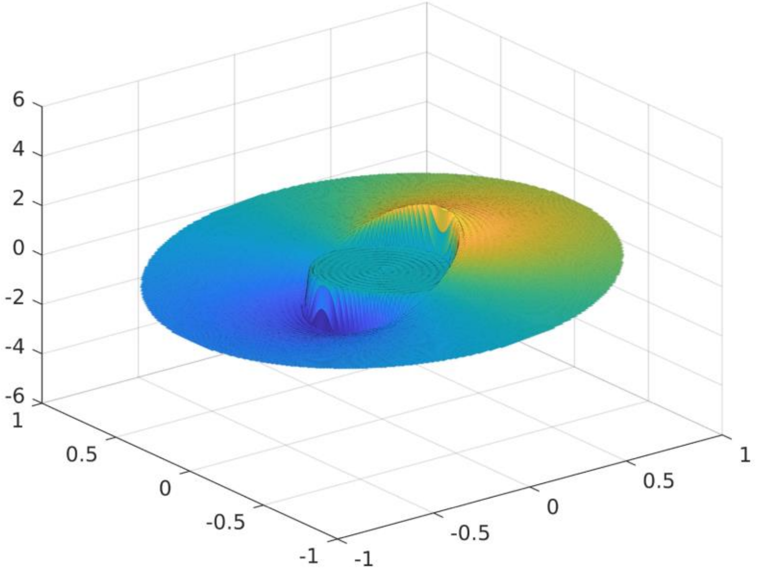
WAVE:
Comparing Numerical Methods on Cylindrical Wave Equation Solver Performance



APC523 Final Project

Michael D. Walker (mw6136)
Philip Satterthwaite (ps1639)
Michael Schroeder (ms2774)
PhD Candidates
Mech. & Aero. Engineering

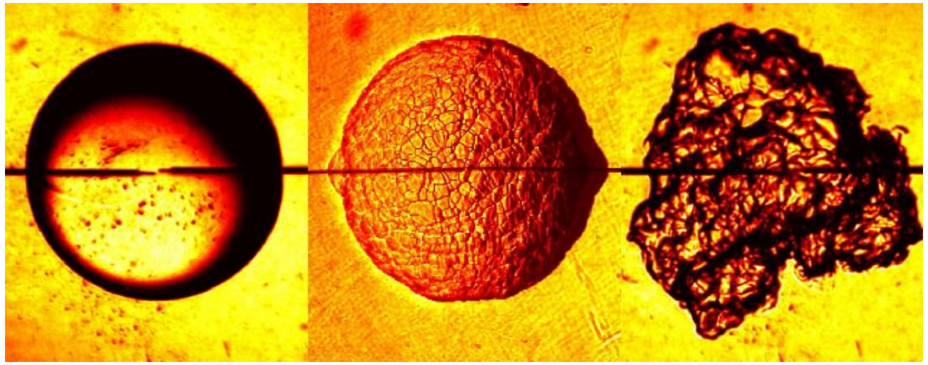
PRINCETON UNIVERSITY

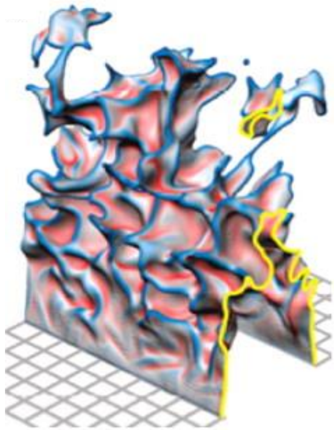
Kanagawa-oki Nami Ura. *The Great Wave off Kanagawa.*

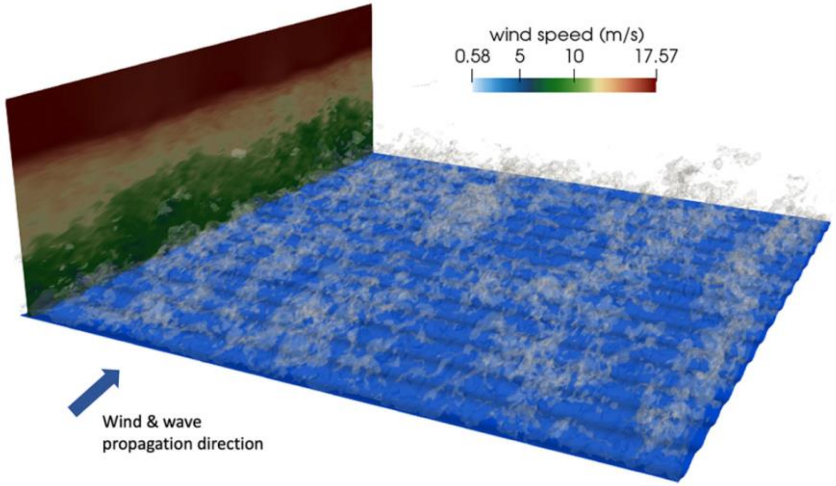
Motivation

The ability to understand and predict surface deformation is of great importance to a broad range of applications in fluid dynamics. Particular to our research (lab: CTRFL) are the ocean surface perturbed by wind, and combustion processes (modeled as an iso-surface of fast-reacting chemistry) perturbed by turbulence. Understanding these deformations requires solving high-order partial differential equations, a task that is often impossible to do mathematically and must be done numerically.

We seek to build a model of surface deformation by solving the scalar wave equation in cylindrical coordinates. Thereby resolving waves on the surface of a fluid generated by the motion of the walls. Further, we will investigate the trade-offs between speed, cost, accuracy, and stability of various numerical methods.







Law et al. *Three faces of a flame*. Driscoll, J. F. *Progr. Energy Combust. Sci.*, 2008. Deskos et al. *J. Atmos. Sci.*, 2021.

Contributions

Michael Walker (mw6136):
- Developed the full analytical solution shown in the Appendices.
- Created the `FFT_solver` using Spectral methods, and further developed improvements in speed and stability.
- Contributed the relevant sections of the report, majority of background & conclusions, most of the presentation slides, and the `README` for documenting and running the code.

Philip Satterthwaite (ps1639):
- Researched viability of time-stepping methods
- Developed code for Forward Euler and RK2 methods
- Contributed to relevant sections of the report and presentation

Michael Schroeder (ms2774):
- Coded analytical solution
  - Created plots and exported data
- Implemented, helped derive, and analyzed finite difference discretization method
- Analyzed applicability of iterative methods to this problem
- Contributed to relevant sections of the report and presentation

Problem Statement

For a small deformation of any planar circular elastic material, the surface deformation φ satisfies the wave equation in plane-polar coordinates where the driving force of the wave motion for a material initially at rest can be specified through the boundary and initial conditions. Seeking to model waves on the surface of the fluid generated by the motion of the walls, we choose the following boundary and initial conditions.

$$\frac{\partial^2 \varphi}{\partial t^2} = c^2 \left[\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial \varphi}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 \varphi}{\partial \theta^2} \right]$$

Boundary Conditions:

$\varphi(r = R, \theta, t) = A\cos(\omega t)\cos(\theta)$

$\varphi(r = 0, \theta, t) \rightarrow \text{finite}$

$\varphi(r, \theta + 2\pi, t) = \varphi(r, \theta, t)$

Initial Conditions:

$\varphi(r, \theta, t = 0) = 0$

$\partial\varphi/\partial t\,(r, \theta, t = 0) = 0$

We will solve for φ for all t, θ, and r using the method of separation of variables and eigenfunction expansions, then use properties of orthogonality to solve for coefficients.

Analytical Solution

Separation of variables and eigenfunction expansion gives the final solution:

$$\varphi(r,\theta,t) = A\left(\frac{r}{R}\right)^2 \cos(\omega t)\cos(\theta) + \tilde{\varphi}(r,\theta,t)$$

$$\tilde{\varphi}(r,\theta,t) = \sum_{m=1}^{\infty}\left(\frac{1}{c^2\lambda_{n,m}-\omega^2}\frac{2A}{R^2}\frac{\int_0^1 \left(\omega^2\eta^2+3c^2\right)\eta J_1\left(\sqrt{\lambda_{1,m}}R\eta\right)d\eta}{J_2{}^2\left(\sqrt{\lambda_{1,m}}R\right)}\cos(\omega t)\ \ldots\right.$$

$$-\frac{2A}{J_2{}^2\left(\sqrt{\lambda_{1,m}}R\right)}\left[\int_0^1 \eta^3 J_1\left(\sqrt{\lambda_{1,m}}R\eta\right)d\eta\ \ldots\right.$$

$$\left.\left.+\frac{1}{c^2\lambda_{n,m}-\omega^2}\frac{1}{R^2}\int_0^1\left(\omega^2\eta^2+3c^2\right)\eta J_1\left(\sqrt{\lambda_{1,m}}R\eta\right)d\eta\right]\cos\left(c\sqrt{\lambda_{1,m}}t\right)\right)\cos(\theta)J_1\left(\sqrt{\lambda_{1,m}}r\right)$$

By inspection of $\tilde{\varphi}(r,\theta,t)$, the coefficient in the summation goes to infinity and "blows up" as the denominator $c^2\lambda_{n,m}-\omega^2 \to 0$, thus a resonance condition exists when $c^2\lambda_{n,m}=\omega^2$. $c\sqrt{\lambda_{n,m}}$ can be interpreted as an eigenfrequency which, when excited, results in resonance.

See Appendix of project report for full derivation.

Discretization and Finite Difference

Define: $r = i\Delta r,\ \theta = j\Delta\theta, \text{ and } t = n\Delta t$

$$\varphi(r,\theta,t) = \varphi(i\Delta r, j\Delta\theta, n\Delta t) = \varphi\left(r_i, \theta_j, t^{(n)}\right) \equiv \varphi_{i,j}^{(n)}$$

Using forward difference (explicit method) in r and central differences in θ and t

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial\varphi}{\partial r}\right) = \frac{1}{r_i}\left(r_{i+1/2}\frac{\varphi_{i+1,j}^{(n)} - \varphi_{i,j}^{(n)}}{\Delta r^2} - r_{i-1/2}\frac{\varphi_{i,j}^{(n)} - \varphi_{i-1,j}^{(n)}}{\Delta r^2}\right)$$

$$\frac{1}{r^2}\frac{\partial^2\varphi}{\partial\theta^2} = \frac{1}{{r_i}^2}\left(\frac{\varphi_{i,j+1}^{(n)} - 2\varphi_{i,j}^{(n)} + \varphi_{i,j-1}^{(n)}}{\Delta\theta^2}\right)$$

$$\frac{\partial^2\varphi}{\partial t^2} = \frac{\varphi_{i,j}^{n+1} - 2\varphi_{i,j}^{n} + \varphi_{i,j}^{n-1}}{\Delta t^2}$$

where

$$r_{i+1/2} = 0.5\left(r_{i+1} + r_i\right)$$

$$\boxed{\varphi_{i,j}^{n+1} = \Delta t^2 c^2\left[\frac{1}{r_i}\left(r_{i+1/2}\frac{\varphi_{i+1,j}^{(n)} - \varphi_{i,j}^{(n)}}{\Delta r^2} - r_{i-1/2}\frac{\varphi_{i,j}^{(n)} - \varphi_{i-1,j}^{(n)}}{\Delta r^2}\right) + \frac{1}{{r_i}^2}\left(\frac{\varphi_{i,j+1}^{(n)} - 2\varphi_{i,j}^{(n)} + \varphi_{i,j-1}^{(n)}}{\Delta\theta^2}\right)\right] + 2\varphi_{i,j}^{n} - \varphi_{i,j}^{n-1}}$$

CFL condition: $c^2\dfrac{\Delta t}{\min(\Delta r^2, \Delta\theta^2)} \leq \dfrac{1}{2}$

Fourier Spectral Methods

2-D Fourier Transform in polar coordinates and its inverse:

$$\mathcal{F}(k_r, k_\theta) = \int_0^\infty \int_{-\pi}^{\pi} f(r,\theta) \exp(-irk_r \cos(k_\theta - \theta)) r \, dr d\theta$$

$$f(r,\theta) = \frac{1}{(2\pi)^2} \int_0^\infty \int_0^{2\pi} \mathcal{F}(k_r, k_\theta) \exp(i\langle k_r, k_\theta \rangle \cdot \langle r, \theta \rangle) r \, dk_r \, dk_\theta$$

The Laplacian Fourier pair in polar coordinates:

$$\nabla^2 f(r,\theta) = \frac{\partial^2 f}{\partial r^2} + \frac{1}{r}\frac{\partial f}{\partial r} + \frac{1}{r^2}\frac{\partial^2 f}{\partial \theta^2} \quad \longleftrightarrow \quad {k_r}^2 \mathcal{F}(k_r, k_\theta) + \frac{{k_\theta}^2}{r^2} \mathcal{F}(k_r, k_\theta)$$

Transform the scalar wave equation ($\mathcal{F}(\phi) = \hat{\phi}$) to obtain the **Helmholtz equation** with a time-forcing term:

$$\frac{\partial^2 \varphi}{\partial t^2} = c^2 \left[\frac{1}{r}\frac{\partial}{\partial r}\left(r \frac{\partial \varphi}{\partial r} \right) + \frac{1}{r^2}\frac{\partial^2 \varphi}{\partial \theta^2} \right] \quad \longleftrightarrow \quad \frac{\partial^2 \hat{\varphi}}{\partial t^2} = -c^2 \left({k_r}^2 \hat{\varphi} + \frac{{k_\theta}^2}{r^2} \hat{\varphi} \right)$$

1-D Helmholtz equation is a strict eigenvalue problem, where the eigenvalue k is the wavenumber $\quad \nabla^2 f = -k^2 f$

Fourier Spectral Methods (cont.)

Evolve the equation in time by choosing a simple Forward Euler iterative scheme:

$$\hat{\varphi}^{(n+1)} = 2\hat{\varphi}^{(n)} - \hat{\varphi}^{(n-1)} + (\Delta t)^2 \left[-c^2 \left({k_r}^2 + \frac{{k_\theta}^2}{r^2} \right) \right] \hat{\varphi}^{(n)}$$

The solution is obtained by then computing the inverse transform ($\mathcal{F}^{-1}(\hat{\phi}) = \phi$). The discrete Fast Fourier Transform (FFT) is computed using the `SciPy` package (`fft`, `ifft`, `fftfreq`) in `Python`.

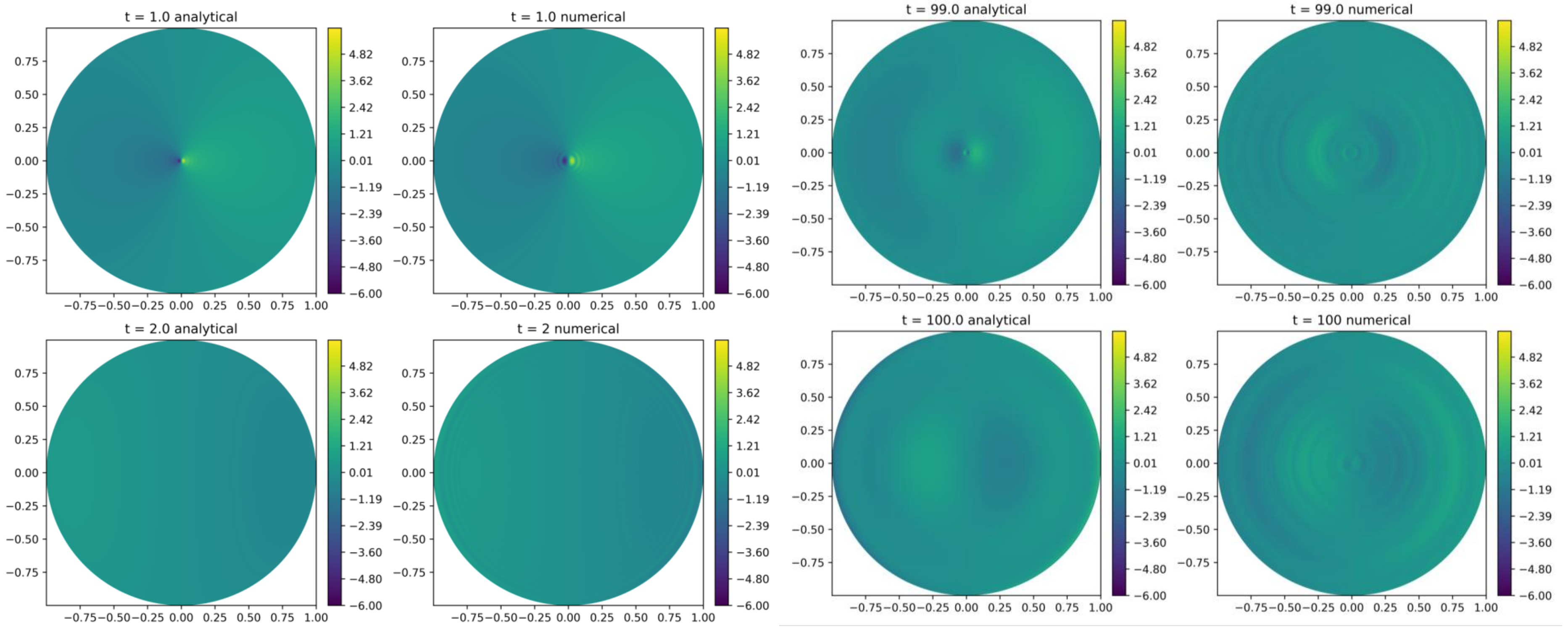
```
# Calculate frequencies
freq_r = fftfreq(num_r, dr)
freq_theta = fftfreq(num_theta, dtheta)

# Calculate Laplacian in frequency domain
k_r_squared = (2 * np.pi * freq_r) ** 2
k_theta_squared = (2 * np.pi * freq_theta) ** 2
laplacian = -c ** 2 * (k_r_squared[:, np.newaxis] + k_theta_squared)

# Update phi in frequency domain using Forward Euler
phi_hat_next = 2 * phi_hat_curr - phi_hat_prev + dt ** 2 * laplacian * phi_hat_curr

# Transform back to spatial domain
phi_next = np.real(ifft(phi_hat_next, axis=0))
```

Results

Finite Difference Discretization Method

- Reasonably accurately captures analytical solution
  - Short runtimes
  - Stable

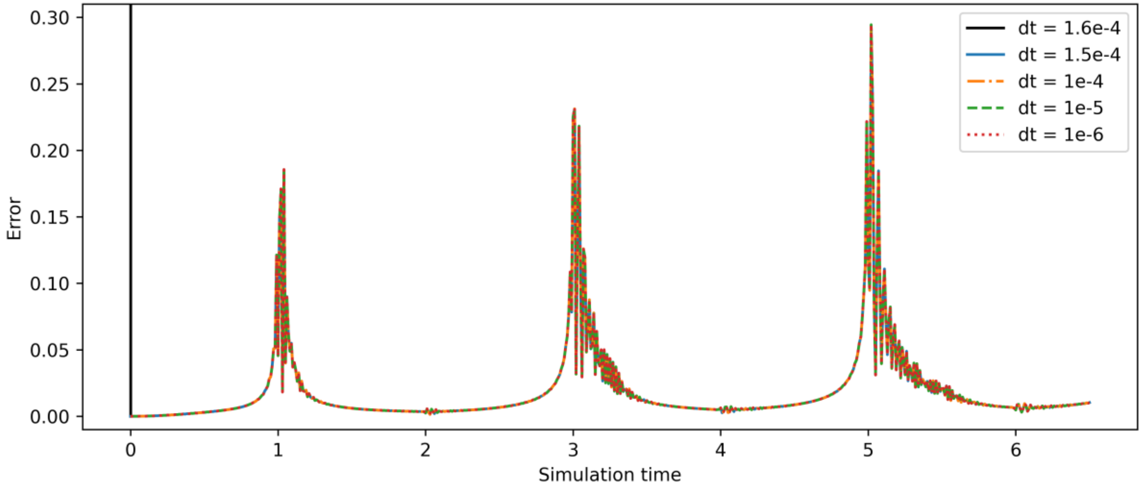
- Numerical accuracy decreases with time

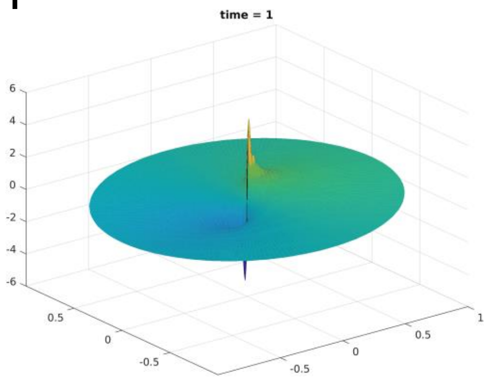


Finite Difference Discretization Method (cont.)

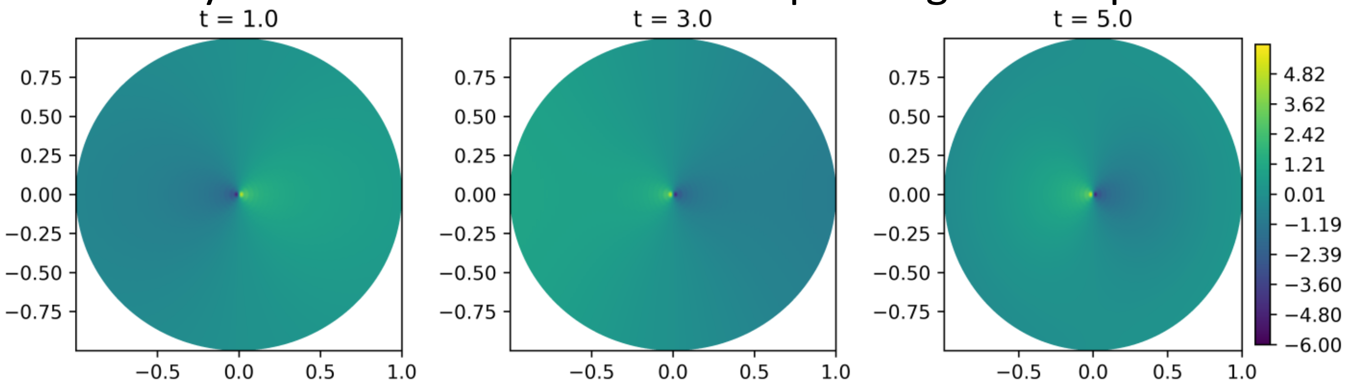
- Low mean square error
  - $MSE = \frac{1}{N^2}\sum_{i,j}\left(u_{i,j}^{anal.}(t) - u_{i,j}^{num.}(t)\right)^2$
- Stable solution for timesteps less than $1.5 \cdot 10^{-4}$
- The spikes in MSE correspond to large, high order derivatives in the analytical solution
  - Quickly changing peaks

Mean Square Error vs. Time at Multiple Δts



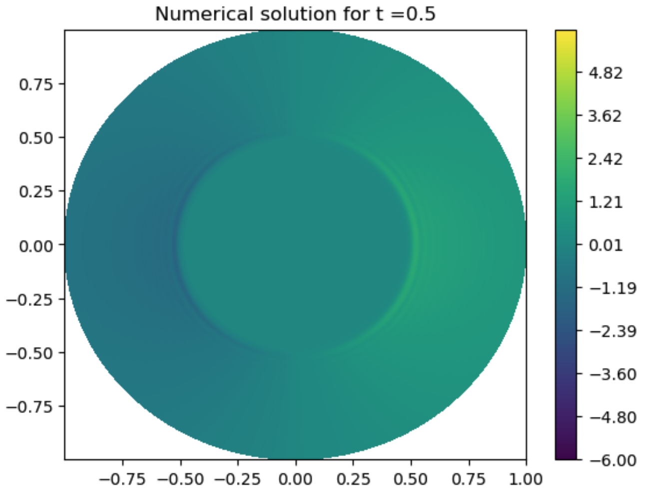


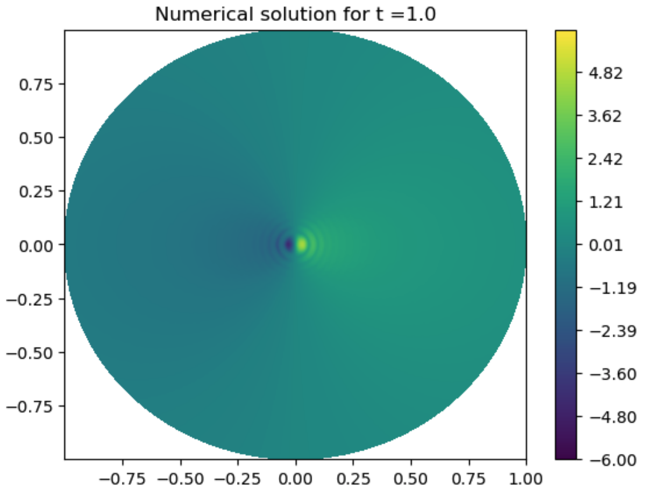
Analytical Solution at Times Corresponding to MSE peaks



Finite Difference Discretization Summary

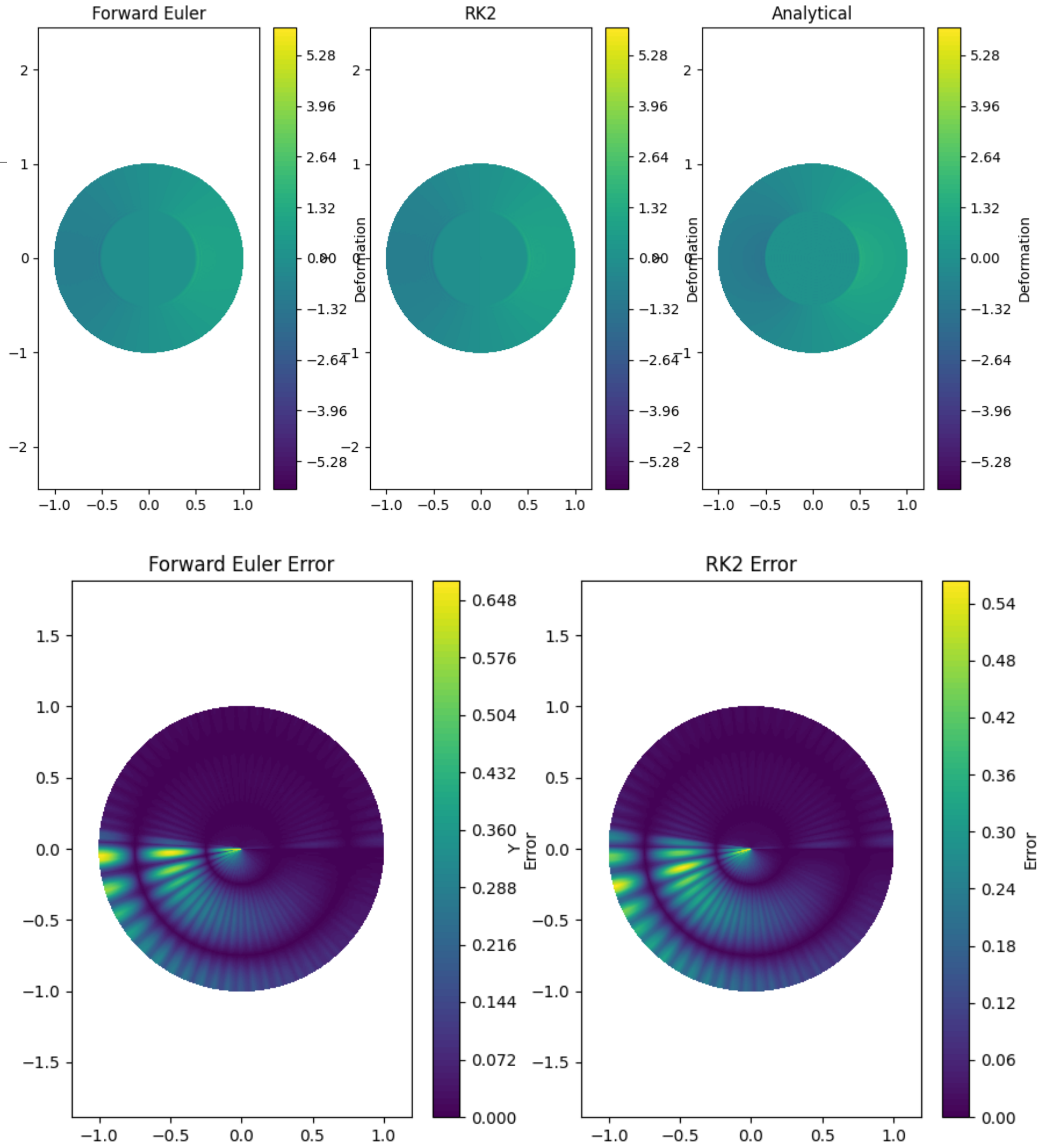
- Quick, accurate method
- Low error that increases in time
  - Increasing rippling effect
- Significantly quicker than analytical solution
- Runtime scales with Δt^{-1}
  - Lowest stable Δt should be used
    - Here, $1.5 \cdot 10^{-4}$





Time Stepping Methods

- Due to the complexity of the PDE, times-stepping methods become needlessly convoluted with respect to other (more stable) methods
- RK2 and Forward Euler methods were studied
- Due to the periodic nature of the PDE and boundary conditions, these methods often failed to converge and stability was highly dependent on spatial and temporal discretization.
- At $t = 0.5$, both converged to a qualitatively correct answer, but with significant error due to a still sub-optimal spatial discretization.

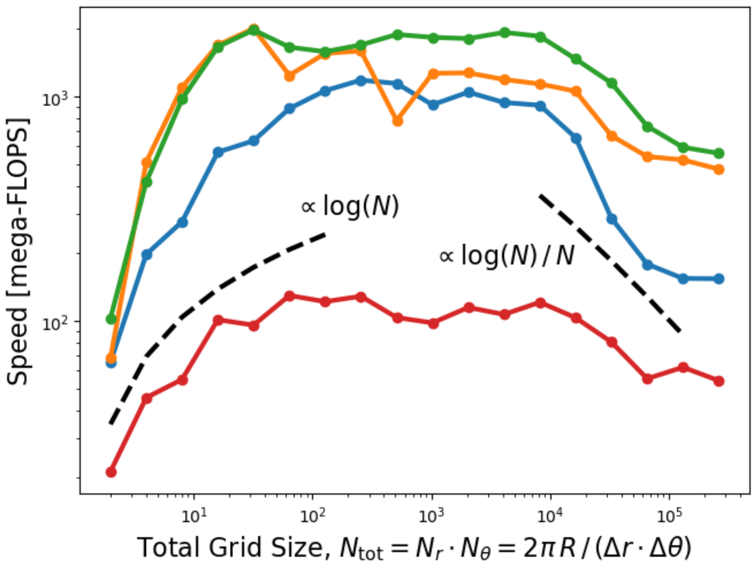


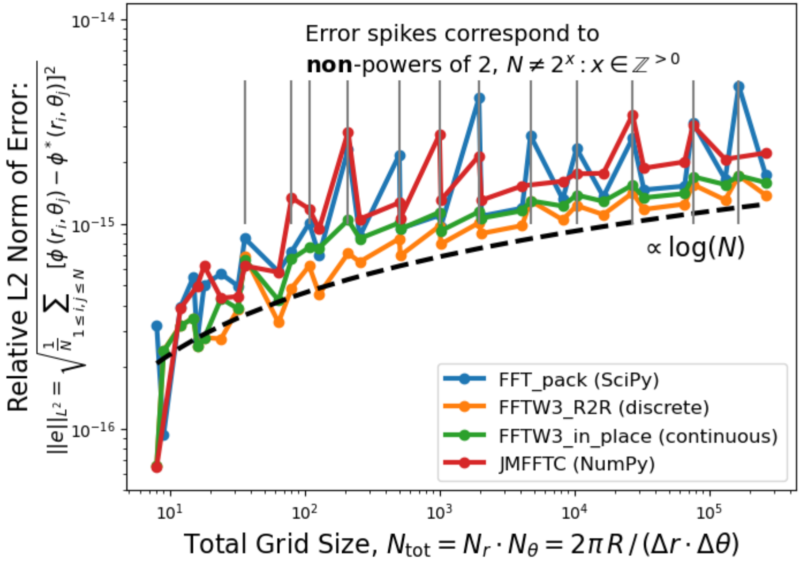
Fourier Spectral Methods

- Compared to other Fourier algorithms using the `benchFFT` software program on Princeton TIGER cluster (CPU-only HPE Linux on 408 2.4 GHz Skylake nodes, 40 cores/node).

- The Fourier transform and inverse have complexity $O(N\log(N))$. Scaling shows $O(\log(N))$ when plotted against the total grid size (excludes complexity of looping over grid) for 1-D transforms of real data up to double-precision.

- For large N (beyond $N = 2^{14}$) performance suffers due to memory allocation $O(\log(N)/N)$. Significant increases in error for discretizations not positive integer powers of 2.





Performance Engineering

- The accuracy of this numerical simulation is determined largely by the spatial resolution requirements. The timestep and numerical stability was therefore limited by this discretization and the speed of sound.

- Performance was optimized by limiting the number of looping functions and using `NumPy` vectorization wherever possible.

- Whenever major looping was required, `Numba` just-in-time compiler was used in “no Python mode”.

Future Work

- Numerical Improvements: **Speed** (spatially varying timesteps, residual averaging), **Accuracy** (higher-order smoothing, higher-order differencing, differed correction factors), **Stability** (high-order expansions and second-order term in time derivative)

- Now that the various numerical methods have been evaluated, this approach can be coupled to the *G*-equation, to provide a deterministic description of flame iso-surface perturbations in turbulent flows.

- Implement parallelization using protocols like `OpenMP` and `MPI`.

- Methods for more efficient mesh discretization (such as adaptive mesh refinement) were beyond the scope of this project, but could also be implemented to improve accuracy, particularly in the supersonic case.

Lessons Learned

- Gained significant proficiency in virtually all numerical method topics of this course. While not relevant to this project, the course content in massively parallel architectures was also extremely relevant to our research.

- Collaborate on writing code in a “linear” way when possible, such that merge conflicts are straightforward to resolve.

- Ideas of testing and performance must be kept in mind from the beginning (test-driven development!)

- We work in a computational group (the Computational Turbulent Reacting Flow Laboratory) and this course genuinely helped us understand the numerical methods implemented in our existing solvers, like `NGA`, `PDRs`, `SUNDIALS`, `HYPRE`, `LAPACK`, and `FFTW`, giving us more exposure to its syntax and code format that will be greatly useful to us in our future research.

Appendix:
Analytical Solution

Wave Equation Classification

The wave equation is a second-order linear partial differential equation (PDE) for the description of waves or standing wave fields. The scalar wave equation describes the mechanical wave propagation of any scalar φ

$$\frac{\partial^2\varphi}{\partial t^2} = c^2\left(\frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2} + \frac{\partial^2\varphi}{\partial z^2}\right) = c^2\,\nabla^2\,\varphi$$

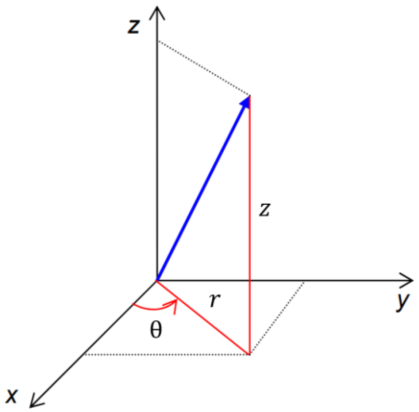
1-D: $$\frac{\partial^2\varphi}{\partial t^2} = c^2\,\frac{\partial^2\varphi}{\partial x^2}$$

$$A\frac{\partial^2\varphi}{\partial t^2} + B\frac{\partial^2\varphi}{\partial x\partial t} + C\frac{\partial^2\varphi}{\partial x^2} + D\frac{\partial\varphi}{\partial t} + E\frac{\partial\varphi}{\partial x} + F\varphi = 0$$

Projected in the 2nd order PDE general form gives $A = 1$, $B = 0$, and $C = -c^2$, and thus the discriminant $B^2 - 4AC > 0$ defines a hyperbolic PDE with a well-posed initial value problem.

Cylindrical Wave Equation

For circularly propagating waves, it is useful to project the wave equation in cylindrical coordinates.



$$y = r\sin\theta$$
$$x = r\cos\theta$$
$$r^2 = x^2 + y^2$$
$$\theta = \tan^{-1}(y/x)$$

$\longrightarrow$

$$\frac{\partial r}{\partial x} = \frac{x}{r} = \cos(\theta) \quad \frac{\partial r}{\partial y} = \frac{y}{r} = \sin(\theta)$$
$$\frac{\partial \theta}{\partial x} = -\frac{\sin(\theta)}{r} \quad \frac{\partial \theta}{\partial y} = \frac{\cos(\theta)}{r}$$

$\longrightarrow$

$$\frac{\partial \varphi}{\partial x} = \frac{\partial \varphi}{\partial r}\frac{\partial r}{\partial x} + \frac{\partial \varphi}{\partial \theta}\frac{\partial \theta}{\partial x}$$
$$\frac{\partial \varphi}{\partial y} = \frac{\partial \varphi}{\partial r}\frac{\partial r}{\partial y} + \frac{\partial \varphi}{\partial \theta}\frac{\partial \theta}{\partial y}$$

$$\nabla^2 = \frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r} + \frac{1}{r^2}\frac{\partial^2}{\partial \theta^2} + \frac{\partial^2}{\partial z^2}$$

$$\boxed{\frac{1}{c^2}\frac{\partial^2 \varphi}{\partial t^2} = \frac{\partial^2 \varphi}{\partial r^2} + \frac{1}{r}\frac{\partial \varphi}{\partial r} + \frac{1}{r^2}\frac{\partial^2 \varphi}{\partial \theta^2} + \cancel{\frac{\partial^2 \varphi}{\partial z^2}}}$$

This remains a hyperbolic PDE, but with a singular point at the origin $r = 0$. The study of surface deformation requires only plane-polar coordinates, so the $\partial^2\varphi/\partial z^2$ term may be neglected. We now seek an analytical solution to the 2-D equation, given well-posed boundary and initial conditions.

Analytical Solution

Separation of variables:

$$\varphi(r,\theta,t) = \underbrace{\mathcal{R}(r)}\,\underbrace{\Theta(\theta)}\,\underbrace{\mathcal{T}(t)}$$

$$\frac{1}{c^2}\frac{\mathcal{T}''}{\mathcal{T}} = -\lambda$$

The Bessel Equation:

$$r^2\mathcal{R}'' + r\mathcal{R}' + (\lambda r^2 - n^2)\mathcal{R} = 0$$

whose solution is the Bessel function:

$$\mathcal{R}(r) = cJ_n(\sqrt{\lambda}r) + \cancel{dY_n(\sqrt{\lambda}r)}$$

$$\Theta'' = -\alpha^2\Theta \to \Theta(\theta) = a\cos(\alpha\theta) + b\sin(\alpha\theta)$$

$$\to \alpha_n = n \in \mathbb{Z}^{\geq 0} \quad \text{an integer}$$

$$\Theta_n(\theta) = a_n\cos(n\theta) + b_n\sin(n\theta)$$

$$\boxed{\mathcal{R}(r)\,\Theta(\theta) = [a_{n,m}\cos(n\theta) + b_{n,m}\sin(n\theta)]J_n(\sqrt{\lambda_{n,m}}r)}$$

See Appendix of project report for full derivation.

Analytical Solution (cont.)

Rescaling to achieve homogeneous boundary conditions: $\tilde{\varphi} \equiv \varphi - A\left(\frac{r}{R}\right)^2 \cos(\omega t)\cos(\theta)$

Stationary Problem:

$$f(r,\theta,t) \equiv A\left(\frac{1}{R^2}\cos(\omega t)\cos(\theta)(\omega^2 r^2 + 3c^2)\right)$$

Eigenvalue expansion:

$$f(r,\theta,t) = \sum_{n=0}^{\infty}\sum_{m=0}^{\infty}[c_{n,m}(t)\cos(n\theta) + d_{n,m}(t)\sin(n\theta)]J_n(\sqrt{\lambda_{n,m}}r)$$

Solving for coefficients with orthogonality (inner product):

$$c_{n,m}(t) = \frac{\langle f(r,\theta,t), J_n(\sqrt{\lambda_{n,m}}r)\cos(n\theta)\rangle_r}{\langle J_n(\sqrt{\lambda_{n,m}}r)\cos(n\theta), J_n(\sqrt{\lambda_{n,m}}r)\cos(n\theta)\rangle_r} = \frac{\int_0^R\int_0^{2\pi} f(r,\theta,t) r J_n(\sqrt{\lambda_{n,m}}r)\cos(n\theta)dr\,d\theta}{\int_0^R\int_0^{2\pi} r {J_n}^2(\sqrt{\lambda_{n,m}}r)\cos^2(n\theta)dr\,d\theta}$$

$$d_{n,m}(t) = \frac{\langle f(r,\theta,t), J_n(\sqrt{\lambda_{n,m}}r)\sin(n\theta)\rangle_r}{\langle J_n(\sqrt{\lambda_{n,m}}r)\sin(n\theta), J_n(\sqrt{\lambda_{n,m}}r)\sin(n\theta)\rangle_r} = \frac{\int_0^R\int_0^{2\pi} f(r,\theta,t) r J_n(\sqrt{\lambda_{n,m}}r)\sin(n\theta)dr\,d\theta}{\int_0^R\int_0^{2\pi} r {J_n}^2(\sqrt{\lambda_{n,m}}r)\sin^2(n\theta)dr\,d\theta}$$

$$\boxed{f(r,\theta,t) = \frac{2A}{R^2}\cos(\omega t)\sum_{m=1}^{\infty}\frac{\int_0^1(\omega^2\eta^2 + 3c^2)\eta J_1(\sqrt{\lambda_{1,m}}R\eta)d\eta}{{J_2}^2(\sqrt{\lambda_{1,m}}R)}\cos(\theta)J_1(\sqrt{\lambda_{n,m}}r)}$$

See Appendix of project report for full derivation.

Analytical Solution (cont.)

Expanding $\tilde{\phi}$:

$$\tilde{\varphi}(r,\theta,t) = \sum_{n=0}^{\infty}\sum_{m=1}^{\infty}[a_{n,m}(t)\cos(n\theta) + b_{n,m}(t)\sin(n\theta)]J_n(\sqrt{\lambda_{n,m}}r)$$

$$\frac{\partial^2\tilde{\varphi}}{\partial t^2} - \frac{A}{R^2}\cos(\omega t)\cos(\theta)(\omega^2 r^2 + 3c^2) = c^2\left[\frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial\tilde{\varphi}}{\partial r}) + \frac{1}{r^2}\frac{\partial^2\tilde{\varphi}}{\partial\theta^2}\right]$$

Solving for coefficients with orthogonality (inner product) and initial conditions:

$$\begin{aligned}
a_{n,m}(0) &= \frac{\langle -A\,\frac{r^2}{R^2}\cos(\theta), J_n(\sqrt{\lambda_{n,m}}r)\cos(n\theta)\rangle_r}{\langle J_n(\sqrt{\lambda_{n,m}}r)\cos(n\theta), J_n(\sqrt{\lambda_{n,m}}r)\cos(n\theta)\rangle_r} \\
&= \frac{\int_0^R\int_0^{2\pi} -A(\frac{r}{R})^2\cos(\theta)rJ_n(\sqrt{\lambda_{n,m}}r)\cos(n\theta)\,dr\,d\theta}{\int_0^R\int_0^{2\pi} rJ_n{}^2(\sqrt{\lambda_{n,m}}r)\cos^2(n\theta)\,dr\,d\theta} \\
&= \frac{\int_0^R -A(\frac{r}{R})^2 rJ_n(\sqrt{\lambda_{n,m}}r)\,dr\int_0^{2\pi}\cos(\theta)\cos(n\theta)\,d\theta}{\frac{1}{2}\pi R^2 J_2{}^2(\sqrt{\lambda_{n,m}}R)}
\end{aligned}$$

$$b_{n,m}(0) = 0$$

$$\begin{aligned}
a_{1,m}(t) &= \kappa_3\cos(c\sqrt{\lambda_{n,m}}t) + \kappa_4\sin(c\sqrt{\lambda_{n,m}}t) \\
&\quad + \frac{1}{c^2\lambda_{n,m} - \omega^2}\frac{2A}{R^2}\frac{\int_0^1(\omega^2\eta^2 + 3c^2)\,\eta J_1\left(\sqrt{\lambda_{1,m}}R\eta\right)d\eta}{J_2{}^2\left(\sqrt{\lambda_{1,m}}R\right)}\cos(\omega t)
\end{aligned}$$

See Appendix of project report for full derivation.

Analytical Solution (cont.)

Thus the final solution:

$$\varphi(r,\theta,t) = A\left(\frac{r}{R}\right)^2 \cos(\omega t)\cos(\theta) + \tilde{\varphi}(r,\theta,t)$$

$$\tilde{\varphi}(r,\theta,t) = \sum_{m=1}^{\infty}\left(\frac{1}{c^2\lambda_{n,m}-\omega^2}\frac{2A}{R^2}\frac{\int_0^1\left(\omega^2\eta^2+3c^2\right)\eta J_1\left(\sqrt{\lambda_{1,m}}R\eta\right)d\eta}{{J_2}^2\left(\sqrt{\lambda_{1,m}}R\right)}\cos(\omega t)\ \ldots\right.$$

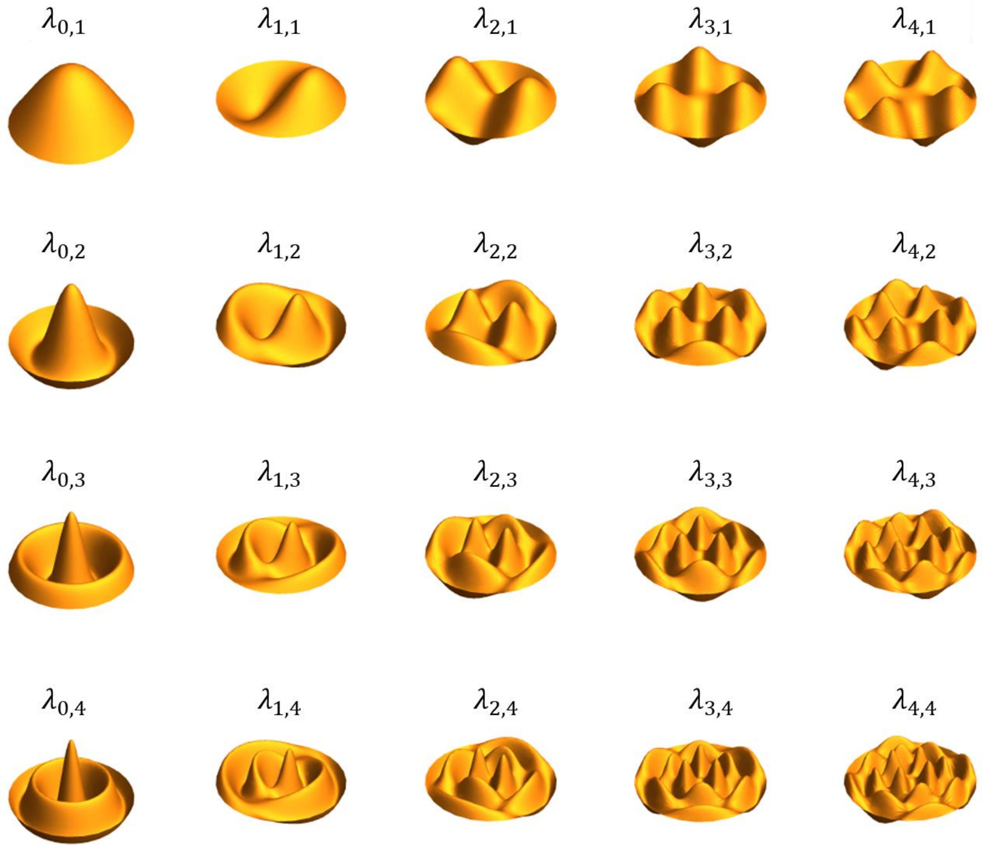
$$-\frac{2A}{{J_2}^2\left(\sqrt{\lambda_{1,m}}R\right)}\left[\int_0^1 \eta^3 J_1\left(\sqrt{\lambda_{1,m}}R\eta\right)d\eta\ \ldots\right.$$

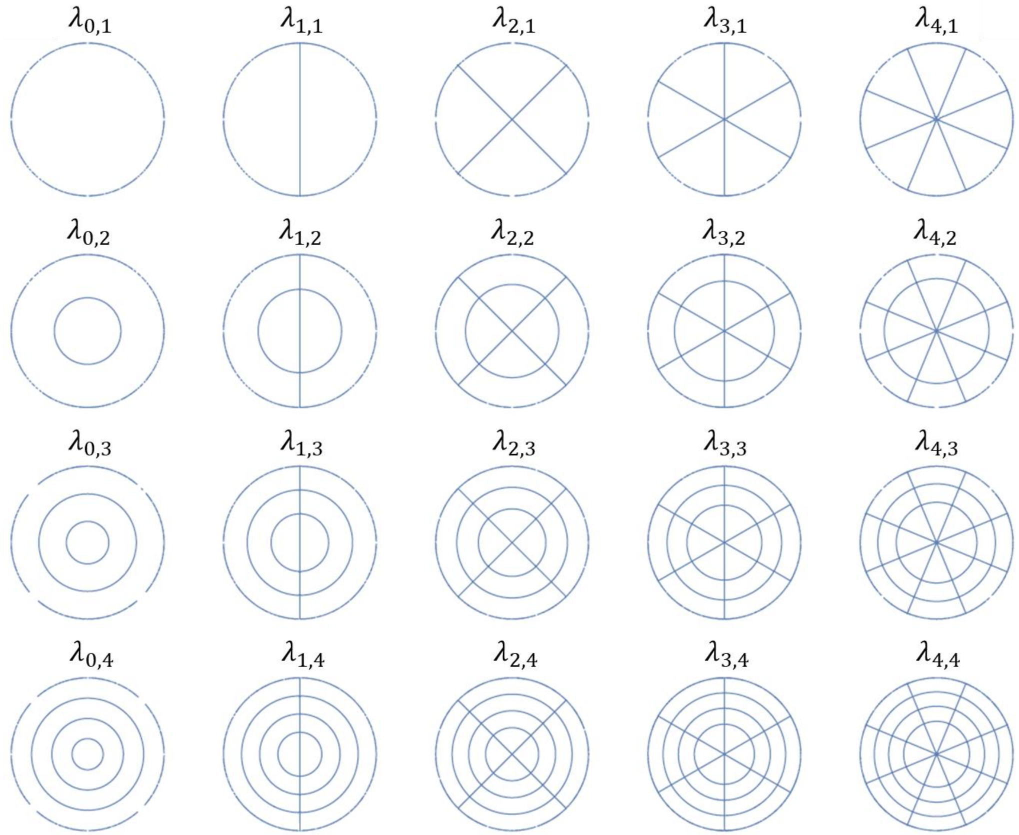
$$\left.\left.+\frac{1}{c^2\lambda_{n,m}-\omega^2}\frac{1}{R^2}\int_0^1\left(\omega^2\eta^2+3c^2\right)\eta J_1\left(\sqrt{\lambda_{1,m}}R\eta\right)d\eta\right]\cos\left(c\sqrt{\lambda_{1,m}}t\right)\right)\cos(\theta)J_1\left(\sqrt{\lambda_{1,m}}r\right)$$

By inspection of $\tilde{\varphi}(r,\theta,t)$, the coefficient in the summation goes to infinity and "blows up" as the denominator $c^2\lambda_{n,m}-\omega^2 \to 0$, thus a resonance condition exists when $c^2\lambda_{n,m}=\omega^2$. $c\sqrt{\lambda_{n,m}}$ can be interpreted as an eigenfrequency which, when excited, results in resonance.

See Appendix of project report for full derivation.

Fundamental Nodes





The whole solution is a linear combination of *standing waves.* The fundamental vibrations (left) are shown for radially derelict boundary conditions ($\phi(r = R, \theta, t) = 0$). The nodal curves (right) show the set for points for which $\phi(r, \theta) = 0 \; \forall t$, dividing the domain into nodal regions. Recall that that $\lambda_{n,m}$ is the eigenmode for the m-th root of the n-th order Bessel function J_n for $m = 1, 2, 3, 4$.