A NUMERICAL IMPLEMENTATION OF THE SPACE-TIME FINITE ELEMENTS METHOD FOR THE 1+1 KLEIN-GORDON EQUATION

HYUN LIM
DEPARTMENT OF PHYSICS
HYUN.LIM@JACKS.SDSTATE.EDU
605.592.0441

AND JUNG-HAN KIMN (ADVISOR)
DEPARTMENT OF MATHEMATICS & STATISTICS
JUNG-HAN.KIMN@SDSTATE.EDU
605.688.5842

SOUTH DAKOTA STATE UNIVERSITY
BROOKINGS, SD 57007

Abstract. We have implemented a fully implicit numerical approach based on space-time finite element methods for the Klein-Gordon equation in the 1(space)+1(time) dimension. The purpose of this paper is to present a stable and parallelizable numerical method. The proposed numerical method is applied to generate successful simulation results of spin-0 particle propagation in a charged scalar field. The time additive Schwarz method is vital to make successful simulations with KSP (Krylov Subspace Methods) solvers. The time parallelizable algorithm is implemented through PETSc(Portable, Extensible, Toolkit for Scientific Computation, developed by Argonne National Laboratory).

1. Introduction.

The Klein-Gordon(KG) equation is a partial differential equation(PDE) that governs the quantum evolution of wave functions for relativistic spinless particles. The KG equation describes a wide variety of physical phenomena. In paper [1], the KG equation includes classical wave systems such as the displacement of a string attached to an elastic bed and quantum systems based on scalar fields. The KG equation is also related to the Dirac equation. Modeling light matter interaction with the relativistic effect can be explained by the Dirac equation, and the KG equation can be applied to the relativistic effect.

For another example, the electron-positron pair creation process in the supercritical breakdown of the fermionic vacuum is a striking prediction of the Dirac equation in [2]. It turns out that a quantum field theory based on the KG equation can predict similar phenomena from the Dirac equation in [3]. Therefore, the KG equation can provide new physical explanations about the universe, and can also provide clues to the Dirac equation because any solution to the Dirac equation is automatically a solution to the KG equation.

In addition, the KG equation has many applications. The KG equation can be modified to a non-homogeneous model and a nonlinear model. A modified version of the KG equation can be used for many engineering models to explain solitary waves, and wave propagations. Paper [4] shows the Klein-Gordon system transmitting monochromatic waves. Paper [5] shows an Einstein-Klein-Gordon system (a massive scalar field coupled to gravity) responding to a numerical simulation of oscillation.

Solutions of the KG equation both homogeneous and non-homogeneous are generally unavailable in [6]. Therefore, numerical simulation is required to estimate solutions. Papers [7] introduces numercial methods to the KG equation and Klein-Gordon-Schrödinger(KGS) equation without a forcing term. Paper [8] shows numerical approaches to the KGS equation without a forcing term.

In this paper, we present to generate successful simulation results of spin-0 particle propagation in a charged scalar field. This problem is defined as a KG equation with forcing term. To efficiently solve the KG equation using numerical methods, we use various numerical procedures. First of all, we consider numerical stability, which is vital to simulate time dependent problems. Therefore, we implement a fully implicit method through space time finite element methods, which solved for a nonhomogeneous wave equation in the 3+1 dimension successfully in [9]. We discretize space and time together for the entire domain using a finite element space which does not separate time and space basis functions.

Because of the large size of the discretized problems, we need to use iterative numerical methods based on Krylov subspace (KSP) methods. A proper preconditioner is also needed to help convergence and increase the speed of convergence. Therefore, we consider the time additive Schwarz preconditioner. Domain decomposition (DD) methods in [10, 11] provide a very natural way of deriving parallel algorithms for the numerical solutions of linear systems and can be used to a preconditioner for iterative methods. When DD algorithms are used, a large number of subproblems can be solved in parallel. The local interaction is through the exchange of information between neighboring subdomains. Our preconditioner will be implemented based on domain decomposition approaches but we decompose the time instead of the domain. We will show the numerical simulation for the KG equation using the time decomposition method.

The paper is organized as follows: in Section 2, we review the background of the KG equation; in Section 3, we introduce our implementation ingredients: the 1+1 space-time finite element discretization and a time parallelizable preconditioner based on the additive Schwarz idea; in Section 4, we numerically test example problems of the KG equation in the 1+1 dimension; in Section 5, we test the 1+1 KG equation with traditional numerical methods; in Section 6, we conclude by discussing future research opportunities.

2. Klein-Gordon equation. The description of phenomena at high energies requires the investigation of relativistic equations. The Klein-Gordon equation is a relativistic version of the Schrödinger equation. The KG equation explains the motion of a quantum scalar or pseudoscalar field in which quanta are spinless particles. The KG equation describes the quantum amplitude for finding a point particle in various places and the relativistic wavefunction.
The non-relativistic equation for the energy of a free particle:

$$\frac{\boldsymbol{p}^2}{2m} = E \tag{2.1}$$

By quantizing the energy of a free particle, we get the non-relativistic Schrödinger equation for a free particle:

$$\frac{\boldsymbol{p}^2}{2m}\Psi = i\hbar\frac{\partial}{\partial t}\Psi \tag{2.2}$$

where, $\boldsymbol{p} = -i\hbar\nabla$ is the momentum operator (∇ being the del operator. In the 3 dimensional Cartesian coordinate system with coordinates (x, y, z), del is defined in terms of a partial derivative operator as $\nabla = \hat{x}\frac{\partial}{\partial x} + \hat{y}\frac{\partial}{\partial y} + \hat{z}\frac{\partial}{\partial z}$)

The Schrödinger equation does not have to be relativistically covariant, meaning that it does not take into account Einstein's Special Relativity in [12]. In order to obtain a relativistic equation, we start by considering a free particle within a relativistic relation. It is natural to try to use the identity from special relativity describing energy:

$$\sqrt{\boldsymbol{p}^2c^2 + m^2c^4} = E \tag{2.3}$$

Then, we insert the quantum mechanical operators for momentum and energy, which yields the equation:

$$\sqrt{(-i\hbar\nabla)^2c^2 + m^2c^4}\Psi = i\hbar\frac{\partial}{\partial t}\Psi \tag{2.4}$$

The square of the above identity:

$$\boldsymbol{p}^2c^2 + m^2c^4 = E^2 \tag{2.5}$$

By quantizing, that gives:

$$((-i\hbar\nabla)^2c^2 + m^2c^4)\Psi = \left(i\hbar\frac{\partial}{\partial t}\right)^2\Psi \tag{2.6}$$

Which simplifies to:

$$-\hbar^2c^2\nabla^2\Psi + m^2c^4\Psi = -\hbar^2\frac{\partial^2}{\partial t^2}\Psi \tag{2.7}$$

So, the Klein-Gordon equation is written by:

$$\left(\frac{1}{c^2}\frac{\partial^2}{\partial t^2} - \nabla^2 + \frac{m^2c^2}{\hbar^2}\right)\Psi(\mathbf{x},t) = 0 \tag{2.8}$$

The KG equation is most often written in natural units in [13]:

$$\left(-\partial_t + \nabla^2\right)\Psi(\mathbf{x},t) = m^2\Psi(\mathbf{x},t) \tag{2.9}$$

And, the KG equation has many variations. The quasilinear KG equation in [14] is given by:

$$u_{tt} - \alpha^2u_{xx} + \gamma^2u = \beta u^3 \tag{2.10}$$

And the nonlinear KG equation in [14] is given by:

$$\sum_{i=1}^{n} u_{x_ix_i} + \lambda u^p = 0 \tag{2.11}$$

See the detail about the general solution forms about the KG equation in Appendices A and B.

3. Spcae Time Finite Element Method.

We investigate a space-time Finite element method similar to using continuous approximation functions in both space and time to explore its use for numerical relativity simulations. The discretization for the KG equation in this paper is an extension of the discretization for the nonhomogeneous wave equation in [9].

Rewrite the Klein-Gordon equation in natural units in the 1+1 dimension:

$$-\frac{\partial^2\Psi}{\partial t^2} + \frac{\partial^2\Psi}{\partial x^2} - \Psi = 0 \tag{3.1}$$

We discretize space and time together for the entire domain using a finite element space which does not discriminate between space and time basis functions and consider iterative solution methods with a time decomposition preconditioner. We take variables $u = \Psi$ and $v = \frac{\partial\Psi}{\partial t}$ instead of Ψ, and separate the equation:

$$\frac{\partial^2u}{\partial x^2} - \frac{\partial v}{\partial t} - u = 0 \tag{3.2}$$

$$-\frac{\partial u}{\partial t} + v = 0 \tag{3.3}$$

That also indicates that the linear system of equations can be written in matrix form as $Ax = b$. Here A is called the **stiffness matrix** and b is called the **force vector**. This is the system we wish to solve numerically.

Finite element space is the space of piecewise polynomial functions $\phi : \Omega\times(0,T] \to \mathfrak{R}$. The weak forms of this equation are:

$$K(u,v,\phi) = \int_\Omega \left(-\nabla u\nabla\phi + \frac{\partial v}{\partial t} + u\phi\right)ds = 0 \tag{3.4}$$

$$G(u,v,\phi) = \int_\Omega \left(-\frac{\partial u}{\partial t}\phi + v\phi\right)ds = 0 \tag{3.5}$$

Using the same method, we solve the spin-0 particle in a charged scalar field:

$$-\frac{\partial^2\Psi}{\partial t^2} + \frac{\partial^2\Psi}{\partial x^2} - \Psi = f \tag{3.6}$$

Where f is forcing function given by:

$$f = \sqrt{\frac{1}{2}}\left(\sin(x-t)\cos(x-t)\right) \tag{3.7}$$

And the weak forms of this equation are:

$$K(u,v,\phi) = \int_\Omega \left(-\nabla u \nabla \phi + \frac{\partial v}{\partial t} + u\phi - f\phi\right) ds = 0 \tag{3.8}$$

$$G(u,v,\phi) = \int_\Omega \left(-\frac{\partial u}{\partial t}\phi + v\phi\right) ds = 0 \tag{3.9}$$

The stiffness matrix A is a $N \times N$ matrix, where N is the total number of nodes in the domain. In the unit square example we are considering, we have $N = W^2$. The stiffness matrix A is defined as:

$$A = \begin{bmatrix} a(\phi_1,\phi_1) & \cdots & a(\phi_1,\phi_{W^2}) \\ \vdots & \ddots & \vdots \\ a(\phi_{W^2},\phi_1) & \cdots & a(\phi_{W^2},\phi_{W^2}) \end{bmatrix} \tag{3.10}$$

We calculate the stiffness matrix using the element stiffness matrices. The detail construction of each submatrices is in Appendix C.

For this boundary value problem, the solution values at the nodes on the boundary $\partial\Omega$ are known, so these nodes need not be included in our system of equations. We set the boundaries:

$$\begin{aligned} u &= u^0 \; on \; \Omega \times t = 0 \\ v &= v^0 \; on \; \Omega \times t = 0 \\ u_n &= 0 \; on \; \Omega \times [0,T] \\ v_n &= 0 \; on \; \Omega \times [0,T] \end{aligned} \tag{3.11}$$

Hence, our stiffness matrix reduces from a $W^2 \times W^2$ matrix to a $(W-2)^2 \times (W-2)^2$ matrix. If we label the stiffness matrix component in row i and column k as a_{ik}, then $a_{ik} = a(\phi_i, \phi_k)$, which can be calculated by adding the effects of all the square elements:

$$a_{ik} = \sum_{T_j \in \Omega} a_j(\phi_i, \phi_k). \tag{3.12}$$

4. Numerical Results. In this section, we present approximated solutions to the 1+1 Klein-Gordon equation using a space-time Finite Elements Method. We implement our numerical test in PETSc(Portable, Extensible, Toolkit for Scientific Computation, developed by Argonne National Laboratory) [15]. From many heuristic tests, we found that using GMRES and the time additive Schwarz method is one of the efficient solving method for this problem. See the detail about the time additive Schwarz method in Appendix D.

4.1. Time-Decomposition Method. Figure 4.1 shows the convergence time test with difference numbers of time subdomains. The green line is the test in the 90 × 90 grid, and the red line is the test in the 60 × 60 grid. The graph shows that large numbers of time subdomains converge fast. This indicates that large time subdomain numbers solve the numerical problem efficiently.

This result also indicates that time decomposition algorithm is promising in future parallel calculation in multi-process machines with large size problems; each process will calculate local problem with ideally one time subdomain and the local problem therefore will be reduced with more number of processes. The result Figure 4.1 shows the global problem might be reduced with more number of time subdomains. Finding optimal number of time subdomains in parallel implementation will be very interesting in our future research.

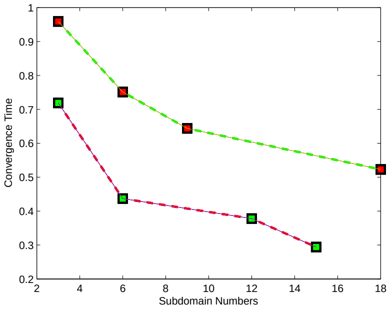


Fig. 4.1: Convergence Time Test with Difference Subdomain Numbers

Figure 4.2 shows mesh division within space-time. We respect the original boundary conditions for the subspace interface conditions : Dirichlet for $t = (t_{n-1} - overlap)$ and an evolution equation determined for $t = (t_n + overlap)$ where t_n is the n-th time decomposition. In this paper, we fixed overlap size as 1. Figure 4.3 shows analytic solutions of the 1+1 KG equation in free space. We test spin-0 particle propagation through space and time.

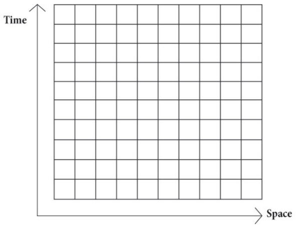
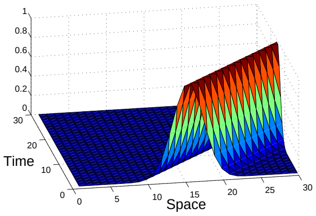


Fig. 4.2: Mesh Grid Division within Space and Time

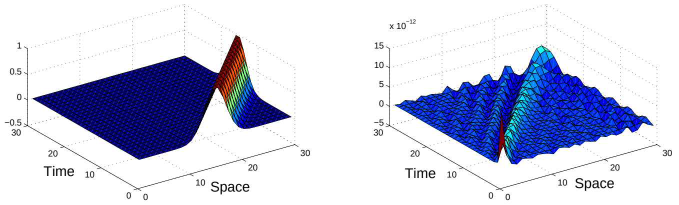
4.2. Test Problems.

4.2.1. Test Problem 1 : Free Space Case. We test the spin-0 particle in free space which is the forcing term is zero as a special case. We implement this problem using Matlab. Figure 4.4 shows numerical solutions of the 1+1 KG equation in free space, and indicate particle propagation. We use GMRES [16] and the time additive Schwarz preconditioner in a 30 × 30 grids. The domain of the problem is split into time subdomains. The local problem of each subdomain is solved by GMRES.



(a) Analytic Solution Curve in the 30x30 grid

Fig. 4.3: Analytic Solutions of the 1+1 Klein-Gordon equation for spin-0 particles in free space



(a) Solution Curve in the 30x30 grid (b) Error Curve the 30x30 grid

Fig. 4.4: Numerical Solutions of 1+1 Klein-Gordon Equation for spin-0 particles in free space

4.2.2. Test Problem 2 : Charged Field Case. We choose the forcing function given by:

$$f = \sqrt{\frac{1}{2}}\left(\sin(x-t)\cos(x-t)\right) \tag{4.1}$$

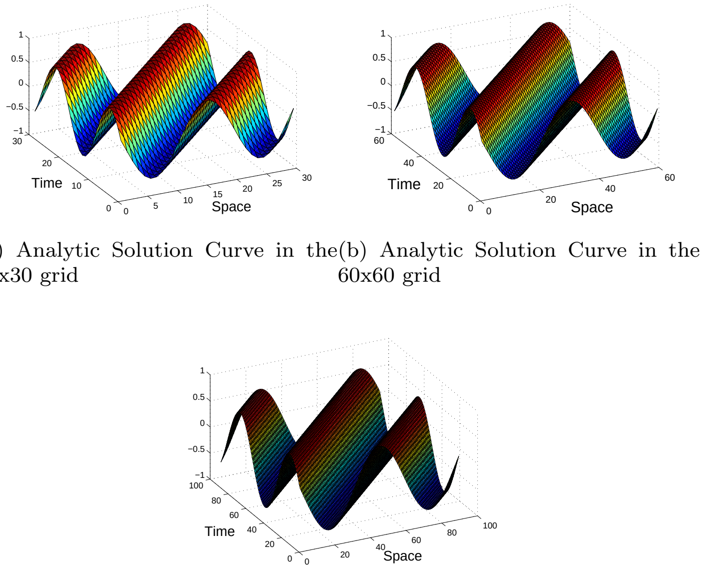
for representing a trigonometric electromagnetic wave. We expect that the solutions indicate an electromagnetic wave.

Figure 4.5 shows analytic solutions of the 1+1 KG equation within a charged scalar field. The solutions indicate that interactions between spin-0 particles and a charged scalar field produce an electromagnetic wave form.

Figure 4.6 shows the solution after 1, 50, 200, and 500 iterations in a 60 × 60 grid. We use the six time additive Schwarz time subdomains. Large iteration numbers provide a stable solution.

Figures 4.7, 4.8, and 4.9 show numerical solutions of the 1+1 KG equation within a charged scalar field in 30 × 30, 60 × 60, and 90 × 90 grids. Spin-0 particles and a charged Klein-Gordon field produce an electromagnetic wave. The solution shapes provide propagated electromagnetic waves through space and time. We use GMRES and the time additive Schwarz method in PETSc.

5. Comparing Results with Traditional Methods. We test numerical approaches to the 1+1 Klein-Gordon equation. We test with the Euler method, Runge-Kutta method, Explicit Difference method, and the Crank-Nicolson method. We test the equation 3.6 to compare with the our charge field results



(a) Analytic Solution Curve in the 30x30 grid

(b) Analytic Solution Curve in the 60x60 grid

(c) Analytic Solution Curve in the 90x90 grid

Fig. 4.5: Analytic Solutions of the 1+1 Klein-Gordon equation for a charged scalar field

5.1. Euler Method. We use the Euler method for solving this system. The 1+1 KG equation is:

$$\frac{\partial^2 \Psi}{\partial t^2} = \frac{\partial^2 \Psi}{\partial x^2} - \Psi + f \tag{5.1}$$

For applying the Euler Method, we need to change the KG equation as follows:

$$\frac{\partial \phi}{\partial t} = g(x, \phi) \tag{5.2}$$

We apply the method of line technique for adapting a partial differential equation. Also, we need to take a new variable ϕ for reducing order:

$$\phi = \begin{bmatrix} \Psi \\ \Psi' \end{bmatrix} \tag{5.3}$$

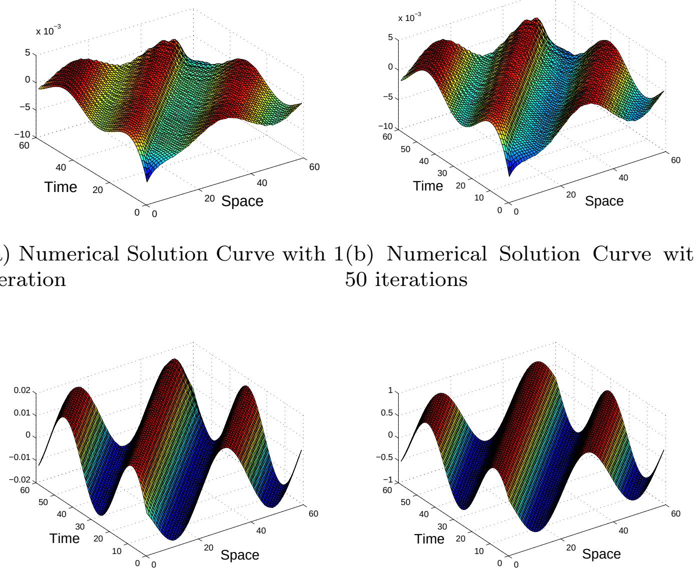
We apply that to the 1+1 KG equation as follows:

$$\phi_{n+1} = \phi_n + hg(x_n, \phi_n) \tag{5.4}$$

where h is step size. And, $g(x, \phi)$ is RHS of the equation (5.1) as evaluated by the finite difference calculation. Figure 5.1 shows numerical solutions of the 1+1 KG equation within a charged scalar field in a 60 × 60 grid. Figure 5.2 shows minimum error comparison data to the Euler method. We test 30, 60, and 120 step numbers.

5.2. Runge-Kutta Method. We use the Runge-Kutta 4th order(RK4) method. We apply the RK4 method to time direction. Using the similar analogy in section 5.1, we take the new variable ϕ instead of Ψ:

$$\phi = \begin{bmatrix} \Psi \\ \Psi' \end{bmatrix} \tag{5.5}$$



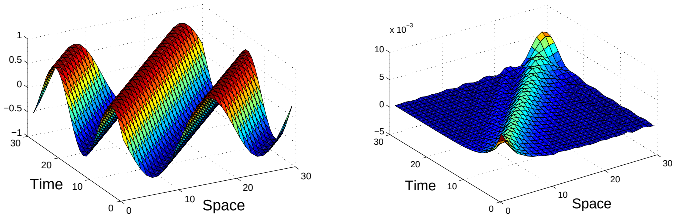
(a) Numerical Solution Curve with 1 iteration

(b) Numerical Solution Curve with 50 iterations

(c) Numerical Solution Curve with 200 iterations

(d) Numerical Solution Curve with 500 iterations

Fig. 4.6: Numerical Solutions of the 1+1 Klein-Gordon equation for a charged scalar field within 60 × 60 grids



(a) Solution Curve in the 30x30 grid

(b) Error Curve in the 30x30 grid

Fig. 4.7: Numerical Solutions of 1+1 Klein-Gordon Equation for a charged scalar field

And, we use the method of line technique which we apply to the 1+1 KG equation as follows:

$$\phi_{n+1} = \phi_n + \frac{1}{6}h(k_1 + k_2 + k_3 + k_4) \tag{5.6}$$

$$k_1 = g(x_n, \phi_n) \tag{5.7}$$

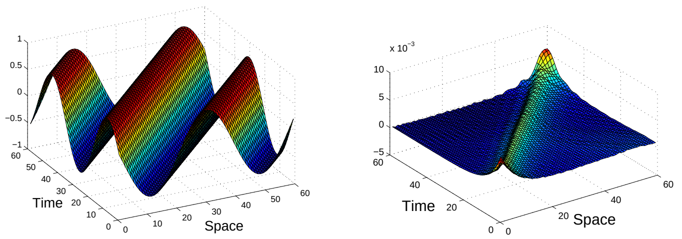
$$k_2 = g(x_n + 0.5h, \phi_n + 0.5hk_1) \tag{5.8}$$

$$k_3 = g(x_n + 0.5h, \phi_n + 0.5hk_2) \tag{5.9}$$

$$k_4 = g(x_n + h, \phi_n + hk_3) \tag{5.10}$$

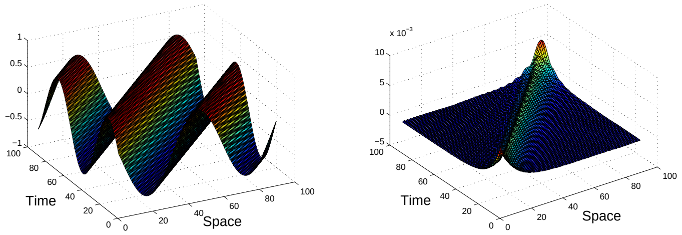
where h is step size. And, $g(x, \phi)$ is RHS of the equation (5.1) as evaluated by the finite difference calculation.

Figure 5.3 shows numerical solutions of the 1+1 KG equation within a charged scalar field in a 60 × 60 grid. Figure 5.4 shows minimum error comparison data to



(a) Solution Curve in the 60x60 grid (b) Error Curve in the 60x60 grid

Fig. 4.8: Numerical Solutions of 1+1 Klein-Gordon Equation for a charged scalar field



(a) Solution Curve in the 90x90 grid (b) Error Curve in the 90x90 grid

Fig. 4.9: Numerical Solutions of 1+1 Klein-Gordon Equation for a charged scalar field

the Runge-Kutta method. We test 30, 60, and 120 step numbers.

5.3. Explicit Difference Method. We use Explicit Difference method. We reconstruct the KG equation as follows:

$$-\frac{\psi_j^{n+1}-2\psi_j^n+\psi_j^{n-1}}{\Delta t^2}+\frac{\psi_{j+1}^n-2\psi_j^n+\psi_{j-1}^n}{\Delta x^2}+\psi_n^j=f \tag{5.11}$$

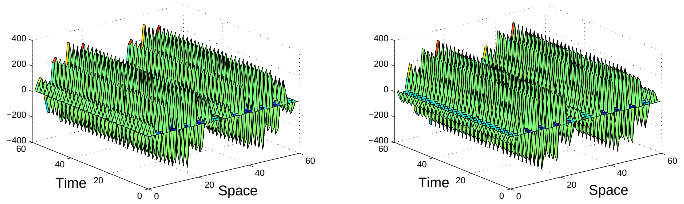
Figure 5.5 shows numerical solutions of the 1+1 KG equation within a charged scalar field in a 60 × 60 grid. Figure 5.6 shows minimum error comparison data to the Explicit Difference method. We test 30, 60, and 120 step numbers.

5.4. Crank-Nicolson Method. We use the Crank-Nicolson method. We reconstruct the KG equation as follows:

$$-\frac{\psi_j^{n+1}-2\psi_j^n+\psi_j^{n-1}}{\Delta t^2}+\frac{1}{2\Delta x^2}((\psi_{j+1}^{n+1}-2\psi_j^{n+1}+\psi_{j-1}^{n+1})+(\psi_{j+1}^n-\psi_j^n+\psi_{j-1}^n))+\psi_n^j=f \tag{5.12}$$

Figure 5.7 shows numerical solutions of the 1+1 KG equation within a charged scalar field in a 60 × 60 grid. Figure 5.8 shows minimum error comparison data to the Crank-Nicolson method. We test 30, 60, and 120 step numbers.

6. Conclusion & Future Research. We present the results of the Klein-Gordon equation using a space-time finite element method in the 1+1 dimension of spin-0 particle propagation in the charged field. We use a time decomposition strategy to implement time parallelizable algorithm. The numerical results show that this method is promising to develop future parallel simulation for the KG equation.



(a) Solution Curve in the 60x60 grid (b) Error Curve in the 60x60 grid

Fig. 5.1: Numerical Solutions of the 1+1 Klein-Gordon Equation with the Euler Method

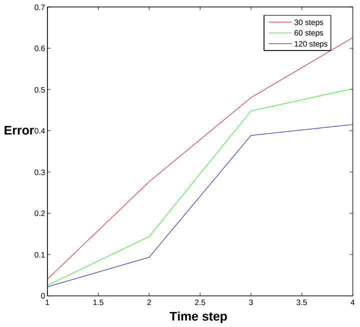


Fig. 5.2: Minimum error comparison data plot of the 1+1 Klein-Gordon Equation with the Euler Method

It will be very interested in future research to investigate the optimal mixed temporal spatial domain decomposition. Another important future research topic is to extend current implementation of the space-time Finite Element method to 2+1 and 3+1 dimensions. These extensions will be needed parallel implementation in High Performance Computing machines.

7. Acknowledgment. I am very indebted to Dr. Matthew Anderson for helpful discussions on the subject of this paper.

Appendix A. Free Spin-0 Particle. The KG equation is:

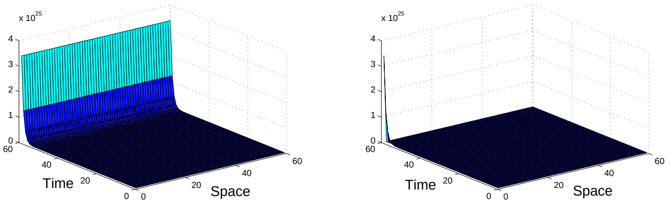
$$\left(\frac{1}{c^2}\frac{\partial^2}{\partial t^2} - \nabla^2 + \frac{m^2c^2}{\hbar^2}\right)\Psi(\mathbf{x},t) = 0 \tag{A.1}$$

A possible ansatz of the solution for a free wave is:

$$\Psi = A\exp\left(-\frac{i}{\hbar}p_\mu x^\mu\right) \tag{A.2}$$

Where $\frac{1}{\hbar p_\mu} = k_\mu$. There exist two possible solutions for a given momentum p, one with positive and one with negative energy. Consequently:

$$\Psi_\pm = A_\pm \exp\left(\frac{i}{\hbar}(px \mp E_p t)\right) \tag{A.3}$$



(a) Solution Curve in the 60x60 grid (b) Error Curve in the 60x60 grid

Fig. 5.3: Numerical Solutions of the 1+1 Klein-Gordon Equation with the Runge-Kutta Method

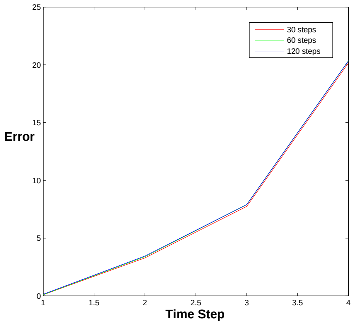


Fig. 5.4: Minimum error comparison data plot of the 1+1 Klein-Gordon Equation with the Runge-Kutta Method

Where $E_p = \pm c\sqrt{p^2 + m^2c^2}$. Discretizing the continuous plane wave by confining the wave to a cubic box (normalization box) with body length L. This leads to:

$$\Psi_{n(\pm)} = A_{n(\pm)} \exp\left(\frac{i}{\hbar}(p_n x \mp E_n t)\right) \tag{A.4}$$

Where:

$$p_n = \frac{2pi}{L}n, \quad n = (n_1, n_2, n_3), \quad n_i \in N \tag{A.5}$$

And:

$$E_p n = c\sqrt{p_n^2 + m^2c^2} = E_n \tag{A.6}$$

The normalization factors $A_{n(\pm)}$ are determined by:

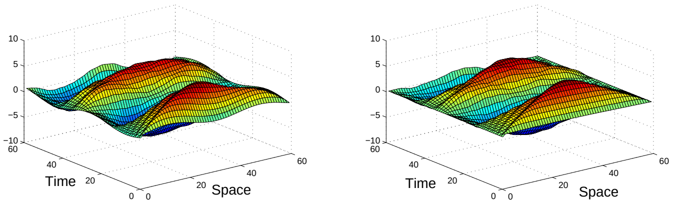
$$\pm \int_{L^3} d^3x\rho = \pm \frac{E_n}{mc^2} A_{n(\pm)}^2 L^3 \tag{A.7}$$

Choosing the phase such that the amplitudes are real:

$$A_{n(\pm)} = \sqrt{\frac{mc^2}{L^3 E_n}} \tag{A.8}$$

And the solutions are in [12]:

$$\Psi_{n(\pm)} = \sqrt{\frac{mc^2}{L^3 E_n}} \exp\left(\frac{i}{\hbar}(p_n x \mp E_n t)\right) \tag{A.9}$$



(a) Solution Curve in the 60x60 grid (b) Error Curve in the 60x60 grid

Fig. 5.5: Numerical Solutions of the 1+1 Klein-Gordon Equation with the Explicit Difference Method

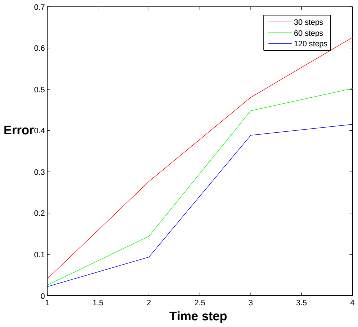


Fig. 5.6: Minimum error comparison data plot of the 1+1 Klein-Gordon Equation with the Explicit Difference Method

We can take the real part of the solution with positive energy for testing particle propagation in free space.

Appendix B. The Charged Klein-Gordon Field. In the case of a charged scalar field (Charged Klein-Gordon field), the current is given by:

$$j_\mu = \frac{ie\hbar}{2m}(\phi^* \partial_\mu \phi - \phi \partial_\mu \phi^*) \tag{B.1}$$

With $\frac{\partial j^\mu}{\partial x^\mu} = 0$ and the total charge is:

$$Q = \frac{ie\hbar}{2mc^2} \int d^3x \left(\phi^* \frac{\partial \phi}{\partial t} - \phi \frac{\partial \phi^*}{\partial t} \right) \tag{B.2}$$

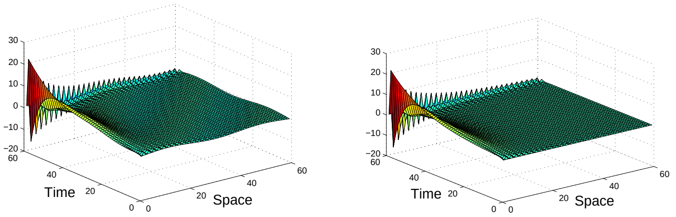
To examine charged fields, we decompose ϕ into real and imaginary components:

$$\phi = \frac{1}{\sqrt{2}}(\phi_1 + i\phi_2), \tag{B.3}$$

Where ϕ_1 and ϕ_2 are real. To demonstrate the new degree of freedom (charge), we need to adjust the solutions' forms. We consider spin-0 particles in a charged field in [12]:

$$\Psi = A_{n(\pm)} \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix} \exp\left(\frac{i}{\hbar}(px - E_n t) \right) \tag{B.4}$$

We expect that the solutions indicate an electromagnetic wave in space-time. So, we need to set the forcing function to examine electromagnetic action, and also choose the real part of the solution with positive energy for simulation.



(a) Solution Curve in the 60x60 grid (b) Error Curve in the 60x60 grid

Fig. 5.7: Numerical Solutions of the 1+1 Klein-Gordon Equation with the Crank-Nicolson Method

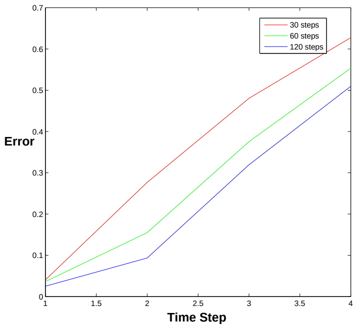


Fig. 5.8: Minimum error comparison data plot of the 1+1 Klein-Gordon Equation with the Crank-Nicolson Method

Appendix C. Calculating Element Stiffness Matrices.

Define $\Omega = [0,1] \times [0,1] \subset \Re^2$, or in other words, let Ω be the unit square. We divide the mesh using uniform squares which can be defined to have equal size h with respect to x and t.

For calculating element stiffness matrices A_j, we consider a single square element. The **element stiffness matrix** for this singular element is represented as:

$$\mathbf{A_j} = \begin{bmatrix} a_j(\phi_A, \phi_A) & a_j(\phi_A, \phi_B) & a_j(\phi_A, \phi_C) & a_j(\phi_A, \phi_D) \\ a_j(\phi_B, \phi_A) & a_j(\phi_B, \phi_B) & a_j(\phi_B, \phi_C) & a_j(\phi_B, \phi_D) \\ a_j(\phi_C, \phi_A) & a_j(\phi_C, \phi_B) & a_j(\phi_C, \phi_C) & a_j(\phi_C, \phi_D) \\ a_j(\phi_D, \phi_A) & a_j(\phi_D, \phi_B) & a_j(\phi_D, \phi_C) & a_j(\phi_D, \phi_D) \end{bmatrix}$$

We use A_j to assemble the stiffness matrix for the entire domain. Because all of the elements in the square of Ω are the same size, we can utilize A_j for all elements, only having to adjust for the orientation of the square. We determine the basis functions explicitly at the nodes:

$$\phi_j(A) = \begin{cases} 1 & \text{if E = A} \\ 0 & \text{if E} \neq \text{A} \end{cases} \quad \text{for E} \in \text{A,B,C,D}$$

$$\phi_j(B) = \begin{cases} 1 & \text{if E = B} \\ 0 & \text{if E} \neq \text{B} \end{cases} \quad \text{for E} \in \text{A,B,C,D}$$

$$\phi_j(C) = \begin{cases} 1 & \text{if E = C} \\ 0 & \text{if E} \neq \text{C} \end{cases} \quad \text{for E} \in \text{A,B,C,D}$$

$$\phi_j(D) = \begin{cases} 1 & \text{if E} = \text{D} \\ 0 & \text{if E} \neq \text{D} \end{cases} \quad \text{for E} \in \text{A,B,C,D}$$

Consider first the basis function ϕ_A, this creates a plane over the element region, which can be represented as $\phi_A = u_A x + v_A t + w_A xt + z_A$, where $u_A, v_A, w_A, z_A \in \Re$. We note that $\phi_A = 1$ when $x = 0$ and $t = 0$ and that $\phi_A = 0$ when $x = h$ and $t = 0$. Also, $\phi_A = 0$ when $x = 0$ and $t = h$, and $\phi_A = 0$ when $x = h$ and $t = h$. We can develop a set of four equations for the four unknowns. Hence:

$$1 = u_A(0) + v_A(0) + w_A(0) + z_A \Rightarrow z_A = 1 \tag{C.1}$$

$$0 = u_A(h) + v_A(0) + w_A(0) + z_A \Rightarrow u_A = -\frac{1}{h} \tag{C.2}$$

$$0 = u_A(0) + v_A(h) + w_A(0) + z_A \Rightarrow v_A = -\frac{1}{h} \tag{C.3}$$

$$0 = u_A(h) + v_A(h) + w_A(h^2) + z_A \Rightarrow w_A = \frac{1}{h^2} \tag{C.4}$$

So that:

$$\phi_A = \frac{1}{h^2}xt - \frac{1}{h}x - \frac{1}{h}t + 1 \quad \text{for} \quad (x,t) \in T_j \tag{C.5}$$

We again have a plane, so $\phi_B = u_B x + v_B t + w_B xt + z_B$. This time, $\phi_B = 1$ at $(h, 0)$ and $\phi_B = 0$ at $(0,0), (0,h)$, and (h,h). Our system of equations solving for u_B, v_B, and w_B becomes:

$$0 = u_B(0) + v_B(0) + w_B(0) + z_B \Rightarrow z_B = 0 \tag{C.6}$$

$$1 = u_B(h) + v_B(0) + w_B(0) + z_B \Rightarrow u_B = \frac{1}{h} \tag{C.7}$$

$$0 = u_B(0) + v_B(h) + w_B(0) + z_B \Rightarrow v_B = 0 \tag{C.8}$$

$$0 = u_B(h) + v_B(h) + w_B(h^2) + z_B \Rightarrow w_B = -\frac{1}{h^2} \tag{C.9}$$

Therefore:

$$\phi_B = -\frac{1}{h^2}xt + \frac{1}{h}x \quad \text{for} \quad (x,t) \in T_j$$

Next, consider ϕ_C. This plane is represented by $\phi_C = u_C x + v_C t + w_C xt + z_C$, with $\phi_C = 1$ at (h,h) and $\phi_C = 0$ at $(0,0), (h,0)$, and $(0,h)$. We solve the system:

$$0 = u_C(0) + v_C(0) + w_C(0) + z_C \Rightarrow z_C = 0 \tag{C.10}$$

$$0 = u_C(h) + v_C(0) + w_C(0) + z_C \Rightarrow u_C = 0 \tag{C.11}$$

$$0 = u_C(0) + v_C(h) + w_C(0) + z_C \Rightarrow v_C = 0 \tag{C.12}$$

$$1 = u_C(h) + v_C(h) + w_C(h^2) + z_C \Rightarrow w_C = -\frac{1}{h^2} \tag{C.13}$$

So that:

$$\phi_C = \frac{1}{h^2}xt \quad \text{for} \quad (x,t) \in T_j$$

Finally, consider ϕ_D. This plane is $\phi_D = u_D x + v_D t + w_D xt + z_D$, with $\phi_D = 1$ at $(0,h)$ and $\phi_D = 0$ at $(0,0), (h,0)$, and (h,h). We solve the system:

$$0 = u_D(0) + v_D(0) + w_D(0) + z_D \Rightarrow z_D = 0 \tag{C.14}$$

$$0 = u_D(h) + v_D(0) + w_D(0) + z_D \Rightarrow u_D = 0 \tag{C.15}$$

$$1 = u_D(0) + v_D(h) + w_D(0) + z_D \Rightarrow v_D = \frac{1}{h} \tag{C.16}$$

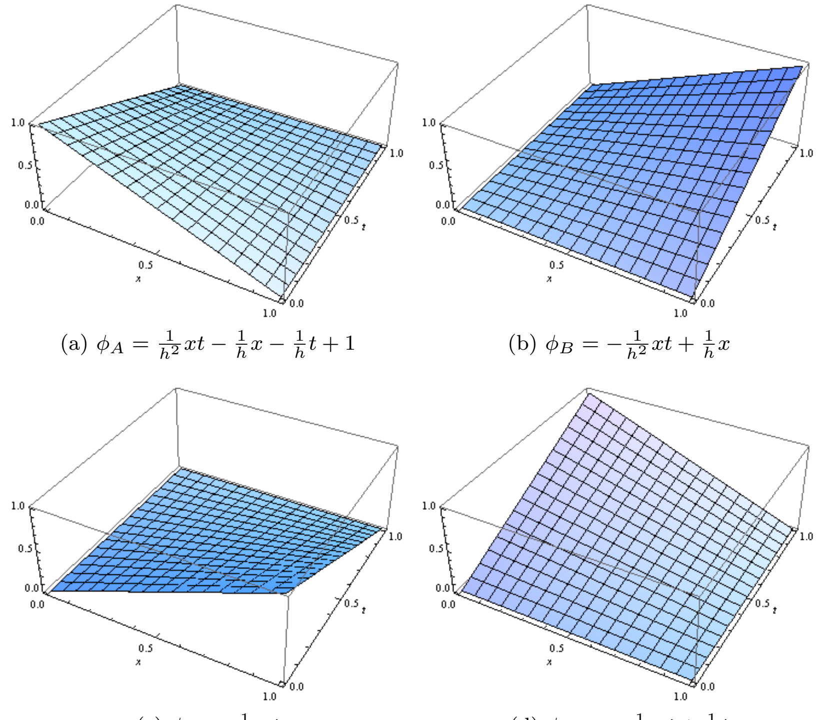
$$0 = u_D(h) + v_D(h) + w_D(h^2) + z_D \Rightarrow w_D = -\frac{1}{h^2} \tag{C.17}$$

So that:

$$\phi_D = -\frac{1}{h^2}xt + \frac{1}{h}t \quad \text{for} \quad (x,t) \in T_j$$

To sum up, the basis functions are:

$$\begin{aligned}
\phi_A &= \frac{1}{h^2}xt - \frac{1}{h}x - \frac{1}{h}t + 1 \\
\phi_B &= -\frac{1}{h^2}xt + \frac{1}{h}x \\
\phi_C &= \frac{1}{h^2}xt \\
\phi_D &= -\frac{1}{h^2}xt + \frac{1}{h}t
\end{aligned}$$



(a) $\phi_A = \frac{1}{h^2}xt - \frac{1}{h}x - \frac{1}{h}t + 1$

(b) $\phi_B = -\frac{1}{h^2}xt + \frac{1}{h}x$

(c) $\phi_C = \frac{1}{h^2}xt$

(d) $\phi_D = -\frac{1}{h^2}xt + \frac{1}{h}t$

Fig. C.1: Quadrilateral four-node finite elements

Our equation needs three types of element stiffness matrices. First, we consider the Poisson term u_{xx},

$$Poisson_j(a_\alpha, a_\beta) = \int_{T_j} \nabla\phi_i \nabla\phi_j d\Omega \tag{C.18}$$

We calculate the stiff matrix in the 1+1 dimension, so that matrix A_j is a 4x4 matrix. We can see that A is an unsymmetric matrix. By the definition of the basis functions, we can also see that many of the $a(\phi_i, \phi_j)$ have values, which means that the total matrix A will be non-symmetrical. Due to this fact, we need to examine the stiffness matrix carefully. In addition, we can show that A is a positive definitive matrix. As illustrated above, this calculation is the Poisson case. Knowing the basis function with respect to the (x,t) coordinates allows us to compute the components of the element stiffness matrix directly. Thus, the time derivation term is not necessary. This elements calculation indicates:

$$a_j(\phi_A,\phi_A) = \int_0^h \int_0^h \left(\frac{\partial\phi_A}{\partial x}\frac{\partial\phi_A}{\partial x} \right) dtdx = \frac{1}{3}$$

$$a_j(\phi_A,\phi_B) = \int_0^h \int_0^h \left(\frac{\partial\phi_A}{\partial x}\frac{\partial\phi_B}{\partial x} \right) dtdx = -\frac{1}{3}$$

$$a_j(\phi_A,\phi_C) = \int_0^h \int_0^h \left(\frac{\partial\phi_A}{\partial x}\frac{\partial\phi_C}{\partial x} \right) dtdx = -\frac{1}{6}$$

$$a_j(\phi_A,\phi_D) = \int_0^h \int_0^h \left(\frac{\partial\phi_A}{\partial x}\frac{\partial\phi_D}{\partial x} \right) dtdx = \frac{1}{6}$$

$$a_j(\phi_B,\phi_B) = \int_0^h \int_0^h \left(\frac{\partial\phi_B}{\partial x}\frac{\partial\phi_B}{\partial x} \right) dtdx = \frac{1}{3}$$

$$a_j(\phi_B,\phi_C) = \int_0^h \int_0^h \left(\frac{\partial\phi_B}{\partial x}\frac{\partial\phi_C}{\partial x} \right) dtdx = \frac{1}{6}$$

$$a_j(\phi_B,\phi_D) = \int_0^h \int_0^h \left(\frac{\partial\phi_B}{\partial x}\frac{\partial\phi_D}{\partial x} \right) dtdx = -\frac{1}{6}$$

$$a_j(\phi_C,\phi_C) = \int_0^h \int_0^h \left(\frac{\partial\phi_C}{\partial x}\frac{\partial\phi_C}{\partial x} \right) dtdx = \frac{1}{3}$$

$$a_j(\phi_C,\phi_D) = \int_0^h \int_0^h \left(\frac{\partial\phi_C}{\partial x}\frac{\partial\phi_D}{\partial x} \right) dtdx = -\frac{1}{3}$$

$$a_j(\phi_D,\phi_D) = \int_0^h \int_0^h \left(\frac{\partial\phi_D}{\partial x}\frac{\partial\phi_D}{\partial x} \right) dtdx = \frac{1}{3}$$

Using these calculated values, we can create the element stiffness matrix with respect to Poisson case as:

$$\mathbf{A_j; Poisson(i,j)} = \begin{pmatrix} \frac{1}{3} & -\frac{1}{3} & -\frac{1}{6} & \frac{1}{6} \\ -\frac{1}{3} & \frac{1}{3} & \frac{1}{6} & -\frac{1}{6} \\ -\frac{1}{6} & \frac{1}{6} & \frac{1}{3} & -\frac{1}{3} \\ \frac{1}{6} & -\frac{1}{6} & -\frac{1}{3} & \frac{1}{3} \end{pmatrix}$$

In similar way, we also need to show two more element matrices forms which are:

$$Heat_j(a_\alpha, a_\beta) = \int_{T_j} \frac{\partial\phi_i}{\partial t}\phi_j d\Omega \qquad \text{(C.19)}$$

Calculating each element of the matrices:

$$a_j(\phi_A, \phi_A) = \int_0^h \int_0^h \left(\frac{\partial \phi_A}{\partial t} \phi_A + \frac{\partial \phi_A}{\partial t} \phi_A \right) dxdt = -\frac{h}{3}$$

$$a_j(\phi_A, \phi_B) = \int_0^h \int_0^h \left(\frac{\partial \phi_A}{\partial t} \phi_B + \frac{\partial \phi_B}{\partial t} \phi_A \right) dxdt = -\frac{h}{6}$$

$$a_j(\phi_A, \phi_C) = \int_0^h \int_0^h \left(\frac{\partial \phi_A}{\partial t} \phi_C + \frac{\partial \phi_C}{\partial t} \phi_A \right) dxdt = \frac{h}{6}$$

$$a_j(\phi_A, \phi_D) = \int_0^h \int_0^h \left(\frac{\partial \phi_A}{\partial t} \phi_D + \frac{\partial \phi_D}{\partial t} \phi_A \right) dxdt = \frac{h}{3}$$

$$a_j(\phi_B, \phi_B) = \int_0^h \int_0^h \left(\frac{\partial \phi_B}{\partial t} \phi_B + \frac{\partial \phi_B}{\partial t} \phi_B \right) dxdt = -\frac{h}{3}$$

$$a_j(\phi_B, \phi_C) = \int_0^h \int_0^h \left(\frac{\partial \phi_B}{\partial t} \phi_C + \frac{\partial \phi_C}{\partial t} \phi_B \right) dxdt = \frac{h}{3}$$

$$a_j(\phi_B, \phi_D) = \int_0^h \int_0^h \left(\frac{\partial \phi_B}{\partial t} \phi_D + \frac{\partial \phi_D}{\partial t} \phi_B \right) dxdt = \frac{h}{6}$$

$$a_j(\phi_C, \phi_C) = \int_0^h \int_0^h \left(\frac{\partial \phi_C}{\partial t} \phi_C + \frac{\partial \phi_C}{\partial t} \phi_C \right) dxdt = \frac{h}{3}$$

$$a_j(\phi_C, \phi_D) = \int_0^h \int_0^h \left(\frac{\partial \phi_C}{\partial t} \phi_D + \frac{\partial \phi_C}{\partial t} \phi_D \right) dxdt = \frac{h}{6}$$

$$a_j(\phi_D, \phi_D) = \int_0^h \int_0^h \left(\frac{\partial \phi_D}{\partial t} \phi_D + \frac{\partial \phi_D}{\partial t} \phi_D \right) dxdt = \frac{h}{3}$$

$$\mathbf{A_j}; \mathbf{Heat(i,j)} = \begin{pmatrix} -\frac{h}{3} & -\frac{h}{6} & \frac{h}{6} & \frac{h}{3} \\ -\frac{h}{6} & -\frac{h}{3} & \frac{h}{3} & \frac{h}{6} \\ -\frac{h}{6} & -\frac{h}{3} & \frac{h}{3} & \frac{h}{6} \\ -\frac{h}{3} & -\frac{h}{6} & \frac{h}{6} & \frac{h}{3} \end{pmatrix}$$

$$Standard_j(a_\alpha, a_\beta) = \int_{T_j} \phi_i \phi_j d\Omega$$

$$a_j(\phi_A, \phi_A) = \int_0^h \int_0^h \phi_A \phi_A dtdx = \frac{h^2}{9}$$

$$a_j(\phi_A, \phi_B) = \int_0^h \int_0^h \phi_A \phi_B dtdx = \frac{h^2}{18}$$

$$a_j(\phi_A, \phi_C) = \int_0^h \int_0^h \phi_A \phi_C dtdx = \frac{h^2}{36}$$

$$a_j(\phi_A, \phi_D) = \int_0^h \int_0^h \phi_A \phi_D dtdx = \frac{h^2}{18}$$

$$a_j(\phi_B, \phi_B) = \int_0^h \int_0^h \phi_B \phi_B dtdx = \frac{h^2}{9}$$

$$a_j(\phi_B, \phi_C) = \int_0^h \int_0^h \phi_B \phi_C dtdx = \frac{h^2}{18}$$

$$a_j(\phi_B, \phi_D) = \int_0^h \int_0^h \phi_B \phi_D dtdx = \frac{h^2}{36}$$

$$a_j(\phi_C, \phi_C) = \int_0^h \int_0^h \phi_C \phi_C dtdx = \frac{h^2}{9}$$

$$a_j(\phi_C, \phi_D) = \int_0^h \int_0^h \phi_C \phi_D dtdx = \frac{h^2}{18}$$

$$a_j(\phi_D, \phi_D) = \int_0^h \int_0^h \phi_D \phi_D dtdx = \frac{h^2}{9}$$

$$\mathbf{A_j; Standard(i, j)} = \begin{pmatrix} \frac{h^2}{9} & \frac{h^2}{18} & \frac{h^2}{36} & \frac{h^2}{18} \\ \frac{h^2}{18} & \frac{h^2}{9} & \frac{h^2}{18} & \frac{h^2}{36} \\ \frac{h^2}{36} & \frac{h^2}{18} & \frac{h^2}{9} & \frac{h^2}{18} \\ \frac{h^2}{18} & \frac{h^2}{36} & \frac{h^2}{18} & \frac{h^2}{9} \end{pmatrix} \tag{C.20}$$

Appendix D. Additive Schwarz Method.

Domain Decomposition methods(DD) solve a boundary value problem by splitting it into smaller boundary value problems on subdomains and iterating to coordinate the solution between adjacent subdomains. The problems in the subdomains are independent, which makes domain decomposition methods suitable for parallel computing. Domain decomposition methods are typically used as preconditioners for Krylov space iterative methods, such as the conjugate gradient method or GMRES. Domain decomposition methods embody large potential for a parallelization of the FEM, and serve a basis for distributed, parallel computations. See the detail about the preconditioner in Appendix E.

Time decomposition methods in this paper can be a variant for time of the additive Schwarz method (ASM). For a domain $\Omega = \cup_i \Omega_i$, ASM can be written as:

$$u^{n+1} = u^n + \sum_i B_i(f - Au^n) \quad \text{with} \quad B_i = R_i^T A_{\Omega_i}^{-1} R_i. \tag{D.1}$$

The additive Schwarz method may be viewed as a generalization of block Jacobi methods. [10, 17, 11]

Appendix E. Preconditioner.

Finite element simulations of moderate size models require solving linear systems with millions of unknowns. Several hours per time step is an average sequential run time thus, the parallel computing is a necessity. The development of scalable parallel numerical methods for large algebraic systems is central in solving very large scale linear systems. The most widely used methods for such problems are Krylov subspace methods such as GMRES and the Conjugate Gradient method.

The key to scalability is preconditioning which accelerates the convergence of the iterative solver. Preconditioning means replacing the system $Ax = b$ to a system that is more easily solved. For example, one might replace $Ax = b$ to $M^{-1}Ax = M^{-1}b$ where M is an approximation of A with the properties:

1. $M^{-1}A$ is well conditioned or has few extreme eigenvalues,
2. $Mx = b$ is easy to solve.

A careful choice of M can often make the condition number of $M^{-1}A$ much smaller than the condition number of A and thus accelerate convergence.

REFERENCES

[1] D. Shirokoff. Renormalized waves and thermalization of the Klein-Gordon Equation. *Physical Review E*, 83(046217):1–13, 2011.

[2] B. Muller Walter Greiner and J. Rafelski. *Quantum Electrodynamics of Strong Fields.* Springer, second edition, 1985.

[3] M. Jiang Y. J. Li R. Grobe Q. Z. Lv, A. C. Su and Q. Su. Pair creation for bosons in electric and magnetic fields. *Physical Review A*, 87(023416):1–11, 2013.

[4] Jorge Eduardo Macías-Díaz. Bistability of a two-dimensional Klein-Gordon system as a reliable means to transmit monochromatic waves: A numerical approach. *Physical Review E*, 78(056603):9, 2008.

[5] Philippe Grandclíement and Gyula Fodor. Numerical simulation of oscillatons: Extracting the radiating tail. *Physical Review D*, 84(065037):1–15, 2011.

[6] V. G. Bagrov and D. Gitman. *Exact Solutions of Relativistic Wave Equations.* Addison Wesley, first edition, 2003.

[7] Li Yang. Numerical Studies of the Klien-Gordon-Schrödinger Equations, June 2006. Master thesis papaer.

[8] Weizhu Bao and Li Yang. Efficient and accurate numerical methods for the Klein-Gordon-Schrödinger Equations. *Journal of Computational Physics*, 225:1863–1893, 2007.

[9] Matthew Anderson and Jung-Han Kimn. A numerical approach to space-time finite elements for the wave equation. *Journal of Computational Physics*, 226:466–476, 2007.

[10] Barry F. Smith, Petter Bjørstad, and William Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations.* Cambridge University Press, 1996.

[11] Andrea Toselli and Olof Widlund. *Domain Decomposition Methods - Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics.* Springer, 2004.

[12] Walter Greiner and D.A. Bromley. *Relativistic Quantum Mechanics. Wave Equation.* Springer, third edition, 2000.

[13] J. J. Sakuriai. *Advanced Quantum Mechanics.* Addison Wesley, first edition, 1967.

[14] Eric W. Wiesstein. Klein-Gordon equation.

[15] Satish Balay, Jed Brown, Kris Buschelman, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc Web page, 2012. http://www.mcs.anl.gov/petsc.

[16] Youcef Saad and Martin H. Schultz. GMRES:A generalized minimal residual algorithm for solving nonsymmetric linear systems. *Society for Industrial and Applied Mathematics*, 7(3):856–869, 1986.

[17] Youcef Saad. *Iterative method for sparse liner system.* Society for Industrial and Applied Mathematics, second edition, 2003.