<center>

ME 500
Numerical Methods in Mechanical Engineering
Assignment 5

Brandon Lampe

December 2, 2015

</center>

1 Theoretical Formulations

1.1 Definitions

spatial discretization   procedure of discretizing the problem domain into an ordered set of nodes and the space between nodes (h).

mesh   is what results from the spatial discretization.

nodes   locations in the mesh where approximate solutions are calculated.

node spacing   (h) the distance between nodes where no approximate solutions are calculated.

finite difference method   a method of obtaining approximate solutions to differential over a problem domain that has discretized via a mesh. Approximate solutions are only obtained at nodal locations, and these approximate solutions are suitable if they show convergence.

stencil   is the assumed form of the approximating equations in $[A]$. Differing rates of convergence may be obtained by assuming different stencils. Additionally, a higher rate of convergence may be obtained by assuming a stencil in the forcing vector $\{f\}$.

local truncation error   (τ) is the residual vector obtained when the exact solution (ϕ^e) is substituted into the approximating equations:

$$\{\tau\} = [A]\{\phi^e\} - \{f\}$$

The finite difference approximating equations have been developed by minimizing the local truncation error. That is, the coefficients of the stencil were obtained by solving for τ with $\{\phi^e\}$ being approximated with a truncated Taylor Series Expansion of ϕ^e_{i-1} and ϕ^e_{i+1} about ϕ^e_i. The order of τ ($\tau\ O(?)$) is the magnitude of the lowest order terms in the Taylor Series Expansion that were not matched exactly.

error   the difference between the exact and analytical solutions. A scalar measure of error (ϵ) may be obtained via a scaled p-norm of the error vector:

$$\{e\} = \{\phi^e\} - \{\phi\}$$
$$\epsilon = \frac{\|\{e\}\|_p}{n^{1/p}}$$

stability   an approximate solution has stability if it remains bounded within the problem domain (does not approach infinity). If stability is assumed, this is analogous to assuming the minimum eigenvalue (λ_{min}) of the approximating equation matrix $[A]$ is greater than zero. Which is analogous to having a condition number (C) that does not approaching infinity:

$$C = \frac{\lambda_{max}}{\lambda_{min}}$$

Stability implies that small changes in the problem data result in small changes in the solution.

<center>1</center>

consistency   an approximate solution has consistency if the local truncation error approaches zero as the spacing between nodes (h) approaches zero. This implies that the approximating equations be of the order of h raised to a power that is greater than or equal to one. E.g., consistency is obtained if:

$$\tau \; O(h^r)$$
$$r \geq 1$$

Where r is the rate of convergence.

convergence   an approximate solution has convergence if the measure of error (ϵ) approaches zero as the spacing between nodes (h) approaches zero. This is what we want!

Lax's Equivalence Theorem   states that if an approximate solution has stability and consistency then it also has convergence.

1.2   Formulation of Finite Difference Approximation to the ODE

The model ODE will be:

$$(k^A \phi^e_{,x})_{,x} + f(x) = 0 \tag{1}$$

Where, ϕ^e is the exact solution. We can also express the ODE in index form when discretized:

$$(k^A \phi^e_{i,x})_{,x} + f_i = 0 \tag{2}$$

Basic steps involved in developing a finite difference algorithm are outline in order below:

1. Assume a form of the approximating equations at a specific node (i) in the domain, this form is the stencil. We have assumed a three-point stencil here:

$$\alpha_{i-1}\phi_{i-1} + \alpha_i \phi_i + \alpha_{i+1}\phi_{i+1} = f_i$$

This form implies the linear ODE will be approximated by setting up a linear system of equations:

$$[A]\{\phi\} = \{f\}$$

   - approximating equations: $[A] \implies$ the α terms we choose to approximate the solution
   - approximate values of the primary variable: $\{\phi\} \implies$ the unknowns we are solving for
   - forcing function: $\{f\} \implies$ vector of known values

2. Show that the solution has consistency, e.g., as $h \to 0$ also $\tau_i \to 0$

3. Show that the solution has stability, e.g., the condition number must no approach infinity

4. Lax's Equivalence Theorem states that if a numerical approximate has consistency and stability then it also has convergence, e.g., as $h \to 0$ also $\epsilon \to 0$. That is, as the mesh is refined the approximate solution will approach the exact solution.

1.2.1   Components of [A]

Note, this section assumes that nodal spacing (h) is constant. The components of [A] are determined by looking at the local truncation error (τ) at a specific node (i), where:

$$\{\tau\} = [A]\{\phi^e\} - \{f\}$$

or in the discretized form:

$$\tau_i = \alpha_{i-1}\phi^e_{i-1} + \alpha_i \phi^e_i + \alpha_{i+1}\phi^e_{i+1} - f_i \tag{3}$$

We now want to approximate the values of ϕ^e_{i-1} and ϕ^e_{i+1} in Equation 3 about the point i and choose to use a Taylor Series Expansion (TSE) to accomplish this. The full TSE is:

$$\phi^e_{i-1} = \sum_{n=0}^{\infty} \frac{\phi^e_{i,n}}{n!} (i - (i-1))^n$$

$$i,n \implies n^{th} \text{ derivative at the } i^{th} \text{ node}$$

$$h = i - (i-1), \text{ which is the nodal spacing, therefore}$$

$$\phi^e_{i-1} = \sum_{n=0}^{\infty} \frac{\phi^{e(n)}_i}{n!} (-h)^n$$

Now, the expanded TSEs of ϕ^e_{i-1} and ϕ^e_{i+1} are:

$$\phi^e_{i-1} = \phi^e_i + \frac{\phi^e_{i,x}}{1!}(-h)^1 + \frac{\phi^e_{i,2x}}{2!}(-h)^2 + \frac{\phi^e_{i,3x}}{3!}(-h)^3 + \frac{\phi^e_{i,4x}}{4!}(-h)^4 + \frac{\phi^e_{i,5x}}{5!}(-h)^5 + ... \tag{4}$$

$$\phi^e_{i+1} = \phi^e_i + \frac{\phi^e_{i,x}}{1!}(h)^1 + \frac{\phi^e_{i,2x}}{2!}(h)^2 + \frac{\phi^e_{i,3x}}{3!}(h)^3 + \frac{\phi^e_{i,4x}}{4!}(h)^4 + \frac{\phi^e_{i,5x}}{5!}(h)^5 + ... \tag{5}$$

Also, we wish to express f_i using Equation 2 from above:

$$f_i = -(k^A \phi^e_{i,x})_{,x} \tag{6}$$

We now substitute Equations 4, 5, and 6 into the discretized form of the local truncation error (Equation 3). Additionally, group like terms as:

$$\tau_i = c_0 \phi^e_i + c_1 \phi^e_{i,x} + c_2 \phi^e_{i,2x} + c_3 \phi^e_{i,3x} + c_4 \phi^e_{i,4x} + c_5 \phi^e_{i,5x} + ... \tag{7}$$

Where the leading c terms in Equation 7 represent a collection of terms that we need to solve for. Recall, from Equation 3 that we assume ϕ^e_i exists and we know what f_i is, therefore we only need to solve for the α terms. Here we have chosen a three-point stencil, which results in three unknown values of alpha: α_{i-1}, α_i, α_{i+1}. To solve for the α values, we determine the values of collections and set c_0, c_1 and c_2 equal to zero:

$$c_0 = \alpha_{i-1} + \alpha_i + \alpha_{i-1} = 0$$

$$c_1 = -h(\alpha_{i-1} + \alpha_{i+1}) = 0$$

$$c_2 = \frac{h^2}{2}(\alpha_{i-1} + \alpha_{i+1}) + k^A = 0$$

$$c_3 = \frac{-h^3}{6}(\alpha_{i-1} + \alpha_{i+1})$$

$$c_4 = \frac{h^4}{24}(\alpha_{i-1} + \alpha_{i+1})$$

By setting c_0, c_1 and c_2 equal to zero we can solve for α_{i-1}, α_i, α_{i+1} using c_3 and c_4. This results in:

- $\alpha_{i-1} = \frac{-k^A}{h^2}$

- $\alpha_i = \frac{2k^A}{h^2}$

- $\alpha_{i+1} = \frac{-k^A}{h^2}$

The remaining nonzero terms in τ_i are now:

$$\tau_i = c_3 \phi^e_{i,3x} + c_4 \phi^e_{i,4x} + c_5 \phi^e_{i,5x} + ... \tag{8}$$

By substituting in the determined values of α into c_3 and c_4, we see that:

$$c_3 = 2k^A(h-h) \implies 0 \text{ for uniform mesh spacing}$$

$$c_4 = \frac{-k^A}{12}(h^2 + h^2 + h^2) \implies c_4 \text{ is of order } h^2$$

3

Therefore, by choosing c_0, c_1 and c_2 equal to zero for a uniform mesh c_3 also became zero, and the lowest nonzero term in τ_i is c_4, which contains terms that are dependent on the problem data (k^A) and nodal spacing (h). When trying to compare numerical schemes, the problem data is a constant; therefore when analyzing the numerical scheme we only look at terms pertaining to the discretization (h). Because of that, we say that this scheme is of second order (h^2), and the theoretical rate of convergence is 2. That is, as $h \to 0$, $\tau_i \to 0$ (consistency). An additional view is that we have matched terms in the TSE exactly up to those containing h^2, therefore we have truncated the TSE at h^2.

This analysis only applies to interior nodes of the mesh and a separate analysis is necessary to determine the rate of convergence on the boundaries. Note that in Equation 7, terms up to c_3 are assumed to exist. This implies that derivatives of the exact solution up to the third derivative must also exist. Therefore, the exact solution must be at least C^3 continuous (it's third derivative must exist) for this theoretical rate of convergence to hold.

Therefore, the components of the approximating equation are:

$$[A] = \begin{bmatrix} x_{11} & x_{12} & 0 & \cdots\cdots\cdots & & 0 \\ \alpha_{i-1} & \alpha_i & \alpha_{i+1} & 0 & \cdots & 0 \\ 0 & \alpha_{i-1} & \alpha_i & \alpha_{i+1} & & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & & \alpha_{i-1} & \alpha_i & \alpha_{i+1} \\ 0 & 0 & \cdots & 0 & x_{n,n-1} & x_{n,n} \end{bmatrix} = \frac{k^A}{h^2} \begin{bmatrix} x_{11} & x_{12} & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & x_{n,n-1} & x_{n,n} \end{bmatrix}$$

Where the x terms in $[A]$ denote components that will be modified in order to satisfy boundary conditions.

1.2.2 Boundary Conditions

Boundary conditions may be satisfied by modifying the forcing function and approximating equations. such as:

$$\frac{k^A}{h^2} \begin{bmatrix} x_{11} & x_{12} & & & & \\ -1 & 2 & -1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & x_{n,n-1} & x_{n,n} \end{bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \vdots \\ \phi_{n-1} \\ \phi_n \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ \vdots \\ \vdots \\ f_{n-1} \\ f_n \end{Bmatrix}$$

The choice of how the modification is completed may vary, but essentially the boundary condition terms must be satisfied. For example, a temperature prescribed boundary condition of $\phi^e(x = 0) = T_0$ may be implemented by:

$$\frac{k^A}{h^2} \begin{bmatrix} 1 & 0 & & & & \\ -1 & 2 & -1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & x_{n,n-1} & x_{n,n} \end{bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \vdots \\ \phi_{n-1} \\ \phi_n \end{Bmatrix} = \begin{Bmatrix} \frac{h^2}{k^A} T_0 \\ f_2 \\ \vdots \\ \vdots \\ f_{n-1} \\ f_n \end{Bmatrix}$$

A similar approach may be used for flux type boundary conditions.

1.3 Formulation of Finite Difference Approximation to One-Degree-of-Freedom ODE with Time as the Independent Variable

For the model problem:

$$C\dot{T}^e + KT^e = f \tag{9}$$

Where

$$C = \text{heat capacitance}$$
$$K = \text{thermal conductivity}$$
$$T^e = \text{ the exact analytical solution (primary variable)}$$

The discretized form of Equation 9 for two different time steps may be represented as:

$$C\dot{T}^e(t^{k+1}) + KT^e(t^{k+1}) = f(t^{k+1}) \tag{10}$$

$$C\dot{T}^e(t^k) + KT^e(t^k) = f(t^k) \tag{11}$$

The general Trapezoidal rule was implemented to provide a range of time integrators such that:

$$T^{k+1} = T^k + s\left[\alpha\dot{T}^{k+1} + (1-\alpha)T^k\right] \tag{12}$$

Where

$$k = \text{ time increment}$$
$$\alpha = \text{ method of time integration}$$
$$T = \text{ the approximate solution}$$

Different values of α provide fundamentally different method of approximating the solution:

$$\alpha = 0 \implies \text{explicit, backwards difference}$$
$$\alpha = 0.5 \implies \text{implicit, central difference}$$
$$\alpha = 1.0 \implies \text{implicit, forward difference}$$

The term implicit here means that the solution process requires solving a system of equations (e.g., the linear algebraic problem). Whereas, an explicit scheme does not necessitate the inversion of a matrix. Additionally, explicit schemes have a critical time step size which cannot be exceed else stability will not be obtained. Whereas, implicit schemes are unconditionally stable, although the solution may be inaccurate if the solution necessitates a time step size smaller than the chosen step size.

1.3.1 Develop Approximating Equations

As was done for finite difference method, we propose a stencil, say:

$$\gamma T^{k+1} + \beta T^k = (1-\alpha)f^k + \alpha f^{k+1} \tag{13}$$

Then the local truncation error at the kth time step is determined by substituting in the exact solution into the approximating equation:

$$\tau^k = \gamma T^e(t^{k+1}) + \beta T^e(t^k) - \left[(1-\alpha)f(t^k) + \alpha f(t^{k+1})\right] \tag{14}$$

Substitute in Equations 10 and 11 for the forcing terms in Equation 14:

$$\tau^k = \gamma T^e(t^{k+1}) + \beta T^e(t^k) - \left[(1-\alpha)(C\dot{T}^e(t^k) + KT^e(t^k)) + \alpha(C\dot{T}^e(t^{k+1}) + KT^e(t^{k+1}))\right] \tag{15}$$

The local truncation error is now in terms of the primary variable (T^e). Now perform a Taylor Series expansion about terms evaluated at t^{k+1}. This is shown in lecture notes 24-15. A similar analysis is performed as was done for the finite difference method which results in:

$$\beta = K(1-\alpha) - \frac{C}{s}$$
$$\gamma = \frac{C}{s} + \alpha K$$
$$s = \text{ time step size}$$

This results in $|\tau^k| \leq 0.5Cs(1-s\alpha)|\ddot{T}^e_{max}|$ and is therefore on the order of s (the time increment size). Note, all other terms in τ^k depend on problem data only.

The system integrator is obtained by writing Equations 10 and 11 in the approximate form and combining them with the time integrator:

$$CT^{\dot{k+1}} + KT^{k+1} = f^{k+1} \tag{16}$$

$$C\dot{T}^k + KT^k = f^k \tag{17}$$

$$T^{k+1} = T^k + s\left[\alpha\dot{T}^{k+1} + (1-\alpha)T^k\right] \tag{18}$$

The combined form of Equations 16, 17, and 18 form the system integrator as shown below:

$$AT^{k+1} = BT^k + F^k \tag{19}$$

$$where$$
$$A = C + \alpha sK$$
$$B = C - (1-\alpha)sK$$
$$F^k = (1-\alpha)sf^k + \alpha sf^{k+1}$$

1.4 Formulation of Finite Difference Approximation to PDE: $C\phi_{,t} - (k\phi_{,x})_{,x} = Q$

2 Analysis of One-Dimensional Steady-State Heat Equation

2.1 Problem Data

- Governing Equation: $\frac{d}{dx}\left(k^A\frac{dT}{dx}\right) + Q(x) = 0$

- Problem Domain: $0 \leq x \leq 10$

- Coeficient Function: $k^A = 20$

- Forcing Function: $Q(x) = \frac{-k^A}{(x+1)^2}$

- Boundary Conditions

  - Dirichlet: $T(0) = 0$
  - Nuemann: $Q^* = \frac{2k^A}{(x+1)} \implies \left(\frac{dT}{dx}\right)_{x=10} = \frac{2}{11}$

2.2 Write Finite Difference Algorithm

My finite difference algorithm is in the attached code listing titled *HW05.prob2*

2.3 Comparison of Results

The analytical solution to this problem is: $T(x) = 2\ ln(x+1)$. Numerical approximations are shown in Figure 1, where approximate solutions only exist at the filled circles. The dotted lines connecting the circles are shown only as a visual aid, and these lines do not represent the numerical approximation.

2.4 Rate of Convergence

For each of the mesh discretizations, using a scaled p-norm, an error norm was calculated between the numerical approximation and the exact solution (Figure 2). The numerical rate of convergence was determined to be approximately one, which agrees with the theoretical rate of convergence for implemented three-point stencil.
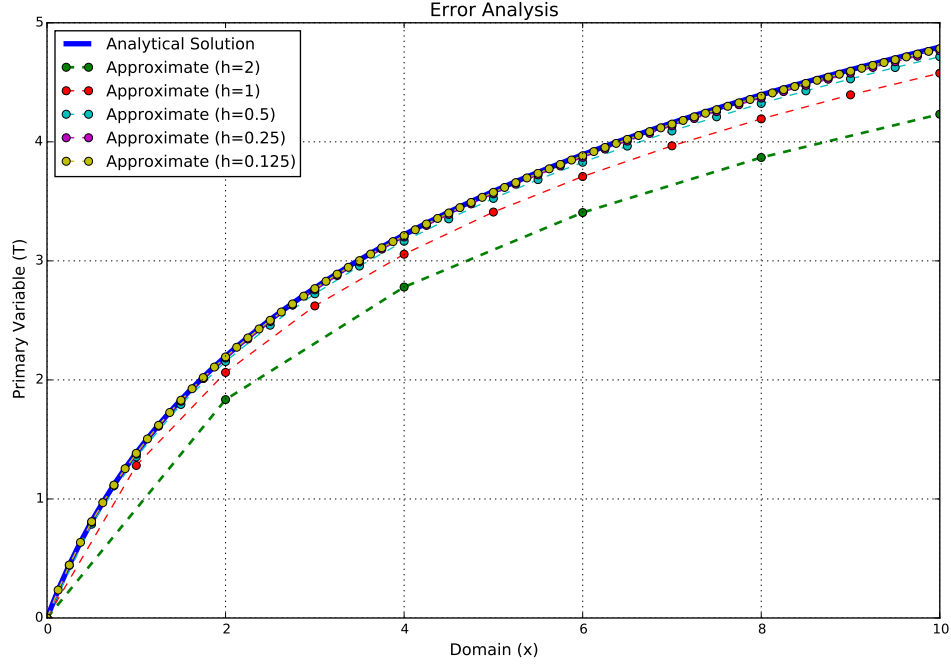
Figure 1: Numerical (approximate) solutions over a range of nodal spacing (h) and the exact analytical solution.

3 Time Integration

The code listing for this problem is provided in section *HW05.prob3*.

3.1 Single-Degree-of-Freedom Problem

3.1.1 Theoretical Analysis of Critical Time Step

Consider the homogeneous ODE for analysis of the crtical time step size (s_{cr}:

$$C\dot{T} + KT = 0 \tag{20}$$

Define the following parameters, as was done in section 8.4 of the course text and in lecture Notes 24-6:

$$s = \text{time step size}$$
$$\bar{s} = s\Psi \implies \text{dimensionless time step size}$$
$$\Psi = \frac{K}{C}$$

For conditionally stable integration (explicit), we can now define a dimensionless critical time step ($\bar{s}_{cr}$):

$$\bar{s}_{cr} = \frac{1}{\frac{1}{2} - \alpha} \tag{21}$$

For unconditionally stable integration methods ($0.5 \leq \alpha \leq 1$), stability is not dependent on the time step size (s). However, for large time steps the numerical approximate may not be very accurate.
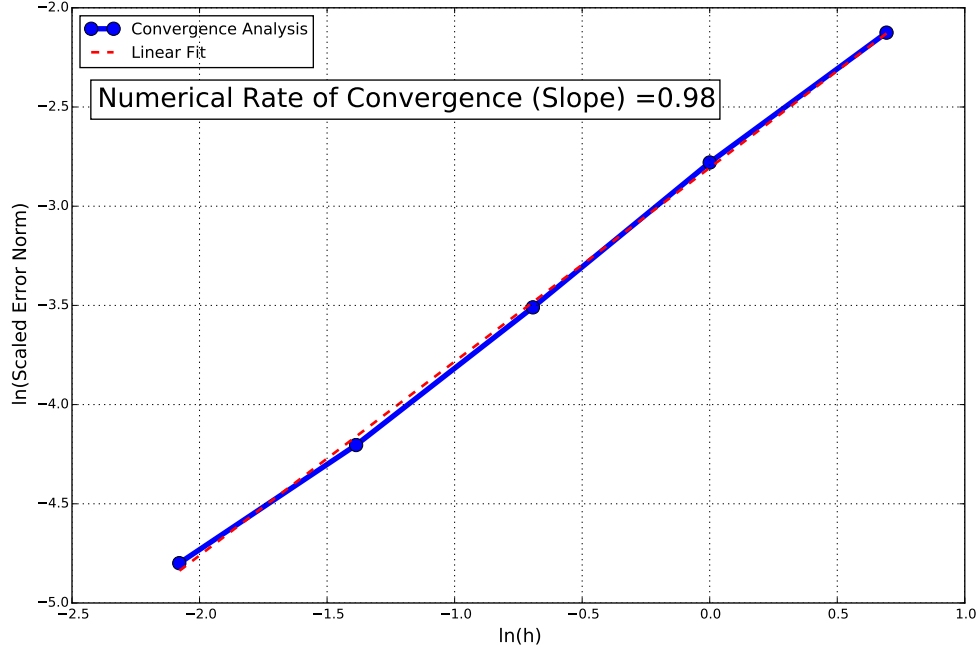
7

Figure 2: Analysis of the numerical rate or convergence.

3.1.2 The Model Problem

Now consider:

$$\dot{T} + \Lambda T = 0 \quad \text{for: } T(0) = 2, \quad 0 \le t \le 8 \tag{22}$$

With $\Lambda = 1$, this results the a critical time step size, $s_{cr} = 2$, for $\alpha = 0$. Results from this analysis are shown in Figures 3. For explicit integration ($\alpha = 0$) it appeared to be unstable for time steps sizes (s) greater than 2, which agrees with the theoretical analysis. The instabilities are readily observed on an enlarged scale (Figure 4). When using explicit integration, time step sizes of 2 and 1 appear to be stable over the problem domain, but they do not provide very good results.

The implicit integration schemes appeared to be stable for all time step sizes across the entire domain, which agrees with theory. Convergence occurs with a time step size of 1 when $\alpha = 0.5$ (Crank-Nicolson), where time step sizes of 0.25 were needed for the other two schemes to achieve the same level of error. The theoretical rate of convergence derived in class showed that the integration method for the Crank-Nicolson scheme was second-order in time and the other schemes were only first order. Therefore, regarding the approximate rates of convergence, my numerical results agree with theory.
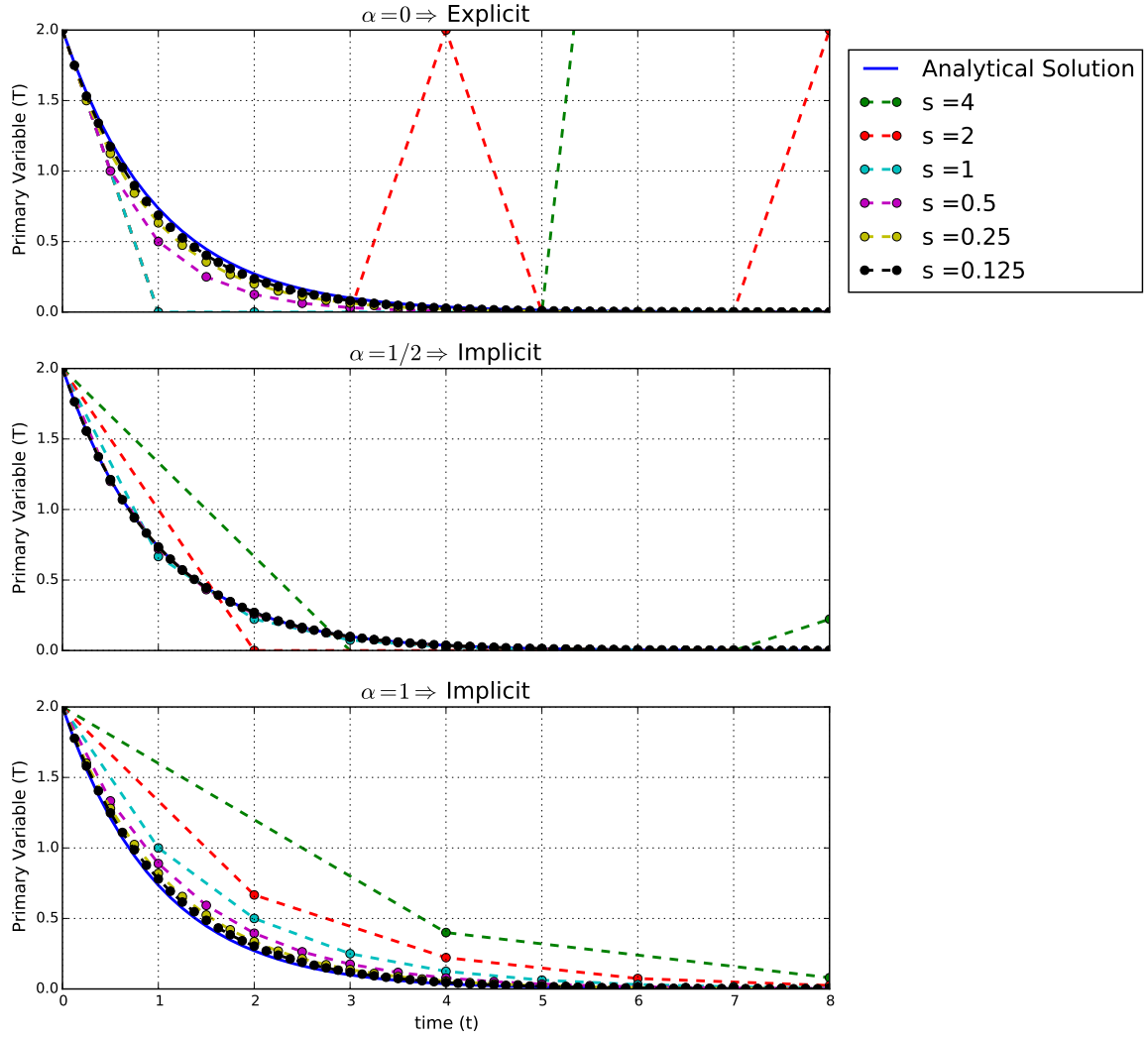
Figure 3: Numerical results over a range of time steps with explicit (top), implicit with $\alpha = 0.5$ (middle), and implicit with $\alpha = 1$ (bottom)..
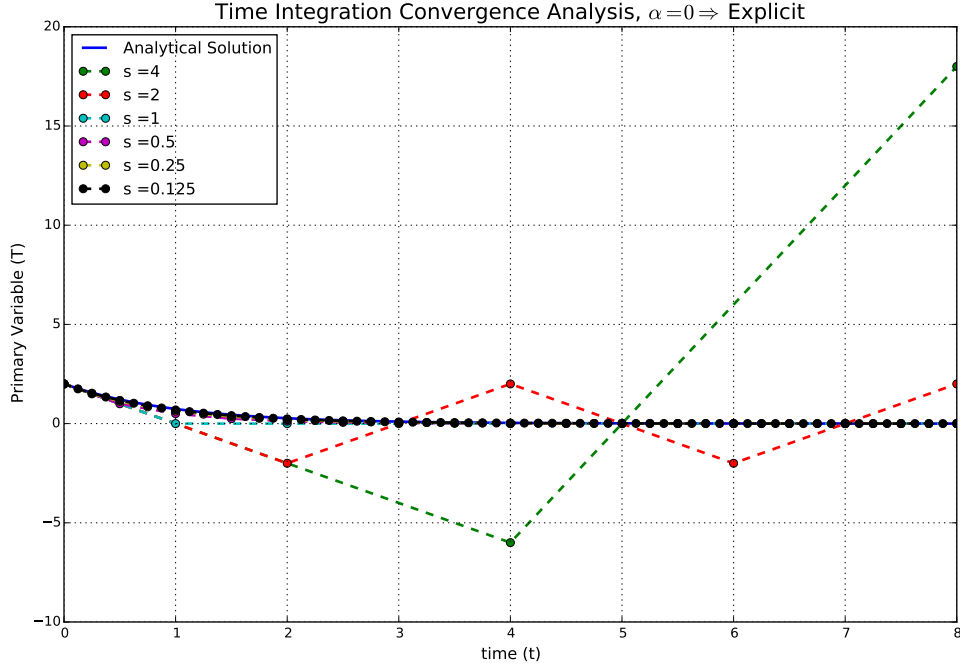
Figure 4: Numerical results for explicit integration (same as top frame in Figure 3) with an alternate vertical scale that allows for displaying the magnitude of numerical instability.

3.2 Single-Degree-of-Freedom with a Forcing Term

Based on the problem data, I calculated the critical time step for $\alpha = 0$ as:

$$\Psi = \frac{F(t)_{max} - K}{C}$$
$$= \frac{10 - 2}{1} = 8$$
$$s_{cr} = \frac{\bar{s}_{cr}}{\Psi}$$
$$= 0.25$$

However, when using a time step size of 0.25 the analytical solution had poor resolution. This is the result of the problem being nonhomogeneous (nonzero forcing function). Therefore, a minimium step size of 0.02 was chosen. The other bound for time step size was calculated as:

$$\frac{t_p}{4} = \frac{2\pi}{\Omega} \frac{1}{4} = 0.157 \tag{23}$$

Therefore, analysis of this problem was performed for: $0.02 \leq s \leq 0.157$. However, the problem statement implies that 0.157 should be the upper bound of s. So I am unsure on this problem, but my results are shown below in Figures 5, 6, and 7. The error for time step sizes appears small, time permitting, a thorough error analysis would involved calculated a scaled error norm for each time step size... but this was not done here.

Visually, the results for α values of 0.5, 0.75 and 1.0 appear to match the analytical solution very well for small time steps ($s = 0.02$). However, for larger times steps ($s = 0.16$) the numerical approximation provided by $\alpha = 1.0$ appears to match the exact solution much better than for the approximation resulting

10

from $\alpha = 0.5$. This is likely because for values of α less than unity, the numerical integration depends on the previous time step, and for an oscillating solution such as in this problem, the previous time step is a poor indicator of the results during the next time step. This indicates that for very dynamic problems, using a fully implicit integration scheme ($\alpha = 1.0$) may be beneficial.
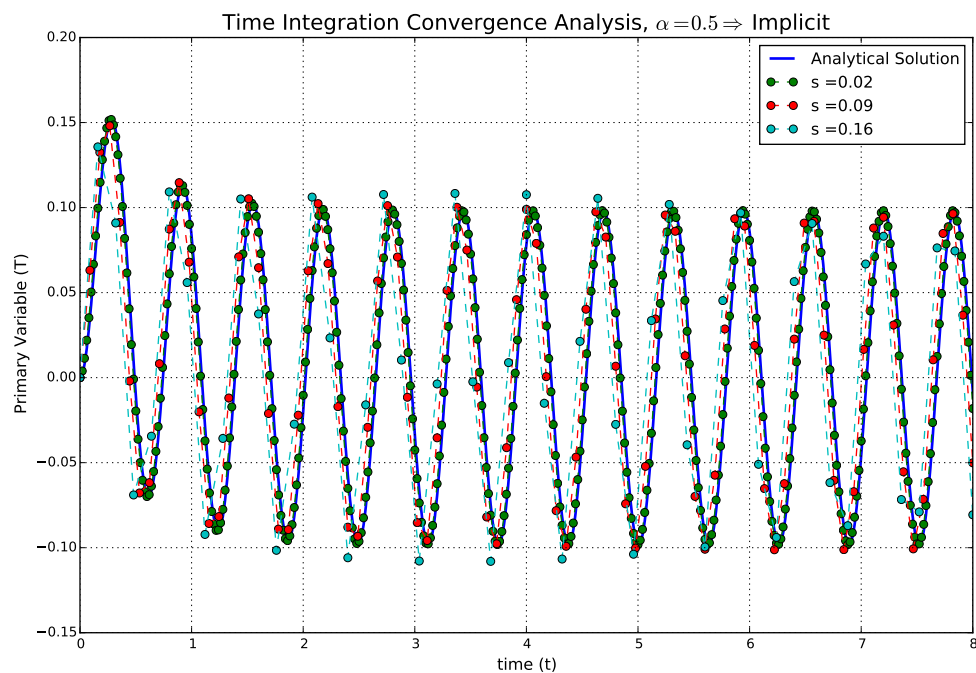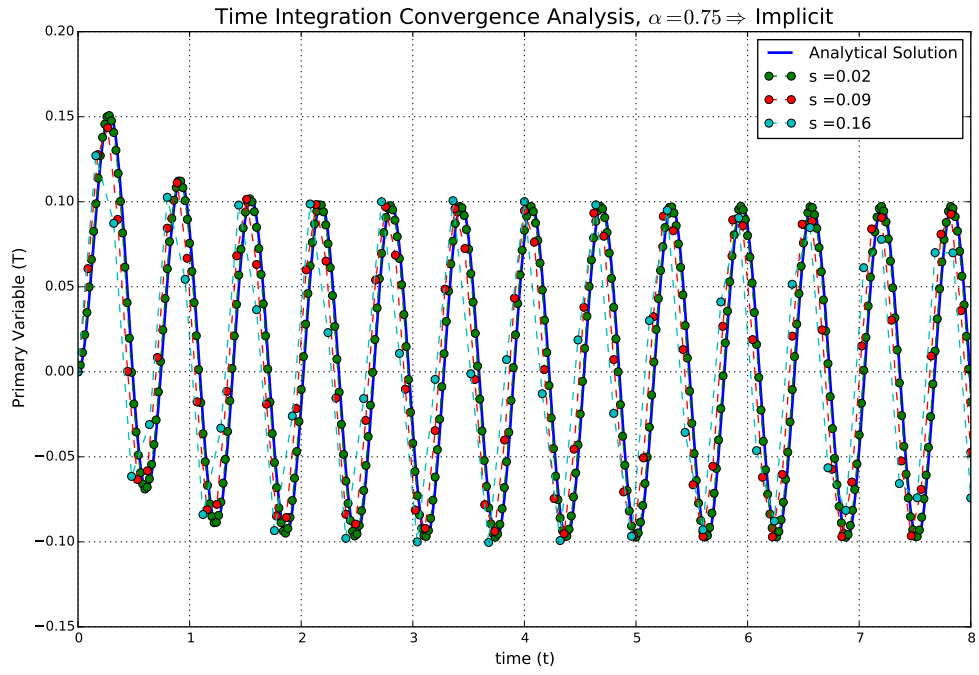

Figure 5: Numerical results for transient problem with $\alpha = 0.5$.

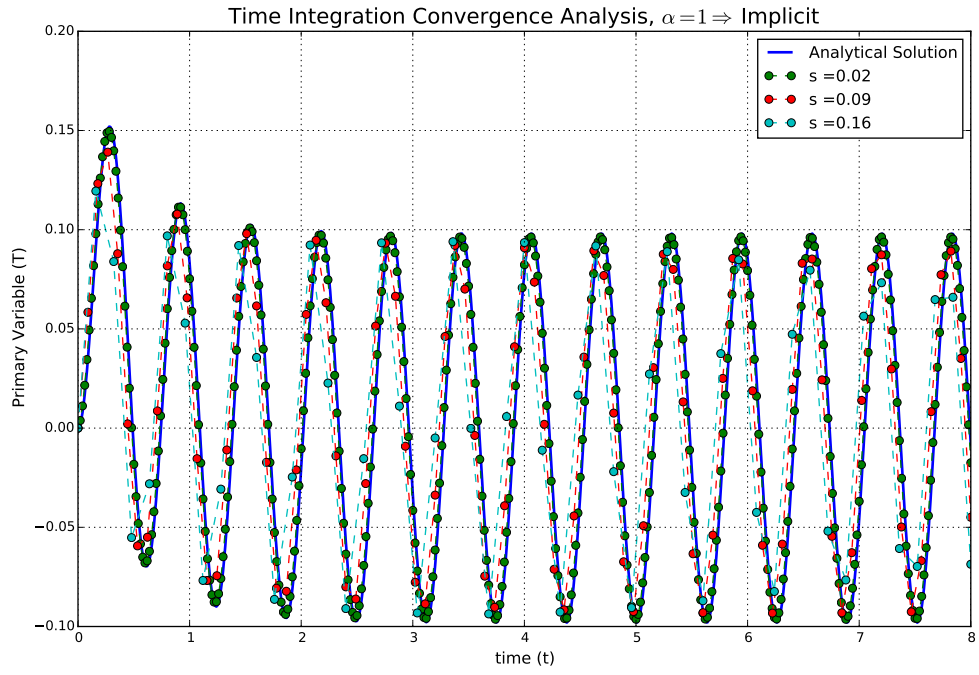Figure 6: Numerical results for transient problem with $\alpha = 0.75$.

Figure 7: Numerical results for transient problem with $\alpha = 1$.

4 Transient Problem

The code listing for this problem is provided in section *HW05.prob4*. Using the relationship derived in class for the critical time step size for explicit integration $\left(s_{cr} = \frac{Ch^2}{2k}\right)$, when a nodal spacing of 0.5 was used, then $s_{cr} = 0.0025$. Because of this small time step size, I chose to present my numerical solutions with results from the implicit scheme with $\alpha = 1$. This choice of α allows me to use a time step size of 0.1, rather than the explicit method requiring a step size of approximately 0.0025 when the same spatial discretization was used.

4.1 Constant Boundary Conditions

Results for the problem *4.i* are shown below in Figure 8. The contradiction between the boundary condition at $x = 0$ and intial condition was handled by overriding the intial condition with the prescribed temperature at $x = 0$. This is evident in the results for time zero in Figure 8. I am not sure if this was the correct method to handle the problem, but this approach seems physically reasonable.
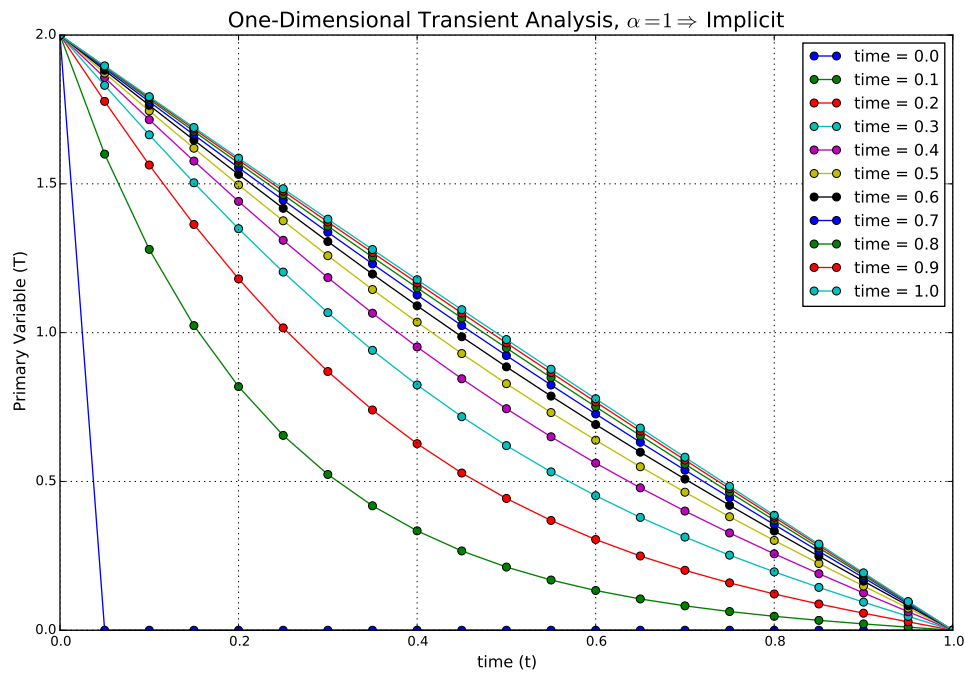

Figure 8: Numerical solution to transient problem with constant boundary conditions.

4.2 Time-Dependent Boundary Condition

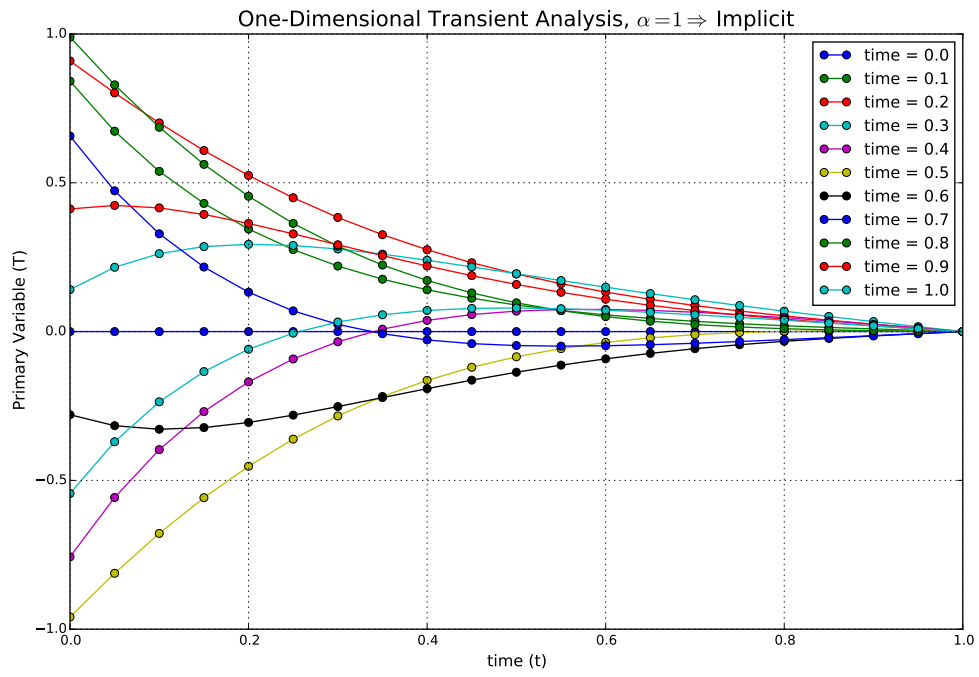Results for the problem *4.ii* are shown below in Figure 9.

Figure 9: Numerical solution to transient problem with time-dependent boundary conditions.