`nearPeriodDiffEqui1`: errors in patch scheme for equilibrium of a 1D heterogeneous diffusion with nearly correct period

A.J. Roberts

September 5, 2023

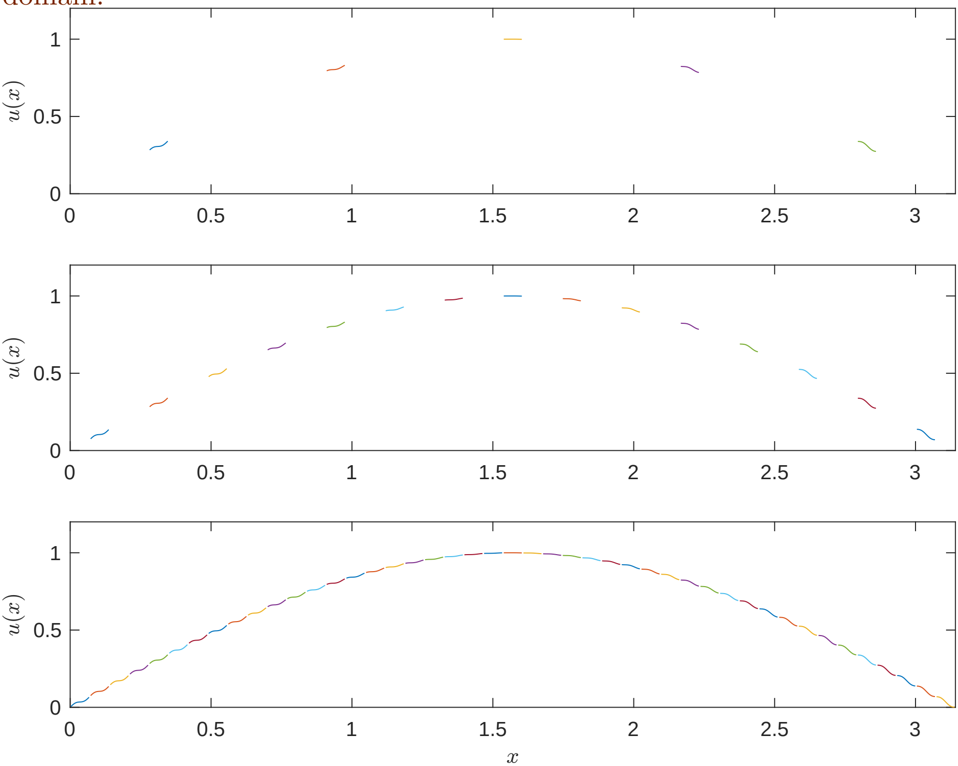
Explore heterogeneous diffusion in 1D on patches to compare with approach in arxiv:2308.07563 by authors CDE. The microscale period ϵ is to be a little different from the cell (patch) size η. Then explore accuracy via forced equilibria. Here we use cells that are patches in the *equation-free patch scheme* (e.g., Roberts et al. 2023; Bunder, Kevrekidis, and Roberts 2021; Samaey, Roberts, and Kevrekidis 2010). We invoke functions from the *Equation-Free Toolbox* (Maclean, Bunder, and Roberts 2021).

Suppose the spatial microscale lattice is at points x_i, with constant spacing dx. With dependent variables $u_i(t)$, let's seek the equilibriun of the forced the microscale lattice diffusion system

$$\frac{\partial u_i}{\partial t} = \frac{1}{dx^2}\delta[c_{i-1/2}\delta u_i] - u_i + f_i, \tag{1}$$

in terms of the centred difference operator δ, and for some time-constant forcing f_i. The system has a microscale heterogeneity via the coefficients $c_{i+1/2}$ which has periodicity ϵ in x. Instead of varying cell-size for fixed heterogeneity period, here we explore results for various periods ϵ at fixed cell-size η.

Figure 1: example solutions of heterogeneous diffusion equilibrium with forcing $f(x) = \sin x$. By symmetry the plot shows half the domain. As shown, solve with $10, 30, 90$ patches/cells. The bottom 90 patch equilibrium is the exact reference solution. The top two equilibria are for patch ratios $r = 1/9, 1/3$ respectively: computation is done only on the fraction r of the domain.



0.1 Code various numbers of patches over domain

Establish system of length 2π. Explore various number of patches.

```
%}
clear all
Xlim = [-pi pi]
lMax = 3
nPatches = 10 *3.^(0:lMax-1)
maxDetune = 100
%{
```

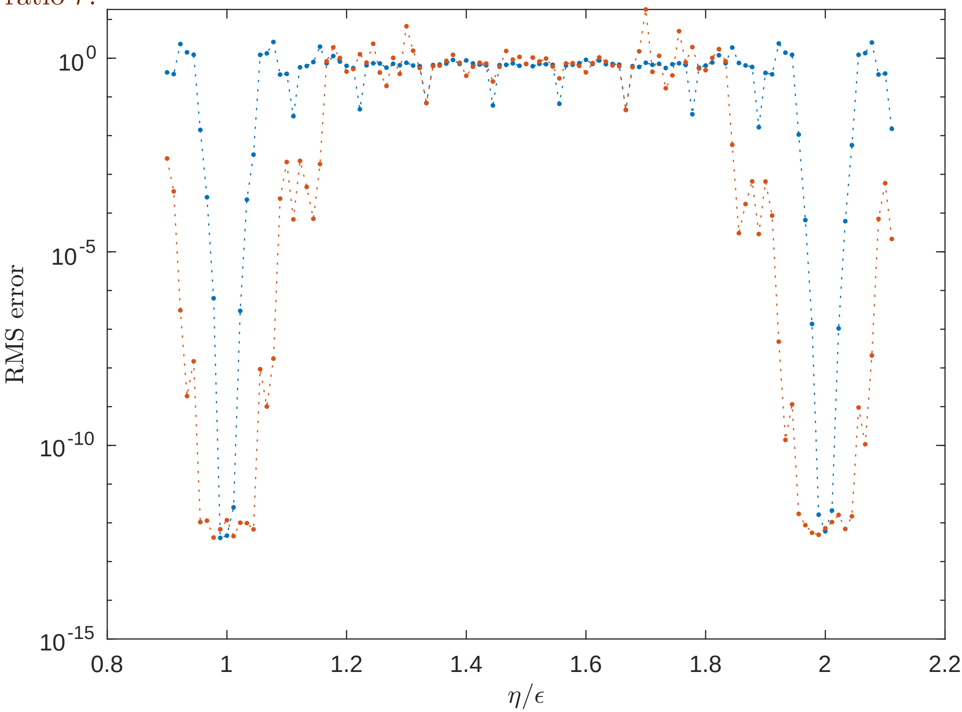
Set up microgrid parameters, and set strength of heterogeneity (abs-value less than one). CDE used cell size $\eta/\epsilon \in [1, 50]$, and at least 4 096 points, so here with, say, 90 cells that is over 45 points per cell (per patch). For computational speed, use less.

```
%}
mPerPatch = 12
eta = diff(Xlim)/nPatches(lMax)
dx = eta/mPerPatch
heteroAmp = 0.9 % 0.9 is close to CDE's (4.1a)
%{
```

Loop over cell to heterogeneity ratios The micro-scale heterogeneity must be 2π-periodic over the macroscale domain, so in general has to be of the following form for integer `nDetune` (positive means smaller period, as shown by CDE, negative means larger).

```
%}
RmsErrs = [];  etaEps = [];
for nDetune = -9:maxDetune
epsilon = 2*pi/(2*pi/eta+nDetune)
%{
```

Figure 2: errors in equilibrium as function of cell-size η to heterogeneity periodicity ϵ: blue dots are $r = 1/9$; red dots are $r = 1/3$. The errors are RMS of the difference between solutions, such as in Figure 1, in the five common patches/cells. The errors are essentially zero in the vicinity of the ideal integer ratio for η/ϵ. The low error valleys appear broader for larger ratio r.



Consequently, the heterogeneity

$$\begin{aligned}\cos(2\pi x/\epsilon) &= \cos\left[2\pi x/\eta + 2\pi(1/\epsilon - 1/\eta)x\right] \\ &= \cos(2\pi x/\eta)\cos(kx) - \sin(2\pi x/\eta)\sin(kx) \\ &\text{for wavenumber } k := 2\pi(1/\epsilon - 1/\eta).\end{aligned}$$

So the discrepancy between cell-size and heterogeneous-period can be viewed as a modulation of precise cell-periodicity by 'macroscale' modulation of wavenumber k. That is, the discrepancy may be viewed as an example of a "functionally graded material". In the patch scheme such modulations are resolved on patches of spacing H provided their wavenumber $k < \pi/H$. That is, patch-resolution requires

$$\begin{aligned} &2\pi(1/\epsilon - 1/\eta) < \pi/H \\ \iff &\eta/\epsilon - 1 < (\eta/H)/2 \\ \iff &\eta/\epsilon < 1 + r/2 \end{aligned}$$

where $r := \eta/H$ is the patch ratio. For example, for $r = \frac{1}{3}, \frac{1}{9}$ we need $\frac{\eta}{\epsilon} < \frac{7}{6}, \frac{19}{18} \approx 1.17, 1.06$ in order to realise accuracy—see Figure 2.

The sides of the valleys for $r = 1/3$ in Figure 2 appear to be affected by higher-harmonic structures developing in the equilibrium solution, and these structures are then relatively poorly resolved.

0.2 Code to create the patch schemes

Establish the global data struct `patches` for the microscale heterogeneous lattice diffusion system (1) solved on 2π-periodic domain. Use spectral interpolation for best accuracy.

```
%}
global patches
us=[];
for nPatch = nPatches
    nSubP = mPerPatch+2;
    configPatches1(@heteroDiff,Xlim,'periodic',nPatch ...
```

```
        ,0,dx,nSubP,'EdgyInt',true);
%{
```

Set the microscale heterogeneity with harmonic mean one, and set the forcing so solution should have amplitude near one. Choose forcing anti-symmetric to ensure solution effectively satisfies BCs of $u = 0$ at $x = 0, \pi$.

```
%}
  xMid = ( patches.x(1:end-1,:,:,:)+patches.x(2:end,:,:,:) )/2;
  patches.cs = 1./(1+ heteroAmp*cos(2*pi*xMid/epsilon) );
%  xMid=squeeze(xMid)
%  xMicro=squeeze(patches.x)
%  cMicro = squeeze(patches.cs)
  patches.f = 2*sin(patches.x); % sign or sin
  xs=squeeze(patches.x);
%{
```

Solve for equilibrium The linear system is of the form $\vec f(\vec u) = J\vec u + \vec f_0$. Get the constant term in the system by evaluating at zero.

```
%}
  u0 = 0*patches.x;
  f0 = patchSys1(1,u0);
%{
```

Put NaNs on the patch-edges, to then find all micro-grid points interior to the patches, and hence are the variables in $\vec u$.

```
%}
  u0([1 end],:,:,:)=nan;
  i=find(~isnan(u0));
  nJac=length(i);
%{
```

Create Jacobian J column by column: since linear we numerically differentiate with unit vectors.

```
%}
```

```
  Jac=nan(nJac);
  for j=1:nJac
    u0(i)=((1:nJac)==j);
    dudt= (patchSys1(0,u0)-f0);
    Jac(:,j)=dudt(i);
  end
  assert(rcond(Jac)>1e-9,'Jacobian seems too ill-conditioned')
%{
```

Solve linear system.

```
%}
  u0(i) = -sparse(Jac)\f0(i);
  ueq = squeeze(u0);
%{
```

Check the residual.

```
%}
  res = patchSys1(1,u0);
  normRes = norm(res(i));
  assert(normRes<1e-8,"norm of the residual is too big")
%{
```

Plot one example equilibrium for comparison

```
%}
    if nDetune==1
    figure(1)
    subplot(lMax,1,find(nPatch==nPatches))
    j=find(xs(2,:)>0);
    plot(xs(:,j),ueq(:,j))
    ylim([min(0,min(min(ueq(:,j)))) max(1.2,max(max(ueq(:,j))))]),
    xlim([0 pi])
    ylabel("$u(x)$")
    if nPatch==nPatches(end),
        xlabel("$x$"), drawnow
        %set(gca,'position',[.2 .2 r r])
```

```
        exportgraphics(gcf,mfilename+"ueq.pdf" ...
        ,'ContentType','vector')
        end%if
    end%if nDetune
%{
```

Use common patches for quantitative comparison

```
%}
    if nPatch==nPatches(1), J=find(xs(2,:)>0);
    else J=3*J-1;
    end
    us = cat(3,us,ueq(:,J));
%{
```

End of the for-loop over the number of patches. Here compute the errors in the patch solutions compared to the full-domain exact solution.

```
%}
end%for nPatch
uErrs = us-us(:,:,end);
rmsErrs = reshape( rms(uErrs,[1 2],'omitnan') ,1,[]);
log10rmsErrs = log10(rmsErrs)
RmsErrs = [RmsErrs; rmsErrs];
etaEps = [etaEps; eta/epsilon];
%{
```

At end of loop over detuning parameters, plot errors as function of the cell to periodicity ratio.

```
%}
end%for nDetune
figure(2)
semilogy(etaEps,RmsErrs,'.:')
xlabel("$\eta/\epsilon$")
ylabel("RMS error")
set(gca,'position',[.2 .2 .64 .64])
exportgraphics(gcf,mfilename+"Err.pdf" ...
```

```
    ,'ContentType','vector')
%{
```

End of the main script.

0.3 heteroDiff(): heterogeneous diffusion

This function codes the lattice heterogeneous diffusion inside the patches. For 2D input arrays `u` and `x` (via edge-value interpolation of `patchSys1`, **??**), computes the time derivative **??** at each point in the interior of a patch, output in `ut`. The array of diffusivities c_i have previously been stored in struct `patches.cs`.

```
%}
function ut = heteroDiff(t,u,patches)
  dx = diff(patches.x(2:3));   % space step
  i = 2:size(u,1)-1;   % interior points in a patch
  ut = nan+u;          % preallocate output array
  ut(i,:,:,:) = diff(patches.cs(:,1,:,:).*diff(u))/dx^2 ...
                -u(i,:,:,:) +patches.f(i,:,:,:);
end% function
%{
```

References

Bunder, J. E., I. G. Kevrekidis, and A. J. Roberts (July 2021). "Equation-free patch scheme for efficient computational homogenisation via self-adjoint coupling". In: *Numerische Mathematik* 149.2, pp. 229–272. DOI: 10.1007/s00211-021-01232-5 (cit. on p. 1).

Maclean, John, J. E. Bunder, and A. J. Roberts (2021). "A toolbox of Equation-Free functions in Matlab/Octave for efficient system level simulation". In: *Numerical Algorithms* 87, pp. 1729–1748. DOI: 10.1007/s11075-020-01027-z (cit. on p. 1).

Roberts, A. J. et al. (Jan. 2023). *Accurate and efficient multiscale simulation of a heterogeneous elastic beam via computation on small sparse patches*. Tech. rep. https://arxiv.org/abs/2301.13145. DOI: 10.48550/arXiv.2301.13145 (cit. on p. 1).

Samaey, G., A. J. Roberts, and I. G. Kevrekidis (2010). "Equation-free computation: an overview of patch dynamics". In: *Multiscale methods: bridging the scales in science and engineering*. Ed. by Jacob Fish. Oxford University Press. Chap. 8, pp. 216–246. ISBN: 978-0-19-923385-4 (cit. on p. 1).