Discretizing Schrodinger Type Operators with Spectral Accuracy on Quantum Graphs

Gracie Conte

University of North Carolina, Chapel Hill
March 2, 2020

Quantum Mechanics 101

The Schrödinger equation

$$iu_t = -u_{xx} + f(x)$$

where $f(x)$ could be:

- Potential energy term: $V(x)u$
- Interaction between particles: $|u|^2 u$

Quantum Mechanics 101

The Schrödinger equation

$$iu_t = -u_{xx} + f(x)$$

where $f(x)$ could be:

- Potential energy term: $V(x)u$
- Interaction between particles: $|u|^2u$

It is useful for modeling waves in thin branching structures

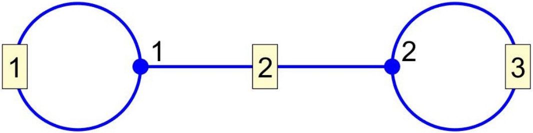
- Qubits in a quantum circuit
- Free electrons orbiting organic molecules
- Electromagnetic waves propagating through dielectric tubes

My Problem

Find time-periodic solutions to

$$iu_t = -u_{xx} - |u|^2 u$$

on



While I'm at it:

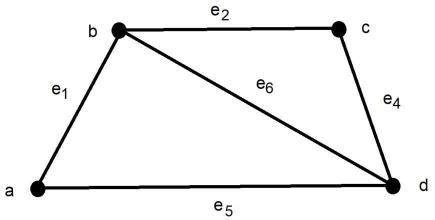
Solve $iu_t = -u_{xx} - |u|^2 u$ on *any* graphs with machine precision

Graphs

- A **Graph**, G, is a pair (V, E) where:
  - $V =$ set of vertices v_j
  - $E =$ set of edges e_j

Graphs

- A **Graph**, G, is a pair (V, E) where:
  - V = set of vertices v_j
  - E = set of edges e_j
- Example
  - $V = \{a, b, c, d\}$
  - $E = \{(a, b), (b, c), (c, d), (b, d)\}$



Graphs

- A **Graph**, G, is a pair (V, E) where:
  - $V =$ set of vertices v_j
  - $E =$ set of edges e_j
- A **Metric Graph** has the additional condition:
  - each edge has a length $l_j \in (0, \infty)$
- A **Quantum Graph** is:
  - a metric graph
  - has a Schrödinger type operator on each edge

Graphs

- A **Graph**, G, is a pair (V, E) where:
  - $V =$ set of vertices v_j
  - $E =$ set of edges e_j
- A **Metric Graph** has the additional condition:
  - each edge has a length $l_j \in (0, \infty)$
- A **Quantum Graph** is:
  - a metric graph
  - has a Schrödinger type operator on each edge
    - Schrödinger's Equation: $iu_t = -u_{xx} + f(x)$

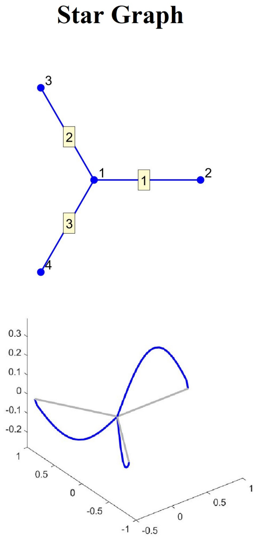
Graphs

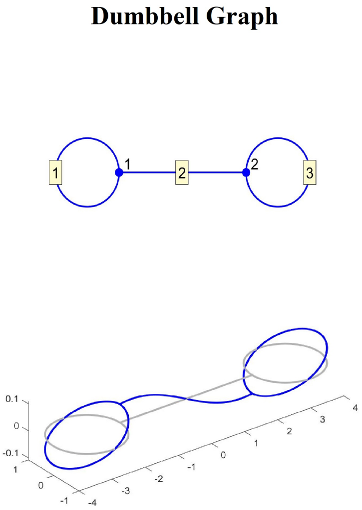
- A **Graph**, G, is a pair (V, E) where:
  - V = set of vertices v_j
  - E = set of edges e_j
- A **Metric Graph** has the additional condition:
  - each edge has a length $l_j \in (0, \infty)$
- A **Quantum Graph** is:
  - a metric graph
  - has a Schrödinger type operator on each edge
    - Schrödinger's Equation: $iu_t = -u_{xx} + f(x)$

↑ Schrödinger type operator

Example: Solutions on Graphs

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL





Vertex Conditions

THE UNIVERSITY *of* NORTH CAROLINA *at* CHAPEL HILL

Possible conditions at a vertex v:

1) Leaf Nodes (Incident to **exactly one** edge)

- Boundary Condition
  - Dirichlet: $u_j(v) = 0$
  - Neumann: $u_j'(v) = 0$
  - Robin: $\alpha_j u_j(v) + u_j'(v) = 0$

Vertex Conditions

Possible conditions at a vertex v:

1) Leaf Nodes (Incident to **exactly one** edge)

- Boundary Condition
  - Dirichlet: $u_j(v) = 0$
  - Neumann: $u_j'(v) = 0$
  - Robin: $\alpha_j u_j(v) + u_j'(v) = 0$

2) Internal Nodes (Incident to **more than one** edge)

- Matching Conditions
  - Continuity Condition: $u_j(v) = u_k(v)$
  - Current Conservation: $u_j'(v) = u_k'(v)$
    - Kirchoff: $\sum_{j=1}^{d_v} u_j'(v) = \sigma u_1(v)$

Numerically Defining Operators: Line

Problem: Solve for u when $x \in [0, \ell]$ in:

$$u_{xx} = f(x), \quad u(0) = a, \ u(\ell) = b$$

Discretized Problem: $D^2\vec{u} = \vec{f}$ where D is the discretized version of $\frac{d}{dx}$

We know D^2 and $\vec{f}$ so we can solve for $\vec{u}$.

Numerically Defining Operators: Line

Problem: Solve for u when $x \in [0, \ell]$ in:

$$u_{xx} = f(x), \quad u(0) = a, \ u(\ell) = b$$

Discretized Problem: $D^2 \vec{u} = \vec{f}$ where D is the discretized version of $\frac{d}{dx}$

We know D^2 and $\vec{f}$ so we can solve for $\vec{u}$.

But how does do we enforce the boundary conditions?

Numerically Defining Operators: BCs

Popular method

- Row replacement
- Linear convergence

Numerically Defining Operators: BCs

Popular method

- Row replacement
- Linear convergence

Our Method

- Project information from n points to $n-2$ points
  - Method name: Rectangular Collocation
- Spectral convergence
  - $e_n \sim (\frac{\ell}{n})^n$

Rectangular Collocation

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

1. Start with:
   - n discretization points that we are currently evaluating at $\{x_k\}_{k=1}^{n}$
   - $n-2$ discretization points we'd like to be working on instead $\{y_k\}_{k=1}^{n-2}$

Rectangular Collocation

1. Start with:
   - n discretization points that we are currently evaluating at $\{x_k\}_{k=1}^{n}$
   - $n-2$ discretization points we'd like to be working on instead $\{y_k\}_{k=1}^{n-2}$
2. Create an interpolating polynomial for the n points

Rectangular Collocation

1. Start with:
   - n discretization points that we are currently evaluating at $\{x_k\}_{k=1}^{n}$
   - $n-2$ discretization points we'd like to be working on instead $\{y_k\}_{k=1}^{n-2}$
2. Create an interpolating polynomial for the n points

   Given discretization points $\{x_k\}_{k=1}^{n}$ the barycentric weights are:

$$w_k = \prod_{\substack{l=1 \\ l \neq k}}^{n} (x_k - x_l)^{-1} \qquad k = 1, ..., n$$

   The unique polynomial interpolating $\{(x_j, f_j)\}_{j=1}^{n}$ is:

$$p_{n-1}(y) = \frac{\sum_{k=1}^{n} (w_k/(y - x_k)) f_k}{\sum_{l=1}^{n} (w_k/(y - x_l))}$$

Rectangular Collocation

1. Start with:
   - n discretization points that we are currently evaluating at $\{x_k\}_{k=1}^{n}$
   - $n-2$ discretization points we'd like to be working on instead $\{y_k\}_{k=1}^{n-2}$
2. Create an interpolating polynomial for the n points
3. Evaluates that polynomial at new $n-2$ discretization points

Rectangular Collocation

1. Start with:
   - n discretization points that we are currently evaluating at $\{x_k\}_{k=1}^{n}$
   - $n-2$ discretization points we'd like to be working on instead $\{y_k\}_{k=1}^{n-2}$
2. Create an interpolating polynomial for the n points
3. Evaluates that polynomial at new $n-2$ discretization points

$$p_{n-1}(\boldsymbol{y}) = P\ p_{n-1}(\boldsymbol{x})$$

Rectangular Collocation

1. Start with:
   - n discretization points that we are currently evaluating at $\{x_k\}_{k=1}^{n}$
   - $n-2$ discretization points we'd like to be working on instead $\{y_k\}_{k=1}^{n-2}$
2. Create an interpolating polynomial for the n points
3. Evaluates that polynomial at new $n-2$ discretization points

$$p_{n-1}(\boldsymbol{y}) = P\ p_{n-1}(\boldsymbol{x})$$

$$\boldsymbol{w}_{n-2,1} = P_{n-2,n}\ \boldsymbol{v}_{n,1}$$

Rectangular Collocation

1. Start with:
   - n discretization points that we are currently evaluating at $\{x_k\}_{k=1}^{n}$
   - $n-2$ discretization points we'd like to be working on instead $\{y_k\}_{k=1}^{n-2}$
2. Create an interpolating polynomial for the n points
3. Evaluates that polynomial at new $n-2$ discretization points
4. Use vector multiplication to define the Barycentric Resampling Matrix: P

Rectangular Collocation

1. Start with:
   - n discretization points that we are currently evaluating at $\{x_k\}_{k=1}^{n}$
   - $n-2$ discretization points we'd like to be working on instead $\{y_k\}_{k=1}^{n-2}$
2. Create an interpolating polynomial for the n points
3. Evaluates that polynomial at new $n-2$ discretization points
4. Use vector multiplication to define the Barycentric Resampling Matrix: P

$$(P_{n-2,n})_{j,k} = \begin{cases} \frac{w_k}{y_j - x_k}\left(\sum_{l=1}^{N} \frac{w_l}{y_j - x_l}\right)^{-1} & y_j \neq x_k \\ 1 & y_j = x_k \end{cases}$$

Rectangular Collocation

1. Start with:
   - n discretization points that we are currently evaluating at $\{x_k\}_{k=1}^{n}$
   - $n-2$ discretization points we'd like to be working on instead $\{y_k\}_{k=1}^{n-2}$
2. Create an interpolating polynomial for the n points
3. Evaluates that polynomial at new $n-2$ discretization points
4. Use vector multiplication to define the Barycentric Resampling Matrix: P
5. Use P to create rectangular differentiation matrices

Rectangular Collocation

1. Start with:
   - n discretization points that we are currently evaluating at $\{x_k\}_{k=1}^{n}$
   - $n-2$ discretization points we'd like to be working on instead $\{y_k\}_{k=1}^{n-2}$
2. Create an interpolating polynomial for the n points
3. Evaluates that polynomial at new $n-2$ discretization points
4. Use vector multiplication to define the Barycentric Resampling Matrix: P
5. Use P to create rectangular differentiation matrices

$$P_{n-2,n}D^2_{n,n} = \text{Projected Second Derivative Matrix}$$

Numerically Defining Operators: Line

Problem: Solve for u when $x \in [0, \ell]$ in:

$$u_{xx} = f(x), \quad u(0) = a, \ \ u(\ell) = b$$

Discretized: $D^2\vec{u} = \vec{f}$ where D is the discretized version of $\frac{d}{dx}$

(*Still need to enforce the boundary conditions*)

Numerically Defining Operators: Line

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

$$\underbrace{\begin{bmatrix} 1 & \dots & 0 \\ & \begin{bmatrix} & & \\ & PD^2 & \\ & & \end{bmatrix} & \\ 0 & \dots & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix}}_{\vec{u}} = \underbrace{\begin{bmatrix} a \\ f_2 \\ \vdots \\ f_{n-1} \\ b \end{bmatrix}}_{\vec{f}}$$

Use some built in commands and solve $L\vec{u} = \vec{f}$ for $\vec{u}$

Numerically Defining Operators: Line

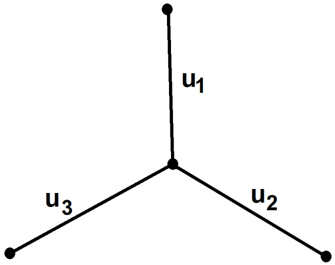
$$\underbrace{\begin{bmatrix} 1 & \dots & 0 \\ & \begin{bmatrix} & & \\ & PD^2 & \\ & & \end{bmatrix} & \\ 0 & \dots & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix}}_{\vec{u}} = \underbrace{\begin{bmatrix} a \\ f_2 \\ \vdots \\ f_{n-1} \\ b \end{bmatrix}}_{\vec{f}}$$

Use some built in commands and solve $L\vec{u} = \vec{f}$ for $\vec{u}$

Now glue some lines together and you have a quantum graph!

Numerically Defining Operators: Graph

Problem: Solve $u_{xx} = f(x)$ when x is in:



$$\begin{cases} u_1(l_1) = u_2(l_2) = u_3(l_3) = 0 & \text{Boundary Condition} \\ u_1(0) = u_2(0) = u_3(0) & \text{Continuity Condition} \\ u_1'(0) + u_2'(0) + u_3'(0) = 0 & \text{Current Conservation (Kirchoff Condition)} \end{cases}$$

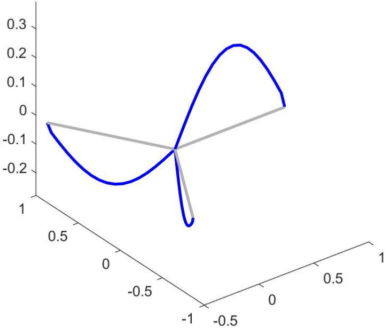
Numerically Defining Operators: Graphs

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

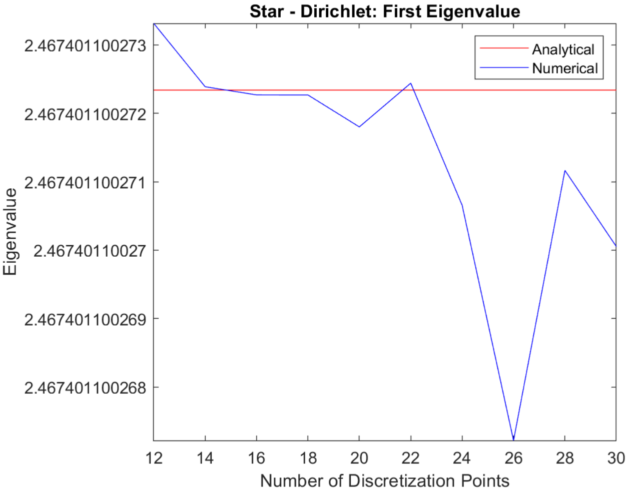
$$L = \begin{bmatrix} \begin{bmatrix} PD^2 & \dots & 0 \\ \vdots & PD^2 & \vdots \\ 0 & \dots & PD^2 \end{bmatrix} \\ \begin{bmatrix} & BC & \end{bmatrix} \\ \begin{bmatrix} & Continuity & \end{bmatrix} \\ \begin{bmatrix} & KC & \end{bmatrix} \end{bmatrix}$$

Numerically Defining Operators: Graphs

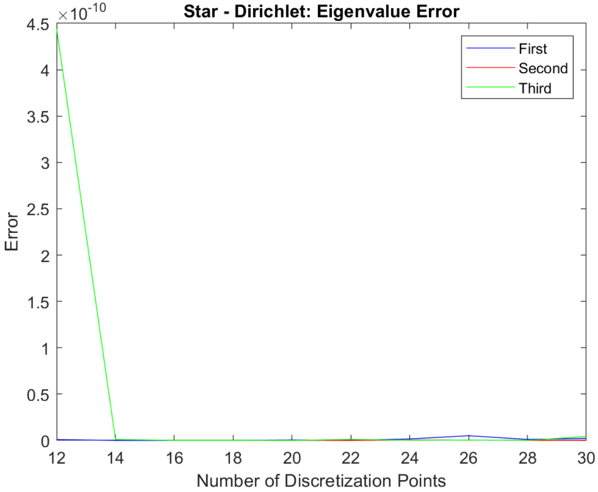
Use built in commands to solve $L\vec{u} = \vec{f}$:



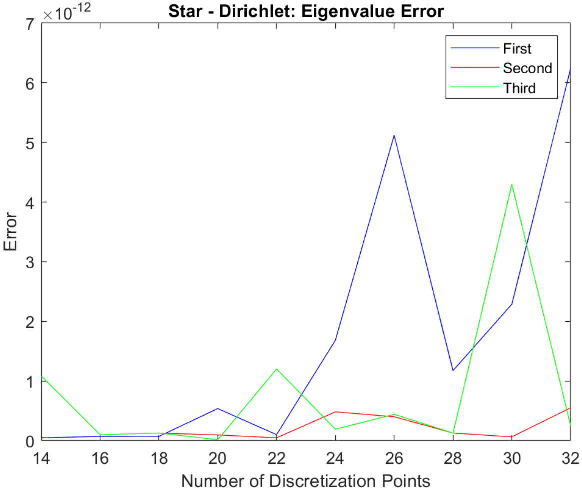
Convergence of Spatial Operator



Convergence of Spatial Operator



Convergence of Spatial Operator



Time Evolution

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

So you want you want to see what happens when time doesn't stand still? How interesting...

Challenges:

- Finding a time-stepper that matches the accuracy of our spatial solver
- Accounting for our spatial solver being on a new domain
- Coping with the non-linearity

Time Evolution: Linear

Step One: Ignore the non-linearity

We know how to solve $iu_t = -u_{xx}$ analytically
(Spoiler: Its solution is an exponential)

Time Evolution: Linear

THE UNIVERSITY *of* NORTH CAROLINA *at* CHAPEL HILL

The Problem:

$$\begin{cases} iu_t = -u_{xx} \\ \text{vertex conditions} \\ u(x,0) = f(x) \end{cases}$$

The Discretized Problem:

$$\begin{cases} \dfrac{d\boldsymbol{u}}{dt} = -iD^2\boldsymbol{u} \\ B\boldsymbol{u} = 0 \\ u(\boldsymbol{x},0) = f(\boldsymbol{x}) \end{cases}$$

this means $\dfrac{d\boldsymbol{u}}{dt} \neq -i \underbrace{\begin{bmatrix} [& PD^2 &] \\ [& B &] \end{bmatrix}}_{L} \boldsymbol{u}$

Time Evolution: Linear

We need to project our solution

$$\widetilde{\boldsymbol{u}} = P_{M,N}\boldsymbol{u}$$

(Note: $P_{M,N} =$ zero matrix with $P_{n-2,n}$'s on its diagonal)

Can recover original solution using this:

$$\begin{bmatrix} P_{M,N} \\ L \end{bmatrix} \boldsymbol{u} = \begin{bmatrix} I_M \\ 0 \end{bmatrix} \widetilde{\boldsymbol{u}}$$

$$\Rightarrow \quad \boldsymbol{u} = \underbrace{\begin{bmatrix} P_{M,N} \\ L \end{bmatrix} \begin{bmatrix} I_M \\ 0 \end{bmatrix}}_{E} \widetilde{\boldsymbol{u}}$$

Time Evolution: Linear

Apply $P_{M,N}$ to both sides of: $\dfrac{d\boldsymbol{u}}{dt} = -iD^2\boldsymbol{u}$

$$\frac{d\widetilde{\boldsymbol{u}}}{dt} = -iP_{M,N}D^2E\widetilde{\boldsymbol{u}}$$

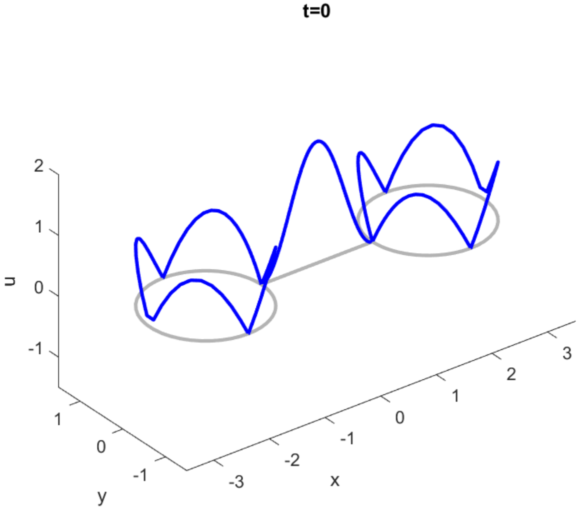
Use analytic knowledge and matrix exponentials to get solution:

$$\widetilde{\boldsymbol{u}} = \exp\left(-itP_{N-mj,N}D^2E\right)\widetilde{\boldsymbol{u}}(x,0)$$

Recover the actual solution:

$$\boldsymbol{u} = E\exp\left(-itP_{M,N}D^2E\right)P_{M,N}\boldsymbol{u}(x,0)$$

Time Evolution: $iu_t = -u_{xx}$



Time Evolution: Nonlinear

Step Two: Admit you have a problem

$$iu_t = -u_{xx} - |u|^2 u$$

Challenges:

- Most well-developed non-linear schemes are only fourth order
- The better schemes haven't been adjusted for DAEs

Time Evolution: Strang Splitting

Rewrite: $u_t = \mathcal{L}u + N(u,t)$

Solve separate problems

$$d_t \boldsymbol{v} = \mathcal{L}\boldsymbol{v} \qquad d_t \boldsymbol{w} = N\boldsymbol{w}$$

$$\boldsymbol{v}_n = e^{\mathcal{L}t}\boldsymbol{u}_{n-1} \qquad \boldsymbol{w}_n = F(Nt)\boldsymbol{u}_{n-1}$$

Second Order Scheme:

$$\boldsymbol{u}_n = e^{\mathcal{L}\frac{\Delta t}{2}} F(N\Delta t) e^{\mathcal{L}\frac{\Delta t}{2}} \boldsymbol{u}_{n-1}$$

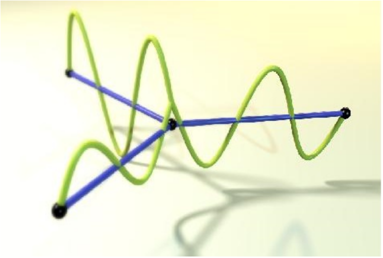
General Scheme:

$$\boldsymbol{u}_n = e^{c_1 \Delta t \mathcal{L}} F(d_1 t N) e^{c_2 \Delta t \mathcal{L}} F(d_2 \Delta t N) \ldots e^{c_k \Delta t L} F(d_k \Delta t N) \boldsymbol{u}_{n-1}$$

where c_i's and d_i's represent fractional time steps

Conclusion

- Developing tools to model Quantum Graphs is essential
- Rectangular Collocation is a superior method for solving PDE's with Schrödinger type operators
- Computationally efficient time stepping scheme come in two pieces
  - Use matrix exponentials for linear term
  - Account for non-linear term with a splitting scheme (I'll get on it)
- Will be ready to look for time periodic orbit solutions soon!



Thanks!

Special thanks to

- My advisor, Jeremy Marzuola
- My collaborator at NJIT, Roy Goodman
- And you guys for coming