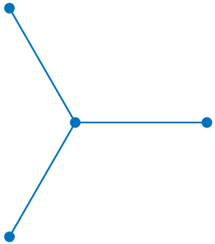
Numerical Methods for Solving Partial Differential Equations on Quantum Graphs

Gracie Conte

University of North Carolina, Chapel Hill
March 21, 2022

My Dissertation Problem

Goal: Solve the nonlinear Schrödinger equation on graphs such as:



My work: Develop a software package that

- has a spatial solver
- evolves the solution forward in time
- finds time-periodic solutions
- creates helpful visuals of the solution

The Schrödinger Equation

$$i\frac{\partial u}{\partial t} = -\frac{\partial^2 u}{\partial x^2} + f(x)$$

$f(x)$ could be:

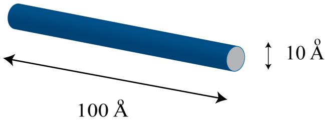
- Potential energy term: $V(x)u$
- Interaction between particles: $|u|^2 u$

Quantum Mechanics 101:

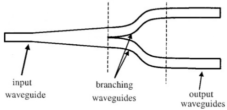
- Developed because light has wave / particle duality
- Calculates the wave function of a quantum mechanical system
- $|u|^2$ tells us the *probability* that a quantum object is at a given location

Real World Scenarios

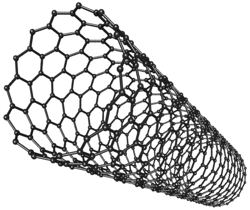
Quantum Wires



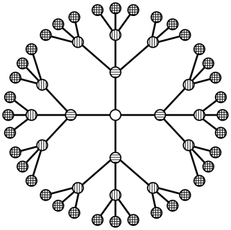
Branching Waveguides



Carbon Nano-structures



Photonic Crystals



Graphs

A **graph** consists of *vertices* and *edges* that connect pairs of vertices.

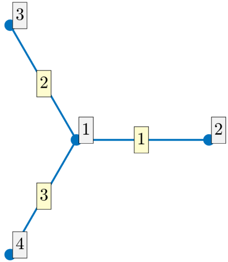
A **metric graph** has an additional condition:

- each edge has a length $0 < \ell_j < \infty$

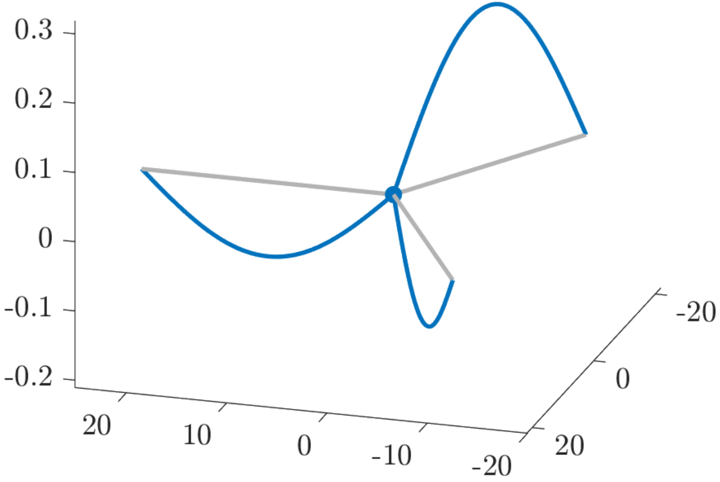
A **quantum graph** has one more condition:

- a Schrödinger-type operator is defined on each edge
  - The Schrödinger equation: $i\frac{\partial u}{\partial t} = -\frac{\partial^2 u}{\partial x^2} + f(x)$

↑ Schrödinger-type operator



Graphs: Solution Example



Graphs: Vertex Conditions

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Possible conditions at a vertex:

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: $u_j(v) = u_k(v)$
     - Kirchhoff: $\left(\sum_{j=1}^{d_v} u_j'(v)\right) + \beta u_1(v) = 0$

Numerically Defining Operators: Discretizations

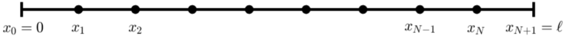
Goal: Find a numerical representation of $f(x)$

Pick points to evaluate $f(x)$ at: $\boldsymbol{x} = [x_0 \;\; x_1 \;\; \dots \;\; x_{N+1}]^T$

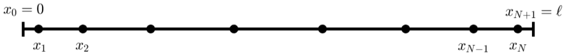
Numerical f is: $\boldsymbol{f} = [f_0 \;\; f_1 \;\; \dots \;\; f_{N+1}]^T$ where $f_j = f(x_j)$

Common choices for $\boldsymbol{x}$:

Uniform:



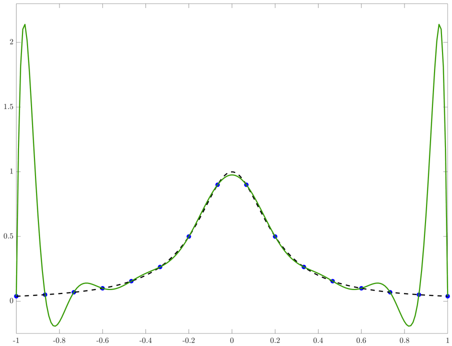
Chebyshev:



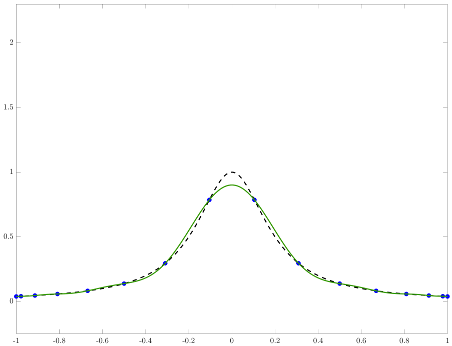
Numerically Defining Operators: Interpolation

Goal: Find a polynomial to approximate $f(x)$ using discretization points $\{x_j\}_{j=0}^{N+1}$

Idea: $f(x) \approx p(x) = \sum_{j=0}^{N+1} f_j l_j(x)$



(a) Uniform points



(b) Chebyshev points

Numerically Defining Operators: The Derivative Matrix

Goal: Define $\frac{df}{dx}$ numerically without calculating $f'(x)$ by hand

Evaluate $f(x)$ at discrete points: $\{(x_j, f_j)\}_{j=0}^{N+1}$

Develop an interpolating polynomial: $f(x) \approx p(x) = \sum_{j=0}^{N+1} f_j l_j(x)$

Approximate the derivative: $f'(x) \approx p'(x) = \sum_{j=0}^{N+1} f_j l_j'(x)$

Therefore $f'(\boldsymbol{x})$ can be estimated numerically by $\boldsymbol{Df}$ where $D_{ij} = l_j'(x_i)$

Numerically Defining Operators: Line Example

Goal: Solve for u in

$$\begin{cases} \dfrac{d^2u}{dx^2} = f(x), \quad x \in (0,\ell) \\ u(0) = a \\ u(\ell) = b \end{cases}$$

Discretized problem:

$$\begin{cases} \boldsymbol{D}^2\boldsymbol{u} = \boldsymbol{f} \\ u_0 = a \\ u_{N+1} = b \end{cases}$$

where $\boldsymbol{D}$ is the discretized version of $\frac{d}{dx}$

But what about the boundary conditions?

Numerically Defining Operators: Boundary Conditions

Popular method: Row Replacement

- Remove top and bottom rows and replace with BC's

  $\Rightarrow$ Linear to quadratic convergence

Better method: Rectangular Collocation

- Create an interpolating polynomial, $p(x)$, for $\{x_j\}_{j=0}^{N+1}$
- Evaluate $p(x)$ at desired points, $\{\chi_j\}_{j=1}^{N}$
- Solve $p(\boldsymbol{\chi}) = \boldsymbol{P}\, p(\boldsymbol{x})$ for $\boldsymbol{P}$

  $\Rightarrow$ Spectral convergence: $e_N \sim (\frac{\ell}{N})^N$

(Driscoll and Hale 2016)

Numerically Defining Operators: Line

Discretized problem with incorporated boundary conditions:

$$\begin{cases} \boldsymbol{P}\boldsymbol{D}^2\boldsymbol{u} = \boldsymbol{P}\boldsymbol{f} \\ u_0 = a \\ u_{N+1} = b \end{cases}$$

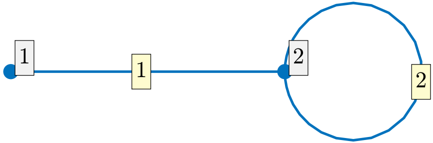
Matrix form:

$$\underbrace{\begin{bmatrix} \begin{bmatrix} & \boldsymbol{P}\boldsymbol{D}^2 & \end{bmatrix} \\ 1 \quad \dots \quad 0 \\ 0 \quad \dots \quad 1 \end{bmatrix}}_{\boldsymbol{L}} \underbrace{\begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_N \\ u_{N+1} \end{bmatrix}}_{\boldsymbol{u}} = \underbrace{\begin{bmatrix} \begin{bmatrix} \boldsymbol{P}\boldsymbol{f} \end{bmatrix} \\ a \\ b \end{bmatrix}}_{\text{`}\boldsymbol{f}\text{' and BCs}}$$

Use built-in commands and solve $\boldsymbol{L}\boldsymbol{u} = \boldsymbol{f}$ for $\boldsymbol{u}$

Numerically Defining Operators: Graph Example

Problem: Solve $\dfrac{d^2u}{dx^2} = f(x)$ when x is in:



given the boundary conditions:

$$\begin{cases} u_1'(0) = 0 & \text{Neumann condition} \\ u_1(0) = u_2(0) = u_2(l_2) & \text{Continuity condition} \\ u_1'(l_1) - u_2'(0) + u_2'(l_2) = 0 & \text{Kirchhoff condition} \end{cases}$$

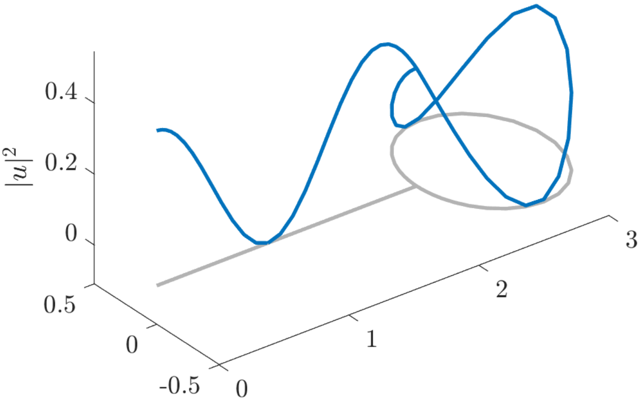
Numerically Defining Operators: Graph Example

Matrix form of discretized problem:

$$\underbrace{\begin{bmatrix} \begin{bmatrix} \boldsymbol{PD}^2 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{PD}^2 \end{bmatrix} \\ [\quad \text{Neumann} \quad] \\ [\quad \text{Continuity} \quad] \\ [\quad \text{Kirchhoff} \quad] \end{bmatrix}}_{\boldsymbol{L}} \underbrace{\begin{bmatrix} u_{1,0} \\ u_{1,1} \\ \vdots \\ u_{1,N+1} \\ u_{2,0} \\ u_{2,1} \\ \vdots \\ u_{2,N+1} \end{bmatrix}}_{\boldsymbol{u}} = \underbrace{\begin{bmatrix} \begin{bmatrix} \boldsymbol{Pf}_1 \\ \boldsymbol{Pf}_2 \end{bmatrix} \\ \begin{bmatrix} \\ \mathbf{0} \\ \\ \end{bmatrix} \end{bmatrix}}_{\text{`}\boldsymbol{f}\text{' and BC's}}$$

Numerically Defining Operators: Graph Example

Use built in commands to solve $\boldsymbol{Lu} = \boldsymbol{f}$:



Numerically Defining Operators: Accuracy

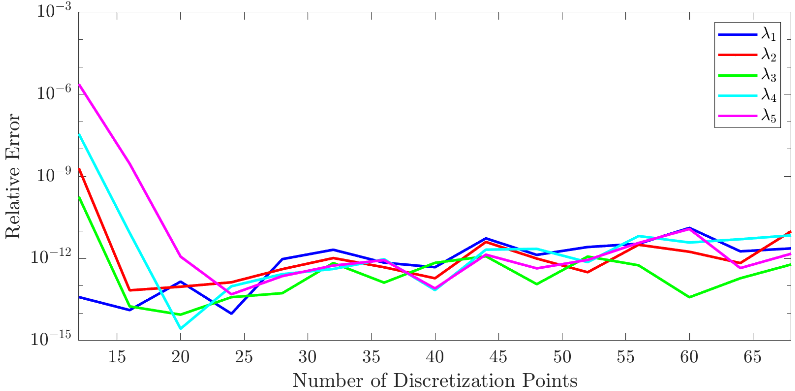


Figure: Accuracy of Eigenvaules for $\boldsymbol{D}^2$

Time Evolution

Goal: Solve for u in

$$\begin{cases} \dfrac{\partial u}{\partial t} = F(t,u), & t > 0 \\ l(u,u') = 0 & \text{(Boundary conditions)} \\ u(x,0) = f(x) & \end{cases}$$

on some graph.

Discretized problem:

$$\begin{pmatrix} \boldsymbol{P} \\ \boldsymbol{0} \end{pmatrix} \frac{d\boldsymbol{u}}{dt} = \begin{pmatrix} \boldsymbol{F}(t,\boldsymbol{u}) \\ \boldsymbol{L}_{BC}\boldsymbol{u} \end{pmatrix}.$$

Challenges:

- Finding a time-stepper that matches the accuracy of our spatial solver
- Coping with non-linear terms
- Doing it in a way that works for operators defined on graphs

Time Evolution: Traditional Runge-Kutta

Traditional discretized problem: $\frac{d\boldsymbol{y}}{dt} = G(t, \boldsymbol{y})$ No boundary conditions!

Scheme: $\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + h\sum_{i=1}^{4} b_i \boldsymbol{k}_i$

where

$$\begin{aligned}
\boldsymbol{k}_1 &= \boldsymbol{G}(t_n, \boldsymbol{y}_n) \\
\boldsymbol{k}_2 &= \boldsymbol{G}(t_n + c_2 h, \boldsymbol{y}_n + h a_{21}\boldsymbol{k}_1) \\
\boldsymbol{k}_3 &= \boldsymbol{G}(t_n + c_3 h, \boldsymbol{y}_n + h(a_{31}\boldsymbol{k}_1 + a_{32}\boldsymbol{k}_2)) \\
\boldsymbol{k}_4 &= \boldsymbol{G}(t_n + c_s h, \boldsymbol{y}_n + h(a_{41}\boldsymbol{k}_1 + a_{42}\boldsymbol{k}_2 + a_{43}\boldsymbol{k}_3)
\end{aligned}$$

Time Evolution: Adapted Runge-Kutta (a valiant effort)

Our discretized problem: $\underbrace{\begin{pmatrix} \boldsymbol{P} \\ \boldsymbol{0} \end{pmatrix}}_{\boldsymbol{B}} \frac{d\boldsymbol{u}}{dt} = \begin{pmatrix} \boldsymbol{F}(t, \boldsymbol{u}) \\ \boldsymbol{L}_{BC}\boldsymbol{u} \end{pmatrix}$

Scheme: $\begin{pmatrix} ? \end{pmatrix} \boldsymbol{u}_{n+1} = \boldsymbol{B}\boldsymbol{u}_n + h\sum_{i=1}^{s} b_i \boldsymbol{k}_i$

where

$$\boldsymbol{k}_1 = \begin{pmatrix} \boldsymbol{F}(t_n, \boldsymbol{u}_n) \\ \boldsymbol{0} \end{pmatrix}$$

$$\boldsymbol{k}_2 = \begin{pmatrix} \boldsymbol{F}(t_n + c_2 h, \boldsymbol{u}_n + h a_{21}\boldsymbol{k}_1) \\ \boldsymbol{0} \end{pmatrix}$$

$$\boldsymbol{k}_3 = \begin{pmatrix} \boldsymbol{F}(t_n + c_3 h, \boldsymbol{u}_n + h(a_{31}\boldsymbol{k}_1 + a_{32}\boldsymbol{k}_2)) \\ \boldsymbol{0} \end{pmatrix}$$

$$\boldsymbol{k}_4 = \begin{pmatrix} \boldsymbol{F}(t_n + c_s h, \boldsymbol{u}_n + h(a_{41}\boldsymbol{k}_1 + a_{42}\boldsymbol{k}_2 + a_{43}\boldsymbol{k}_3)) \\ \boldsymbol{0} \end{pmatrix}$$

Time Evolution: Adapted Runge-Kutta

Our discretized problem: $\underbrace{\begin{pmatrix} \boldsymbol{P} \\ \boldsymbol{0} \end{pmatrix}}_{\boldsymbol{B}} \frac{d\boldsymbol{u}}{dt} = \begin{pmatrix} \boldsymbol{F}(t, \boldsymbol{u}) \\ \boldsymbol{L}_{BC}\boldsymbol{u} \end{pmatrix}$

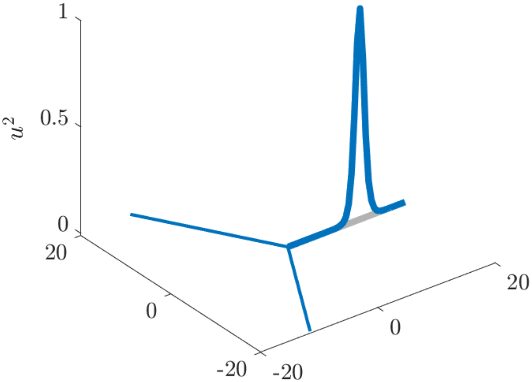
Scheme: $\boldsymbol{C}\boldsymbol{u}_{n+1} = \boldsymbol{B}\boldsymbol{u}_n + h\sum_{i=1}^{s} b_i \boldsymbol{k}_i$

where

$$\begin{aligned}
\boldsymbol{k}_1 &= \begin{pmatrix} \boldsymbol{F}(t_n, \boldsymbol{u}_n) \\ \boldsymbol{0} \end{pmatrix} \\
\boldsymbol{k}_2 &= \begin{pmatrix} \boldsymbol{F}(t_n + c_2 h, \boldsymbol{u}_n + h a_{21}\boldsymbol{\kappa}_1) \\ \boldsymbol{0} \end{pmatrix} \\
\boldsymbol{k}_3 &= \begin{pmatrix} \boldsymbol{F}(t_n + c_3 h, \boldsymbol{u}_n + h(a_{31}\boldsymbol{\kappa}_1 + a_{32}\boldsymbol{\kappa}_2)) \\ \boldsymbol{0} \end{pmatrix} \\
\boldsymbol{k}_4 &= \begin{pmatrix} \boldsymbol{F}(t_n + c_s h, \boldsymbol{u}_n + h(a_{41}\boldsymbol{\kappa}_1 + a_{42}\boldsymbol{\kappa}_2 + a_{43}\boldsymbol{\kappa}_3)) \\ \boldsymbol{0} \end{pmatrix},
\end{aligned}$$

$\boldsymbol{C} = \begin{pmatrix} \boldsymbol{P} \\ \boldsymbol{L}_{BC} \end{pmatrix}$, and $\boldsymbol{C}\boldsymbol{\kappa}_i = \boldsymbol{k}$

Time Evolution: The Action!



Accuracy:

Conservation of Energy: 10^{-09}

Conservation of Mass: 10^{-09}

Conservation of Momentum: 10^{-07}

Time-Periodic Solutions

Objective: Study the dynamics of differential equations

Bifurcation: A dramatic change in the solution's behavior when a parameter makes a small change

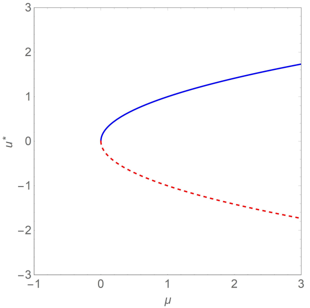
Local: Changes in the stability of an equilibrium

Global: Periodic solutions appear and disappear

Example: $\dfrac{du}{dt} = \mu - u^2$

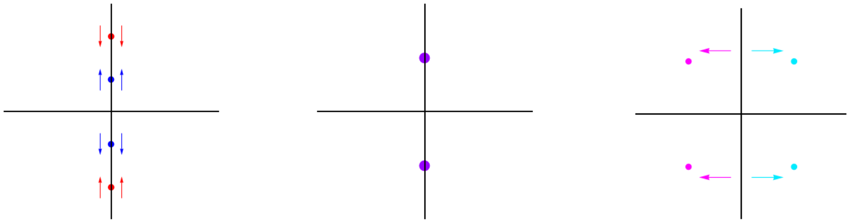
Three cases:

- $\mu < 0$
- $\mu = 0$
- $\mu > 0$

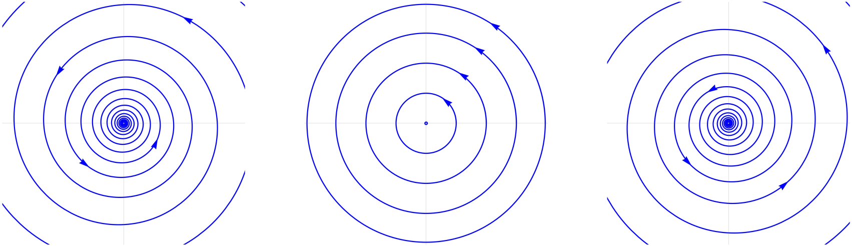


Time-Periodic Solutions: Hamiltonian-Hopf bifurcations

Hamilton-Hopf bifurcations occur when two eigenvalues collide on the imaginary axis then split into the complex plane.



Time-Periodic Solutions: Hamiltonian-Hopf bifurcations



(a) Stable (b) Periodic (c) Unstable

Figure: Hamilton-Hopf bifurcation - solution visual

Time-Periodic Solutions: The Schrödinger equation

NLS equation:

$$i\frac{\partial u}{\partial t} = -\frac{\partial^2 u}{\partial^2 x} + |u|^2 u$$

has stationary state solution:

$$u(x,t) = e^{i\mu t}\psi(x)$$

where $\psi(x)$ solves:

$$\psi''(x) + \psi^3(x) = \mu\psi(x)$$

↑ Bifurcation parameter

Time-Periodic Solutions: The Schrödinger equation

Normal-mode perturbation:

$$u(x,t) = e^{i\mu t}\left[\psi(x) + f_1(x)e^{\lambda t} + \bar{f}_2(x)e^{\bar{\lambda} t}\right], \qquad f_1, f_2 \ll 1$$

Need f_1 and f_2 to satisfy the eigenvalue problem

$$J\mathcal{L}\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \lambda \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

↑

Eigenvalues

where

$$J = -i\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad \mathcal{L} = \begin{bmatrix} \partial_{xx} - \mu + \psi^2(x) & -\psi^2(x) \\ -\psi^2(x) & \partial_{xx} - \mu + \psi^2(x) \end{bmatrix}$$

Time-Periodic Solutions: Adjoint Continuation Method

Goal: Finds time-periodic solution based off the information from the bifurcation.

Idea: Analyze zeros of

$$G(u_0, T) = \frac{1}{2} \int_0^{2\pi} \Big(u(x, T) - u_0(x) \Big)^2 \, dx$$

Requirements:

- Initial condition
- Explicit information about time
- The Jacobian of the objective function

(Ambrose & Wilkening 2010)

Conclusion

- Developing tools to model PDEs on quantum graphs is valuable
- Rectangular Collocation is a superior method for accurately solving PDEs with boundary conditions
- Time evolution must be done carefully to preserve accuracy
- Bifurcation analysis allows us to study a problem's dynamics and find more interesting solutions
- Additional tools are needed to find time-periodic solutions

Thanks!

Special thanks to

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

Questions?