cgmdd update ◁◁+

```
vj:=vj+gil*(-res+tp mat((0,1,1,0,0,1,1,0))*gd);
```

Use the following generalised inverse of `ll` that ensures the sum of the centre four grid values are zero—this ensures the amplitude of the coarse grid model is the grid value at the centre of each element.

▷▷ cgmdd preamble ◁◁+

```
gil:=mat((-17,-11,-5,-3,7,1,-9,-19)
        ,(-11,-17,-7,-1,5,3,-3,-9)
        ,(-5,-7,-9,1,3,5,3,1)
        ,(-3,-1,1,-1,-3,3,5,7)
        ,(7,5,3,-3,-1,1,-1,-3)
        ,(1,3,5,3,1,-9,-7,-5)
        ,(-9,-3,3,5,-1,-7,-17,-11)
        ,(-19,-9,1,7,-3,-5,-11,-17))/16$
```

4.5   Write some of the slow manifold

Just print the evolution, after checking the amplitude equation.

▷▷ cgmdd write ◁◁

```
res_ampl:=(for i:=3:6 sum vj(i,1))-4*u(j);
write gj:=gj;
```

4.6   Transform the coarse grid evolution

To appreciate different aspects of the coarse grid evolution on the slow manifold, we rewrite the dynamics in various ways. First write in terms of coarse grid centred operators. This is the natural expression of the dynamics.

▷▷ cgmdd transform ◁◁

```
write
gjcors:=(gj where {u(j+~p)=>(mu+del/2)^(+2*p) when p>0
                  ,u(j+~p)=>(mu-del/2)^(-2*p) when p<0
                  ,u(j)=>1,mu^2=>1+del^2/4});
```

Write in terms of the next coarser grid centred operators. This is not natural, but may be of interest although I have not reported any of these expressions.

<div align="center">▷▷ cgmdd transform ◁◁+</div>

```
write
gjmega:=(gj where {u(j+~p)=>(mu+del/2)^(+p) when p>0
                  ,u(j+~p)=>(mu-del/2)^(-p) when p<0
                  ,u(j)=>1,mu^2=>1+del^2/4});
```

Write in terms of the original fine grid centred operators. This is the *equivalent fine grid dynamics* and confirms that the analysis indeed accurately reproduces the coarse scale dynamics that take place on the original fine grid.

<div align="center">▷▷ cgmdd transform ◁◁+</div>

```
write
gjfine:=(gj where {u(j+~p)=>(mu+del/2)^(+4*p) when p>0
                  ,u(j+~p)=>(mu-del/2)^(-4*p) when p<0
                  ,u(j)=>1,mu^2=>1+del^2/4});
```

The above forms are organised largely in terms of the coupling parameter γ. We also write results organised in terms of the centred difference operator δ.

<div align="center">▷▷ cgmdd transform ◁◁+</div>

```
dels:=coeff(gjfine,del)$
for p:=0:deg((1+gam)^99,gam) do
    write "del^",p,factorize(part(dels,p+1));
```

5   Coarse grid modelling of nonlinear reaction diffusion

This section modifies the algorithm of the previous Section 4 to model the nonlinear reaction diffusion [20, e.g.][14, §7.1] on a lattice of

$$\dot{u}_i = \delta^2 u_i + \alpha(u_i - u_i^2). \tag{15}$$

We reuse the previous section's code except for truncating in the nonlinear parameter α and including the nonlinear reaction in the residual.

Parameters   Compute the asymptotic series to this order of truncation in the small coupling parameter γ and the small reaction strength α.

▷▷ cgmnrd ◁◁

```
let {gam^3=>0,alfa^2=>0};
```

Improve printed appearance.

▷▷ cgmnrd ◁◁+

```
factor alfa;
```

Overview of the algorithm   Initialise, then iterate to improve the approximation of the slow manifold until the residual is zero to the required error. Do exactly the same algorithm except include some nonlinear terms in the residual of the differential equations.

▷▷ cgmnrd ◁◁+

```
% see caddsamm.pdf for documentation
◁◁ cgmdd preamble ▷▷
◁◁ cgmdd initialisation ▷▷
zero:=tp mat((0,0,0,0,0,0,0,0))$
repeat begin
  ◁◁ cgmdd residual ▷▷
  ◁◁ cgmnrd nonlinear ▷▷
```

```
     ◁◁ cgmdd update ▷▷
     showtime;
   end until res=zero;
   ◁◁ cgmdd write ▷▷
   end;
```

5.1   Nonlinear terms

Include in the residual the nonlinear reaction $\alpha(u_i - u_i^2)$ in the interior grid points of each element.

<div align="center">▷▷ cgmnrd nonlinear ◁◁</div>

```
res:=res+alfa*tp mat((
    0,vj(2,1)-vj(2,1)^2,vj(3,1)-vj(3,1)^2,0,
    0,vj(6,1)-vj(6,1)^2,vj(7,1)-vj(7,1)^2,0));
```

6   Homogenisation of diffusivity variations

A canonical problem in spatially inhomogeneous dynamics is the effective large scale diffusion through a domain with microscopic variations in diffusion coefficient [3, 17, 1, e.g.]. This section adapts the algorithm of Section 4 to transform diffusion on a fine grid, with fine grid variations in coefficient, into an 'homogenised' diffusion on a coarser grid [14, §7.2].

Here consider

$$\dot{u}_i = \delta\big[f_i \delta u_i\big], \tag{16}$$

where the diffusion coefficient governing flux between fine grid points $x_{i-\frac{1}{2}}$ and $x_{i+\frac{1}{2}}$ is $f_i = 1 + \epsilon\kappa_i$. For simplicity, we model the dynamics as a power series in ϵ, the strength of the inhomogeneity.

The computer algebra finds the evolution on the slow manifold is, for example,

$$\dot{U}_j = \gamma^2 \tfrac{1}{16}\big\{\mathcal{K}_{j-1}U_{j-2} - (\mathcal{K}_{j-1} + \mathcal{K}_{j+1})U_j + \mathcal{K}_{j+1}U_{j+2}\big\} + \mathcal{O}(\gamma^3), \tag{17}$$

where the coarse grid effective diffusivity coefficients

$$\mathcal{K}_j = 1 + \epsilon \tfrac{1}{4}\big[\kappa_{2j-2} + \kappa_{2j-1} + \kappa_{2j+1} + \kappa_{2j+2}\big] + \mathcal{O}(\epsilon^2).$$

Beautifully, the effective diffusivity on the coarse grid is *local*: the diffusivity $\mathcal{K}_{j\pm1}$ governing the flux between coarse grid point X_j and $X_{j\pm2}$ depends only upon the fine scale diffusivities between X_j and $X_{j\pm2}$, namely between κ_{2j} and $\kappa_{2j\pm4}$.

Parameters Compute the asymptotic series to this order of truncation in the small coupling parameter γ and the small variations on the diffusivity ϵ. Use small variations in diffusivity as it is then simpler to derive and simpler to see changes to the case of uniform diffusion, rather than look at complicated exact expressions.

<div align="center">▷▷ cghdv ◁◁</div>

```
let {gam^3=>0,eps^3=>0};
```

Improve printed appearance.

<div align="center">▷▷ cghdv ◁◁+</div>

```
factor eps;
```

Set the following flag to one to explore the specific case of zigzag diffusivity. For general diffusivity variations, set it to zero.

<div align="center">▷▷ cghdv ◁◁+</div>

```
zigzag:=0;
```

Overview of the algorithm Initialise, then iterate to improve the approximation of the slow manifold until the residual is zero to the required error. Use much of the basic algorithm of Section 4, the main change is simply to add the variations to the computation of the residuals.

```
                    ▷▷ cghdv ◁◁+
% see caddsamm.pdf for documentation
◁◁ cgmdd preamble ▷▷
◁◁ cghdv preamble ▷▷
◁◁ cgmdd initialisation ▷▷
zero:=tp mat((0,0,0,0,0,0,0,0))$
repeat begin
  ◁◁ cgmdd residual ▷▷
  ◁◁ cghdv variations ▷▷
  ◁◁ cgmdd update ▷▷
  showtime;
end until res=zero;
◁◁ cgmdd write ▷▷
if zigzag then begin
◁◁ cgmdd transform ▷▷
end else write cgdiffusivity:=coeffn(gj,u(j+2),1);
end;
```

In the last few lines above, invoke the transformation of the model in the simpler case of zigzag diffusivity.

6.1   Variation in diffusivity

Include in the residual a varying diffusivity: firstly via a matrix for the intraelement modifications; and secondly via the coupling terms with neighbouring elements.

```
                  ▷▷ cghdv variations ◁◁
res:=res+eps*kap*vj +eps*gamm*tp mat((
    sub(j=j-1,-kp(2*j)*vj(4,1)+kp(2*j)*vj(5,1)),
    0,0,
    kp(2*j)*sub(j=j-1,vj(8,1))-kp(2*j)*sub(j=j+1,vj(1,1)),
    kp(2*j)*sub(j=j+1,vj(1,1))-kp(2*j)*sub(j=j-1,vj(8,1)),
    0,0,
    sub(j=j+1,+kp(2*j)*vj(4,1)-kp(2*j)*vj(5,1))
```

```
        ));
```

The operator `kp` $= \kappa_i$ denotes the variations in diffusivity.

$$\triangleright\triangleright \texttt{ cghdv preamble } \triangleleft\triangleleft$$

```
operator kp;
```

Activate the following to obtain the coarse grid model for a zigzag microscale. Then perhaps work to higher order.

$$\triangleright\triangleright \texttt{ cghdv preamble } \triangleleft\triangleleft\texttt{+}$$

```
if zigzag then let kp(~p)=>(-1)^(p-2*j);
if zigzag then let {gam^5=>0,eps^5=>0};
```

The matrix of diffusivity variations within each element are stored in `kap`.

$$\triangleright\triangleright \texttt{ cghdv preamble } \triangleleft\triangleleft\texttt{+}$$

```
kap:=mat(
 (-kp(2*j-2),+kp(2*j-2),0,-gamd*kp(2*j),+gamd*kp(2*j),0,0,0)
,(+kp(2*j-2),-kp(2*j-2)-kp(2*j-1),+kp(2*j-1),0,0,0,0,0)
,(0,+kp(2*j-1),-kp(2*j-1)-kp(2*j),+kp(2*j),0,0,0,0)
,(-gamd*kp(2*j),0,+kp(2*j),-kp(2*j),0,0,0,+gamd*kp(2*j))
,(+gamd*kp(2*j),0,0,0,-kp(2*j),+kp(2*j),0,-gamd*kp(2*j))
,(0,0,0,0,+kp(2*j),-kp(2*j)-kp(2*j+1),+kp(2*j+1),0)
,(0,0,0,0,0,+kp(2*j+1),-kp(2*j+1)-kp(2*j+2),+kp(2*j+2))
,(0,0,0,+gamd*kp(2*j),-gamd*kp(2*j),0,+kp(2*j+2),-kp(2*j+2))
)$
```

These are the only changes necessary.

7   Derive envelope equation for zigzag patterns

An outstanding issue in modelling is the direct discrete, macroscale modelling of pattern evolution [6, 2, e.g.]. On a lattice, one of the simplest

microscale discrete evolution equations is

$$\dot{u}_i = -4\mu^2 u_i + \alpha u_i^2 = -u_{i-1} - 2u_i - u_{i+1} + \alpha u_i^2. \tag{18}$$

What is the corresponding coarser grid equation governing the local amplitude of the zigzag mode? In particular, does the quadratic nonlinearity stabilise or destabilise the origin? [14, §7.3]

   The computer algebra to construct the slow manifold is copied from Section 4 but with numbers changed by $(-1)^i$ to account for the zigzag neutral mode. The corresponding evolution of the coarse grid order parameters, the amplitudes U_j, then governs the local amplitude of the zigzag mode, rather than the local mean field. Here the coarser scale *modulation* equation

$$\dot{U}_j = \gamma^2 (\tfrac{1}{4}\bar{\delta}^2 + \tfrac{1}{16}\bar{\delta}^4) U_j - \alpha\gamma\tfrac{1}{4}(\bar{\delta}\bar{\mu}U_j)(\bar{\delta}^2 U_j) + \tfrac{1}{2}\alpha^2 U_j^3 + \mathcal{O}(\gamma^2 + \alpha^3). \tag{19}$$

Evaluated at $\gamma = 1$ this coarse grid model predicts that the quadratic nonlinearity in (18) is destabilising as the rational closure of the fine grid dynamics generates the cubic growth term $\tfrac{1}{2}\alpha^2 U_j^3$.

Parameters   Compute the asymptotic series to this order of error in the small coupling parameter γ and in the small nonlinearity α. Omitting terms $\mathcal{O}(\alpha^3)$ gives a basic nonlinear model.

<div align="center">▷▷ eezzp ◁◁</div>

```
    let {gam^3=>0,alfa^3=>0};
```

   Improve printed appearance.

<div align="center">▷▷ eezzp ◁◁+</div>

```
    factor alfa;
```

Overview of the algorithm   Initialise, then iterate to improve the approximation of the slow manifold until the residual is zero to the required error. Note that `tp` denotes the transpose.

```
                    ▷▷ eezzp ◁◁+
% see caddsamm.pdf for documentation
◁◁ cgmdd preamble ▷▷
◁◁ eezzp initialisation ▷▷
zero:=tp mat((0,0,0,0,0,0,0,0))$
repeat begin
  ◁◁ eezzp residual ▷▷
  ◁◁ eezzp update ▷▷
  showtime;
end until res=zero;
◁◁ eezzp write ▷▷
end;
```

7.1 Initialise

The initial approximation is an arbitrary equilibrium point, $g_j = 0$, on the slow subspace of piecewise zigzag solutions $\boldsymbol{v}_j$.

```
                  ▷▷ eezzp initialisation ◁◁
gj:=0$
vj:=u(j)*tp mat((1,-1,1,-1,1,-1,1,-1))$
```

7.2 Compute residual

The evolution on the fine grid is the isolated zigzag formation, modified by the coupling with neighbouring elements, and with a nonlinear effect. Here the quadratic nonlinearity is scaled by γ to consistently simplify this truncation of the model.

```
                   ▷▷ eezzp residual ◁◁
res:=-tp mat((0,df(vj(2,1),t),df(vj(3,1),t),0
             ,0,df(vj(6,1),t),df(vj(7,1),t),0))
    +ll*vj
    +gamm*tp mat((
```

```
            sub(j=j-1,vj(4,1)+vj(5,1)),
            0,0,
            sub(j=j-1,vj(8,1))+sub(j=j+1,vj(1,1)),
            sub(j=j+1,vj(1,1))+sub(j=j-1,vj(8,1)),
            0,0,
            sub(j=j+1,vj(4,1)+vj(5,1))
            ))
    +gam*alfa*tp mat((0,vj(2,1)^2,vj(3,1)^2,0
                      ,0,vj(6,1)^2,vj(7,1)^2,0));
```

Where in the above, the linear operator on an element, decoupled from the other elements is the following.

```
              ▷▷ eezzp initialisation ◁◁+
  ll:=mat((-1,-1,0,gamd,gamd,0,0,0)
          ,(-1,-2,-1,0,0,0,0,0)
          ,(0,-1,-2,-1,0,0,0,0)
          ,(gamd,0,-1,-1,0,0,0,gamd)
          ,(gamd,0,0,0,-1,-1,0,gamd)
          ,(0,0,0,0,-1,-2,-1,0)
          ,(0,0,0,0,0,-1,-2,-1)
          ,(0,0,0,gamd,gamd,0,-1,-1))$
```

7.3   Update approximation to the slow manifold

Update the coarse grid evolution using the solvability condition.

```
              ▷▷ eezzp update ◁◁
  gd:=(1/4)*mat((1,-1,1,-1,1,-1,1,-1))*res;
  gj:=gj+gd(1,1);
```

Update the fine grid, intraelement structure using a generalised inverse `gil`.

```
                  ▷▷ eezzp update ◁◁+
    vj:=vj+gil*(-res+tp mat((0,-1,1,0,0,-1,1,0))*gd);
```

Use the following generalised inverse of `ll` that ensures the sum of the centre four grid values are zero—this ensures the amplitude of the coarse grid model is the grid value at the centre of each element.

```
                  ▷▷ eezzp initialisation ◁◁+
    gil:=mat(
        (-17,11,-5,3,7,-1,-9,19),
        (11,-17,7,-1,-5,3,3,-9),
        (-5,7,-9,-1,3,-5,3,-1),
        (3,-1,-1,-1,3,3,-5,7),
        (7,-5,3,3,-1,-1,-1,3),
        (-1,3,-5,3,-1,-9,7,-5),
        (-9,3,3,-5,-1,7,-17,11),
        (19,-9,-1,7,3,-5,11,-17))/16$
```

7.4   Write some of the slow manifold

Just print the evolution, after checking the amplitude equation.

```
                  ▷▷ eezzp write ◁◁
    res_ampl:=(for i:=3:6 sum (-1)^(i-1)*vj(i,1))-4*u(j);
    write gj:=gj;
```

References

[1] Todd Arbogast and Kirsten J. Boyd. Subgrid upscaling and mixed multiscale finite elements. *SIAM J. Numer. Anal.*, 44:1150–1171, 2006.

[2] H. R. Brand. Phase dynamics—a review and a perspective. *Propagation In Systems Far From Equilibrium*, pages 206–224, 1988.

[3] Achi Brandt. General highly accurate algebraic coarsening. *Elect. Trans. Num. Anal.*, 10:1–20, 2000.

[4] Shiyi Chen and G. D. Doolen. LatticeBoltzmann method for fluid flows. *Annu. Rev. Fluid Mechanics*, 30:329–364, 1998.

[5] J. Cisternas, C. W. Gear, S. Levin, and I. G. Kevrekidis. Equation-free modeling of evolving diseases: Coarse-grained computations with individual-based models. Technical report, [http://arXiv.org/abs/nlin.AO/0310011], 2003.

[6] M. C. Cross and P. C. Hohenberg. Pattern formation outside of equilibrium. *Rev. Mod. Phys.*, 65(3):851–1112, 1993. http://prola.aps.org/abstract/RMP/v65/i3/p851_1.

[7] Johannes Giannoulis, Michael Herrmann, and Alexander Mielke. Lagrangian and hamiltonian two-scale reduction. Technical report, http://arxiv.org/abs/0802.2820v1, 2008.

[8] Alexander Kurganov and Eitan Tadmor. New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations. *J. Computational Physics*, 160:241–282, 2000.

[9] T. MacKenzie and A. J. Roberts. Holistic discretisation of shear dispersion in a two-dimensional channel. In K. Burrage and Roger B. Sidje, editors, *Proc. of 10th Computational Techniques and Applications Conference CTAC-2001*, volume 44, pages C512–C530, March 2003. http://anziamj.austms.org.au/V44/CTAC2001/Mack.

[10] T. MacKenzie and A. J. Roberts. Accurately model the Kuramoto–Sivashinsky dynamics with holistic discretisation. *SIAM J. Applied Dynamical Systems*, 5(3):365–402, 2006. doi:10.1137/050627733 http://epubs.siam.org/SIADS/volume-05/art_62773.html.

[11] Mauro Mobilia, Ivan T. Georgiev, and Uwe C. Tauber. Spatial stochastic predator-prey models. Technical report, `http://arxiv.org/abs/q-bio.PE/0609039`, 2006.

[12] A. J. Roberts. A holistic finite difference approach models linear dynamics consistently. *Mathematics of Computation*, 72:247–262, 2002. `http://www.ams.org/mcom/2003-72-241/S0025-5718-02-01448-5`.

[13] A. J. Roberts. Resolving the multitude of microscale interactions accurately models stochastic partial differential equations. *LMS J. Computation and Maths*, 9:193–221, 2006. `http://www.lms.ac.uk/jcm/9/lms2005-032`.

[14] A. J. Roberts. Conserving self-adjoint rules for coupling in multiscale modelling. Technical report, University of Southern Queensland, 2008.

[15] A. J. Roberts. Model dynamics on a multigrid across multiple length and time scales. Technical report, http://arxiv.org/abs/0802.1098, 2008.

[16] A. J. Roberts and I. G. Kevrekidis. General tooth boundary conditions for equation free modelling. *SIAM J. Scientific Computing*, 29(4):1495–1510, 2007.

[17] G. Samaey, I. G. Kevrekidis, and D. Roose. The gap-tooth scheme for homogenization problems. *SIAM Multiscale Modeling and Simulation*, 4:278–306, 2005. doi:10.1137/030602046.

[18] Matthew J. Simpson, Alistair Merrifield, Kerry A. Landman, and Barry D. Hughes. Simulating invasion with cellular automata: Connecting cell-scale and population-scale properties. *Physical Review E*, 76:021918, 2007.

[19] T. P. Witelski and A. J. Bernoff. Self-similar asymptotics for linear and nonlinear diffusion equations. *Studies in Applied Maths.*, 100:153–193, 1998.

[20] D. J. Wollkind, V. S. Manoranjan, and L. Zhang. Weakly nonlinear analyses of prototype reaction-diffusion model equations. *SIAM Review*, 36(2):176–214, 1994.