# Numerical Models for the Time-Dependent Heat and Stokes' Equations

Molly Q. Feldman*, James H. Adler,† & Scott P. MacLachlan†

August 27, 2013

**Abstract**

In the following research, we attempt to approximate the time-dependent heat equation and the time-dependent Stokes' equation with discrete computational models. In doing so, we attempt to balance three main goals:

1. the speed of computation,
2. the accuracy of conserved physical quantities,
3. the accuracy of the solution equation

This paper contains a mathematical overview of numerical solutions for partial differential equations as well as an in-depth explanation of the computational models, all of which were written in Python (specific libraries used are explained in Section 2.1).

**Note** : As this paper is written for a variety of audiences, those familiar with the premise of partial differential equations, the Galerkin discretization, and the FOSLS discretization can skip all sections marked with a $(*)$. Similarly, those who would prefer only a background, non-technical explanation of the research should skip all sections *not* marked with a $(*)$. Otherwise, we only assume the reader has a basic understanding of multivariable calculus.

# 1 Introduction.

## 1.1 A Brief Aside to Physics $(*)$.

Partial differential equations (PDEs) are integral to the modeling of physical phenomena. Take the heat equation as an example. The heat equation represents the dispersal of temperature over time. In typical PDE form, it is expressed as follows,

$$\begin{aligned} \frac{\partial u}{\partial t} - \nabla^2 u &= 0 \text{ on } \Omega, \\ u &= 0 \text{ on } \partial\Omega, \end{aligned} \tag{1}$$

where $\Omega$ is the domain, $\partial\Omega$ is the boundary of the domain, $t$ is time, and $u$ is the temperature.

In comparison, the time-dependent Stokes' equation models fluid dynamics with a variable pressure, $p$. It consists of the following system

$$\begin{aligned} \frac{\partial \vec{u}}{\partial t} + \nabla p - \nabla^2 u &= 0 \text{ on } \Omega, \\ \nabla \cdot \vec{u} &= 0 \text{ on } \Omega, \\ \vec{n} \cdot \vec{u} &= 0 \text{ on } \partial\Omega, \end{aligned} \tag{2}$$

*Department of Mathematics & Statistics, Swarthmore College, Swarthmore, PA, USA (mfeldma2@swarthmore.edu).
†Department of Mathematics, Tufts University, Medford, MA, USA

where $\vec{u}$ is the velocity of the fluid and $\vec{n}$ is the normal vector to boundary.

The fundamental goal of this research is to solve for scalar $u$ in the heat equation and vector $\vec{u}$ and pressure $p$ in the Stokes' equation. The manner in which we solve these PDEs is the focus of the following sections.

## 1.2 Analytical vs. Numerical Methods(∗).

In a typical introductory partial differential equations class students are taught to compute solutions for PDEs analytically. However, an analytical approach is not practical in this case. First of all, most computers lack the memory, processing power etc. to complete the necessary steps of an analytical calculation. Secondly, there are more complex equations (such as those that relate to magnetohydrodynamics) which may not have any analytical solution. In those cases, a numerical model is the only solution method available. The numerical model that we utilize in our project is the Finite-Element Method (FEM). In order to minimize the introductory portions of this paper, we assume that the reader has a basic understanding of the FEM from this point on.[‡]

## 1.3 Discretization Techniques and Associated Linear Systems.

The FEM is a technique which converts continuous equations (such as those in systems (1) and (2)) into their discrete equivalents. This research works with two such methods: the Galerkin FEM and the First-Order System Least-Squares (FOSLS) FEM. A key difference between them is that FOSLS has less restrictions on the choices of basis functions and function spaces than Galerkin. This simplicity is a major benefit of the FOSLS approach.

The main purpose of these discretizations is to generate a solvable linear system from a given PDE. In our project, such a linear system is an expression in the form $Lu = 0$, where $L$ is a linear PDE and $u$ is the solution to the equation. From the FEM, in order to solve a PDE numerically, we need to generate a weak equation representation. This is accomplished in the Galerkin approach by hitting $Lu = 0$ with a test function, $v$, on both sides, resulting in the expression $Lu \cdot v = 0 \cdot v$. Integrating both sides gives us the weak equation

$$\langle Lu, v\rangle = \langle 0, v\rangle.^{\S} \tag{3}$$

In comparison, FOSLS works by attempting to minimize the $L_2$ norm of the residual of the PDE, i.e. $||Lu - 0||_{L_2}$. Through this process, we obtain the final weak equation

$$\langle Lu, Lv\rangle = \langle 0, Lv\rangle. \tag{4}$$

The FOSLS weak equation differs from the Galerkin weak equation only in the formulation of the test function.

The exact equations for the computational models follow directly from the generalized form of equations (3) and (4). In the subsequent equations, the $^h$ superscript denotes the discrete variable and the $\phi$'s represent the test functions associated with the subscripted variables. Unless otherwise noted, all of the following equations hold for all time $t$.

Restricting the heat equation (equation (1)) to the finite-element space, applying the Galerkin discretization, and integrating by parts, we obtain the equation

$$\langle \frac{\partial u^h}{\partial t}, v^h\rangle + \langle \nabla u^h, \nabla v^h\rangle = 0. \tag{5}$$

However, equation (5) is only a semi-discretized form of the heat equation, as $t$ is still a continuous quantity. To achieve the fully discretized weak equation, we approximate the time derivative using a backward Euler discretization, i.e.

$$\frac{u_k^h - u_{k-1}^h}{dt}, \tag{6}$$

[‡]Give some good reference for a document to read to refresh one's knowledge of the FEM

[§]Unless otherwise noted, the inner product operator $\langle a, b\rangle$ is short-hand notation for $\int_\Omega \langle a, b\rangle\, dx$, the $L_2$ norm

where the subscripts denote the instances of $u^h$ at times $k$ and $k-1$ and $dt$ is the time step. Thus the final weak form for the heat equation is

$$\frac{1}{dt}\langle u_k^h, v^h\rangle + \langle \nabla u_k^h, \nabla v^h\rangle = \frac{1}{dt}\langle u_{k-1}^h, v^h\rangle. \tag{7}$$

In order to obtain a similar result for FOSLS, system (1) must first be converted into a first-order system. To do so the quantity $V = \nabla u$ is introduced. This yields the following system of equations

$$\begin{aligned} \frac{\partial u}{\partial t} - \nabla \cdot V &= 0 \text{ on } \Omega, \\ V - \nabla u &= 0 \text{ on } \Omega, \\ \nabla \times V &= 0 \text{ on } \Omega, \\ u &= 0 \text{ on } \partial\Omega, \\ \tau \cdot V &= 0 \text{ on } \partial\Omega, \end{aligned} \tag{8}$$

where $\tau$ is the tangent vector to the boundary. The auxiliary equations added to system (1) to obtain system (8) are consistent with the system and guarantee that the FOSLS form has a unique solution.

To obtain the semi-discretized weak form of system (8), the system is restricted to the finite-element space and the FOSLS method is applied, resulting in

$$\langle \frac{\partial u^h}{\partial t} - \nabla \cdot V^h, \frac{\partial \phi_u^h}{\partial t} - \nabla \cdot \phi_V^h\rangle + \langle V^h - \nabla u^h, \phi_V^h - \nabla \phi_V^h\rangle + \langle \nabla \times V^h, \nabla \times \phi_V^h\rangle = 0. \tag{9}$$

Again, the backward Euler discretization is utilized to obtain the FOSLS weak equation

$$\frac{1}{dt}\langle u_k^h - \nabla \cdot V^h, \phi_u^h - \nabla \cdot \phi_V^h\rangle + \langle V^h - \nabla u_k^h, \phi_V^h - \nabla \phi_u^h\rangle + \langle \nabla \times V^h, \nabla \times \phi_V^h\rangle = \frac{1}{dt}\langle u_{k-1}^h, \phi_u^h - \nabla \cdot \phi_V^h\rangle. \tag{10}$$

The Stokes' weak equations are determined in a similar way. The basic time-dependent Stokes' equation for fluid flow on a domain with Dirichlet boundary conditions is cited in system (2). Analogously to the heat equation, applying the Galerkin discretization to system (2) yields the following semi-discretized form

$$\langle \frac{\partial \vec{u}^h}{\partial t}, \vec{v}^h\rangle + \langle \nabla \vec{u}^h, \nabla \vec{v}^h\rangle + \langle \nabla p^h, \vec{v}^h\rangle - \langle \vec{u}^h, \nabla q^h\rangle = 0. \tag{11}$$

To obtain the fully discretized form for both the Galerkin and FOSLS forms of Stokes', the backwards Euler discretization is again applied. Thus, the full weak equation for Galerkin Stokes' is

$$\frac{1}{dt}\langle \vec{u}_k^h, \vec{v}^h\rangle + \langle \nabla \vec{u}_k^h, \nabla \vec{v}^h\rangle + \langle \nabla p^h, \vec{v}^h\rangle - \langle \vec{u}_k^h, \nabla q^h\rangle = \frac{1}{dt}\langle \vec{u}_{k-1}^h, \vec{v}^h\rangle, \tag{12}$$

where the $k$ subscript again demarcates the instances of $\vec{u}^h$ at times $k$ and $k-1$ and $dt = \Delta k$.

The FOSLS formulation of Stokes' can be computed by first introducing the quantity $\underline{\underline{V}} = \nabla \vec{u}$ in order to make Stokes' a first-order system. This quantity differs for $V$ in that it is equal to the gradient of vector rather than a scalar. Thus, $\underline{\underline{V}}$ is known as a tensor and has a value equal to $\begin{pmatrix} \frac{\partial u_1}{\partial x} & \frac{\partial u_2}{\partial x} \\ \frac{\partial u_1}{\partial y} & \frac{\partial u_2}{\partial y} \end{pmatrix}$, where

$\vec{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$. This definition then leads to the following set of equations

$$\begin{aligned} \frac{\partial \vec{u}}{\partial t} + \nabla p - \nabla \cdot \underline{\underline{V}} &= 0 \text{ on } \Omega, \\ \underline{\underline{V}} - \nabla \vec{u} &= 0 \text{ on } \Omega, \\ \nabla \cdot \vec{u} &= 0 \text{ on } \Omega, \\ \nabla \times \underline{\underline{V}} &= 0 \text{ on } \Omega, \\ \nabla \text{tr} \underline{\underline{V}} &= 0 \text{ on } \Omega, \\ \vec{n} \cdot \vec{u} &= 0 \text{ on } \partial\Omega, \\ \vec{n} \times \underline{\underline{V}} &= 0 \text{ on } \partial\Omega, \end{aligned} \tag{13}$$

where `tr` is the trace function.

We then arrive at the fully discretized weak equation for FOSLS,

$$\begin{aligned} &\frac{1}{dt}\langle \vec{u}_k^h + \nabla p - \nabla \cdot \underline{\underline{V}}^h, \phi_{\vec{u}}^h + \nabla \phi_p^h - \nabla \cdot \phi_{\underline{\underline{V}}}^h \rangle + \langle \underline{\underline{V}}^h - \nabla \vec{u}_k^h, \phi_{\underline{\underline{V}}}^h - \nabla \phi_{\vec{u}}^h \rangle + \langle \nabla \cdot \vec{u}_k^h, \nabla \cdot \phi_{\vec{u}}^h \rangle \\ &+ \langle \nabla \times \underline{\underline{V}}^h, \nabla \times \phi_{\underline{\underline{V}}}^h \rangle + \langle \nabla \text{tr} \underline{\underline{V}}^h, \nabla \text{tr} \phi_{\underline{\underline{V}}}^h \rangle = \frac{1}{dt}\langle \vec{u}_{k-1}^h, \phi_{\vec{u}}^h + \nabla p^h - \nabla \cdot \phi_{\underline{\underline{V}}}^h \rangle. \end{aligned} \tag{14}$$

The previous equations and systems are key as they are directly implemented in the source code. When using the *FEniCS* libraries, the left-hand sides of these quantities are defined as bilinear forms, `a`, and the right-hand sides as linear functionals, `L`. Given `a` and `L`, the associated linear system is directly generated and solved. Other libraries utilize similar techniques to generate the main linear system.

## 1.4 Direct and Interative Solvers $(*)$.

Next, we briefly discuss how the resulting in linear systems are solved. The computational models that follow use three types of solvers: 1) direct solvers, 2) iterative solvers, and 3) iterative solvers with preconditioners.

A computational direct solver computes the actual solution of the linear system, assuming the computer performs exact arithmetic. There are two caveats to this method. First, the solution is only exact for the discrete model and not for the continuous one. However, none of the available solving techniques would provide an exact continuous solution given a discrete model. Second, direct solvers can be extremely slow and computationally intensive. Thus, accuracy is the main benefit of a direct solver, not speed.

In comparison, iterative solvers attempt to compute a "good enough" solution in the fastest possible time. They utilize a solution technique which entails estimating a solution, checking to see if it is within a certain threshold, and then recalculating the estimation given the result of the comparsions. Two iterative solvers are used in this project's models: Krylov solvers and multigrid methods. Although the details of each approach is beyond the scope of this introduction, Krylov and multigrid solvers differ in how they condition the matrix of the linear system. The specific solvers we use are a Ruge-Stuben algebraic multigrid solver as well as two preconditioned Krylov solvers. A preconditioner is used with a Krylov solver to better condition the matrix to make it easier to invert. The specific Krylov/preconditioner pairings we work with are a conjugate gradient (CG) solver with an AMG preconditioner and a generalized minimal residual method (GMRES) solver with an AMG preconditioner.

The take-away message from this brief account is that there are many ways to solve linear systems each with their own strengths. A direct solver allows for the least amount of approximation possible, but can be time-prohibitive. A pure AMG solver allows for a faster approximation technique but may perform more iterations than necessary, given a more general initial condition. Finally, a preconditioned Krylov solver theoretically requires less iterations than an AMG solver, but is problem dependent.

## 1.5 Energy Laws.

In the previous sections, we have shown how to solve for $u$ and $\vec{u}$ in the heat and Stokes' equations. As these equations model physical phenomena, they must have associated energy laws. This fact is utilized in this project to measure the quality of the generated discrete solutions. Take the heat equation, written in the form

$$\frac{\partial u}{\partial t} - \nabla^2 u = 0, \tag{15}$$

as an example. If equation (15) is multiplied on both sides by $u$ and integrated by parts, the following equality is obtained

$$\frac{\partial}{\partial t}\frac{1}{2}\int_\Omega u^2\,dx = -\int_\Omega \nabla u \cdot \nabla u\,dx. \tag{16}$$

This energy law also applies for the time-dependent Stokes' equation, substituting $\vec{u}$ for $u$ and hitting $\nabla \cdot u = 0$ with $p$, although the related derivation is outside the scope of this section. The left-hand portion of equation (16) is similar to the derivative of the integral of the Newtonian kinetic energy equation. It in fact *is* the Newtonian equation for the Stokes' version.

Because equation (16) is obtained directly from the heat and Stokes' equations, it is proven that the energy law holds in the continuous case. However, $u/\vec{u}$ can only be calculated in this project's models in a discrete environment. Thus, it is necessary to show that both sides of equation (16) can be computed discretely in order to later prove that the energy law holds discretely for the Galerkin and FOSLS discretizations.

The general FEM represents $u$ discretely as $u = \sum_{i=1}^{I}\sum_{j=1}^{J} c_{i,j}\psi_{i,j}$, where the $c_{i,j}$'s are nodal coefficients and the $\psi_{i,j}$'s are the typical basis functions. Substituting this quantity into the left-hand side of equation (16) results in the following equality

$$\frac{\partial}{\partial t}\frac{1}{2}\int_\Omega \left[\left(\sum_{i=1}^{I}\sum_{j=1}^{J} c_{i,j}\psi_{i,j}\right)\left(\sum_{i=1}^{I}\sum_{j=1}^{J} c_{i,j}\psi_{i,j}\right)\right] dx = -\int_\Omega \nabla u \cdot \nabla u\,dx. \tag{17}$$

Through applying some calculus and linear algebra techniques, the left-hand side of equation (17) is represented as $\frac{\partial}{\partial t}\frac{1}{2}\vec{c}^T \cdot M\vec{c}$, where $\vec{c}$ is a vector of the $c_{i,j}$ coefficients and $M$ is a matrix with entries of the form $M_{i,j} = \langle \psi_i, \psi_j \rangle$. The matrix, $M$, is typically called the mass matrix of the system. As the basis functions are known, $M$ is easily calculated discretely. The second variable, $\vec{c}$, is discretely obtained by evaluating the discrete $u$ at each time step, resulting in a coefficient vector. So, the left-hand side of equation (16) is calculated discretely.

The process for the right-hand side is similar. If $u = \sum_{i=1}^{I}\sum_{j=1}^{J} c_{i,j}\psi_{i,j}$ is substituted for $u$ in the right-hand side of equation (16) and a set of operations is performed, the equation

$$-\int_\Omega \nabla u \cdot \nabla u\,dx = \vec{c}^T \cdot A\vec{c} \tag{18}$$

is obtained, where $A$ is the stiffness matrix of the Poisson operator. As the elements of $A$ are equal to $\langle \nabla\psi_i, \nabla\psi_j \rangle$ for all $i$ and $j$, $A$ can be discretely computed from the known basis functions. Thus, it has been shown that both sides of equation (16) can be calculated discretely.

As stated previously, the above energy law also allows us to determine the quality of this project's Galerkin and FOSLS calculations of $u/\vec{u}$. This is accomplished by comparing either the left or right-hand sides of equation (16) calculated discretely with our $u/\vec{u}$, with the same quantity computed with the true solution[¶]. More information about the specifics of comparing the results can be found in Section 1.6 and the graphs themselves in Sections 3 and 4. Ultimately, the energy laws provide an "equal footing" on which

[¶]The true solution is a known function of $t$, which is the exact analytical solution of the PDE, can be computed using the initial conditions for the model.

to compare the discrete $u/\vec{u}$ with the analytical solutions, as both sides of the energy law must hold in all cases.

Although it was previously shown that both sides of equation (16) can be computed discretely, it is necessary to prove that the *equality* still holds for the Galerkin and FOSLS discretizations. The semi-discretized Galerkin form of the heat equation can be written as $\langle \frac{\partial u^h}{\partial t}, v^h \rangle = -\langle \nabla u^h, \nabla v^h \rangle$. Remembering the inner-product short-hand notation and letting $v^h = u^h$, this equation is in fact the energy law. Thus, the equality holds implicitly. Analogous proof can be found for the Galerkin Stokes' equation as well. This direct proof is a benefit of the Galerkin model.

Unfortunately, the FOSLS discretizations must satisfy an additional set of requirements in order for the energy law equality to hold[‖]. Given the solution functions of the FOSLS heat equation (equation (10)), $u$ and $V$, a solution can be found for the following discrete weak linear equation

$$\begin{aligned} \langle \frac{\partial {\phi_u}^h}{\partial t} - \nabla \cdot {\phi_V}^h, x^h \rangle &= \langle u^h, x^h \rangle, \\ \langle {\phi_V}^h - \nabla {\phi_u}^h, y^h \rangle &= \langle V^h, y^h \rangle, \\ \langle \nabla \times {\phi_V}^h, z^h \rangle &= 0, \end{aligned} \tag{19}$$

where $x$, $y$, and $z$ are test functions defined on their associated function spaces. If this system is satisfied, then integrating by parts yields the discrete energy law. This then maintains the necessary equality from equation (16). Testing whether or not system (19) holds can be accomplished numerically. In fact, this project attempts to solve system (19) using the FOSLS output of the original equation, to determine the values of $u^h$ and $V^h$, and then both the Galerkin and FOSLS discretizations to determine appropriate values for $\phi_u^h$ and $\phi_V^h$. The results of this portion of the research are called the FOSLS$^2$ method and the FOSLS Least-Squares method. The FOSLS Least-Squares method solves the above over-determined system directly. The FOSLS$^2$ approach assumes system (19) originates from a continuous system of PDE and uses FOSLS to approximation the values of $\phi_u^h$ and $\phi_V^h$. Although the previous assumption does not hold, the FOSLS$^2$ method is simply additional support to show that system (19) is in fact solvable. The FOSLS Stokes' equation has an analogous system

$$\begin{aligned} \langle \frac{\partial \phi_{\vec{u}}^h}{\partial t} + \nabla \phi_p^h - \nabla \cdot \phi_{\vec{V}}^h, x^h \rangle &= \langle \vec{u}^h, x^h \rangle, \\ \langle \phi_{\vec{V}}^h - \nabla \phi_{\vec{u}}^h, y^h \rangle &= \langle V^h, y^h \rangle, \\ \langle \nabla \cdot \phi_{\vec{u}}^h, z^h \rangle &= \langle p^h, z^h \rangle, \\ \langle \nabla \times \phi_{\vec{V}}^h, w^h \rangle &= 0, \\ \langle \nabla \mathrm{tr} \phi_{\vec{V}}^h, \gamma^h \rangle &= 0, \end{aligned} \tag{20}$$

where $x$, $y$, $z$, $w$, and $\gamma$ are test functions defined on their associated function spaces. This too can be shown to generate the necessary equality through integration by parts. This project's computational models utilize the FOSLS Stokes' equation to generate the necessary values for $\vec{u}$ and $\underline{\underline{V}}$ and then the Galerkin model to solve for $\phi_{\vec{u}}^h$ and $\phi_{\underline{\underline{V}}}^h$. Given that systems (19) and (20) hold, the Galerkin and FOSLS discretizations can express equation (16) directly, maintaining the necessary energy laws for their related systems.

## 1.6 Obtaining Results.

As noted previously, this research compares the computational models by comparing specific discrete quantities. For the Galerkin and FOSLS discretizations, there are three graphs: the left-hand side of equation (16) plotted at each time step, the right-hand side at each time step, and the quantity $\frac{1}{2}u^2$ at each time step. The final quantity is chosen to compare the energy of each system, given its relationship to the Newtonian

[‖] Please see the related "FOSLS Energy Notes" for a robust explanation of the necessary requirements.

kinetic energy equation. We also graph these quantities on a logarithmic scale in order to compare the results while disregarding the scaling of the dependent axis.

Each computational model has three variables: $n$, which indicates an $n$ by $n$ FEM mesh where $n = \frac{1}{h}$, $dt$, the time step, and *reps*, the total number time steps. It is generally known from the FEM that as $n$ increases and/or $dt$ decreases, the approximation becomes more accurate. This is explicitly tested by computing different approximations where $n$ remains constant and $dt$ changes and vice versa. A wide variety of $n$ and $dt$ were originally experimented with, from $n = 8\,(2^3)$ to $n = 256\,(2^8)$ and $dt = \frac{1}{32}\,(\frac{1}{2^5})$ to $dt = \frac{1}{655536}\,(\frac{1}{2^{17}})$. The goal is to attempt to determine the limiting behavior of the approximations. The surrounding hypothesis is that the Galerkin approximation should be on order $h^2$, where $h = \frac{1}{n}$, and the FOSLS approximation should be around order $h^2$ or order $h$. Thus, the choice of $n$ and $dt$ as powers of 2 allows the limiting behavior to be easily determined. In addition, each model runs from time $t = 0$ seconds to $t \approx 0.1$ seconds. This range was determined experimentally by observing that all approximations approach zero after 0.1 seconds. This then means that *reps* is equal to $\frac{0.1}{dt}$ for each model.

Two quantities are also graphed for the FOSLS$^2$ and FOSLS Least-Squares approximations. The $L_2$ norm of each model is first calculated and then scaled in two different ways: 1) $\frac{\text{norm}}{\text{energy}}$ and 2) $\frac{\text{norm}}{h*\text{energy}}$ where the energy is the energy computed from the given $n$, $dt$, and *reps* under the FOSLS approximation. These models are also run under a variety of $n$, $dt$, and *reps* values in order to determine their limiting behavior.

# 2 Computational Models.

## 2.1 Computational Libraries.

To create the subsequent computational models, two libraries are utilized:

1. *NumPy* and *SciPy* (`www.scipy.org`)- *NumPy* is a sub-library of *SciPy*, a general library for scientific computing, which directly enhances the ability of Python to deal with matrix computations and solving linear systems through both direct and indirect solvers. It allows for the solution of over-determined linear systems.

2. The *FEniCS* Project (`www.fenicsproject.org`)- this library is specifically designed for solving PDEs with the FEM and has multiple direct and iterative solvers. Its sub-library *DOLFIN* deals with the majority of the linear solvers, with multiple pre-conditioner options. However, it does not support the solution of over-determined systems. An additional benefit of the FEniCS library is a built in graphing technique that allows you to graph any FEniCS function object. This is helpful for debugging as it allows for the visualization of $u$ and $V$ at each discrete time step.

Each model utilizes *NumPy* and *FEniCS* in a different combination. However, there are some universal usages throughout the models. For example, *NumPy* objects (e.g. arrays, matrix operators etc.) contain the vast majority of the discrete data for the modle. *FEniCS* objects, on the other hand, deal with the abstract mathematical concepts and the solving mechanism. The most complex combination of these two libraries is the FOSLS Least-Squares implementation.

## 2.2 Heat Equation Energy Law Models.

The purpose of the following section is to determine whether or not the Galerkin and FOSLS discretizations provides an accurate approximations to the heat equation energy law. In both models, the finite-element domain, $\Omega$, is the unit square (i.e. $\Omega = [0,1]\times[0,1]$). The finite-element boundary conditions are the intrinsic conditions to the heat equation. In other words, $u = 0$ on the left ($x = 0$) and right ($x = 1$) borders of the domain and $V = 0$ on the top ($y = 1$) and bottom ($y = 0$) borders. Additionally, both models start with the same initial condition, $u_0(x, y) = 100\sin(\pi x)\sin(2\pi y)$. This then generates an analytical solution of

$$u(x, y) = 100\sin(\pi x)\sin(2\pi y)e^{-5\pi^2 t}. \tag{21}$$

With this equation it is then possible to determine if the Galerkin and FOSLS models approach the analytical solution as $n$ increases and $dt$ decreases. This also allows the limiting behavior of the approximations to be determined (i.e. are they order $h^2$ or order $h$ models).

In the following tables, the relative error is equal to the quantity $\frac{||u_t^* - u_t||}{||u_t^*|}$, where $u_t^*$ is equation (21) at time $t$ and $u_t$ is the value of the approximation of $u$ at time $t$, given the specified $n$ and $dt$.

| $n$ | $dt = \frac{1}{512}$ | $dt = \frac{1}{1024}$ | $dt = \frac{1}{2048}$ |
|---|---|---|---|
| 16 | 2.550e-4 | 5.058e-4 | 3.396e-4 |
| 32 | 3.913e-3 | 8.898e-4 | 1.738e-4 |
| 64 | 4.259e-3 | 1.065e-3 | 2.550e04 |

Table 1: **Galerkin Model:** Relative Error of the Approximation at Initial Time Step, $t = dt$

| $n$ | $dt = \frac{1}{512}$ | $dt = \frac{1}{1024}$ | $dt = \frac{1}{2048}$ |
|---|---|---|---|
| 16 | 1.411e-1 | 2.822e-2 | 4.904e-2 |
| 32 | 2.250e-1 | 9.620e-2 | 3.457e-2 |
| 64 | 1.065e-3 | 1.171e-1 | _____ |

Table 2: **Galerkin Model:** Relative Error of the Approximation at Final Time Step, $t = 0.1015625$

| $n$ | $dt = \frac{1}{512}$ | $dt = \frac{1}{1024}$ | $dt = \frac{1}{2048}$ | $dt = \frac{1}{4096}$ |
|---|---|---|---|---|
| 16 | 4.846e-2 | 2.416e-2 | 1.209e-2 | 6.059e-3 |
| 32 | 2.241e-2 | 1.090e-2 | 5.386e-3 | 2.692e-3 |
| 64 | 1.009e-2 | 4.443e-3 | 2.107e-3 | 1.038e-3 |

Table 3: **FOSLS Model:** Relative Error of the Approximation at Initial Time Step, $t = dt$

| $n$ | $dt = \frac{1}{512}$ | $dt = \frac{1}{1024}$ | $dt = \frac{1}{2048}$ | $dt = \frac{1}{4096}$ |
|---|---|---|---|---|
| 16 | 1.352e1 | 1.426e1 | 1.457e1 | 1.491e1 |
| 32 | 2.337e0 | 2.341e0 | 2.400e0 | 2.454e0 |

Table 4: **FOSLS Model:** Relative Error of the Approximation at Final Time Step, $t = 0.1015625$

The following tables and plots present data for three computed quantities: the quantity $\frac{1}{2}||u||^2$ (called the "Energy" of the system), the left-hand side of equation (16), and the right-hand side of equation (16). The "Approximation" values are equal to the specified quantity computed with the $u$ obtained from the Galerkin/FOSLS discretization at time $t$. The "Analytical" values are equal to the specified quantity computed with equation (21) at time $t$.

| | Approximation | Analytical | Approximation | Analytical | Approximation | Analytical |
|---|---|---|---|---|---|---|
| **n** | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{2048}}$ | $\mathbf{dt = \frac{1}{2048}}$ |
| **16** | 1.003e3 | 1.031e3 | 1.100e3 | 1.135e3 | 1.153e3 | 1.191e3 |
| **32** | 1.031e3 | 1.031e3 | 1.128e3 | 1.135e3 | 1.182e3 | 1.191e3 |
| **64** | 1.037e3 | 1.031e3 | 1.135e3 | 1.135e3 | 1.189e3 | 1.191e3 |

Table 5: **Galerkin Model:** "Energy" $(\frac{1}{2}||u||^2)$ Calculated at $t = dt$

| | Approximation | Analytical | Approximation | Analytical | Approximation | Analytical |
|---|---|---|---|---|---|---|
| **n** | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{2048}}$ | $\mathbf{dt = \frac{1}{2048}}$ |
| **16** | 6.964e-2 | 5.541e-2 | 5.539e-2 | 5.541e-2 | 4.912e-2 | 5.541e-2 |
| **32** | 8.247e-2 | 5.541e-2 | 6.603e-2 | 5.541e-2 | 5.878e-2 | 5.541e-2 |
| **64** | 8.604e-2 | 5.541e-2 | _____ | 5.541e-2 | _____ | 5.541e-2 |

Table 6: **Galerkin Model:** "Energy" $(\frac{1}{2}||u||^2)$ Calculated at $t = 0.1015625$

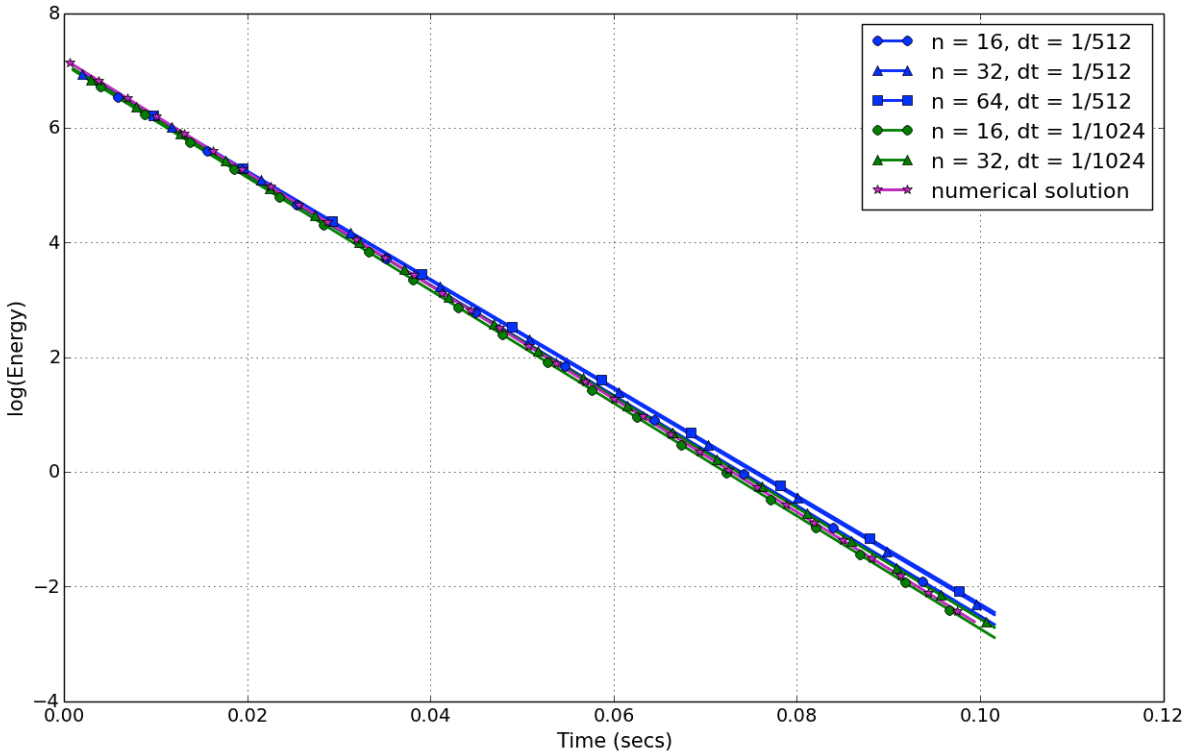


Figure 1: Galerkin Approximation of the Time-Dependent Heat Equation (Logarithmic Scale Energy Plot)

| | Approximation | Analytical | Approximation | Analytical | Approximation | Analytical |
|---|---|---|---|---|---|---|
| **n** | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{2048}}$ | $\mathbf{dt = \frac{1}{2048}}$ |
| **16** | -1.061e5 | -1.017e5 | -1.136e5 | -1.120e5 | -1.177e5 | -1.176e5 |
| **32** | -1.072e5 | -1.017e5 | -1.146e5 | -1.120e5 | -1.187e5 | -1.176e5 |
| **64** | -1.075e5 | -1.017e5 | -1.149e5 | -1.120e5 | -1.190e5 | -1.176e5 |

Table 7: **Galerkin Model:** Derivative of the Energy ($\frac{\partial}{\partial t}\frac{1}{2}||u||^2$) at $t = dt$

| | Approximation | Analytical | Approximation | Analytical | Approximation | Analytical |
|---|---|---|---|---|---|---|
| **n** | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{2048}}$ | $\mathbf{dt = \frac{1}{2048}}$ |
| **16** | -7.264e0 | -5.469e0 | -5.719e0 | -5.469e0 | -5.012e0 | -5.469e0 |
| **32** | -8.579e0 | -5.469e0 | -6.710e0 | -5.469e0 | -5.903e0 | -5.469e0 |
| **64** | -8.913e0 | -5.469e0 | _____ | -5.469e0 | _____ | -5.469e0 |

Table 8: **Galerkin Model:** Derivative of the Energy ($\frac{\partial}{\partial t}\frac{1}{2}||u||^2$) at $t = 0.1015625$

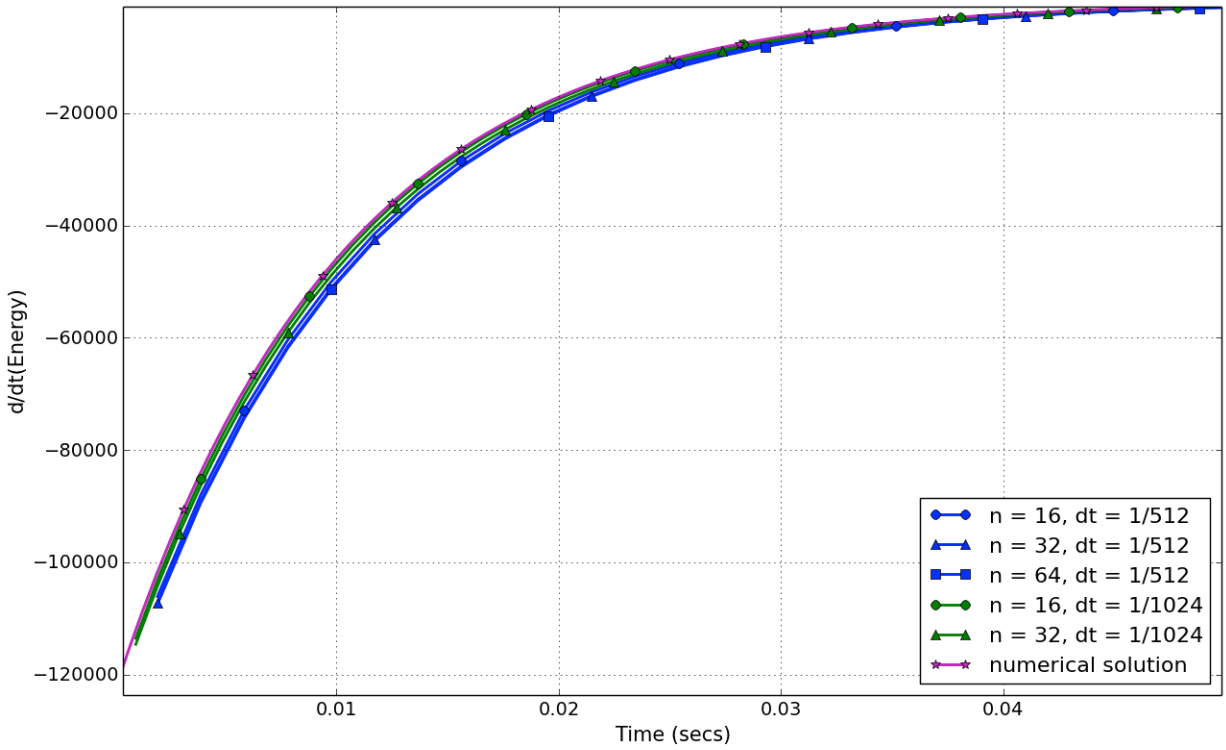


Figure 2: Galerkin Approximation of the Time-Dependent Heat Equation (Derivative Plot)

| | Approximation | Analytical | Approximation | Analytical | Approximation | Analytical |
|---|---|---|---|---|---|---|
| **n** | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{2048}}$ | $\mathbf{dt = \frac{1}{2048}}$ |
| **16** | -1.011e5 | -1.017e5 | -1.108e5 | -1.120e5 | -1.162e5 | -1.176e5 |
| **32** | -1.023e5 | -1.017e5 | -1.119e5 | -1.120e5 | -1.173e5 | -1.176e5 |
| **64** | -1.025e5 | -1.017e5 | -1.122e5 | -1.120e5 | -1.175e5 | -1.176e5 |

Table 9: **Galerkin Model:** Right-Side of Equation (16) $(-||\nabla u||^2)$ at $t = dt$

| | Approximation | Analytical | Approximation | Analytical | Approximation | Analytical |
|---|---|---|---|---|---|---|
| **n** | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{2048}}$ | $\mathbf{dt = \frac{1}{2048}}$ |
| **16** | -7.018e0 | -5.469e0 | -5.582e0 | -5.469e0 | -4.951e0 | -5.469e0 |
| **32** | -8.183e0 | -5.469e0 | -6.552e0 | -5.469e0 | -5.832e0 | -5.469e0 |
| **64** | -8.503e0 | -5.469e0 | -6.820e0 | -5.469e0 | -6.076e0 | -5.469e0 |

Table 10: **Galerkin Model:** Right-Side of Equation (16) $(-||\nabla u||^2)$ at $t = 0.1015625$

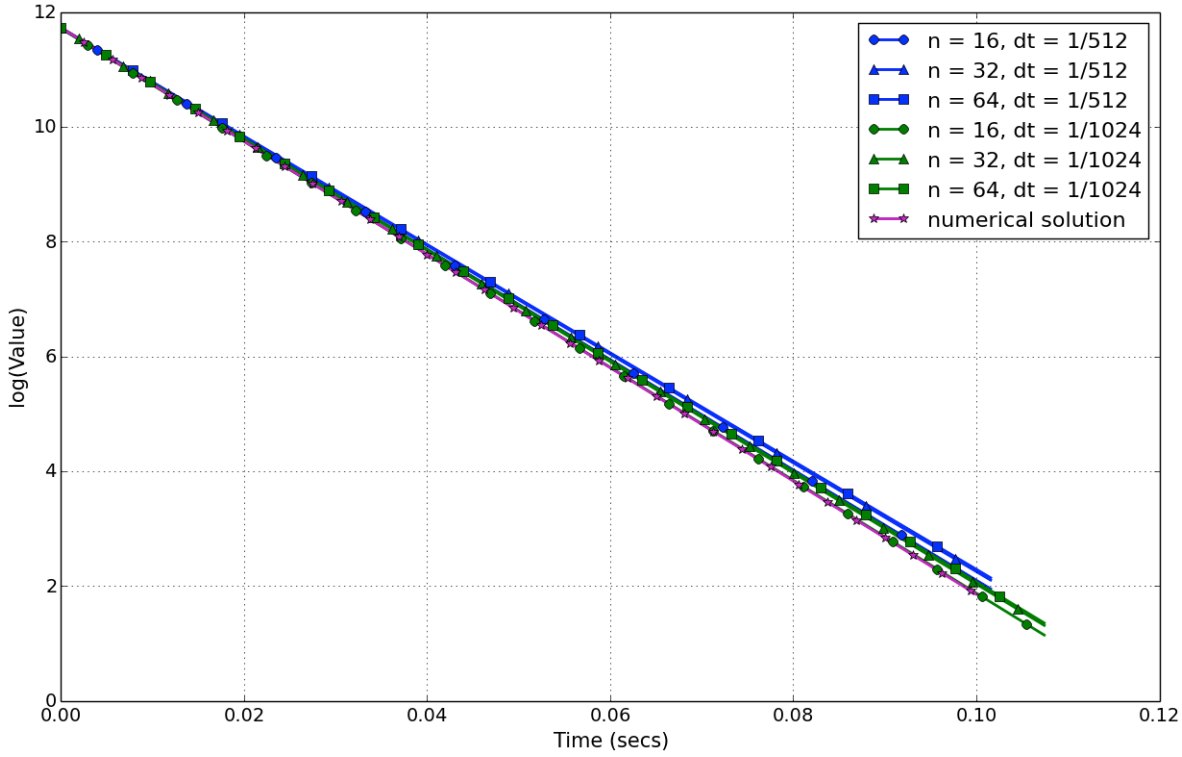


Figure 3: Galerkin Approximation of the Time-Dependent Heat Equation (Logarithmic Scale $-||\nabla u||^2$ Plot)

| | Approximation | Analytical | Approximation | Analytical | Approximation | Analytical |
|---|---|---|---|---|---|---|
| **n** | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{2048}}$ | $\mathbf{dt = \frac{1}{2048}}$ |
| **16** | 1.095e3 | 1.031e3 | 1.152e3 | 1.135e3 | 1.181e3 | 1.191e3 |
| **32** | 1.066e3 | 1.031e3 | 1.149e3 | 1.135e3 | 1.193e3 | 1.191e3 |
| **64** | 1.048e3 | 1.031e3 | 1.141e3 | 1.135e3 | 1.192e3 | 1.191e3 |

Table 11: **FOSLS Model:** "Energy" $(\frac{1}{2}||u||^2)$ Calculated at $t = dt$

| | Approximation | Analytical | Approximation | Analytical | Approximation | Analytical |
|---|---|---|---|---|---|---|
| **n** | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{2048}}$ | $\mathbf{dt = \frac{1}{2048}}$ |
| **16** | 1.112e1 | 5.541e-2 | 1.227e1 | 5.541e-2 | 1.290e1 | 5.541e-2 |
| **32** | 5.919e-1 | 5.541e-2 | 5.831e-1 | 5.541e-2 | 5.945e-1 | 5.541e-2 |
| **64** | 1.480e-1 | 5.541e-2 | 1.251e-1 | 5.541e-2 | _____ | 5.541e-2 |

Table 12: **FOSLS Model:** "Energy" $(\frac{1}{2}||u||^2)$ Calculated at $t = 0.1015625$

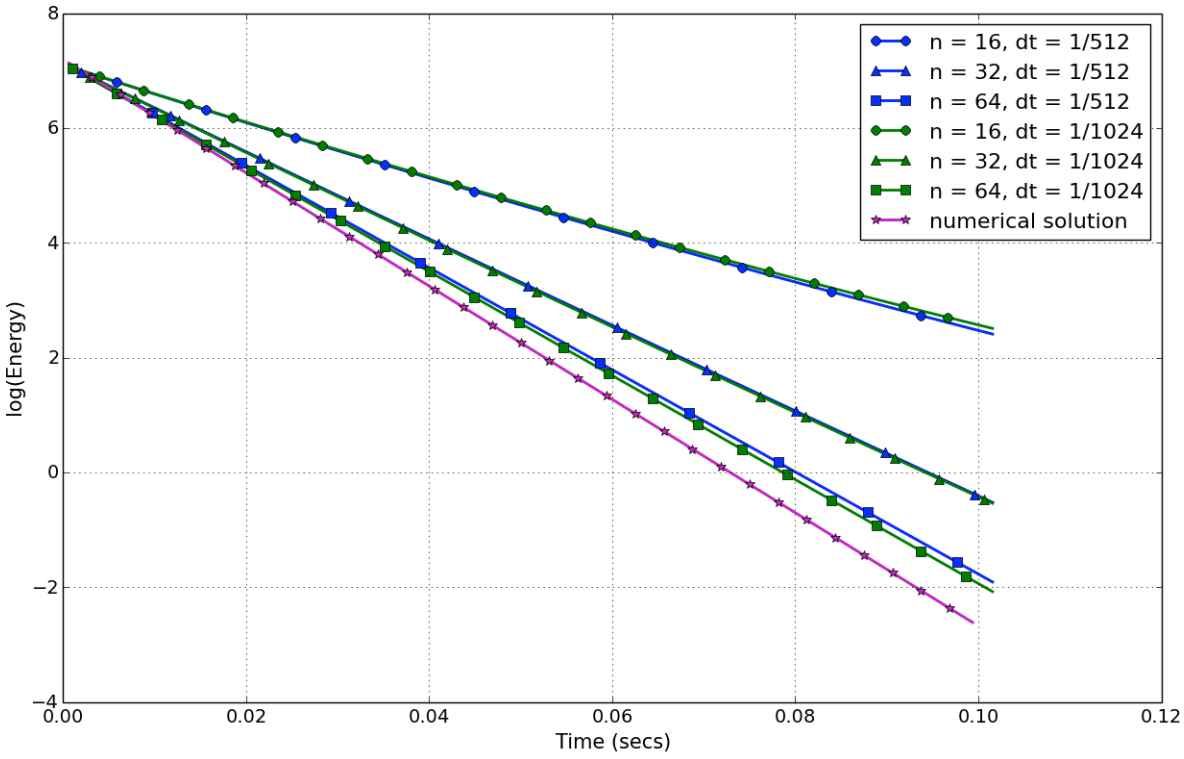


Figure 4: FOSLS Approximation of the Time-Dependent Heat Equation (Logarithmic Scale Energy Plot)

| | Approximation | Analytical | Approximation | Analytical | Approximation | Analytical |
|---|---|---|---|---|---|---|
| **n** | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{2048}}$ | $\mathbf{dt = \frac{1}{2048}}$ |
| **16** | -5.840e4 | -1.017e5 | -5.996e4 | -1.120e5 | -6.034e4 | -1.176e5 |
| **32** | -8.196e4 | -1.017e5 | -9.352e4 | -1.120e5 | -9.577e4 | -1.176e5 |
| **64** | -1.022e5 | -1.017e5 | -1.086e5 | -1.120e5 | -1.120e5 | -1.176e5 |

Table 13: **FOSLS Model:** Derivative of the Energy $(\frac{\partial}{\partial t}\frac{1}{2}||u||^2)$ at $t = dt$

| | Approximation | Analytical | Approximation | Analytical | Approximation | Analytical |
|---|---|---|---|---|---|---|
| **n** | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{2048}}$ | $\mathbf{dt = \frac{1}{2048}}$ |
| **16** | -4.745e2 | -5.469e0 | -4.890e2 | -5.469e0 | -4.957e2 | -5.469e0 |
| **32** | -4.696e1 | -5.469e0 | -4.403e1 | -5.469e0 | -4.313e1 | -5.469e0 |
| **64** | -1.437e1 | -5.469e0 | -1.185e1 | -5.469e0 | _____ | -5.469e0 |

Table 14: **FOSLS Model:** Derivative of the Energy $(\frac{\partial}{\partial t}\frac{1}{2}||u||^2)$ at $t = 0.1015625$

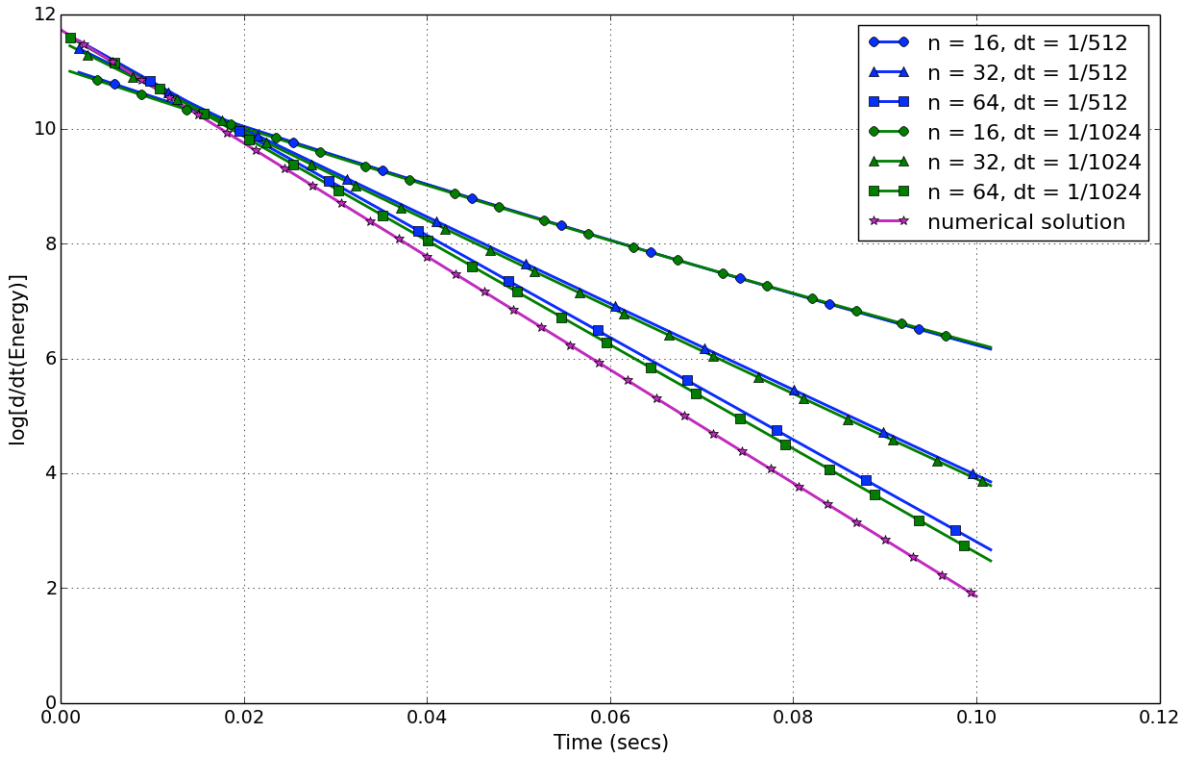


Figure 5: FOSLS Approximation of the Time-Dependent Heat Equation (Logarithmic Scale Derivative Plot)

| | Approximation | Analytical | Approximation | Analytical | Approximation | Analytical |
|---|---|---|---|---|---|---|
| **n** | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{2048}}$ | $\mathbf{dt = \frac{1}{2048}}$ |
| **16** | -1.105e5 | -1.017e5 | -1.161e5 | -1.120e5 | -1.191e5 | -1.176e5 |
| **32** | -1.058e5 | -1.017e5 | -1.140e5 | -1.120e5 | -1.184e5 | -1.176e5 |
| **64** | -1.036e5 | -1.017e5 | -1.128e5 | -1.120e5 | -1.179e5 | -1.176e5 |

Table 15: **FOSLS Model:** Right-Side of Equation (16) $(-||\nabla u||^2)$ at $t = dt$

| | Approximation | Analytical | Approximation | Analytical | Approximation | Analytical |
|---|---|---|---|---|---|---|
| **n** | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{2048}}$ | $\mathbf{dt = \frac{1}{2048}}$ |
| **16** | -1.689e3 | -5.469e0 | -2.009e3 | -5.469e0 | -2.208e3 | -5.469e0 |
| **32** | -7.304e1 | -5.469e0 | -9.404e1 | -5.469e0 | -1.253e2 | -5.469e0 |
| **64** | -1.471e1 | -5.469e0 | -1.256e1 | -5.469e0 | -1.224e1 | -5.469e0 |

Table 16: **FOSLS Model:** Right-Side of Equation (16) $(-||\nabla u||^2)$ at $t = 0.1015625$

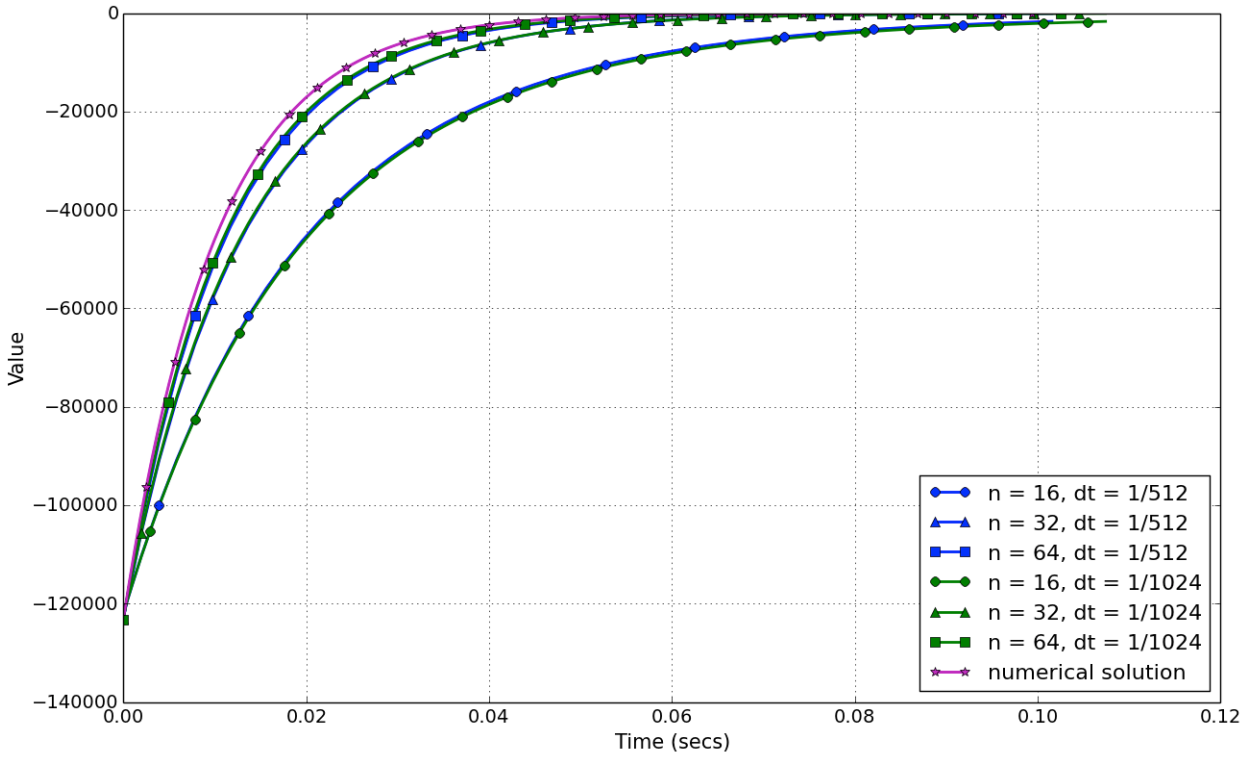


Figure 6: FOSLS Approximation of the Time-Dependent Heat Equation $(-||\nabla u||^2$ Plot)

## 2.3 Stokes' Equation Energy Law Models.

| | Approximation | Analytical | Approximation | Analytical | Approximation | Analytical |
|---|---|---|---|---|---|---|
| **n** | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{2048}}$ | $\mathbf{dt = \frac{1}{2048}}$ |
| **16** | 8.784e-1 | 8.501e-1 | 1.040e0 | 1.031e0 | 1.137e0 | 1.135e0 |
| **32** | 8.786e-1 | 8.501e-1 | 1.040e0 | 1.031e0 | 1.138e0 | 1.135e0 |
| **64** | 8.786e-1 | 8.501e-1 | 1.040e0 | 1.031e0 | 1.138e0 | 1.135e0 |

Table 17: **Galerkin Model:** Energy ($\frac{1}{2}||u||^2$) Calculated at $t = dt$

| | Approximation | Analytical | Approximation | Analytical | Approximation | Analytical |
|---|---|---|---|---|---|---|
| **n** | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{2048}}$ | $\mathbf{dt = \frac{1}{2048}}$ |
| **16** | 1.360e-8 | 2.457e-9 | 6.069e-9 | 2.457e-9 | 3.907e-9 | 2.457e-9 |
| **32** | 1.365e-8 | 2.457e-9 | 6.091e-9 | 2.457e-9 | 3.922e-9 | 2.457e-9 |
| **64** | 1.365e-8 | 2.457e-9 | 6.092e-9 | 2.457e-9 | 3.923e-9 | 2.457e-9 |

Table 18: **Galerkin Model:** Energy ($\frac{1}{2}||u||^2$) Calculated at $t = 0.1015625$

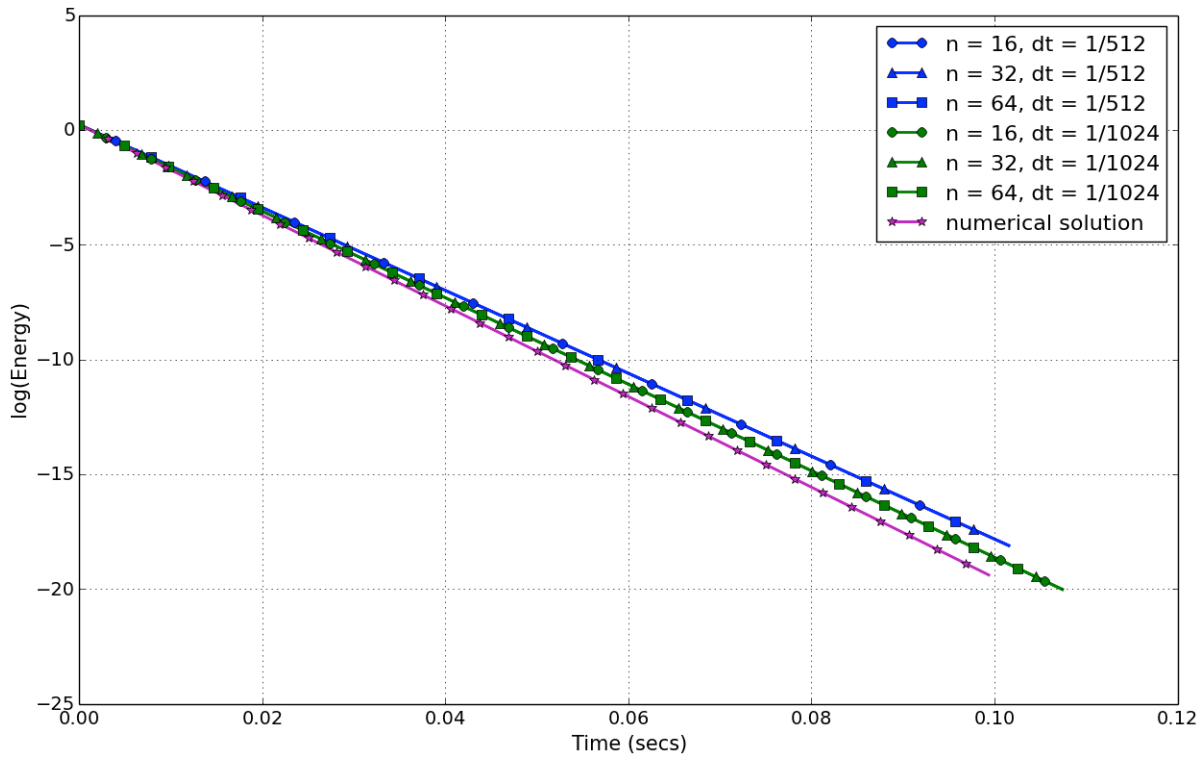


Figure 7: Galerkin Approximation of the Time-Dependent Stokes' Equation (Logarithmic Scale Energy Plot)

| | Approximation | Analytical | Approximation | Analytical | Approximation | Analytical |
|---|---|---|---|---|---|---|
| **n** | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{2048}}$ | $\mathbf{dt = \frac{1}{2048}}$ |
| **16** | -1.901e2 | -1.678e2 | -2.151e2 | -2.035e2 | -2.299e2 | -2.241e2 |
| **32** | -1.901e2 | -1.678e2 | -2.152e2 | -2.035e2 | -2.300e2 | -2.241e2 |
| **64** | -1.901e2 | -1.678e2 | -2.152e2 | -2.035e2 | -2.300e2 | -2.241e2 |

Table 19: **Galerkin Model:** Derivative of the Energy ($\frac{\partial}{\partial t}\frac{1}{2}||u||^2$) at $t = dt$

| | Approximation | Analytical | Approximation | Analytical | Approximation | Analytical |
|---|---|---|---|---|---|---|
| **n** | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{512}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{1024}}$ | $\mathbf{dt = \frac{1}{2048}}$ | $\mathbf{dt = \frac{1}{2048}}$ |
| **16** | -2.944e-6 | -4.84912e-7 | -1.256e-6 | -4.84912e-7 | -7.900e-7 | -4.84912e-7 |
| **32** | -2.954e-6 | -4.84912e-7 | -1.260e-6 | -4.84912e-7 | -7.928e-7 | -4.84912e-7 |
| **64** | -2.954e-6 | -4.84912e-7 | -1.260e-6 | -4.84912e-7 | -7.930e-7 | -4.84912e-7 |

Table 20: **Galerkin Model:** Derivative of the Energy ($\frac{\partial}{\partial t}\frac{1}{2}||u||^2$) at $t = 0.1015625$

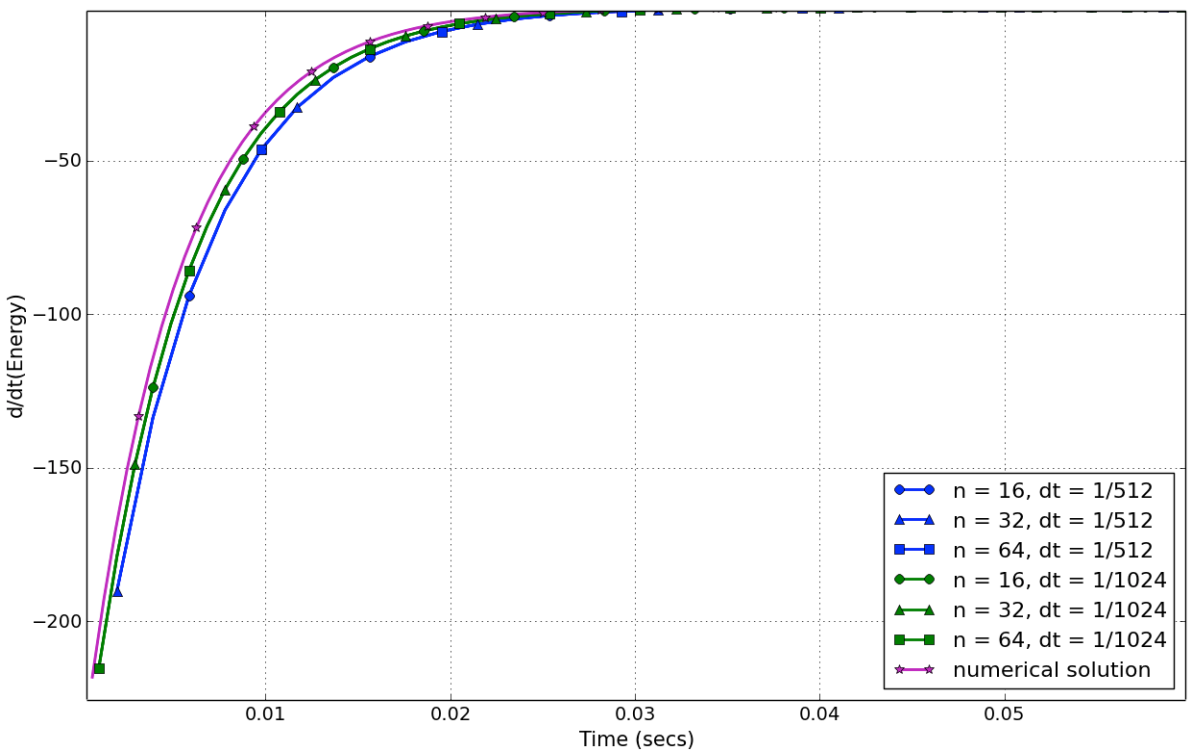


Figure 8: Galerkin Approximation of the Time-Dependent Stokes' Equation (Derivative Plot)

## 2.4 FOSLS$^2$ and FOSLS Least-Squares.

## Acknowledgements.

This research was funded by the David Baltimore/Broad Foundation Endowment at Swarthmore College for undergraduate off-campus scientific research.