Pseudospectra of the patch scheme in various scenarios shows mostly insensitive

A. J. Roberts[*]

May 22, 2023

Contents

1 Introduction

Recall that eigenvalues λ of a matrix A are complex numbers $z \in \mathbb{C}$ such that the *resolvent* $(zI - A)^{-1}$ does not exist—is 'infinite' (Trefethen and Embree 2005, p.3).

Definition 1. *For a given $N \times N$ matrix A, the ϵ-*pseudospectrum *of A is the (open) set $z \in \mathbb{C}$ such that (Trefethen and Embree 2005, pp.13–17) any of the following four equivalent conditions hold:*[1]

[*]School of Mathematical Sciences, University of Adelaide, South Australia. https://profajroberts.github.io, http://orcid.org/0000-0001-8930-1552

[1]Define $\|\vec v\| := \sqrt{\vec v \cdot \vec v}$, and $\|A\| := \max_{\|\vec v\|=1} \|A\vec v\|$.

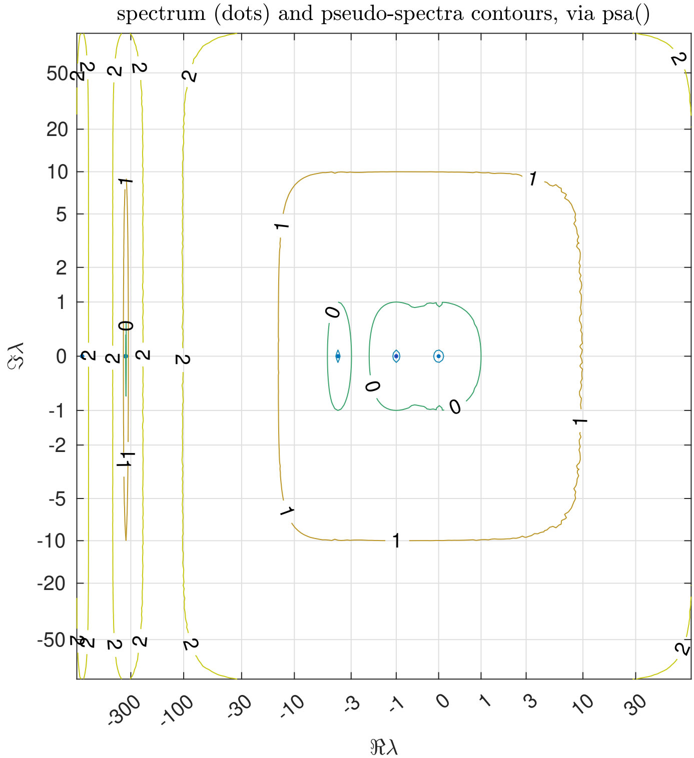


Figure 1: pseudo-spectra of a patch scheme applied to heterogeneous diffusion in 1D space (Appendix A). Plotted are the contours of the reciprocal-resolvent $1/(zI-A)^{-1}$ for $\epsilon = 10^{-1:2}$. The interior of these contours is the corresponding ϵ-pseudospectrum. The asinh-scaling of each axis deforms the circular contours.

- $\|(zI-A)^{-1}\| > \epsilon^{-1}$;
- *z is an eigenvalue of $A+E$ for some matrix E with $\|E\| < \epsilon$;*
- $\min_{\|\vec v\|=1} \|(zI-A)\vec v\| < \epsilon$;
- *the smallest singular value of $zI-A$ is $< \epsilon$.*

The ϵ-pseudospectrum is denoted $\sigma_\epsilon(A)$.

So-called normal matrices have the nicest properties: a matrix A is termed *normal* if it has a complete set of orthogonal eigenvectors (Trefethen and Embree 2005, p.18). That is, if real then A is normal iff it is symmetric. Equivalently, a matrix A is normal if it commutes with its adjoint: $A^\dagger A = AA^\dagger$.

Theorem 2. *Define the ϵ-disk $B_\epsilon := \{z \in \mathbb{C} : |z| < \epsilon\}$. Then $\sigma_\epsilon(A) \supseteq \sigma(A) + B_\epsilon$ for each matrix A. Equality holds iff A is normal.*

That is, for a normal matrix A, the ϵ-pseudospectrum is the union of ϵ-disks centred on each eigenvalue. This is the best case. For example, Figure 1 plots, for $\epsilon = 10^{-1:2}$, the ϵ-pseudospectrum of a patch scheme applied to heterogeneous

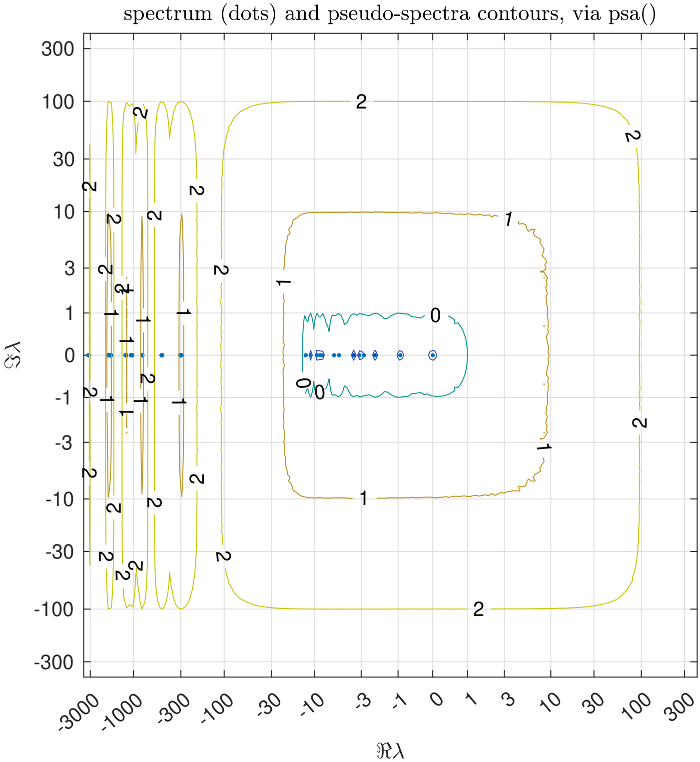


Figure 2: pseudo-spectrum of a patch scheme applied to heterogeneous diffusion in 2D space (Appendix B). Plotted are the contours of the reciprocal-resolvent $1/(zI - A)^{-1}$ for $\epsilon = 10^{-1:2}$. Figures 1 and 2 use spectral, edgy interpolation: finite order and/or centre-cross interpolation appears similar.

diffusion in 1D space: the plot is consistent with the pseudospectra being the union of circular disks (do not be misled by the nonlinear scaling of both axes).

For a non-normal matrix, the ϵ-pseudospectrum is larger than (a superset of) the union of ϵ-disks centred on each eigenvalue.

2 Patch schemes

Possibly the most significant property of the ϵ-pseudospectra for a patch scheme is the property that $\sigma_\epsilon(A)$ are the eigenvalues of $A+E$ for some E with $\|E\| < \epsilon$. Let matrix A denote the linearisation matrix of the chosen patch scheme wrapped around a given microscale system. Then matrix E represents 'perturbation' effects of the linear system due to nonlinearity, forcing, spatial variations, etc. Consequently, the ϵ-pseudospectra indicates whether a chosen patch scheme is likely to induce any artificial instability in real application where such effects are typical.

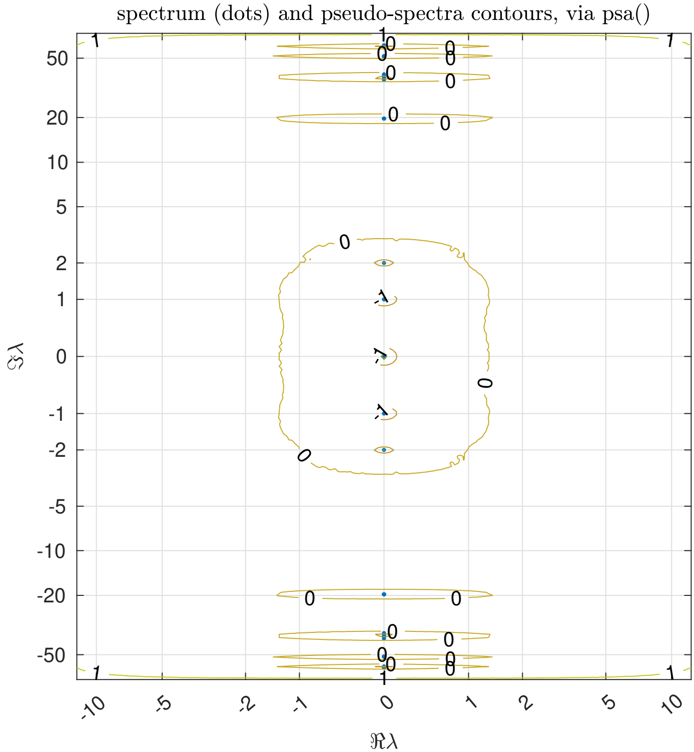


Figure 3: pseudo-spectrum of a patch scheme applied to ideal wave system in 1D space (Appendix C): $h_t = u_x$ and $u_t = h_x$. Plotted are the contours of the reciprocal-resolvent $1/(zI - A)^{-1}$ for $\epsilon = 0.1, 1$. Spectra are for the toolbox's staggered patches with spectral interpolation.

Figures 1 and 2 indicate that the patch scheme implemented in the toolbox[2] for heterogeneous diffusion in 1D and 2D are robust.

Wave-like systems Wave systems are challenging because it is excruciatingly easy for perturbations to tip the patch-wrapped system into unphysical instability. Figure 3 are ϵ-pseudospectra for a staggered patch scheme in 1D ideal waves. The spectra indicate the patch-wrapped waves are reasonably robust to perturbations: the ϵ-pseudospectra appear to be circles as for normal matrices; and the microscale subpatch waves appear no more sensitive to perturbations than the macroscale waves.

However, waves in multiple space dimensions are trickier. Figure 4 plots the pseudo-spectra for an older patch scheme applied to weakly dissipative waves. The shape of the pseudospectra appears non-circular, suggesting the matrix is non-normal. The plot suggests that some of the sub-patch modes may be a little sensitive to perturbations in that a $\epsilon = 1$ contour (labelled 0)

[2]Maclean, Bunder, and Roberts 2021; Roberts, Maclean, and Bunder 2019–2023.

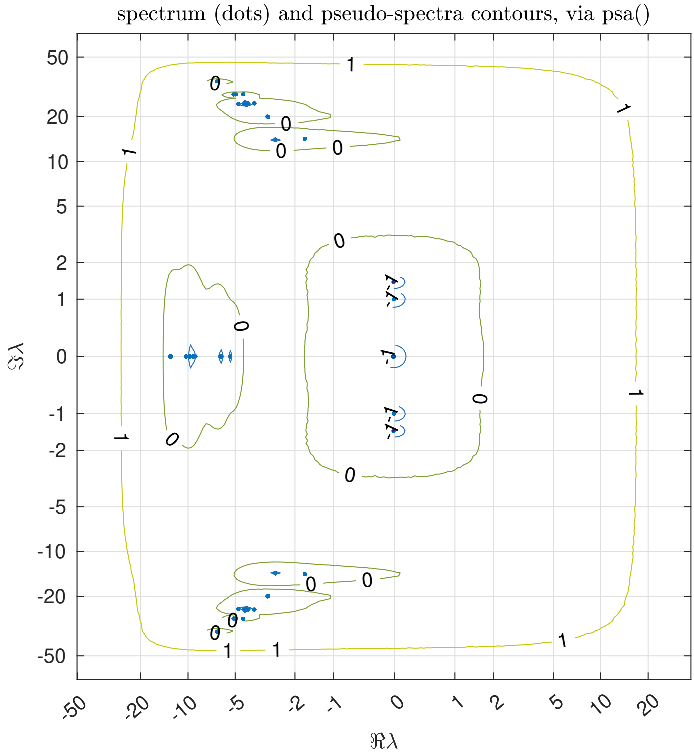


Figure 4: pseudo-spectrum of a patch scheme applied to weakly dissipative waves in 2D space: here $\epsilon = 0.1, 1, 10$. The scheme is staggered patches, with centre-node to patch-edge interpolation. Some sub-patch waves may be a little sensitive.

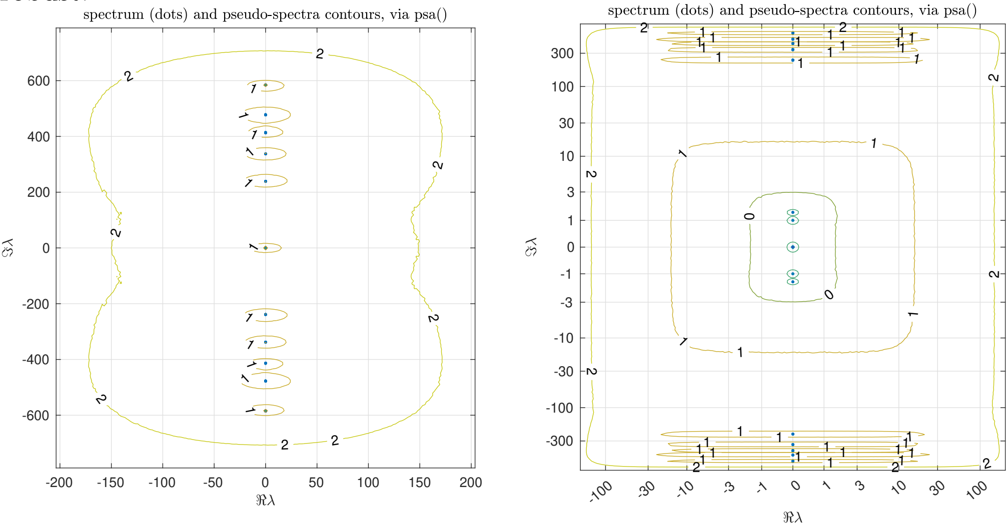
pokes out into instability.

Figure 5 shows what happens when we change the inter-patch interpolation to that *from the centre-cross values* in a patch to the patch-edges. The left panel suggests the patch-wrapped wave system is normal. The right panel expands the domain of the macroscale waves near zero to further indicate the robust properties of this patch-wrapping.

The toolbox (Maclean, Bunder, and Roberts 2021; Roberts, Maclean, and Bunder 2019–2023) implements this more robust centre-cross interpolation for multi-D space.

ToDo explore pseudospectra of implemented polynomial interpolation for bounded domains.

Figure 5: pseudo-spectrum of a patch scheme applied to heterogeneous waves in 2D space: left, unscaled; *right,* scaled. *Here $\epsilon = 10^{-1:2}$. The staggered patch scheme interpolates centre-cross values to the patch edges. These ϵ-pseudospectra suggest that this patch scheme for waves is normal and so robust.*



A heteroDiffPseudoSpectra1: computational homogenisation of a 1D heterogeneous diffusion by simulation on small patches

?? shows an example simulation in time generated by the patch scheme applied to macroscale diffusion propagation through a medium with microscale heterogeneity. The inter-patch coupling is realised by quartic interpolation of the patch's next-to-edge values to the patch opposite edges. Such coupling preserves symmetry in many systems, and quartic appears to be the lowest order that generally gives good accuracy.

Suppose the spatial microscale lattice is at points x_i, with constant spacing dx. With dependent variables $u_i(t)$, simulate the microscale lattice diffusion system

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

in terms of the centred difference operator δ. The system has a microscale heterogeneity via the coefficients $c_{i+1/2}$ which we assume to have some given known periodicity. **??** shows one patch simulation of this system: observe the effects of the heterogeneity within each patch.

First establish the microscale heterogeneity has micro-period `mPeriod` on the lattice, and random log-normal values, albeit normalised to have harmonic mean one. This normalisation then means that macroscale diffusion on a domain of length 2π should have near integer decay rates, the squares of $0, 1, 2, \ldots$. Then the heterogeneity is repeated to fill each patch, and phase-shifted for an ensemble.

```
mPeriod = 3%randi([2 5])
% set random diffusion coefficients
cHetr=exp(0.9*randn(mPeriod,1));
%cHetr = [3.966;2.531;0.838;0.331;7.276];
cHetr = cHetr*mean(1./cHetr) % normalise
```

Establish the global data struct `patches` for the microscale heterogeneous lattice diffusion system (1) solved on 2π-periodic domain. When simulating an ensemble of configurations, `nSubP` (the number of points in a patch) need not be dependent on the period of the heterogeneous diffusion.

```
edgyInt = true
nSubP = (2-edgyInt)*mPeriod+1+edgyInt
```

Compute Jacobian and its spectrum Let's explore the Jacobian dynamics for a range of orders of interpolation, all for the same patch design and heterogeneity. Here use a smaller ratio, and more patches, as we do not plot.

```
nPatch = 5
ratio = 0.2;

leadingEvals = [];
for ord = 0:2:0
  ordInterp = ord
  configPatches1(@heteroDiff,[-pi pi],nan,nPatch ...
      ,ord,ratio,nSubP,'EdgyInt',edgyInt ,'hetCoeffs',cHetr);
```

Form the Jacobian matrix, linear operator, by numerical construction about a zero field. Use `i` to store the indices of the micro-grid points that are interior to the patches and hence are the system variables.

```
  u0 = zeros(nSubP,1,1,nPatch);
  u0([1 end],:,:,:)=nan; u0=u0(:);
```

```
  i = find(~isnan(u0));
  nJac = length(i)
  Jac = nan(nJac);
  for j = 1:nJac
    u0(i) = ((1:nJac)==j);
    dudt = patchSys1(0,u0);
    Jac(:,j) = dudt(i);
  end
  nonSymmetric = norm(Jac-Jac')
%  assert(nonSymmetric<5e-9,'failed symmetry')
  Jac(abs(Jac)<1e-12) = 0;
```

Find the eigenvalues of the Jacobian: the spectral interpolation is effectively exact for the macroscale; quadratic interpolation is usually quantitatively in error; quartic interpolation appears to be the lowest order for reliable quantitative accuracy.

The number of zero eigenvalues, `nZeroEv`, indicates the number of decoupled systems in this patch configuration.

```
  evals = eig(Jac);
  eval = -sort(-real(evals));
  nZeroEv = sum(eval(:)>-1e-5)
  leadingEvals = [leadingEvals eval(1:2*nPatch)]
```

Pseuso-spectrum of the Jacobian

```
figure(1)
if 0
    opts.npts=200; opts.levels=-8:2
    eigtool(Jac,opts,1)
else
multiscalePseudoSpectra(Jac,1,1,200)
figfn=mfilename;
set(gcf,'PaperUnits','centimeters' ...
      ,'PaperPosition',[0 0 14 14] ...
      ,'renderer','Painters')
print('-depsc2',figfn)
matlab2tikz([figfn '.tex'],'showInfo',false ...
    ,'noSize',true,'parseStrings',false,'showWarnings',false ...
    ,'extraCode',['\tikzsetnextfilename{' figfn '}'] ...
```

```
    ,'extraAxisOptions','\extraAxisOptions' ...
    ,'checkForUpdates',false)
end
```

End of the for-loop over orders of interpolation.

```
end
```

End of the main script.

B heteroDiffPseudoSpectra2: pseudospectrum of computational homogenisation of a 2D diffusion

This section extends to 2D the 1D code discussed in **??**. First set random heterogeneous diffusivities of random period in each of the two directions. Crudely normalise by the harmonic mean so the decay time scale is roughly one.

```
mPeriod = [3 3];
cHetr = exp(1*randn([mPeriod 2]));
cHetr = cHetr*mean(1./cHetr(:))
```

B.1 Compute Jacobian and its spectrum

Let's explore the Jacobian dynamics for a range of orders of interpolation, all for the same patch design and heterogeneity. Except here use a small ratio as we do not plot.

```
edgyInt = true
nSubP = (2-edgyInt)*mPeriod+1+edgyInt
nPatch = [3 3] +2*edgyInt
ratio = 0.2
nLeadEvals=prod(nPatch)+max(nPatch);
leadingEvals=[];
```

Evaluate eigenvalues for spectral as the base case for polynomial interpolation of order $2, 4, \ldots$.

```
maxords = 0;
for ord = maxords:2:maxords
    ordInterp = ord
```

Configure with same parameters, then because they are reset by this configuration, restore coupling.

```
    configPatches2(@heteroDiff2,[-pi pi -pi pi],nan,nPatch ...
        ,ord,ratio,nSubP,'EdgyInt',edgyInt,'hetCoeffs',cHetr);
```

Find which elements of the 6D array are interior micro-grid points and hence correspond to dynamical variables.

```
    u0 = zeros([nSubP,1,1,nPatch]);
    u0([1 end],:,:) = nan;
    u0(:,[1 end],:) = nan;
    i = find(~isnan(u0));
```

Construct the Jacobian of the scheme as the matrix of the linear transformation, obtained by transforming the standard unit vectors.

```
    nJac = length(i)
    Jac = nan(nJac);
    sizeJacobian = size(Jac)
    for j = 1:nJac
      u0(i) = (j==(1:nJac));
      dudt = patchSys2(0,u0);
      Jac(:,j) = dudt(i);
    end
```

Test for symmetry, with error if we know it should be symmetric.

```
    notSymmetric = norm(Jac-Jac')
    if notSymmetric>1e-7, disp("failed symmetry"), end
    Jac(abs(Jac)<1e-12) = 0;
```

Find all the eigenvalues (as `eigs` is unreliable).

```
    evals = eig(Jac);
    biggestImag = max(abs(imag(evals)));
    if biggestImag>0, biggestImag=biggestImag, end
```

Sort eigenvalues on their real-part with most positive first, and most negative last. Store the leading eigenvalues in `egs`, and write out when computed all orders. The number of zero eigenvalues, `nZeroEv`, gives the number of decoupled systems in this patch configuration.

```
    [~,k] = sort(real(evals),'Descend');
    evals = evals(k);
    if ord==0, nZeroEv=sum(abs(evals(:))<1e-5), end
    leadingEvals=[leadingEvals evals(nZeroEv*(1:nLeadEvals))]
```

Pseuso-spectrum of the Jacobian

```
figure(1)
multiscalePseudoSpectra(Jac,1,1,200)
figfn=mfilename;
set(gcf,'PaperUnits','centimeters' ...
      ,'PaperPosition',[0 0 14 14] ...
      ,'renderer','Painters')
print('-depsc2',figfn)
matlab2tikz([figfn '.tex'],'showInfo',false ...
    ,'noSize',true,'parseStrings',false,'showWarnings',false ...
    ,'extraCode',['\tikzsetnextfilename{' figfn '}'] ...
    ,'extraAxisOptions','\extraAxisOptions' ...
    ,'checkForUpdates',false)
```

End of the for-loop over orders of interpolation.

```
end
```

End of the main script.

C `waveIdealPS`: simulate a water wave PDE on patches

?? shows an example simulation in time generated by the patch scheme applied to an ideal wave PDE (Cao and Roberts 2013). The inter-patch coupling is realised by spectral interpolation of the mid-patch values to the patch edges.

Establish the global data struct `patches` for the PDEs (linearised) solved on 2π-periodic domain, with eight patches, each patch of half-size ratio 0.1, with seven/eleven micro-grid points within each patch, and spectral interpolation (-1) of 'staggered' macroscale patches to provide the edge-values of the inter-patch coupling conditions.

```
global patches
nPatch = 8
ratio = 0.1
```

```
nSubP = 11 %of the form 4*n-1
Len = 2*pi;
configPatches1(@idealWavePDE,[0 Len],nan,nPatch,-1,ratio,nSubP);
```

Identify which micro-grid points are h or u values on the staggered micro-grid. Also store the information in the struct `patches` for use by the time derivative function.

```
tmp = mod( (1:nSubP)'+(1:nPatch) ,2);
patches.hPts = find(tmp==0);
patches.uPts = find(tmp==1);
```

Form the Jacobian matrix, linear operator, by numerical construction about a zero field. Use `i` to store the indices of the micro-grid points that are interior to the patches and hence are the system variables.

```
  u0 = zeros(nSubP,1,1,nPatch);
  u0([1 end],:,:,:)=nan; u0=u0(:);
  i = find(~isnan(u0));
  nJac = length(i)
  Jac = nan(nJac);
  for j = 1:nJac
    u0(i) = ((1:nJac)==j);
    dudt = patchSys1(0,u0);
    Jac(:,j) = dudt(i);
  end
  Jac(abs(Jac)<1e-12) = 0;
```

Find the eigenvalues of the Jacobian: the spectral interpolation is effectively exact for the macroscale; quadratic interpolation is usually quantitatively in error; quartic interpolation appears to be the lowest order for reliable quantitative accuracy.

The number of zero eigenvalues, `nZeroEv`, indicates the number of decoupled systems in this patch configuration.

```
  evals = eig(Jac);
  [~,k] = sort(abs(evals));
  evals=evals(k);
  nZeroEv = sum(abs(evals(:))<1e-5)
  leadingEvals = evals(1:2*nPatch)
```

Pseuso-spectrum of the Jacobian

```
figure(1)
multiscalePseudoSpectra(Jac,1,1,200)
figfn=mfilename;
set(gcf,'PaperUnits','centimeters' ...
      ,'PaperPosition',[0 0 14 14] ...
      ,'renderer','Painters')
print('-depsc2',figfn)
matlab2tikz([figfn '.tex'],'showInfo',false ...
    ,'noSize',true,'parseStrings',false,'showWarnings',false ...
    ,'extraCode',['\tikzsetnextfilename{' figfn '}'] ...
    ,'extraAxisOptions','\extraAxisOptions' ...
    ,'checkForUpdates',false)
```

References

Cao, Meng and A. J. Roberts (May 2013). "Multiscale modelling couples patches of wave-like simulations". In: *Proceedings of the 16th Biennial Computational Techniques and Applications Conference, CTAC-2012*. Ed. by Scott McCue et al. Vol. 54. ANZIAM J. Pp. C153–C170. DOI: 10.21914/anziamj.v54i0.6137 (cit. on p. 11).

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 pp. 4, 5).

Roberts, A. J., John Maclean, and J. E. Bunder (2019–2023). *Equation-Free function toolbox for Matlab/Octave*. Tech. rep. [https://github.com/uoa1184615/EquationFreeGit] (cit. on pp. 4, 5).

Trefethen, Lloyd N. and Mark Embree (2005). *Spectra and Pseudospectra: the Behaviour of Nonnormal Matrices and Operators*. Princeton University Press. ISBN: 978-0-691-11946-5 (cit. on pp. 1, 2).