Finite Difference based wave propagation

Sajid Ali[1]

[1]Applied Physics
Northwestern Univ

8 Feb 2019

Outline

Basics 1

- The Helmholtz equation is

$$\nabla^2\psi + k^2 n^2(x,y,z)\psi = 0$$

  where $k = \frac{2\pi}{\lambda}$ and $n(x,y,z)$ is the refractive index at the given co-ordinates.
- Approximate the wavefunction with plane wave and osciallating parts,

$$\psi(x,y,z) = u(x,y,z)exp(-ikz)$$

Basics 2

- Substituting this in the Helmholtz equation and neglecting the derivative of u along the direction of propagation, we get the following equation.

$$\text{-2ik}\frac{\partial}{\partial z}u + (\frac{\partial^2}{\partial x} + \frac{\partial^2}{\partial y})u + k^2(n^2 - 1) = 0$$

- Define [1] $a = \frac{-i}{2k}$ and $F(x, y, z) = \frac{-i}{2k}(n^2(x, y, z) - 1)$
- The PDE can now be written as

$$\frac{\partial u}{\partial z} = \text{a}(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) + \text{F(x,y,z)}u$$

[1] [Fuh06]

Outline

Laplacian operator, 1D

- For a function f defined on a discrete grid, we can approximate the value of the function f on the points $x_{i\pm1}$, using the value of f at x_i using Taylor series.

$$f(x_{i\pm1}) = f(x_i) \pm hf'(x_i) + \frac{h^2}{2}f''x + O(h^3)$$

- The second derivative can thus be approximated as

$$f''(x_i) \approx \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2}$$

Lapacian operator, 1D

- As per the last approximation, we can use the following matrix operator to calculate the laplacian in 1D.

$$A = \frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix}$$

Laplacian operator, 2D

- The discrete laplacian operator in 2D can de derived in a method similar to the 1D case. In 2 dimensions, the second derivate can be approximated as

$$f''(x_{i,j}) \approx \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{h_x^2} + \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{h_y^2}$$

Laplacian operator, 2D

- As per the last approximation, we can use the following block matrix operator to calculate the laplacian in 2D.

$$A = \begin{bmatrix} A_{xy} & I_y & & & \\ I_y & A_{xy} & I_y & & \\ & \ddots & \ddots & \ddots & \\ & & I_y & A_{xy} & I_y \\ & & & I_y & A_{xy} \end{bmatrix}$$

- where I_y is a diagonal matrix with all entries set to $\frac{1}{h_y^2}$

Laplacian operator, 2D

- And

$$A_{xy} = \begin{bmatrix} \left\{\frac{-2}{h_x^2} + \frac{-2}{h_y^2}\right\} & \frac{1}{h_x^2} & & & \\ \frac{1}{h_x^2} & \left\{\frac{-2}{h_x^2} + \frac{-2}{h_y^2}\right\} & \frac{1}{h_x^2} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{1}{h_x^2} & \left\{\frac{-2}{h_x^2} + \frac{-2}{h_y^2}\right\} & \frac{1}{h_x^2} \\ & & & \frac{1}{h_x^2} & \left\{\frac{-2}{h_x^2} + \frac{-2}{h_y^2}\right\} \end{bmatrix}$$

Dirichlet Boundary Conditions

- Dirichlet boundary conditions are applied which physically translates to the effect that otuermost pixels never change in value.
- With Dirichlet BC conditions, the operator A in 1D looks like :

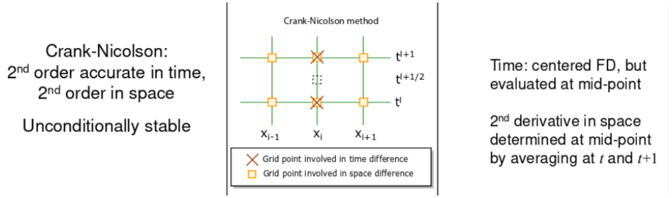
$$A = \frac{1}{h^2}\begin{bmatrix} 1 & 0 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 0 & 1 \end{bmatrix}$$

- The 2D operator changes similarly.

Outline

Integration schemes in 1D

- Fuhse et.al. suggested the use of implicit, second order schemes [2] for stability and accuracy.
- For 1D, Crank-Nicolson method is proposed, the stencil for which is shown below[3].



Crank-Nicolson:
2nd order accurate in time,
2nd order in space

Unconditionally stable

Time: centered FD, but evaluated at mid-point

2nd derivative in space determined at mid-point by averaging at t and t+1

Crank-Nicolson method
t^{l+1}
$t^{l+1/2}$
t^{l}
x_{i-1}
x_i
x_{i+1}
Grid point involved in time difference
Grid point involved in space difference

- This reduces the problem to a tridiagonal matrix solve which is trivial to solve.

---

[2] [Fuh06]
[3] MIT OCW 2.29, Numerical Fluid Mechanics

Integration schemes in 2D

- In 2D Fuhse et.al suggested the use of Alternating Direction Implicit[4].
- ADI is a 2 step process in which alternates between implicit and explicit derivates along the two dimensions. Each step now involves a tridiagonal matrix solution.

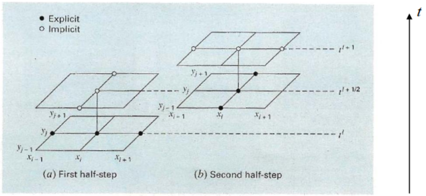


FIGURE 30.10
The two half-steps used in implementing the alternating-direction implicit scheme for solving parabolic equations in two spatial dimensions.

[4][Fuh06]

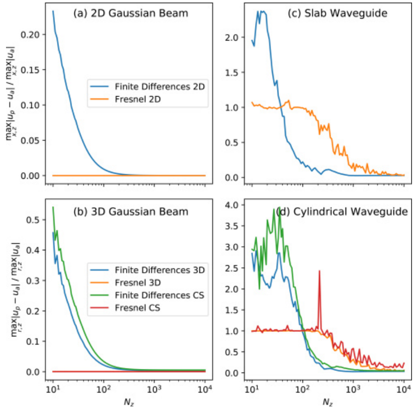
Previous work

- Fuhse et.al. chose IDL to implement the programs.
- A more accurate version of the aforementioned scheme was implemented by Melchoir et.al.[5] in C++ with a Python front end. Although this was efficient, this implementation scheme involved writing the discretiztion routines for CN and ADI and thereby locked in the developers to these integration schemes.
- The schemes are straightforward to implement using the scipy.sparse libraries[6].

[5] [MS17]
[6] https://github.com/s-sajid-ali/xwp

FD vs Multislice

- Multislice is better for free space propagation while FD gives better results for propagation through matter. [7]



[7] [MS17][SO91]

PETSc overview

- What is PETSc ?

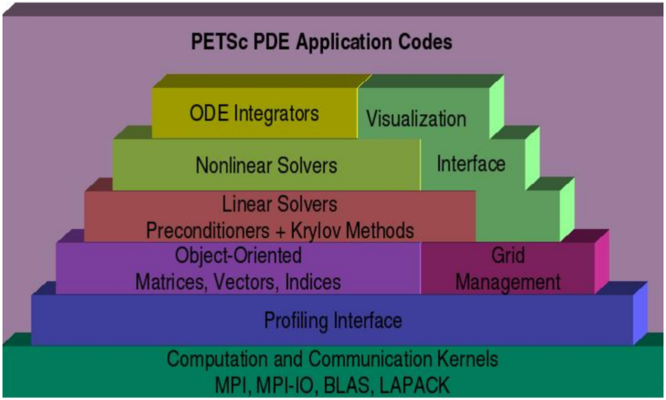
  PETSc, pronounced PET-see (the S is silent), is a suite of data structures and routines for the scalable (parallel) solution of scientific applications modeled by partial differential equations[8]
- Why use PETSc ?

  PETSc can solve (in parallel) the linear system we have at hand using a large number of time-stepping integrations schemes, linear solver schemes and preconditioners which can be dynamically configured at runtime! Moreover, higher order finite volume schemes can be explored as well. An optimization library is also included[9].

[8] https://www.mcs.anl.gov/petsc/

[9] Finite difference is the defualt method to evaluate the jacobian, but since PETSc is written in C, a source-to-source AD tool like Tapende can be used to give a function that evaluates the jacobian.

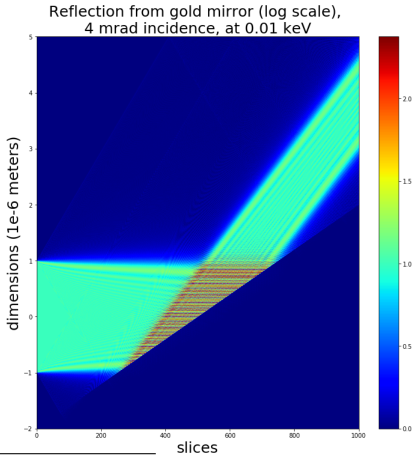
PETSc architecture[10]


PETSc PDE Application Codes
ODE Integrators
Visualization
Nonlinear Solvers
Interface
Linear Solvers
Preconditioners + Krylov Methods
Object-Oriented
Matrices, Vectors, Indices
Grid
Management
Profiling Interface
Computation and Communication Kernels
MPI, MPI-IO, BLAS, LAPACK

[10] https://www.mcs.anl.gov/petsc/documentation/tutorials/CSCS2010.pdf

Results 1D

- Reproduce the plot of x-ray reflectivity from Kenan et.al[11], using FD/PETSc.



[11] [LWJ17]

References I

[Fuh06] Christian Fuhse. *X-ray waveguides and waveguide-based lensless imaging*. PhD thesis, University of Göttingen, 2006.

[LWJ17] Kenan Li, Michael Wojcik, and Chris Jacobsen. Multislice does it all;calculating the performance of nanofocusing x-ray optics. *Opt. Express*, 25(3):1831–1846, Feb 2017.

[MS17] Lars Melchior and Tim Salditt. Finite difference methods for stationary and time-dependent x-ray propagation. *Opt. Express*, 25(25):32090–32109, Dec 2017.

[SO91] R. Scarmozzino and R. M. Osgood. Comparison of finite-difference and fourier-transform solutions of the parabolic wave equation with emphasis on integrated-optics applications. *J. Opt. Soc. Am. A*, 8(5):724–731, May 1991.