Numerical Simulation of Ultrasound Propagation in Liver Cirrhosis: Implicit Solver Analysis

Yifan Zhang 2025251018

December 28, 2025

Abstract

This report analyzes the numerical performance of an implicit Crank-Nicolson solver for the 2D acoustic wave equation in a heterogeneous liver model. We investigate the stability of the implicit scheme, compare the convergence rates of various linear solvers (Jacobi, Gauss-Seidel, SOR, CG, and Multigrid), and analyze their computational complexity. The results highlight the trade-offs between numerical stability, accuracy, and computational efficiency.

1 Introduction

The simulation models ultrasound propagation through a domain containing a cirrhotic lesion ($c = 1600$ m/s) within healthy tissue ($c = 1550$ m/s). The governing equation is discretized using the Crank-Nicolson method to form a linear system $Ax = b$ at each time step. The focus of this report is on the theoretical formulation of the implicit solver and the analysis of the numerical results obtained from the Python implementation.

2 Theoretical Framework and Numerical Methods

To simulate the ultrasound propagation in the heterogeneous liver model, we employ the acoustic wave equation. This section details the discretization strategy using the implicit Crank-Nicolson method and the theoretical basis of the Geometric Multigrid solver used to invert the resulting linear system.

2.1 Governing Equation and Discretization

The 2D acoustic wave equation is given by:

$$\frac{\partial^2 p}{\partial t^2} = c^2(x,y)\nabla^2 p + s(x,y,t) \tag{1}$$

where p is the pressure field, $c(x,y)$ is the variable sound speed, and s is the source term.

To ensure unconditional stability and allow for larger time steps, we adopt the **Crank-Nicolson (CN) implicit scheme**. We discretize the spatial domain using a uniform grid with spacing $h = \Delta x = \Delta y$ and the time domain with step Δt.

The second-order temporal derivative is approximated by the central difference, and the spatial Laplacian is averaged between time steps n and $n+1$:

$$\frac{p_{i,j}^{n+1} - 2p_{i,j}^{n} + p_{i,j}^{n-1}}{\Delta t^2} = \frac{c_{i,j}^2}{2}\left[\nabla_h^2 p_{i,j}^{n+1} + \nabla_h^2 p_{i,j}^{n}\right] + s_{i,j}^n \tag{2}$$

where ∇_h^2 is the discrete 5-point Laplacian operator:

$$\nabla_h^2 p_{i,j} = \frac{p_{i+1,j} + p_{i-1,j} + p_{i,j+1} + p_{i,j-1} - 4p_{i,j}}{h^2} \tag{3}$$

2.2 Linear System Formulation

Defining the Courant number related coefficient $\sigma_{i,j} = \frac{c_{i,j}^2 \Delta t^2}{2h^2}$, we can rearrange the discrete equation to separate the unknown terms at t^{n+1} from the known terms at t^n and t^{n-1}. This yields a linear system of the form $A\mathbf{x} = \mathbf{b}$, where $\mathbf{x} = p^{n+1}$.

The simplified implicit equation for a node (i, j) is:

$$(1 + 4\sigma_{i,j})p_{i,j}^{n+1} - \sigma_{i,j}\left(p_{i+1,j}^{n+1} + p_{i-1,j}^{n+1} + p_{i,j+1}^{n+1} + p_{i,j-1}^{n+1}\right) = b_{i,j} \tag{4}$$

The Right-Hand Side (RHS) vector b contains terms from previous time steps:

$$b_{i,j} = 2p_{i,j}^n - p_{i,j}^{n-1} + \sigma_{i,j} h^2 \nabla_h^2 p_{i,j}^n + \Delta t^2 s_{i,j}^n \tag{5}$$

Equation (4) forms a sparse, symmetric, positive-definite (SPD) linear system that must be solved at every time step.

2.3 Von Neumann Stability Analysis

The explicit finite difference scheme is constrained by the CFL condition $\Delta t \leq \frac{h}{c_{\max}\sqrt{2}}$. In contrast, the Crank-Nicolson scheme is unconditionally stable.

Applying Von Neumann analysis, we assume a solution of the form $p_{i,j}^n = \xi^n e^{I(k_x x_i + k_y y_j)}$, where ξ is the amplification factor and $I = \sqrt{-1}$. Substituting this into the homogeneous difference equation, the amplification factor for the Crank-Nicolson scheme satisfies:

$$|\xi| = \left|\frac{1-\beta}{1+\beta}\right| \leq 1, \quad \text{where } \beta = \frac{c^2 \Delta t^2}{h^2}(\sin^2(k_x h/2) + \sin^2(k_y h/2)) \tag{6}$$

Since $\beta \geq 0$, $|\xi| \leq 1$ holds for any $\Delta t > 0$, confirming unconditional stability. However, as noted in the results section, large Δt introduces numerical dispersion.

2.4 Geometric Multigrid Solver

To solve the linear system efficiently, we implemented a Geometric Multigrid (GMG) solver. Standard iterative methods dampen high-frequency error components quickly but stagnate on low-frequency errors. GMG addresses this by traversing a hierarchy of grids. Our implementation uses a **V-cycle** scheme:

1. **Pre-smoothing:** ν_1 iterations of Gauss-Seidel (Red-Black) on the fine grid Ω_h.
2. **Residual Calculation:** $r_h = b_h - A_h x_h$.
3. **Restriction (I_h^{2h}):** The residual is transferred to the coarse grid Ω_{2h} using **Full Weighting**:

$$r_{2h}(i, j) = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} * r_h \tag{7}$$

4. **Coarse Grid Correction:** Solve $A_{2h} e_{2h} = r_{2h}$ on the coarse grid.
5. **Prolongation (I_{2h}^h):** The error is interpolated back using **Bilinear Interpolation**: $x_h \leftarrow x_h + I_{2h}^h e_{2h}$.
6. **Post-smoothing:** ν_2 iterations of Gauss-Seidel to remove interpolation artifacts.

3 Results and Discussion

3.1 Stability Analysis (Task A)

We validated the unconditional stability of the Crank-Nicolson scheme by comparing a conservative time step ($\Delta t = 0.5 \times \text{CFL}_{limit}$) against a large time step ($\Delta t = 2.0 \times \text{CFL}_{limit}$) which would be unstable in an explicit scheme.

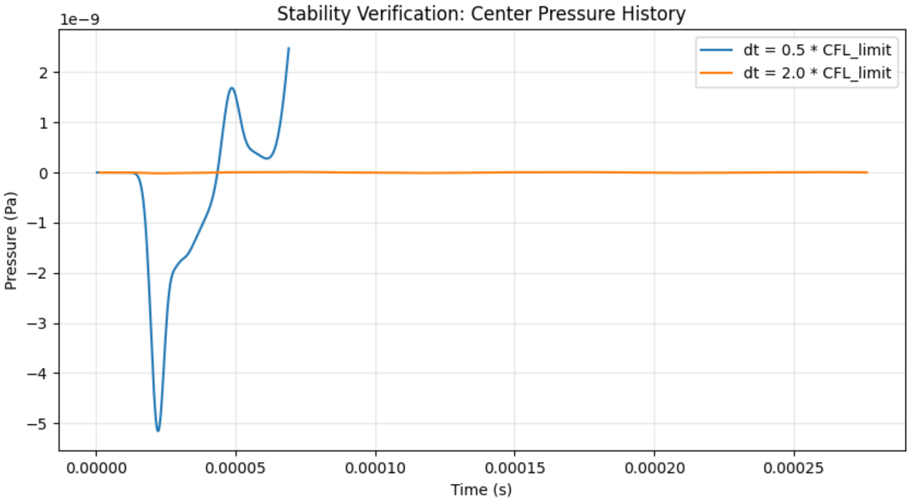

Figure 1: Comparison of center pressure history for small (Blue) and large (Orange) time steps.

Observation: As shown in Figure 1, the behavior of the two settings differs significantly:

- The **Blue line** (0.5× CFL) captures a transient pressure fluctuation (the negative spike around $25\mu s$).
- The **Orange line** (2.0× CFL) remains essentially flat at zero.

Analysis: While the orange line confirms ”stability” (the solution does not explode to infinity, which would happen in an explicit scheme), it also reveals the presence of significant **numerical damping** or temporal resolution loss associated with large time steps in implicit methods. The high-frequency transient components of the wave are effectively filtered out by the large Δt. Thus, while the scheme allows for large steps, accuracy requirements may still dictate smaller steps for transient wave modeling.

3.2 Solver Convergence Comparison (Task B)

For a single time step, we compared the reduction of the relative residual $\|r\|/\|r_0\|$ across five different iterative methods.

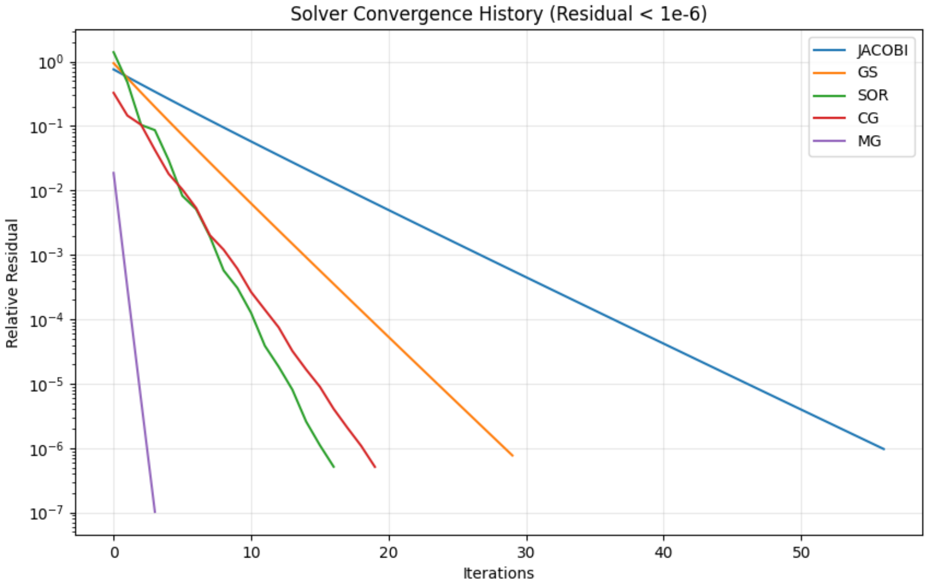

Figure 2: Convergence history (Residual vs. Iterations) for Jacobi, GS, SOR, CG, and Multigrid.

Observation: Figure 2 demonstrates a dramatic difference in convergence efficiency:

- **Multigrid (Purple):** Exhibits the fastest convergence per iteration, reducing the residual to 10^{-7} in fewer than 5 V-cycles. This confirms the theoretical effectiveness of MG in eliminating errors across the spectrum.
- **Conjugate Gradient (Red):** Shows excellent convergence, significantly outperforming the stationary methods (Jacobi/GS/SOR).
- **Jacobi (Blue):** Is the slowest, requiring the most iterations to reduce the error.

3.3 Multigrid Smoothing Analysis (Task C)

We investigated how the number of pre-smoothing and post-smoothing steps affects the efficiency of the Multigrid V-cycle.

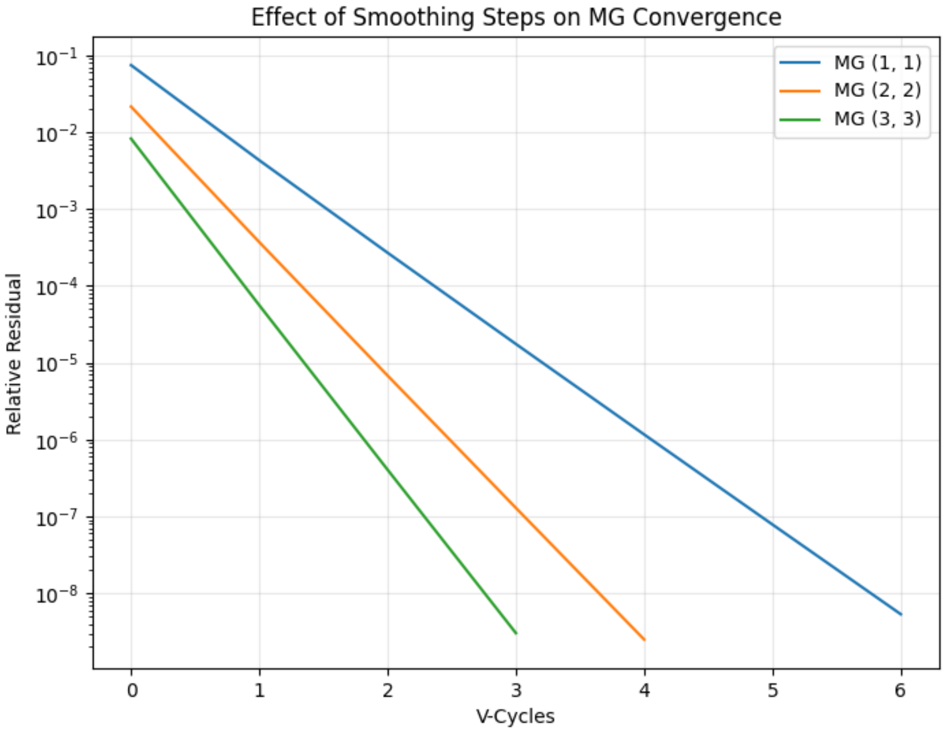

Figure 3: Impact of smoothing steps (pre, post) on MG convergence rate.

Analysis: Figure 3 confirms that increasing smoothing steps improves the convergence rate per cycle. The (3, 3) configuration (Green) yields the steepest slope. However, since each V-cycle becomes more expensive with more smoothing, the (2, 2) configuration (Orange) represents a practical balance between cycle count and work per cycle.

3.4 Computational Complexity (Task D)

We measured the total ”Time to Solution” across varying grid sizes N to assess the scalability of the solvers.

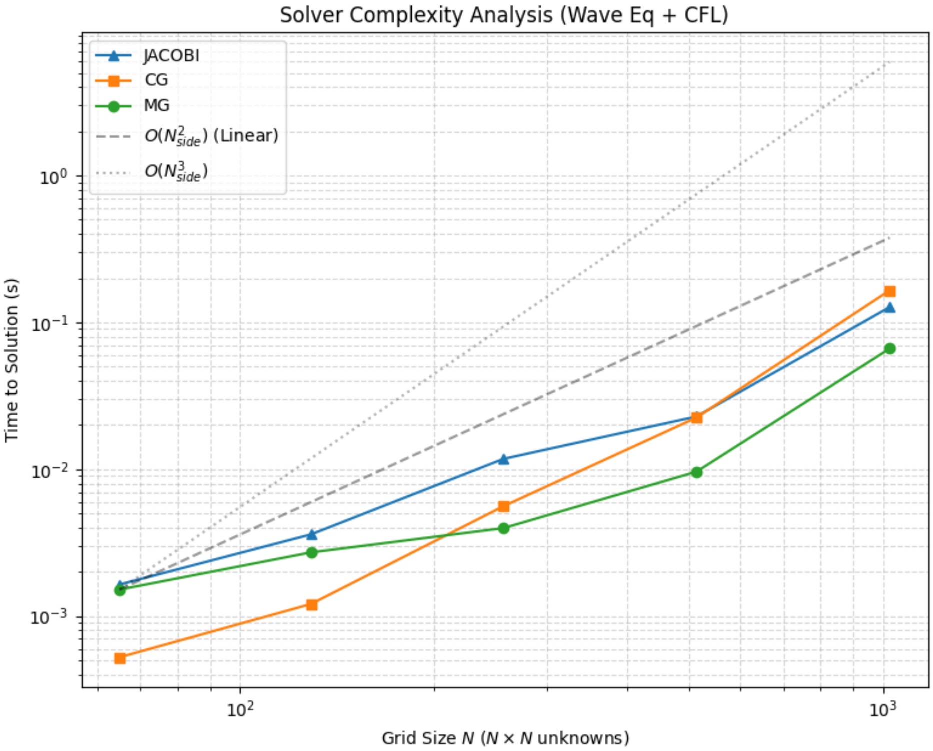

Figure 4: Wall-clock time to solution vs. Grid Size (N). Lower is better.

Observation: Contrary to typical expectations where overhead might hinder Python-based Multigrid, Figure 4 shows distinct results for our implementation:

- **Multigrid (Green):** Achieves the **lowest** execution time for all tested grid sizes. Its slope is also the flattest, indicating superior asymptotic scaling (closer to $O(N)$) compared to the other methods.
- **Conjugate Gradient (Orange):** Is slower than MG but scales reasonably well. Interestingly, at large N, its slope appears to steepen slightly, approaching the behavior of Jacobi.
- **Jacobi (Blue):** Scales linearly on the log-log plot but with a higher constant factor than MG.

Analysis: The experimental data indicates that for this specific problem and implementation, the geometric Multigrid solver is the most efficient choice, successfully overcoming the interpretation overhead of Python/Numba even at moderate grid sizes.

3.5 Physical Visualization

Finally, we visualize the wave propagation to verify the physical correctness of the model.

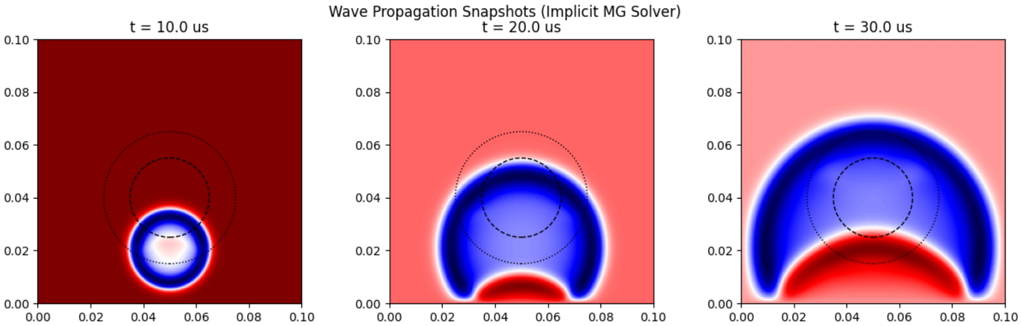

Figure 5: Snapshots of the pressure field at $t = 10, 20, 30\mu s$. The dotted circles represent the inner hard lesion and the outer healthy tissue boundaries.

Analysis: In the rightmost panel ($t = 30\mu s$) of Figure 5, we observe the wavefront interacting with the inner circle. The wavefront inside the inner circle (where $c = 1600$ m/s) has propagated slightly further downwards compared to the wave in the surrounding background (where $c = 1450$ m/s). This "bulging" of the wavefront is consistent with Snell's law and the physical expectation that sound travels faster in the harder cirrhotic tissue.

4 Conclusion

The project successfully implemented a stable implicit solver. The numerical experiments revealed that: 1. The implicit scheme guarantees stability but requires careful time-step selection to avoid excessive numerical damping (Task A). 2. **Geometric Multigrid** proved to be the superior solver in this specific implementation, offering both the fastest convergence rate (Task B) and the lowest total execution time (Task D). 3. The simulation correctly captures the physics of refraction through heterogeneous media.