Discretisation of nonlinear Schrodinger PDE with periodic potential—modified for coupling via averages over cores and action regions

AJR

July 8, 2016

List of pdfcomments

Contents

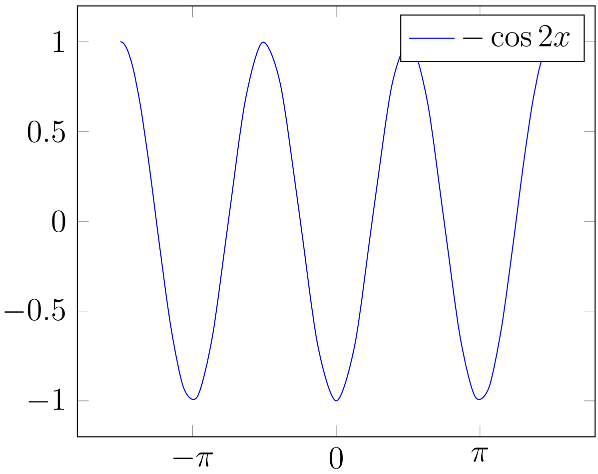

Figure 1: Wannier potential $V(x) = A\cos 2x$ for $A = -1$

1 Introduction

The immediate aim is to model the dynamics in just one space dimension of the nonlinear Schrodinger PDE (1) with periodic potential. That is, we seek to model the field $u(x,t)$ that solves

$$-i\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - V(x)u - \sigma|u^2|u \tag{1}$$

where here, for example and as illustrated by Figure 1, we could take potential

- $V(x) := \nu[1-\cos(2\pi x/H)]\pi^2/2/H^2$,
- or $V(x) := A\cos(2\pi x/H)\pi^2/H^2$ to agree with Alfimov et al. (2002) when $H=\pi$ (but we all use negative A to get localisation about positions $x=jH$),

The plan is to have 'amplitudes' that measures what goes on in the jth well which is centred about $X_j = jH$. To do this we notionally divide space into overlapping elements; the jth element being $E_j := [X_{j-1}-H/2, X_{j+1}+H/2]$ which is centred on X_j and of total length $3H$. Each element is notionally divided into three regions:

- the core region is $[X_j - H/2, X_j + H/2]$; and
- two action regions $[X_{j\pm1}-H/2, X_{j\pm1}+H/2]$.

The subgrid field in each element, $u_j(x,t)$, is coupled to its neighbours by

$$\begin{aligned}\frac{1}{H}\int_{X_{j\pm1}-H/2}^{X_{j\pm1}+H/2} u_j(x,t)\,dx &= \gamma\frac{1}{H}\int_{X_{j\pm1}-H/2}^{X_{j\pm1}+H/2} u_{j\pm1}(x,t)\,dx \\ &\quad + (1-\gamma)\frac{1}{H}\int_{X_j-H/2}^{X_j+H/2} u_j(x,t)\,dx,\end{aligned} \tag{2}$$

Then we construct a model in powers of γ that parametrises the coupling between elements, and hence the coupling between wells. For simplicity, we construct the model in powers of the strength of the well because that is an easy thing to do: quick and dirty.

For example, and following Alfimov et al. (2002), for spacing $H=\pi$, this algorithm constructs the model of linear dynamics, $\sigma=0$, that

$$-i\dot U_j = \tfrac18 A^2 U_j + \gamma(1-\tfrac18 A^2)\tfrac{1}{\pi^2}(U_{j-1}-2U_j+U_{j+1}) + \mathcal{O}\big(\gamma^2, A^3, \sigma\big). \tag{3}$$

The subgrid fields of the slow manifold are complicated, even at this low order of truncation: the first few terms are

$$u_j = U_j - \tfrac14 A\cos 2\theta\, U_j + \gamma\left[\tfrac{1}{\pi}\theta\mu\delta + (\tfrac{1}{2\pi^2}\theta^2 - \tfrac{1}{24})\delta^2\right] U_j + \mathcal{O}\big(\gamma^2 + A^2\big), \tag{4}$$

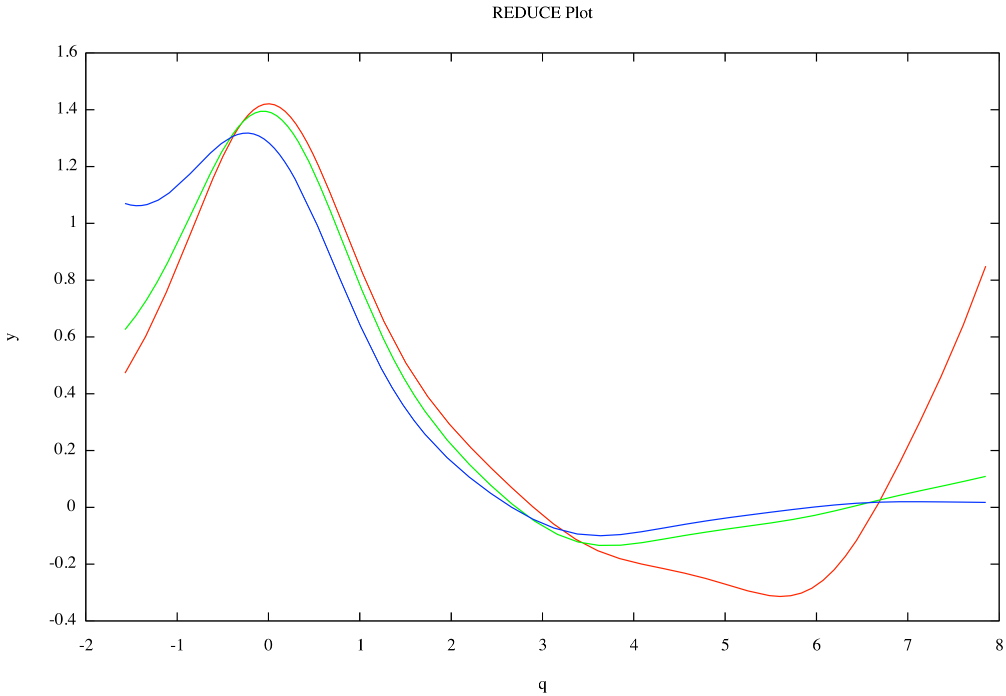

Figure 2: slow manifold subgrid fields for $U_0 = 1$ and all other $U_j = 0$, evaluated at potential strength $A = -1$, for the approximation with errors $\mathcal{O}\big(\gamma^4, A^3, \sigma\big)$: red, $u_0(\theta)$ on $E_0 = [-\frac{3}{2}\pi, \frac{3}{2}\pi]$; green, $u_1(\theta)$ on $E_1 = [-\frac{1}{2}\pi, \frac{5}{2}\pi]$; blue, $u_2(\theta)$ on $E_2 = [\frac{1}{2}\pi, \frac{7}{2}\pi]$.

for centred mean and difference operators, μ and δ. Figure 2 plots an example: there appears to be little penetration through the potential barriers.

The evolution (3) shows two effects:

- the term $\frac{1}{8}A^2U_j$ changes the frequency of solutions on the slow manifold as the potential barriers grow (presumably we could change the potential $V(x)$ so this detuning was zero, if we wish);

- the coefficient $\frac{1}{\pi^2}(1-\frac{1}{4}A^2)$ of the discrete diffusion $\delta^2 U_j$ confirms that increasing potential barriers decrease the communication (tunnelling) between the wells.

We should be able to compare these results to those of Alfimov et al. (2002).

Theory Should have a section on invariant manifold theory that supports the slow manifold as some coarse model of the PDE.

2 Computer algebra construction

The computer algebra code uses Reduce, available freely.[1]

Execute with `in_tex "nlsmod.tex"$`

Improve appearance of printing.

```
on div; on revpri; off allfac;
factor hh,i,nu,aa,sigma,df;
```

The following empowers complex conjugation operator so we just solve one nlS PDE.

```
operator cc;
let { cc(~u*~v)=>cc(u)*cc(v)
    , cc(~u/~v)=>cc(u)/cc(v)
    , cc(~u+~v)=>cc(u)+cc(v)
    , cc(~u^~p)=>cc(u)^p
    , df(cc(~v),~u)=>cc(df(v,u))
    , cc(i)=>-i, cc(-i)=>i
    , cc(~u)=>u when numberp(u)
    , cc(cc(~u))=>u
```

[1] http://www.reduce-algebra.com

```
    , cc(q)=>q
    , cc(cos(~u))=>cos(u)
    , cc(sin(~u))=>sin(u)
    , cc(sign(~u))=>sign(u)
    , cc(nu)=>nu
    , cc(aa)=>aa
    , cc(bb)=>bb
    , cc(sigma)=>sigma
    , cc(gamma)=>gamma
    , cc(pi)=>pi
    , cc(hh)=>hh
    };
```

The following empowers using the sign function to get dependence upon $|\theta|$: it transforms all high powers to just the first or the zeroth; and its derivative is zero upon ignoring the possible delta-function.

```
let { sign(~u)^2=>1
    , df(sign(~u),~v)=>0 };
```

2.1 Subgrid variable

Make subgrid structures a function of element phase $\theta = \pi(x - X_j)/H$ for element size H; denote phase θ by `q`. The the jth element with centre grid point $x = X_j$ has neighbouring grid points at $\theta = \pm\pi$ in the local element coordinate.

```
depend q,x;
let df(q,x)=>pi/hh;
```

2.2 Operators to find updates to approximations

These 'quick and dirty' linear operators are not the best, but they are good enough to achieve the aim of satisfying the PDE and coupling conditions.

Procedure `mean` computes the mean over the jth element, precisely $\texttt{mean}(f) := \frac{1}{H}\int_{-H/2}^{H/2} f\,dx$, equivalently $\frac{1}{\pi}\int_{-\pi/2}^{\pi/2} f\,d\theta$: it finds solvability conditions; and is currently used for the amplitude.

```
operator mean;  linear mean;
let { mean(1,q)=>1
    , mean(q^~~p,q)=>(pi/2)^p*(1+(-1)^p)/2/(p+1)
    , mean(sign(q),q)=>0
    , mean(sign(q)*q^~~p,q)=>(pi/2)^p*(1-(-1)^p)/2/(p+1)
    , mean(cos(~~m*q),q)=>2*sin(m*pi/2)/m/pi
    , mean(sin(~a),q)=>0
    , mean(q^~~p*cos(~~m*q),q)=>(
      +(pi/2)^(p-1)*sin(m*pi/2)*(1+(-1)^p)/2
      -p*mean(q^(p-1)*sin(m*q),q) )/m
    , mean(q^~~p*sin(~~m*q),q)=>(
      -(pi/2)^(p-1)*cos(m*pi/2)*(1-(-1)^p)/2
      +p*mean(q^(p-1)*cos(m*q),q) )/m
    };
```

Procedure `meanr` computes the mean over the $(j+1)$th core of the jth field, precisely $\texttt{meanr}(f) := \frac{1}{H}\int_{H/2}^{3H/2} f\,dx = \frac{1}{\pi}\int_{\pi/2}^{3\pi/2} f\,d\theta$. Correspondingly for `meanl`.

```
operator meanr;  linear meanr;
let { meanr(1,q)=>1
    , meanr(q^~~p,q)=>(pi/2)^p*(3^(p+1)-1)/2/(p+1)
    , meanr(sign(q)*~~a,q)=>meanr(a,q)
    , meanr(cos(~~p*q),q)=>(+sin(3*pi*p/2)-sin(pi*p/2))/p/pi
    , meanr(cos(~~p*q)*q^~~r,q)
      =>(+sin(3*pi*p/2)*3^r-sin(pi*p/2))*(pi/2)^r/p/pi
```

```
      -meanr(sin(p*q)*q^(r-1),q)*r/p
    , meanr(sin(~~p*q),q)=>(-cos(3*pi*p/2)+cos(pi*p/2))/p/pi
    , meanr(sin(~~p*q)*q^~~r,q)
      =>(-cos(3*pi*p/2)*3^r+cos(pi*p/2))*(pi/2)^r/p/pi
      +meanr(cos(p*q)*q^(r-1),q)*r/p
    };
operator meanl;  linear meanl;
let { meanl(1,q)=>1
    , meanl(q^~~p,q)=>(pi/2)^p*(3^(p+1)-1)/2/(p+1)*(-1)^p
    , meanl(sign(q)*~~a,q)=>-meanl(a,q)
    , meanl(cos(~~p*q),q)=>(sin(3*pi*p/2)-sin(pi*p/2))/p/pi
    , meanl(cos(~~p*q)*q^~~r,q)
      =>(+sin(3*pi*p/2)*3^r-sin(pi*p/2))*(-pi/2)^r/p/pi
      -meanl(sin(p*q)*q^(r-1),q)*r/p
    , meanl(sin(~~p*q),q)=>(cos(3*pi*p/2)-cos(pi*p/2))/p/pi
    , meanl(sin(~~p*q)*q^~~r,q)
      =>(cos(3*pi*p/2)*3^r-cos(pi*p/2))*(-pi/2)^r/p/pi
      +meanl(cos(p*q)*q^(r-1),q)*r/p
    };
```

The above were tested via the procedures

```
%procedure test(a); int(a,q,-pi/2,pi/2)/pi-mean(a,q); end;
%procedure test(a); int(a,q,pi/2,3*pi/2)/pi-meanr(a,q); end;
%procedure test(a); int(a,q,-3*pi/2,-pi/2)/pi-meanl(a,q); end;
```

The linear operator `solv` used above solves $\mathcal{L}u = -\frac{\pi^2}{H^2}u_{\theta\theta} = \text{RHS}$ such that $u(0,t) = 0$ and $u(\pi,t) = u(-\pi,t)$.

```
operator solv;  linear solv;
let { solv(q^~~p,q)=>(hh/pi)^2*( -q^(p+2)
        +q*pi^(p+1)*(1-(-1)^p)/2 )/(p+2)/(p+1)
    , solv(1,q)=>(hh/pi)^2*(-q^2)/2
    , solv(sign(q)*q^~~p,q)=>(hh/pi)^2*( -q^(p+2)*sign(q)
        +q*pi^(p+1)*(1+(-1)^p)/2 )/(p+2)/(p+1)
```

```
    , solv(sign(q),q)=>(hh/pi)^2*sign(q)*(-q^2)/2
    , solv(cos(~~m*q),q)=>(cos(m*q)-1)*(hh/pi/m)^2
    , solv(q^~~p*cos(~~m*q),q)=>q^p*(cos(m*q)-1)*(hh/pi/m)^2
    , solv(sin(~~m*q),q)=>sin(m*q)*(hh/pi/m)^2
    , solv(q^~~p*sin(~~m*q),q)=>q^p*sin(m*q)*(hh/pi/m)^2
    };
```

2.3 Initialise the slow manifold

The slow manifold is that the subgrid field depends upon the evolving amplitude $U_j(t) := \frac{1}{H}\int_{-H/2}^{H/2} u_j(x,t)\,dx$. When the total integral of u is conserved, then we expect the total sum of U_j to correspondingly be conserved.

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

The initial subgrid field and evolution is the subspace of piecewise constant fields. This code only generates the slow manifold tangent to this slow subspace: interactions between between multiple modes in the same potential well will need significantly extended code.

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

2.4 Iterate to satisfy nlS PDE and coupling

Iterate in a loop until residuals are zero to specified order of error. The independent small parameters are:

- γ, parametrises the inter-element coupling;
- ν or A, the strength of the potential wells;
- σ, the strength of the nonlinearity.

We will probably link some of these small parameters at sometime.

Set an Euler transform parameter (van Dyke 1964, e.g.), probably should depend upon potential strength, but do not yet know how. For ?? try

```
Eu:=0;
```

```
let { gamma^2=>0, sigma=>0, nu=>0, aa^3=>0 };
for it:=1:99 do begin
```

Compute the residual of the PDE (1) and coupling conditions (2): these drive updates to the approximations. Also trace print the algebraic length of the residuals so we can see how the iteration is proceeding.

```
%write
  potl:=( nu*(1-cos(2*q))/2
         +aa*cos(2*q)
        )*pi^2/hh^2;
%write
  upde:=trigsimp(
    +i*df(uj,t)+df(uj,x,x)-potl*uj-sigma*cc(uj)*uj^2
    ,combine);
  ampj:=mean(uj,q);
%write
  urcc:=(1+Eu/(1-Eu)*gamma)*meanr(uj,q)
    -gamma/(1-Eu)*sub(j=j+1,ampj)
    -(1-gamma)*ampj;
%write
  ulcc:=(1+Eu/(1-Eu)*gamma)*meanl(uj,q)
    -gamma/(1-Eu)*sub(j=j-1,ampj)
    -(1-gamma)*ampj;
```

Use the defined linear operators to update the approximate slow manifold subgrid field and evolution.

```
%write
```

```
  gj:=gj+i*(gd:=mean(upde,q)-(urcc+ulcc)/hh^2);
%write
  uj:=uj+solv(upde-gd,q)+q*(-urcc+ulcc)/2/pi;
```

Fix the amplitude: although better to do this in `solv`, to be flexible we can do it here. This code fixes U_j to be the mean over non-overlapping elements.

```
%write
  uj:=uj-(uamp:=mean(uj,q)-uu(j));
  write lengthResiduals:=map(length(~a)
    ,{upde,urcc,ulcc,uamp});
```

Terminate the iteration when all residuals are zero, to specified error, and print an information number.

```
  showtime;
  if {upde,urcc,ulcc,uamp}={0,0,0,0}
  then write it:=it+100000;
  abortterms:=4000;
  if (foreach j in lengthResiduals sum j)>abortterms
  then rederr({"more than",abortterms,"terms in residuals"});
end;
write gj:=gj;
```

3 Post-processing

3.1 Equivalent differential equation maybe

Determine the equivalent differential equation for amplitudes that vary slowly over the wells.

```
if 1 then begin
let hh^9=>0;
```

```
depend uu,x; depend vv,x;
taylor:={ uu(j)=>uu
    , uu(j+~p)=>uu+(for n:=1:9 sum
               df(uu,x,n)*(hh*p)^n/factorial(n))
    }$
write migde:=(-i*gj where taylor);
end;
```

3.2 Optionally plot subgrid fields

Optionally plot some fully coupled subgrid fields of the linear problem, $\sigma = 0$, for a potential strength $\nu = 3$, say. Or set $A, B \in \{-1, -5, -15\}$ to compare with Alfimov et al. (2002). Set $H = \pi$ so axis scaling works. For expressions with many terms, it would be quicker to output to a file and draw graph in Matlab/Octave/Scilab (I have a bash script that would help edit).

```
load_package gnuplot;
% length less than 999 is enough to plot
write mustbelessthan999:=length(uj);
if length(uj)<999 then begin
  hh:=pi;
  gamma:=1; sigma:=0; nu:=3; aa:=-1;
  uj0:=coeffn(uj,uu(j),1)$
  uj1:=sub(q=q-pi,coeffn(uj,uu(j-1),1))$
  uj2:=sub(q=q-2*pi,coeffn(uj,uu(j-2),1))$
  ujs:=map(max(-1,min(2,~a)),{uj0,uj1,uj2});
  plot(ujs,q=(-pi/2 .. 5*pi/2));
end;
```

3.3 Try to match with Wannier results

What does the interaction look like for specific values? Seems to agree moderately well with first column of Table I of Alfimov et al. (2002), but as yet unclear if the differences will go to zero or not as higher order terns are computed.

```
on rounded; print_precision 4$
gamma:=1; sigma:=0; nu:=3; aa:=-1;
idUdt:=i*gj;
clear gamma; clear aa;
partialsum:=for j:=0:9 sum gamma^j;
hatw01PartialSums:=coeffn(i*gj,uu(j),1)*partialsum;
hatw11PartialSums:=coeffn(i*gj,uu(j+1),1)*partialsum;
```

In these $\hat\omega_{n,\alpha}$: $\mathcal{O}\big(A^2\big)$ coefficients are mostly ??; $\mathcal{O}\big(A^4\big)$ coefficients are ??.

But the convergence of the coefficients in γ appears quite slow, the terms decay maybe like $(??)^n\gamma^n$. Suggest may be a convergence limiting singularity at $\gamma \approx ??$. Could try an Euler transform, $\gamma = \gamma'/(1 - E + E\gamma')$ equivalently $\gamma' = (1 - E)\gamma/(1 - E\gamma)$ for say $E = \frac{2}{3}$ or a bit more conservatively $E = \frac{1}{2}$.

Fin.

```
end;
```

Acknowledgement thanks to CSU and AMSI.

References

Alfimov, G. L., Kevrekidis, P. G., Konotop, V. V. & Salerno, M. (2002), 'Wannier functions analysis of the nonlinear Schrödinger equation with a periodic potential', *Phys. Rev. E* **66**(046608), 1–6.

van Dyke, M. (1964), 'Higher approximations in boundary-layer theory. Part 3. parabola in uniform stream', *J. Fluid Mech.* **19**, 145–159.