

YaleNUS College

Developing Mathematical Software for Efficient Representation of Functions and Numerical Integration: julia and ApproxFun

Dion Ho

05 September 2018

The Schrodinger Equation

6 The time-dependent linear Schrodinger equation

The time-dependent linear Schrodinger equation is a complex partial differential equation used to model quantum systems with wavefunctions which change with time.

The problem is given as follows:

$$\text{Partial Differential Equation, PDE: } [\partial_t + i(-i\partial_x)^2]q(x, t) = q_t - iq_{xx} = 0$$

$$\text{Initial Condition, IC: } q(x, 0) = q_0(x)$$

$$\text{Boundary Condition, BC: } q_x(0, t) + \beta q(0, t) = h(t).$$

Solving this equation requires computing the **wavefunction**: $\Psi(\mathbf{x}, t)$, though we denoted it $q(\mathbf{x}, t)$ instead.

Formula for the wavefunction

The wavefunction

$$\begin{aligned} 2\pi q(x, t) = & \int_{-\infty}^{\infty} e^{i\lambda x - i\lambda^2 t} \hat{q}_0(\lambda) d\lambda \\ & + \int_{\delta D^+} \frac{2i\lambda}{\beta + i\lambda} (e^{i\lambda x - i\lambda^2 t}) \int_0^\tau e^{i\lambda^2 s} h(s) ds d\lambda \\ & + \int_{\delta D^+} \frac{\beta - i\lambda}{\beta + i\lambda} (e^{i\lambda x - i\lambda^2 t}) \hat{q}_0(-\lambda) d\lambda \\ & - \int_{\delta D^+} \frac{\beta - i\lambda}{\beta + i\lambda} (e^{i\lambda x - i\lambda^2 t}) (e^{i\lambda^2 \tau}) \hat{q}(-\lambda; \tau) d\lambda \end{aligned}$$

Direct (Symbolic) Integration

- ▶ $\int x \, dx = \frac{x^2}{2} + C$
- ▶ $\int \sin(x) \, dx = -\cos(x) + C$
- ▶ $\int xe^x \, dx = xe^x + \int e^x \, dx = xe^x + e^x + C$
- ▶ $\int \frac{1}{x(x-1)} \, dx = \int \left(\frac{1}{x-1} - \frac{1}{x} \right) \, dx = \log|x-1| - \log|x| + C$

Direct Integration vs Numerical Integration

- Direct integration produces **EXACT VALUES**.

- ▶ $\int x \, dx = \frac{x^2}{2} + C$

- ▶ $\int \sin(x) \, dx = -\cos(x) + C$

- ▶ $\int xe^x \, dx = xe^x + \int e^x \, dx = xe^x + e^x + C$

- ▶ $\int \frac{1}{x(x-1)} \, dx = \int \left(\frac{1}{x-1} - \frac{1}{x} \right) \, dx = \log|x-1| - \log|x| + C$

- Numerical integration produces **APPROXIMATIONS**.

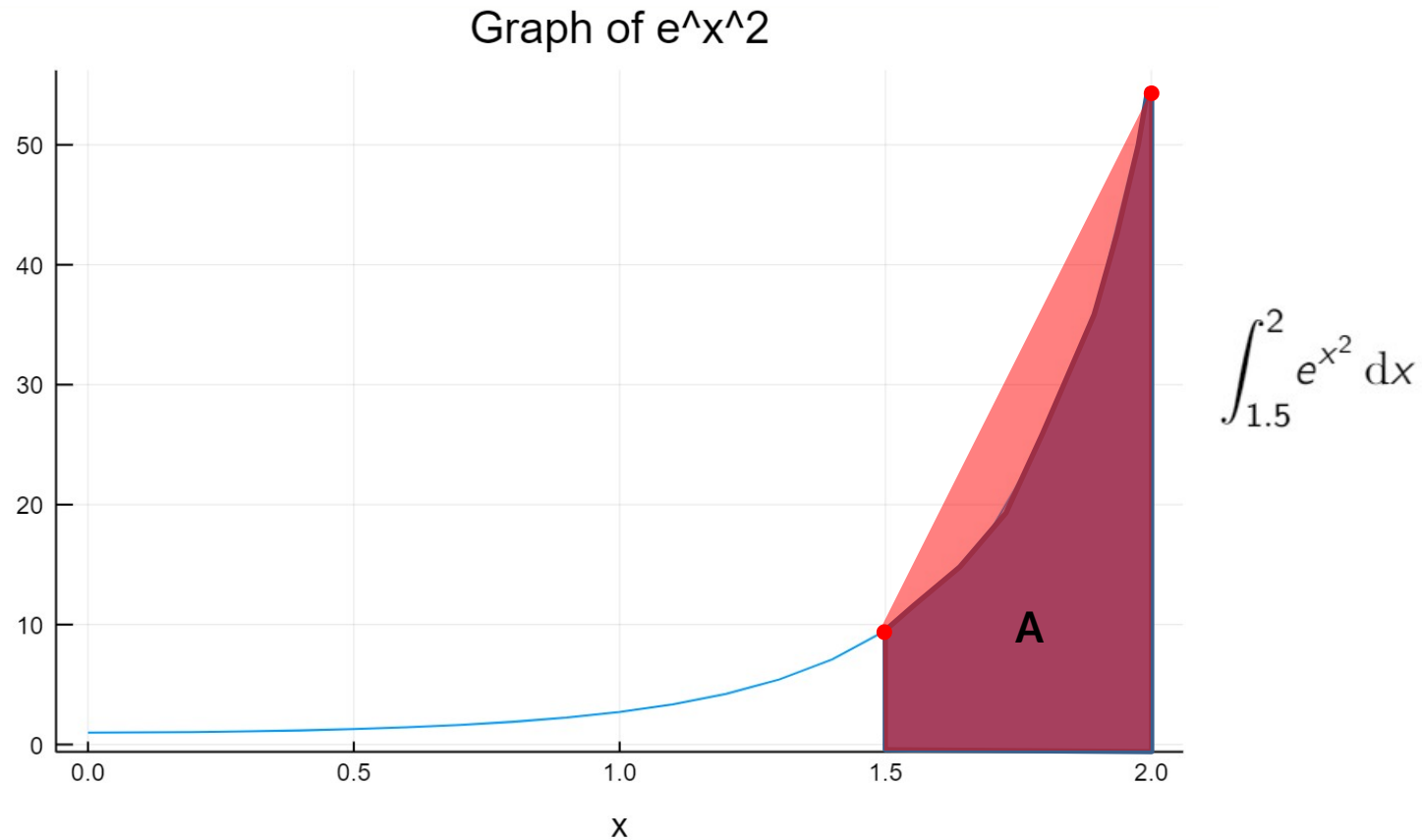
The Problem with Direct Integration

Direct integration is not feasible, or even possible, for every function.

▶ $\int \sqrt{\tan(x)} dx$

▶ $\int e^{x^2} dx$

Integration as Area Under the Curve



Newton-Cotes Rules

- ▶ Trapezoidal rule:

$$\int_b^a f(x) dx \approx \frac{a-b}{2}(f_a + f_b)$$

- ▶ Simpson's rule:

$$\int_b^a f(x) dx \approx \frac{a-b}{3}(f_a + 4f_c + f_b)$$

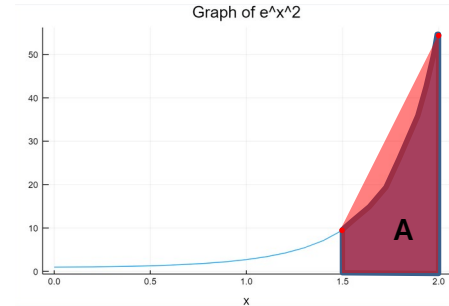
- ▶ Simpson's $\frac{3}{8}$ rule:

$$\int_b^a f(x) dx \approx \frac{3(a-b)}{8}(f_a + 3f_c + 3f_d + f_b)$$

- ▶ Boole's rule:

$$\int_b^a f(x) dx \approx \frac{2(a-b)}{45}(7f_a + 32f_c + 12f_d + 32f_e + 7f_b)$$

f_a denotes $f(a)$.



Polynomial Approximation

$$\int_b^a f(x) dx \approx$$

Taylor Polynomial Approximation

Taylor series expansion of $f(x)$ at number a :

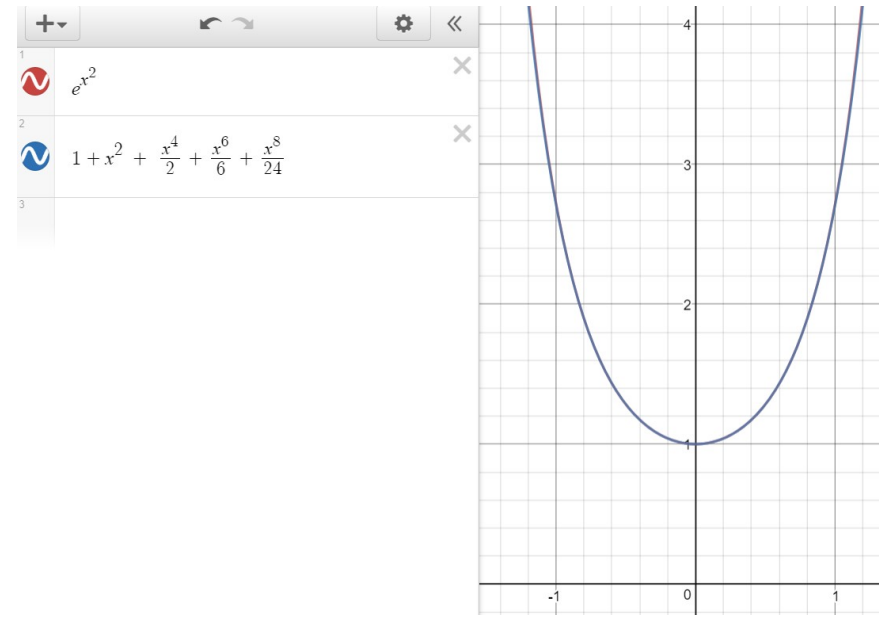
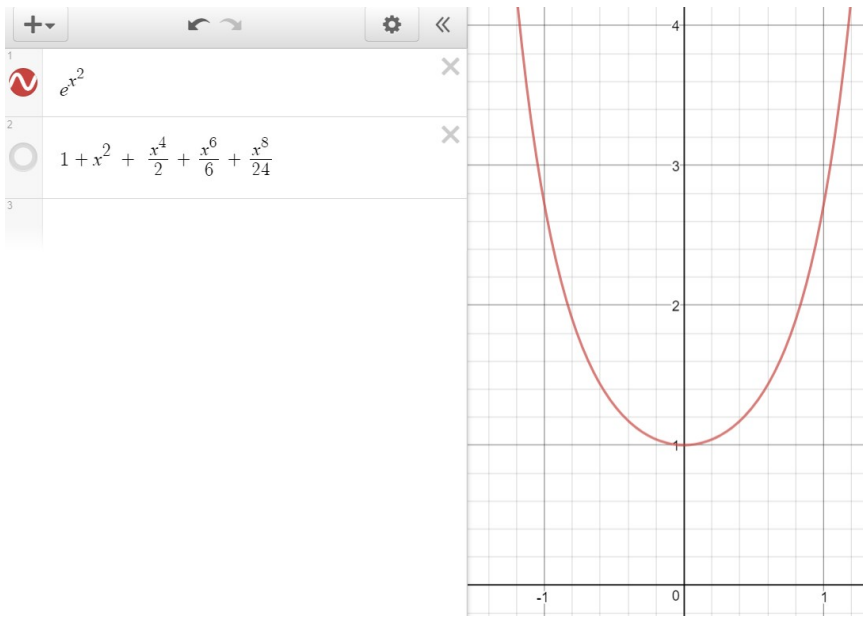
$$f(x) = f(a) + \frac{f'(a)}{1!}(x - a) + \frac{f''(a)}{2!}(x - a)^2 + \dots$$



Truncation

For example:

Taylor Polynomial Approximation (example)



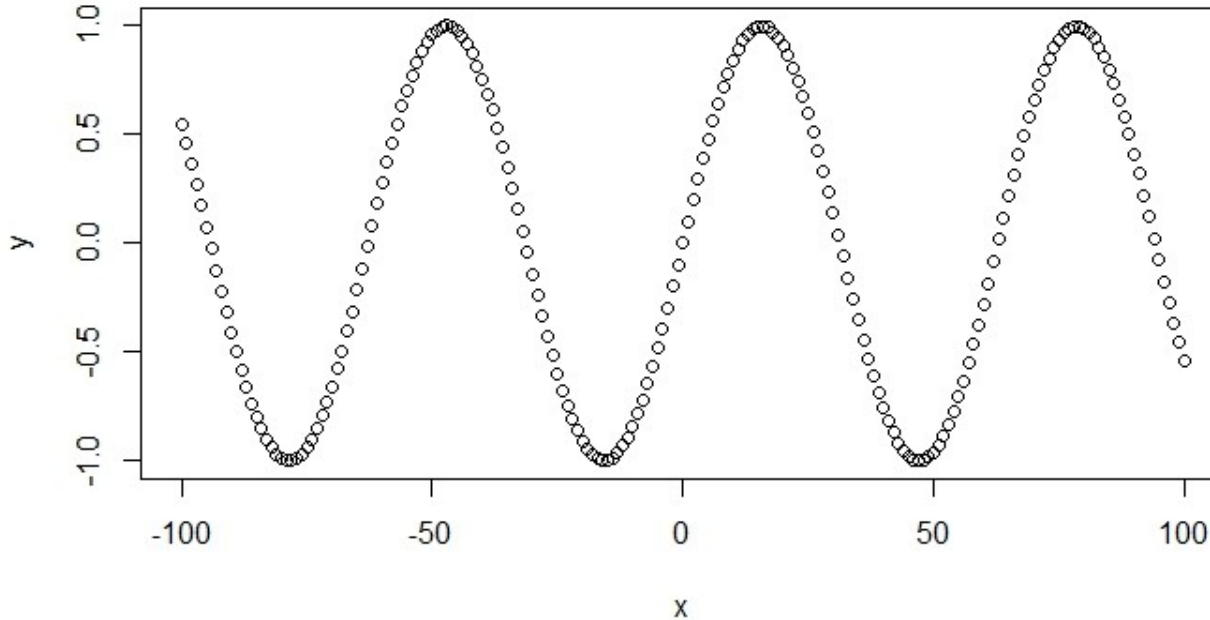
$$e^{x^2} \approx 1 + x^2 + \frac{x^4}{2} + \frac{x^6}{6} + \frac{x^8}{24}$$

Taylor Polynomial Approximation (example)

$$e^{x^2} \approx 1 + x^2 + \frac{x^4}{2} + \frac{x^6}{6} + \frac{x^8}{24}$$

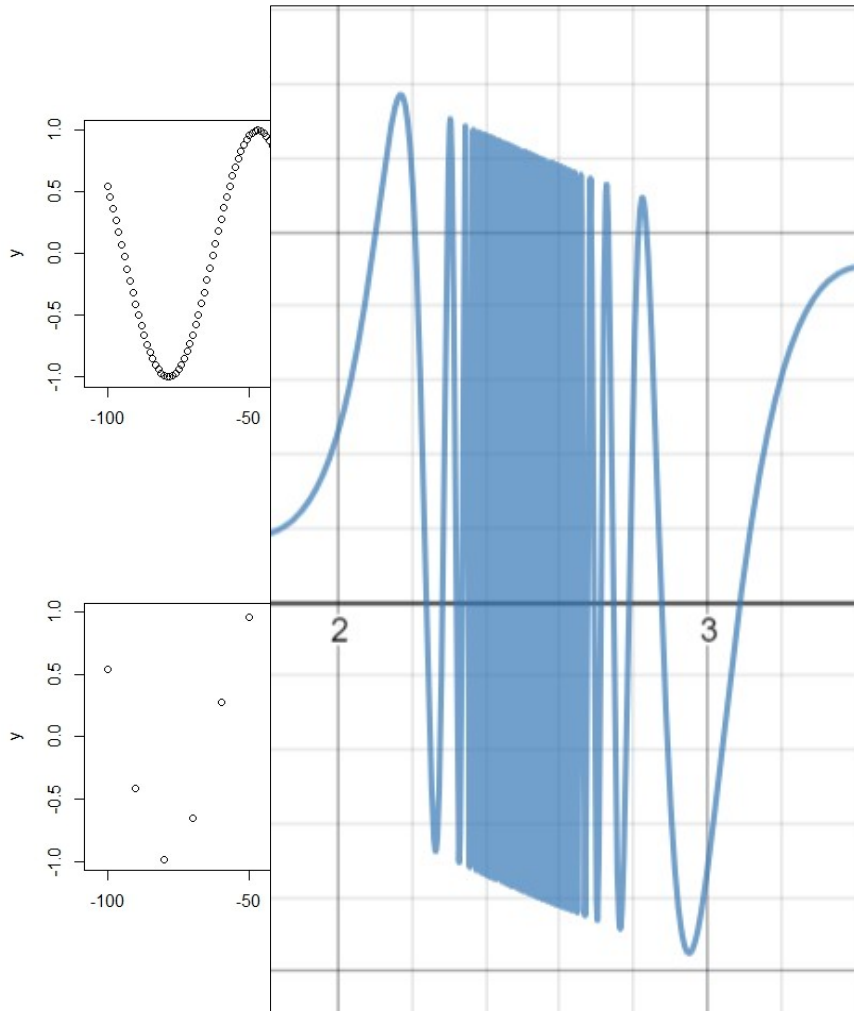
▶ Actual value: $\int_{-1}^1 e^{x^2} dx = 2.925 \dots$

Polynomial Interpolation

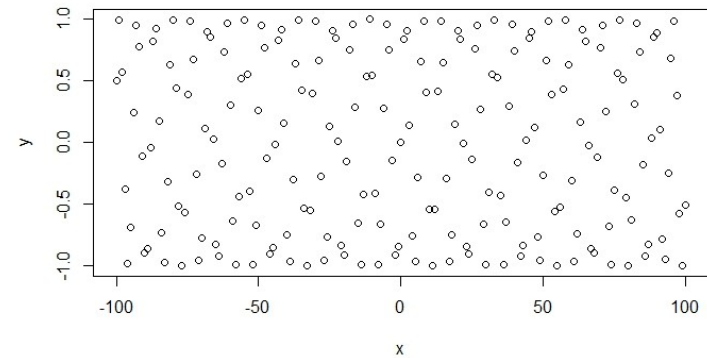


- Given a set of nodes, called *interpolation nodes*, which lie on the function to be approximated, a polynomial which intersects all the nodes will likely be a good approximation.
- The polynomial is formed by **solving a system of linear equations.**

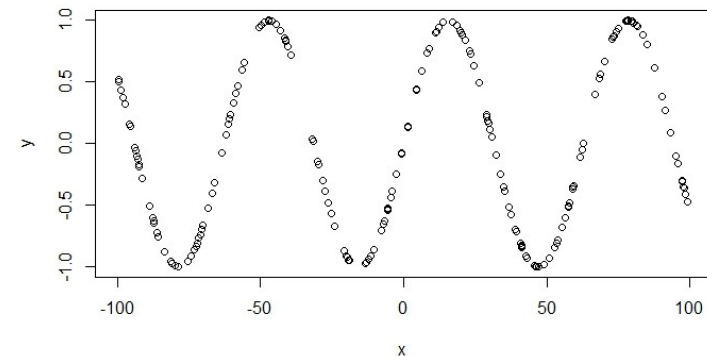
Polynomial Interpolation



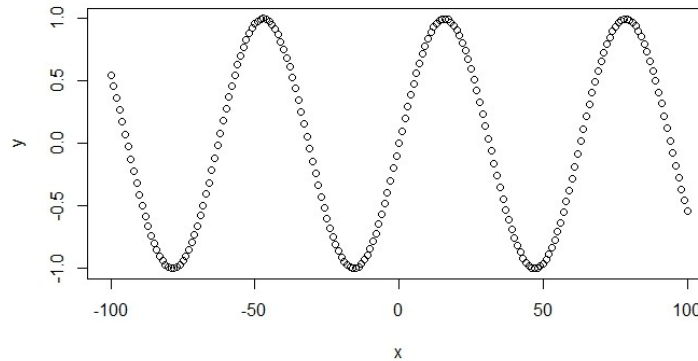
200 nodes for a highly oscillatory function



200 badly distributed nodes



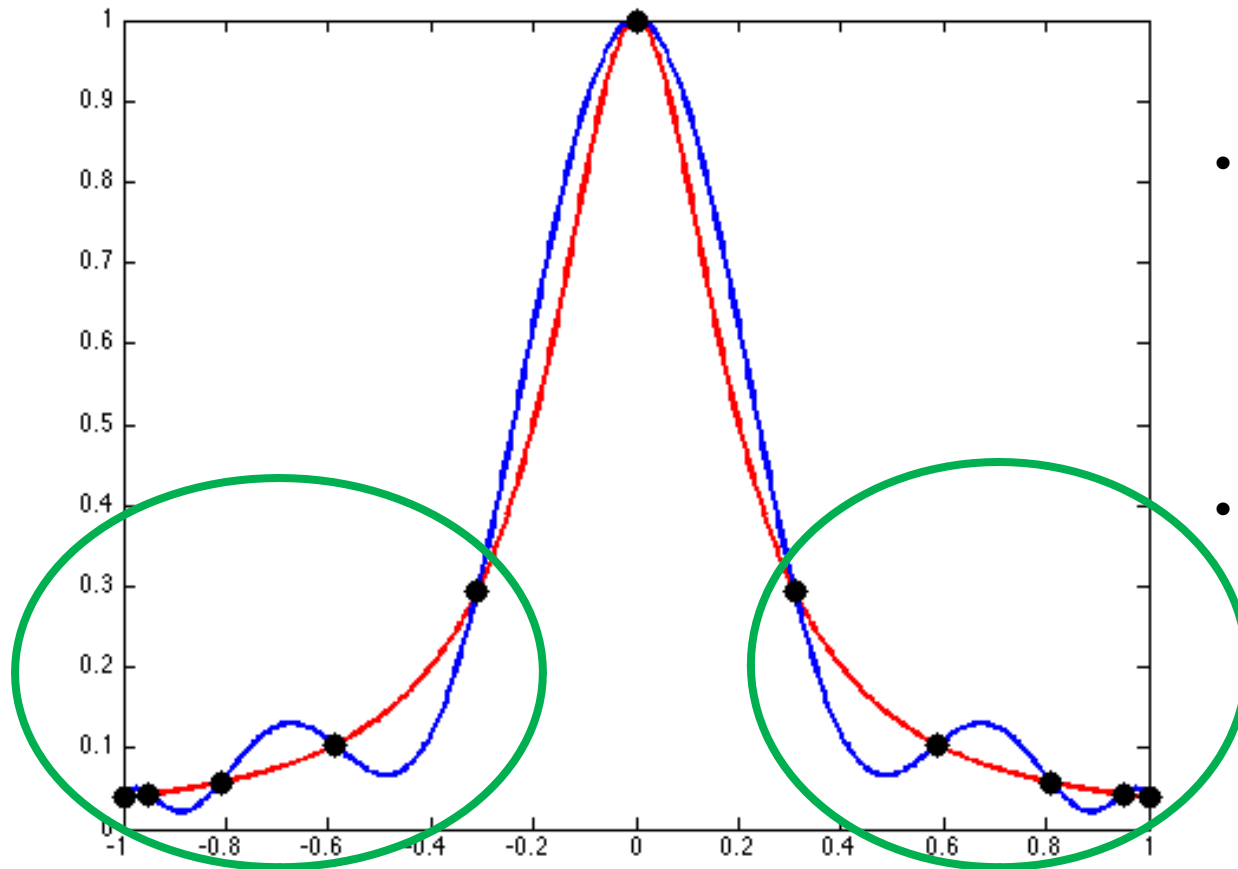
Choice of Distribution of Interpolation Nodes



200 equi-spaced nodes: $x = -100, -99, -98, \dots, 100$

- Equi-spaced nodes result in a relatively poor polynomial approximation due to the *Runge Phenomenon*.
- The use of ***Chebyshev nodes*** averts the Runge Phenomenon and results in a better approximation.

Chebyshev Nodes

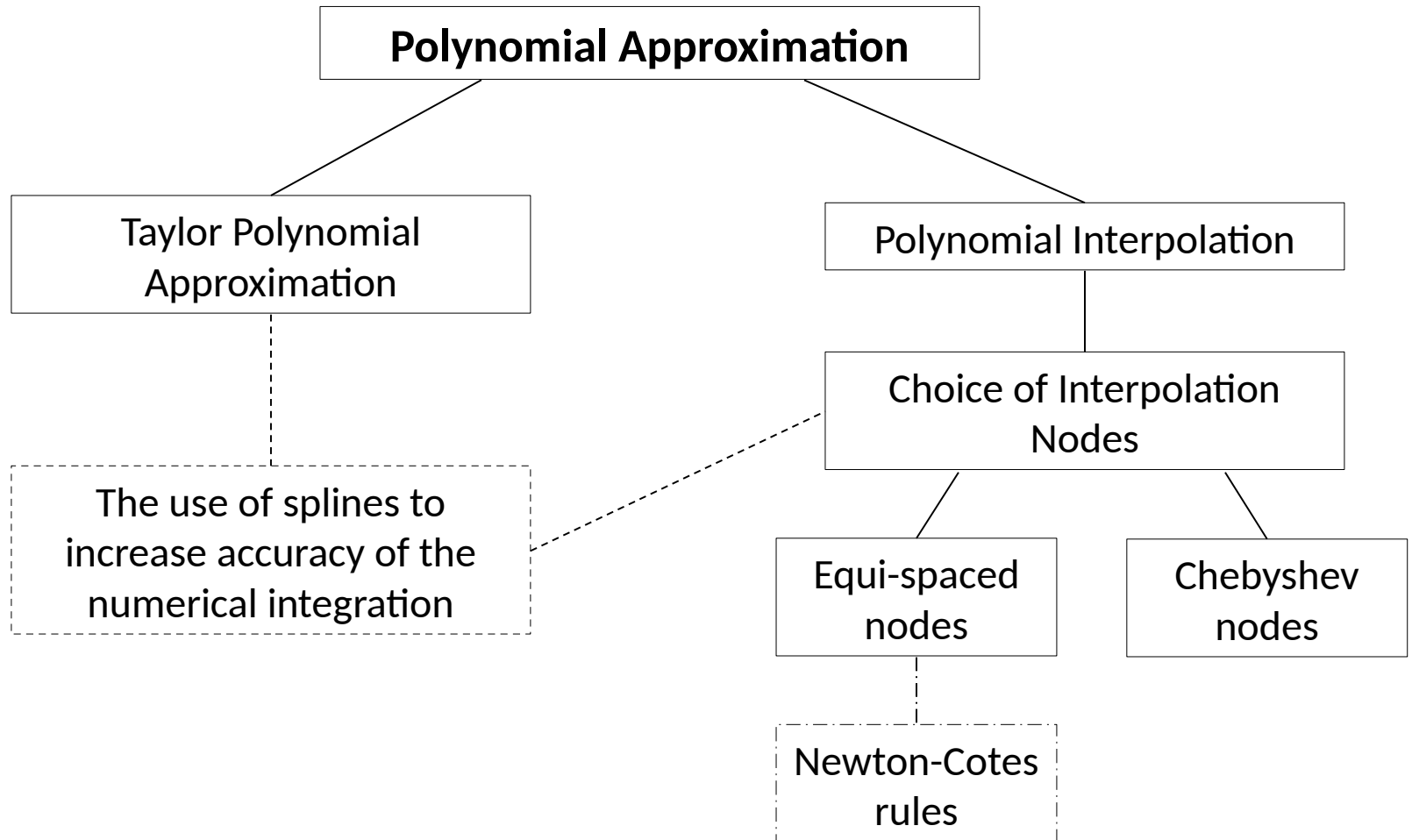


Graph taken from

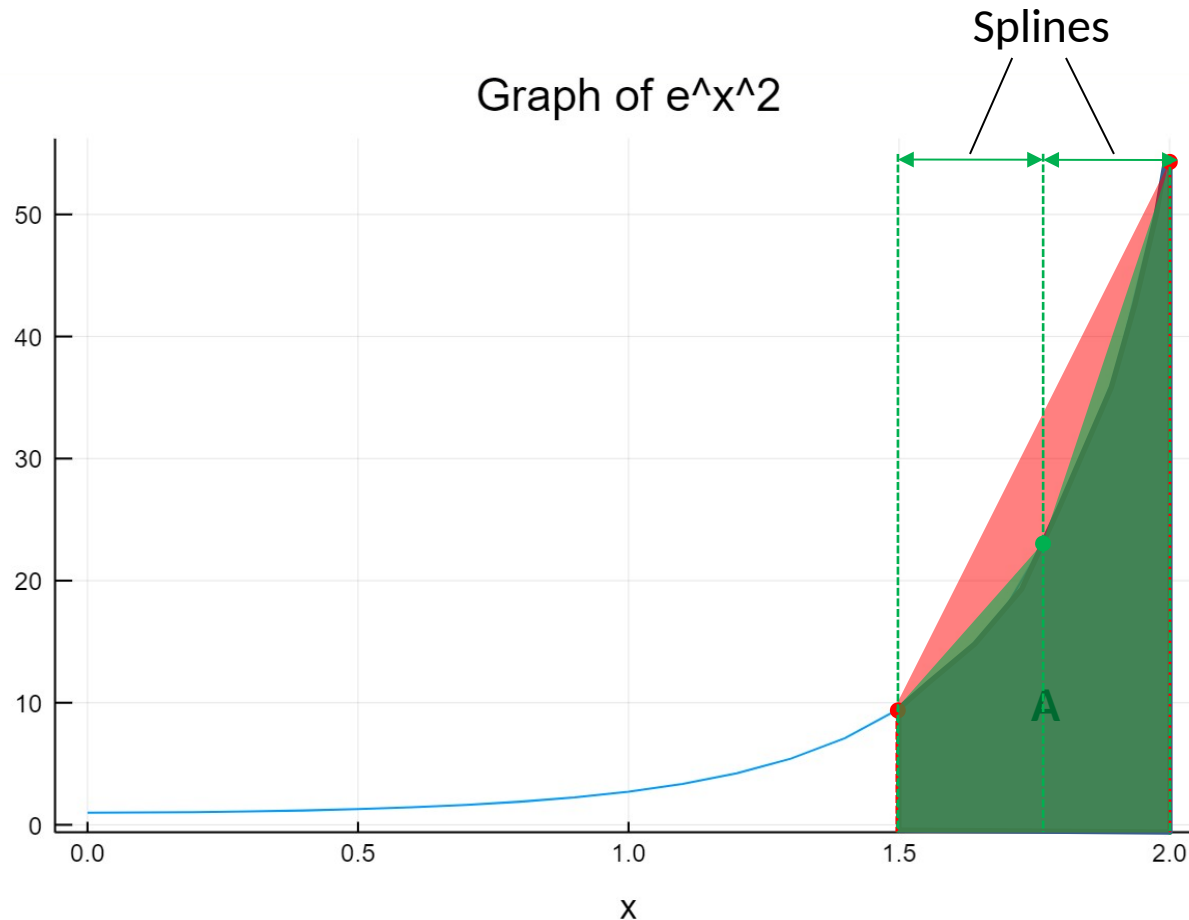
https://math.boisestate.edu/~calhoun/teaching/matlab-tutorials/lab_11/html/lab_11.html

- The nodes shown are Chebyshev nodes; they cluster about the extreme ends of the graph.
- The Runge Phenomenon manifests as **oscillations at the extreme ends.**

Numerical Integration: The many considerations



Splines and Composition



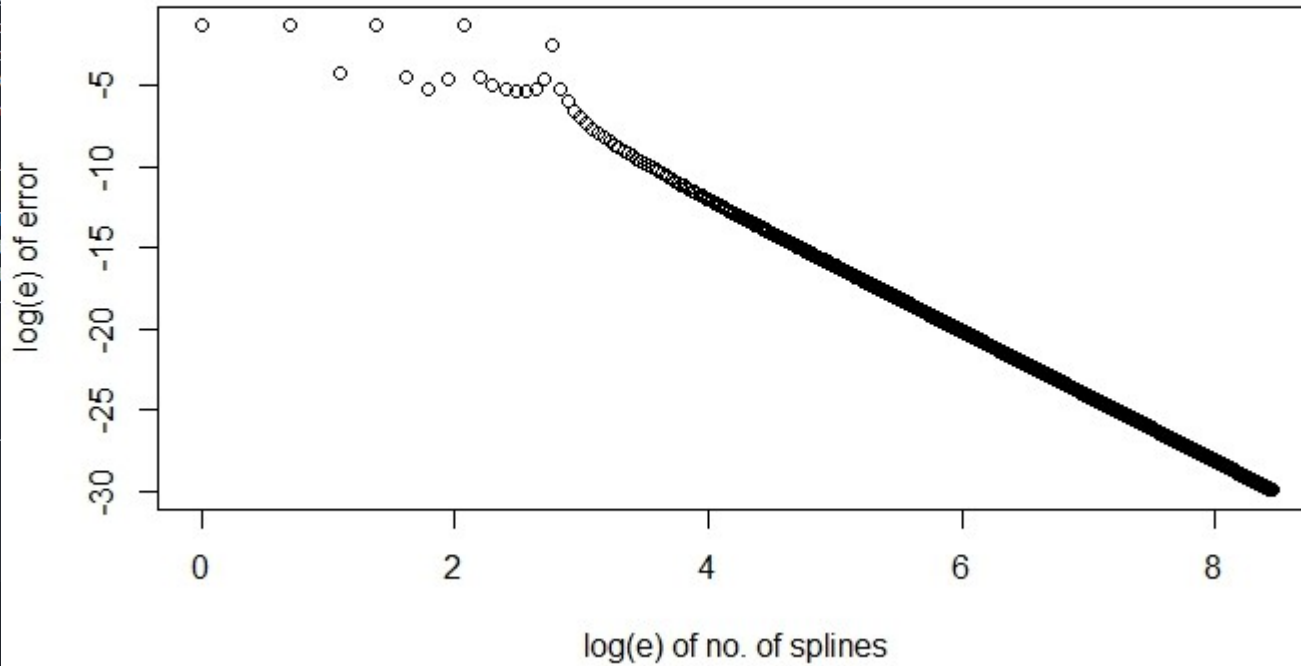
- The summation of multiple splines is called *composition*.
- The greater the number of splines used, the more accurate the numerical integration.

My Work

Quadrature functions.jl — D:\Personal\student-approxfun\julia files — Atom

File Edit View Julia Selection Find Packages Help

s2s38



```
WARNING: Atom.jl
julia> Arguments
A is the array of
X can be the ord
Alternatively, X
X must be an arr

IMPT: A and X mu
julia> booletss
BenchmarkTools.
memory estimat
allocs estimat
-----
minimum time:
median time:
mean time:
maximum time:
-----
samples:
evals/sample:

julia> 
```

or contour integration.	
latory parts only.	BEST
hly oscillatory parts, and	
or contour integration.	
latory parts only.	
ighly oscillatory parts, nes.	BEST

```
35 ans=ans+(4*h/3)*F(y+(2*r-1)*h) + (2*h/3)*F(y+2*r*h)
36 end
37 end
38
```

Quadrature functions.jl 1:1

CR

Formula for the wavefunction

$$\begin{aligned} 2\pi q(x, t) &= \int_{-\infty}^{\infty} e^{i\lambda x - i\lambda^2 t} \hat{q}_0(\lambda) d\lambda \\ &+ \int_{\delta_{D+}} \frac{2i\lambda}{\beta + i\lambda} (e^{i\lambda x - i\lambda^2 t}) \int_0^\tau e^{i\lambda^2 s} h(s) ds d\lambda \\ &+ \int_{\delta_{D+}} \frac{\beta - i\lambda}{\beta + i\lambda} (e^{i\lambda x - i\lambda^2 t}) \hat{q}_0(-\lambda) d\lambda \\ &- \int_{\delta_{D+}} \frac{\beta - i\lambda}{\beta + i\lambda} (e^{i\lambda x - i\lambda^2 t}) (e^{i\lambda^2 \tau}) \hat{q}(-\lambda; \tau) d\lambda \end{aligned}$$

The time-dependent linear Schrodinger equation simply serves as a litmus test for our algorithm!

The Algorithm

ApproxFun (Chebyshev nodes) but with some wrapper functions coded around it for increased efficiency and to gear ApproxFun to perform **numerical contour integration**.

ApproxFun.jl

docs stable docs latest build passing coverage 90% chat on gitter

ApproxFun is a package for approximating functions. It is in a similar vein to the Matlab package [Chebfun](#) and the Mathematica package [RHPackage](#).

The [ApproxFun Documentation](#) contains detailed information, or read on for a brief overview of the package.

The [ApproxFun Examples](#) contains many examples of using this package, in Jupyter notebooks and Julia scripts.

Introduction

Take your two favourite functions on an interval and create approximations to them as simply as:

```
using LinearAlgebra, SpecialFunctions, Plots, ApproxFun
x = Fun(identity, 0..10)
f = sin(x^2)
g = cos(x)
```

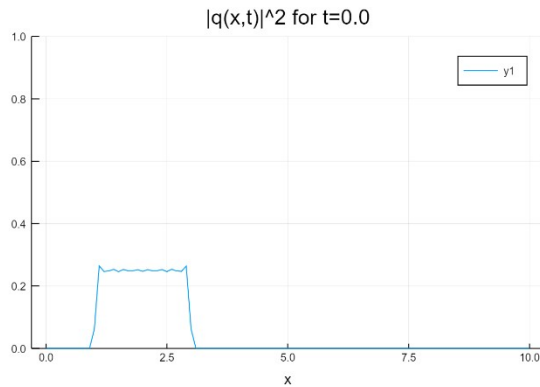
Evaluating `f(.1)` will return a high accuracy approximation to `sin(0.01)`. All the algebraic manipulations of functions are supported and more. For example, we can add `f` and `g^2` together and compute the roots and extrema:

```
h = f + g^2
r = roots(h)
rp = roots(h')
```

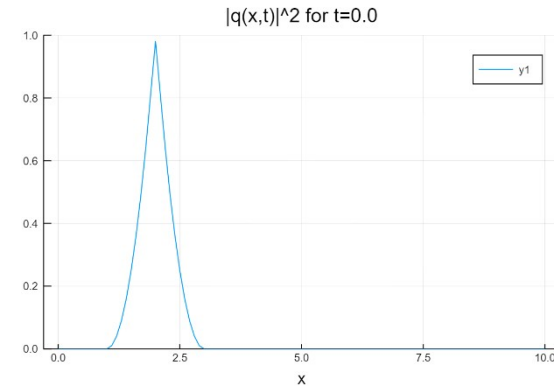
These are
contours on the
complex plane.

The Four Types of Initial Condition

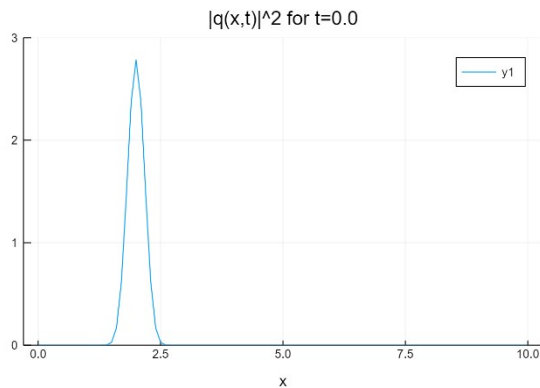
Non-continuous



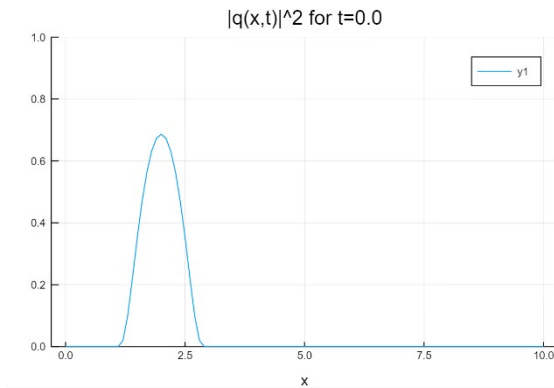
Continuous but non-differentiable



Continuous and n-differentiable

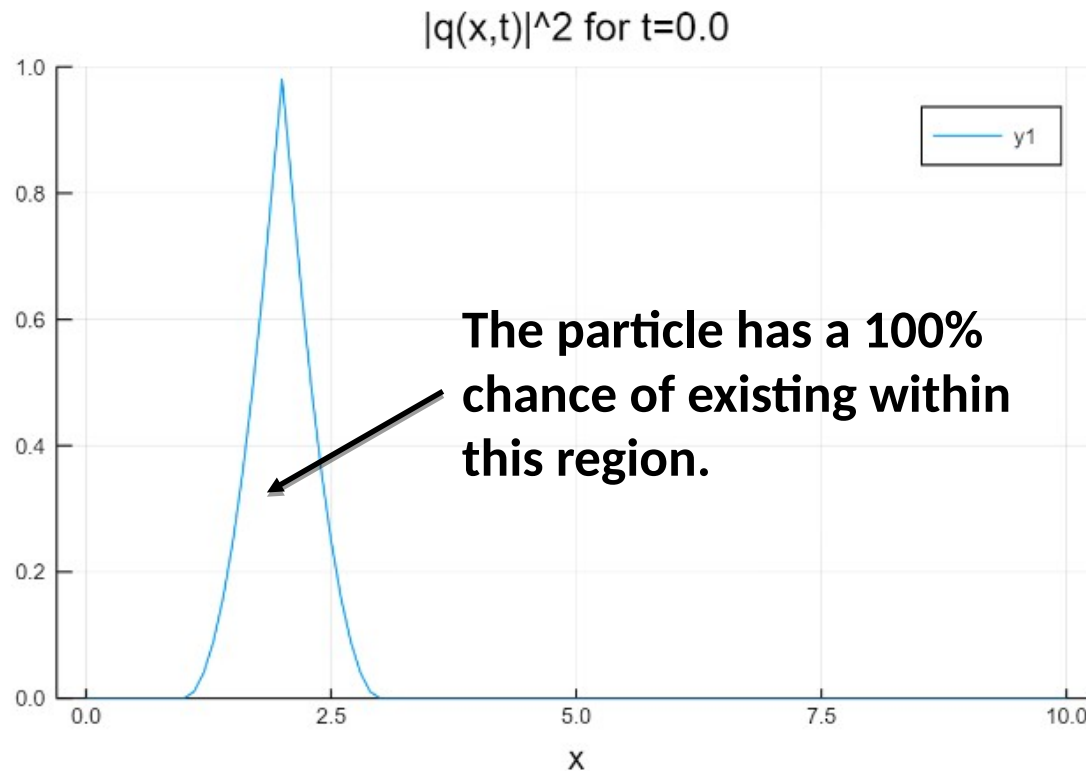


Continuous and infinitely differentiable (mollifier)



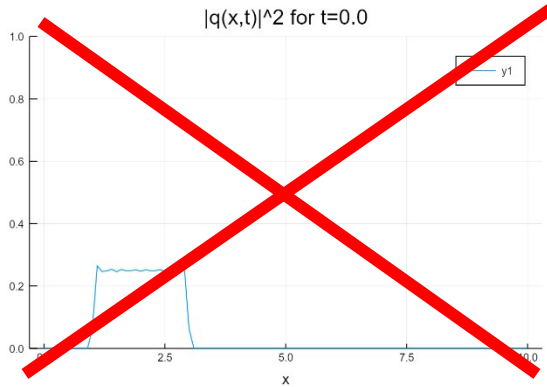
The Probability Density Function

We plotted $|\psi(x,t)|^2$ which is a probability density function telling us the probability of finding the quantum particle in a region of x - values.



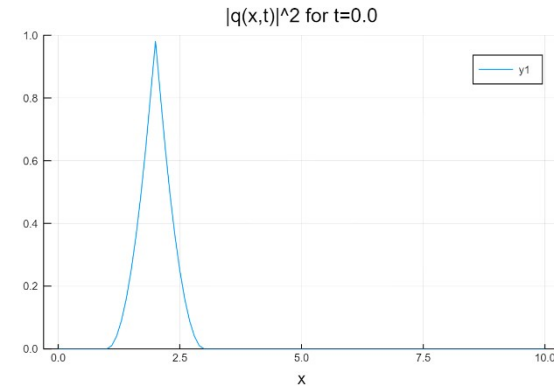
The Four Types of Initial Condition

Non-continuous

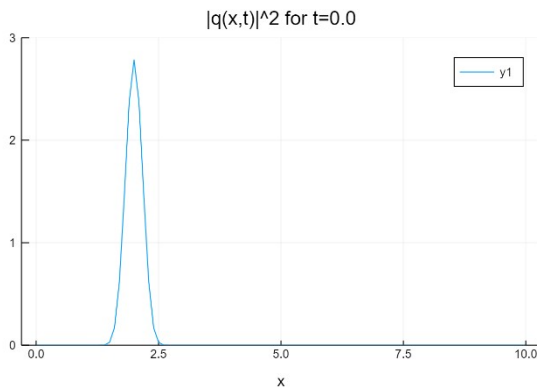


Graphs are inaccurate

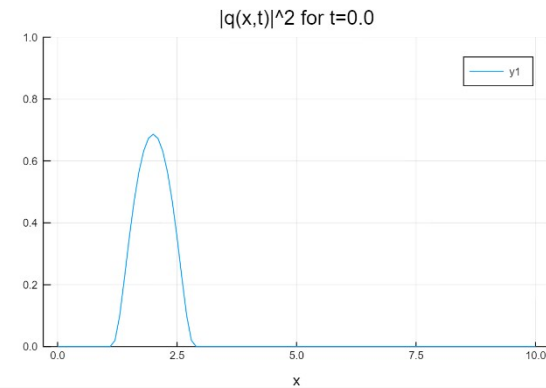
Continuous but non-differentiable

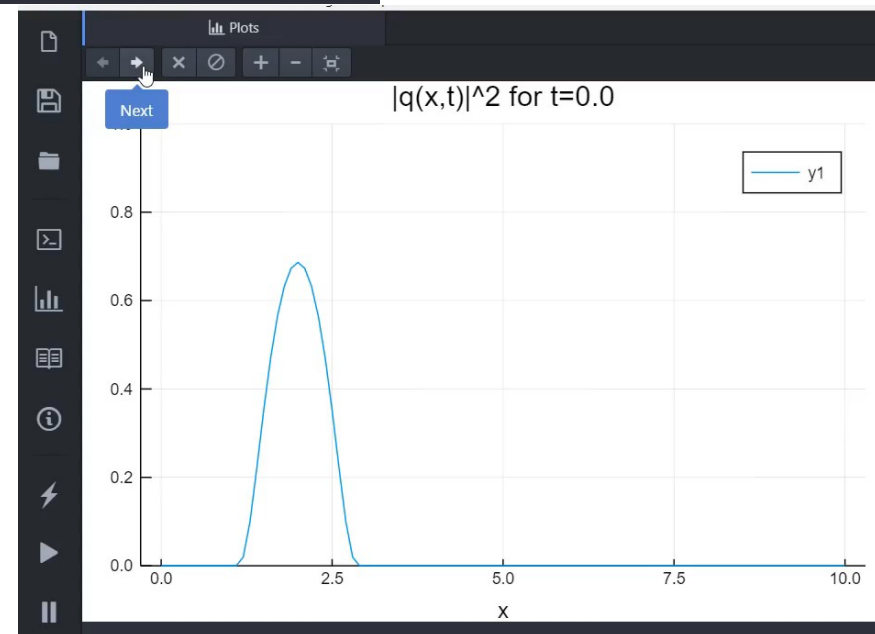
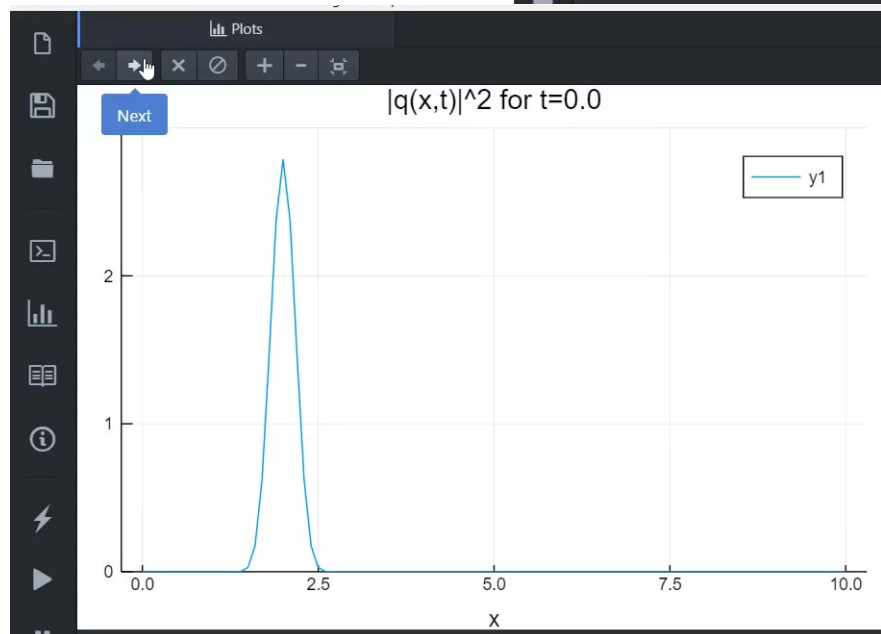
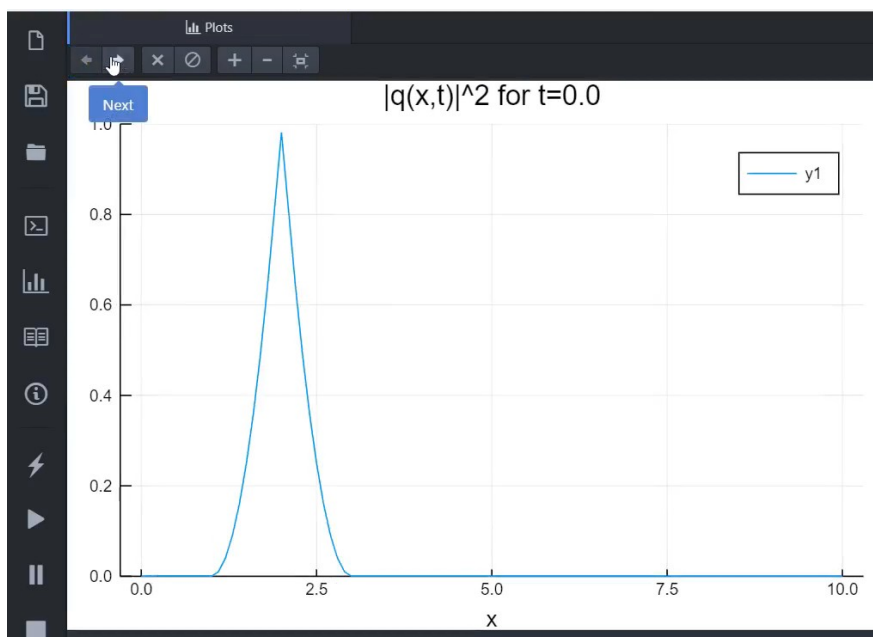


Continuous and n-differentiable



Continuous and infinitely differentiable (mollifier)





Other Potential Applications of this Algorithm

- Many problems require numerical integration, but the numerical integration algorithm best suited to each problem may be different.
- This algorithm can be used to help solve problems on heat flow. Toh Wei Yang's project dealt with the heat equation, which models heat flow:

Heat Equation with Dynamic Boundary Condition Reduces to Fractional Linear Ordinary Differential Equation

Toh Wei Yang | Email: weiyang.toh@u.yale-nus.edu.sg

Abstract

The heat equation is a partial differential equation that describes how heat is distributed in a region over time. In this project, we seek to solve the heat equation on the half line where the boundary condition at one end evolves with time using the Fokas method. We show that the problem reduces to a fractional linear ordinary differential equation (FLODE) with a variable coefficient. Drawing from ideas in fractional calculus, we obtain a solution to the FLODE through the Frobenius method, thus solving the heat equation.

Introduction

Consider the following heat equation with dynamic boundary condition

$$\begin{aligned} \phi_t + \phi_{xx} &= 0, & (x, t) \in (0, \infty) \times (0, T), \\ \phi(x, 0) &= \psi(x), & x \in [0, \infty), \\ \phi_x(0, t) + f(t)\phi(0, t) &= 0, & t \in [0, T], \end{aligned}$$

where T is a positive constant.

Through the Fokas method, we find the solution to be given by

$$2\pi\phi(x, t) = \int_{-\infty}^{\infty} e^{i\lambda x - \lambda^2 t} \hat{\phi}(\lambda) d\lambda - \int_{\partial D^+} e^{i\lambda x - \lambda^2 t} F(\lambda, T) d\lambda \quad (1)$$

where

$$F(\lambda, T) = \int_0^T e^{i\lambda^2 s} \phi(0, s) ds + i\lambda \int_0^T e^{i\lambda^2 s} \psi(0, s) ds,$$

and $D^+ = \{\lambda \in \mathbb{C}^+ : \Re(\lambda) < 0\}$.

Our goal is to express the solution in terms of known data. Through a process known as Dirichlet-to-Neumann Map, we are able to express the solution solely in terms of one boundary value, and reduce the problem to simply solving for that boundary value. We have effectively reduced the problem to solving just for $\psi(0, s)$ in

$$\hat{\phi}(-i\sqrt{-\lambda}) - \int_0^T e^{i\lambda^2 s} \sqrt{-\lambda} \psi(0, s) ds. \quad (2)$$

Fractional Integral and Fractional Derivative

Definition 1. For $0 < \alpha < 1$, the Liouville left-sided fractional integral on \mathbb{R} is defined as

$$(I_{\lambda}^{\alpha} \psi)(x) := \frac{1}{\Gamma(\alpha)} \int_{-\infty}^x \frac{\psi(t) dt}{(x-t)^{1-\alpha}}. \quad (3)$$

Definition 2. For $0 < \alpha < 1$, the Caputo derivative is defined as

$$({}^C D_{\lambda}^{\alpha} \psi)(x) := (I_{\lambda}^{1-\alpha} D) \psi(x) \quad (4)$$

where $D = \frac{d}{dx}$.

Property 1. For $0 < \alpha < 1$ and $\Re(\beta) > 1$,

$$({}^C D_{\lambda}^{\alpha} (t^{\beta-1})) (x) = \frac{\Gamma(\beta)}{\Gamma(\beta-\alpha)} (x)^{\beta-1-\alpha}. \quad (5)$$

In particular,

$$({}^C D_{\lambda}^{\alpha}) (x) = 0.$$

Fourier Transform of Fractional Integrals and Derivatives

Theorem 2. Suppose ψ is a function in the Schwartz space such that

$$\psi(x) = \begin{cases} \hat{\psi}(x) & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$(\mathcal{F} D_{\lambda}^{\alpha} \psi)(x) = \frac{\hat{\psi}(x)}{(-ix)^{\alpha}}$$

where $\hat{\psi}(x) = (\mathcal{F} \psi)(x)$.

Corollary 2.1. Suppose that ψ and α are as in the Theorem 2 and $\psi(0) = 0$. Then

$$(\mathcal{F}^2 D_{\lambda}^{\alpha} \psi)(x) = (-ix)^{\alpha} \hat{\psi}(x).$$

α -analyticity and Power Rule

Definition 3. Let $\alpha \in (0, 1)$ and $f(x)$ be a real function defined on some interval $[a_0, b_0]$ and $x_0 \in [a_0, b_0]$. Then $f(x)$ is said to be α -analytic at x_0 if there exists an interval $N(x_0)$ such that for all $x \in N(x_0)$, $f(x)$ can be expressed as $\sum_{n=0}^{\infty} a_n (x-x_0)^{\alpha n}$.

Proposition 3. Let $\alpha \in (0, 1)$. If $f(x)$ is α -analytic at a_0 with convergence radius ρ , then

$$({}^C D_{\lambda}^{\alpha} f)(x) = \left(D_{\lambda}^{\alpha} \left(\sum_{n=0}^{\infty} a_n (x-a_0)^{\alpha n} \right) \right) (x) = \sum_{n=0}^{\infty} a_n ({}^C D_{\lambda}^{\alpha} (x-a_0)^{\alpha n})(x).$$

Theorem 4. Let $\alpha \in (0, 1)$ and let $f(x) = \psi(x)$ where $\psi(x)$ is as defined in Theorem 2 and such that $f(x) = \sum_{n=0}^{\infty} a_n x^{\alpha n}$. Then,

$$({}^C D_{\lambda}^{\alpha} f)(x) = \sum_{n=0}^{\infty} a_n \frac{\Gamma(n\alpha + 1)}{\Gamma(n-1)\alpha + 1} x^{n\alpha-1}.$$

Solving the Fractional Linear Ordinary Differential Equation via the Frobenius Method

By taking the Fourier inverse of the RHS of Equation 2 and using Theorem 2, Equation 2 reduces to a Fractional Linear Ordinary Differential Equation of the form

$$({}^C D_{\lambda}^{\alpha} \hat{\psi})(t) - f(t)\hat{\psi}(t) = g(t) \quad (6)$$

where $g(t) = \psi(0, t)$ and $f(t) = \frac{1}{2t} \int_0^T e^{-i\lambda^2 s} \sqrt{-\lambda} \psi(0, s) ds$. Suppose that $\hat{\psi}(t)$ is α -analytic about the α -ordinary point 0. We seek the series solution of the form

$$\hat{\psi}(t) = \sum_{n=0}^{\infty} a_n t^{\alpha n}.$$

By Theorem 4,

$$({}^C D_{\lambda}^{\alpha} \hat{\psi})(t) = \sum_{n=0}^{\infty} a_n \frac{\Gamma(n\alpha + 1)}{\Gamma(n-1)\alpha + 1} t^{n\alpha-1}.$$

Further suppose that $f(t)$ and $g(t)$ are also α -analytic about 0. Let $f(t) = \sum_{n=0}^{\infty} b_n t^{\alpha n}$ and $g(t) = \sum_{n=0}^{\infty} c_n t^{\alpha n}$. We can express the coefficients a_{n+1} in terms of a_n , b_n , and c_n by the following recurrence relation:

$$a_{n+1} = \frac{\Gamma(n\alpha + 1)}{\Gamma(n\alpha + 1 - \alpha)} \left(\sum_{k=0}^n a_k b_{n-k} + c_n \right)$$

with $a_0 = 0$ by necessity of Corollary 2.1. We can thus compute the coefficients of $\hat{\psi}(0, t)$, which will then lead us to the solution of the heat equation.

Plots of Solutions to FLODE

Figure 1: Plot of $\psi(t)$ at 10th order of approximation where $f(t) = t - |t - 1/2|^2 + t^2$ and $g(t) = t + |t - 2t + 1|^2$

Figure 2: Plot of $\phi(t)$ at 10th order of approximation where $f(t) = t - 2t + |t|^2 + t^2$ and $g(t) = t + |t - 2t + 1|^2$

Applications of Fractional Differential Equations

In systems where anomalous dynamics are present, fractional differential equations are more accurate than differential equations with classical derivatives in modeling anomalous processes. For example, the Porous Medium Equation (PME) which models non-linear heat flow, and gas flow in porous medium, has been extended into fractional forms to account for anomalous diffusion which then have concrete applications such as in the study of moisture dispersion in porous construction materials.

References

- [1] A. S. Fokas, A Unified Approach to Boundary Value Problems, CBMS-SIAM, 2008.
- [2] A. A. Kilbas, H. M. Srivastava, and J. J. Trujillo, Theory and Applications of Fractional Differential Equations, Elsevier B.V., 2006.
- [3] L. Pincicnik, Analytical studies of a time-fractional porous medium equation, derivation, approximation and applications, Communications in Nonlinear Science and Numerical Simulation, 24(1-3), 2015.

Acknowledgement

I would like to thank David Smith for his help and guidance throughout this research project.

I would also like to express my gratitude to CIFE, and especially to Ms. Zhiana Sandeva for leading the Summer Research Programme.

Lastly, I would like to thank the JY Pillay Global-Asia Programme and the Dean of Faculty Office at Yale-NUS College for their generous funding to this research.

Acknowledgements

- Assistant Professor David Andrew Smith, Yale-NUS College
- JY Pillay Global-Asia Programme
- Dean of Faculty Office, Yale-NUS College
- Centre for International and Professional Experience, Yale-NUS College, especially Ms Zhana Sandeva.

References

References

- [1] M. J. Ablowitz and A. S. Fokas, *Complex variables*, Cambridge Texts in Applied Mathematics, Cambridge University Press, 1997.
- [2] A. S. Fokas, *A unified approach to boundary value problems*, CBMS-SIAM, 2008.
- [3] Wouter Den Haan, <http://econ.lse.ac.uk/staff/wdenhaan/numerical/>, filename: functionapproximation.pdf, last modified: 2011-08-29 16:02.
- [4] J.H. Mathews and K.D. Fink, *Numerical methods using matlab*, Featured Titles for Numerical Analysis Series.
- [5] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, *Numerical recipes in c (2nd ed.): The art of scientific computing*, pp. 190–195, Cambridge University Press, New York, NY, USA, 1992.
- [6] L.N. Trefethen, *Finite difference and spectral methods for ordinary and partial differential equations*, pp. 260–268, The author, 1996.