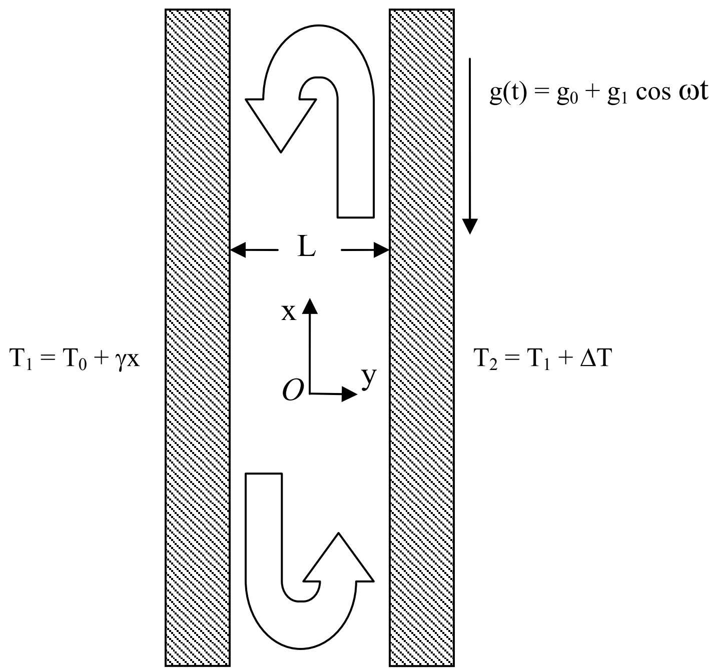
Convection in a Differentially Heated Narrow Slot

By Teja Muppirala
Advisor: Dr. Cho Lik Chan

University of Arizona, Spring/Summer 2002

Convection in a differentially heated narrow slot of fluid can display many different types of behavior depending on the properties of the fluid, and the properties of the slot. The basic principle is that fluid near the hot wall will rise due to density changes, and that fluid near the cool wall will fall. In a finite slot, as the heated fluid accumulates at the top, the accumulated pressure forces it to turn and come back down the cool side. This sets up a circulation as shown:



The temperature may be constant, or linearly increasing with x. The gravity as well may be constant or sinusoidal. The origin of the y coordinate is taken to be the center of the slot, and the x dimension is infinitely long. This will be viewed as 2-dimensional, so no property varies in z.

Determining the convection is both a heat transfer and a fluid mechanics problem, and as such we will need both the equations for fluid motion and the equation for heat transfer. The assumptions we are using are

1. Boussinesq Fluid (density variation only affects the body force term)
2. Viscosity, thermal conductivity, and heat capacity are constant
3. Density varies with temperature as $\rho = \rho_0[1 - \beta_0(T - T_{ref})]$

The governing equations then simplify to

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

$$\rho_0\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}\right) = -\frac{\partial p}{\partial x} + \rho_0\beta_0 g_x(T - T_{ref}) + \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$$

$$\rho_0\left(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}\right) = -\frac{\partial p}{\partial y} + \rho_0\beta_0 g_y(T - T_{ref}) + \mu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right)$$

$$\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = \kappa\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right)$$

The first equation is simply conservation of mass. As noted, since this is a Boussinesq fluid, there is no density dependence. The middle two equations are simplified Navier-Stokes equations, one for each dimension. The last is the heat equation.

For generality and further simplification, we non-dimensionalize these equations using the length scale of the slot, the temperature difference, and the fluid viscosity.

The dimensionless equations are

$$\frac{\partial u^*}{\partial x^*}+\frac{\partial v^*}{\partial y^*}=0$$

$$\frac{\partial u^*}{\partial t}+u^*\frac{\partial u^*}{\partial x^*}+v^*\frac{\partial u^*}{\partial y^*}=-\frac{\partial p^*}{\partial x^*}+\mathrm{Gr}_1 T^*+\frac{\partial^2 u^*}{\partial x^{*2}}+\frac{\partial^2 u^*}{\partial y^{*2}}$$

$$\frac{\partial v^*}{\partial t}+u^*\frac{\partial v^*}{\partial x^*}+v^*\frac{\partial v^*}{\partial y^*}=-\frac{\partial p^*}{\partial y^*}+\mathrm{Gr}_2 T^*+\frac{\partial^2 v^*}{\partial x^{*2}}+\frac{\partial^2 v^*}{\partial y^{*2}}$$

$$\frac{\partial T^*}{\partial t}+u^*\frac{\partial T^*}{\partial x^*}+v^*\frac{\partial T^*}{\partial y^*}=\frac{1}{\mathrm{Pr}}\left(\frac{\partial^2 T^*}{\partial x^{*2}}+\frac{\partial^2 T^*}{\partial y^{*2}}\right)$$

$$x^*=\frac{x}{L},\quad y^*=\frac{y}{L}$$

$$u^*=\frac{u}{\upsilon / L},\quad v^*=\frac{v}{\upsilon / L}$$

$$T^*=\frac{T-T_{ref}}{\Delta T},\quad p^*=\frac{p-p_0}{\rho(\upsilon / L)^2}$$

T_{ref} is taken as the average temperature of the two walls, which exists at the origin. All the equations used hereafter will be dimensionless, and the star will be omitted. Gr and Pr are the Grashof and Pradtl numbers, defined as:

$$\mathrm{Gr}_1=\frac{g_x\beta_0 L^3\Delta T}{\upsilon^2}\quad \mathrm{Gr}_2=\frac{g_y\beta_0 L^3\Delta T}{\upsilon^2}\quad \mathrm{Pr}=\frac{\upsilon}{\kappa}$$

We also define the Raleigh number as the product of the Grashof and Prandtl numbers

$$\mathrm{Ra}=\mathrm{GrPr}=\frac{g\beta_0 L^3\Delta T}{\upsilon\kappa}$$

The final goal of this research is to understand the flow and stability of the case in which γ and g_1 are non-zero. However, another goal is to provide a compilation of all the other cases, both steady state, and transient, so their results are easily accessible. Also, as one might guess, introducing nonzero γ and g_1 in the problem to such a degree that it is best to start out with the basic case and successively build on it.

With all the definitions and equations laid out, the first case we shall examine is the steady state solution of $\gamma = g_1 = 0$, constant gravity and constant temperature walls. For a finite slot, if the height to width is reasonably large, we can approximate the flow in the middle regions as approximately that as if the slot was infinitely long. In that case, we begin by assuming a parallel flow solution, so the only flow is in the x direction. Because the slot is infinitely long, and the wall temperature is constant, we may assume the temperature is only a function of y.

$$\vec{V} = \vec{V}(x, y) = (u, v, w) = (u, 0, 0)$$

$$\frac{\partial u}{\partial x} + \cancel{\frac{\partial v}{\partial y}} = 0$$

$$\cancel{\frac{\partial u}{\partial t}} + u\cancel{\frac{\partial u}{\partial x}} + v\cancel{\frac{\partial u}{\partial y}} = -\frac{\partial p}{\partial x} + \mathrm{Gr}_1 T + \cancel{\frac{\partial^2 u}{\partial x^2}} + \frac{\partial^2 u}{\partial y^2}$$

$$\cancel{\frac{\partial v}{\partial t}} + u\cancel{\frac{\partial v}{\partial x}} + v\cancel{\frac{\partial v}{\partial y}} = -\frac{\partial p}{\partial y} + \cancel{\mathrm{Gr}_2 T} + \cancel{\frac{\partial^2 v}{\partial x^2}} + \cancel{\frac{\partial^2 v}{\partial y^2}}$$

$$\cancel{\frac{\partial T}{\partial t}} + u\cancel{\frac{\partial T}{\partial x}} + \cancel{v\frac{\partial T}{\partial y}} = \frac{1}{\mathrm{Pr}}\left(\cancel{\frac{\partial^2 T}{\partial x^2}} + \frac{\partial^2 T}{\partial y^2}\right)$$

Our four equations have considerably simplified down to

$$\frac{\partial u}{\partial x} = 0 \rightarrow u \neq u(x)$$

$$0 = -\frac{\partial p}{\partial x} + \mathrm{Gr}_1 T + \frac{d^2 u}{dy^2} \quad (1)$$

$$0 = -\frac{\partial p}{\partial y} \rightarrow p \neq p(y)$$

$$0 = \frac{d^2 T}{dy^2} \quad (2)$$

We can immediately solve (2) for T. With our slot geometry and variables defined as they are, the boundary conditions required are that

$$T\left(-\tfrac{1}{2}\right) = -\tfrac{1}{2}$$
$$T\left(\tfrac{1}{2}\right) = \tfrac{1}{2}$$

therefore

$$T = y$$

Replacing back into (5)

$$0 = -\frac{dp}{dx} + \mathrm{Gr}_1 y + \frac{d^2 u}{dy^2}$$

We know that u is only a function of y, so therefore the dp/dx term can only be a constant. Treating it as such, we integrate up this equation to give us the expression for u.

$$u = \int\left(\int \frac{dp}{dx} - \mathrm{Gr}_1 y\, dy\right) dy$$

After integrating twice, we are left with a cubic polynomial expression for the velocity profile, with 3 values to determine.

$$u = -\mathrm{Gr}_1 \frac{y^3}{6} + \frac{dp}{dx}\frac{y^2}{2} + Ay + B \qquad (3)$$

The no slip condition at the wall fixes our boundary conditions so that the velocity is zero at either end.

$$u\left(\tfrac{1}{2}\right) = -\mathrm{Gr}_1 \frac{\left(\tfrac{1}{2}\right)^3}{6} + \frac{dp}{dx}\frac{\left(\tfrac{1}{2}\right)^2}{2} + A\left(\tfrac{1}{2}\right) + B = 0$$

$$u\left(-\tfrac{1}{2}\right) = -\mathrm{Gr}_1 \frac{\left(-\tfrac{1}{2}\right)^3}{6} + \frac{dp}{dx}\frac{\left(-\tfrac{1}{2}\right)^2}{2} + A\left(-\tfrac{1}{2}\right) + B = 0$$

$$0 = \frac{1}{4}\frac{dp}{dx} + 2B \qquad B = -\frac{1}{8}\frac{dp}{dx}$$

$$0 = \frac{-\mathrm{Gr}_1}{24} + A \qquad A = \frac{\mathrm{Gr}_1}{24}$$

$$u = -\mathrm{Gr}_1 \frac{y^3}{6} + \frac{dp}{dx}\frac{y^2}{2} + \frac{\mathrm{Gr}_1}{24} y - \frac{1}{8}\frac{dp}{dx} \qquad (4)$$

To determine the pressure term, we integrate (4) along the y direction. Mass must be conserved, and so this integral should be zero.

$$\int_{-1/2}^{1/2} u \,\mathrm{dy} = \int_{-1/2}^{1/2} -\mathrm{Gr}_1 \frac{y^3}{6} + \frac{dp}{dx}\frac{y^2}{2} + \frac{\mathrm{Gr}_1}{24} y - \frac{1}{8}\frac{dp}{dx}\,\mathrm{dy}$$

$$= \frac{dp}{dx}\left[\frac{y^3}{6} - \frac{y}{8}\right]_{-1/2}^{1/2} = \frac{dp}{dx}\left(\frac{-1}{12}\right) = 0$$

$$\frac{dp}{dx} = 0 \qquad (5)$$

$$\boxed{\begin{aligned} u &= -\mathrm{Gr}_1 \frac{y^3}{6} + \frac{\mathrm{Gr}_1}{24} y \\ T &= y \end{aligned}}$$

The result that the pressure gradient is zero is a direct result of having chosen our temperature reference to be the midpoint temperature. The rising and falling of the fluid then is attributed to buoyancy forces created by temperature changes, and no pressure gradient is necessary to drive the fluid. Another characteristic of this result is that u is an odd function. By the symmetry of the problem, this is what one would expect. There is no reason for the velocity to favor one side over the other. Also the Grashof number is the only parameter that affects the flow. The thermodynamic properties do not matter.

However, we know that in the laboratory they do matter. Although the walls in this case were said to be constant temperature, the buildup of hot fluid at the top and cold fluid at the bottom creates a slight gradient in temperature along the walls. We will solve this case, and then graphically compare this cubic profile with the more complicated profiles we will develop.

Again we assume a parallel flow, the only difference is that now the temperature in no longer just a function of y. It also varies linearly in x.

$$\vec{V} = \vec{V}(x, y) = (u, v, w) = (u, 0, 0)$$

$$T(x, y) = \Theta(y) + \gamma x$$

$$\frac{\partial u}{\partial x} + \cancel{\frac{\partial v}{\partial y}} = 0$$

$$\cancel{\frac{\partial u}{\partial t}} + u\cancel{\frac{\partial u}{\partial x}} + v\cancel{\frac{\partial u}{\partial y}} = -\frac{\partial p}{\partial x} + \mathrm{Gr}_1 T + \cancel{\frac{\partial^2 u}{\partial x^2}} + \frac{\partial^2 u}{\partial y^2}$$

$$\cancel{\frac{\partial v}{\partial t}} + u\cancel{\frac{\partial v}{\partial x}} + v\cancel{\frac{\partial v}{\partial y}} = -\frac{\partial p}{\partial y} + \cancel{\mathrm{Gr}_2 T} + \cancel{\frac{\partial^2 v}{\partial x^2}} + \cancel{\frac{\partial^2 v}{\partial y^2}}$$

$$\cancel{\frac{\partial T}{\partial t}} + u\frac{\partial T}{\partial x} + v\cancel{\frac{\partial T}{\partial y}} = \frac{1}{\mathrm{Pr}}\left(\cancel{\frac{\partial^2 T}{\partial x^2}} + \frac{\partial^2 T}{\partial y^2}\right)$$

Reducing to

$$0 = -\frac{dp}{dx} + \mathrm{Gr}_1\Theta + \mathrm{Gr}_1\gamma x + \frac{d^2 u}{dy^2} \qquad (7)$$

$$\gamma u = \frac{1}{\mathrm{Pr}}\frac{d^2\Theta}{dy^2} \qquad (8)$$

Then replacing (8) in (7)

$$0 = -\frac{dp}{dx} + \mathrm{Gr}_1\Theta + \gamma \mathrm{Gr}_1 x + \frac{1}{\gamma\ \mathrm{Pr}}\frac{d^4\Theta}{dy^4}$$

$$\gamma\ \mathrm{Pr}\frac{dp}{dx} - \gamma^2 Ra\ x = \gamma Ra\ \Theta + \frac{d^4\Theta}{dy^4} \qquad (9)$$

Θ is only a function of y, therefore the right hand side of this equation must be a constant. In fact we can take it to be zero, by the knowledge Θ is odd, we know that that constant must be zero. The pressure distribution can then be found as

$$\frac{dp}{dx} = \gamma \mathrm{Gr}_1 x \quad p = \frac{\gamma \mathrm{Gr}_1}{2}x^2 + p_0 \qquad (10)$$

We can immediately write down the solutions for Θ and u, of course taking the real part

$$\Theta = \mathrm{Re}\left[A\sin\alpha y + B\cos\alpha y + C\sinh\alpha y + D\cosh\alpha y\right]$$

$$\mathrm{u} = \frac{1}{\gamma \mathrm{Pr}}\mathrm{Re}\left[-\alpha^2 A\sin\alpha y - \alpha^2 B\cos\alpha y + \alpha^2 C\sinh\alpha y + \alpha^2 D\cosh\alpha y\right]$$

$$\alpha = \sqrt[4]{-\gamma Ra}$$

By requiring that the functions be odd, the cos and cosh terms drop out. The only remaining work is to find the constants. Using the initial conditions

$$\Theta\left(\tfrac{1}{2}\right) = \tfrac{1}{2} = A\sin\alpha\left(\tfrac{1}{2}\right) + C\sinh\alpha\left(\tfrac{1}{2}\right)$$

$$\mathrm{u}\left(\tfrac{1}{2}\right) = 0 = -\alpha^2 A\sin\alpha\left(\tfrac{1}{2}\right) + \alpha^2 C\sinh\alpha\left(\tfrac{1}{2}\right)$$

$$\frac{1}{2}\alpha^2 = 2\alpha^2 C\sinh\left(\frac{\alpha}{2}\right) \Rightarrow C = \frac{1}{4\sinh\left(\frac{\alpha}{2}\right)}$$

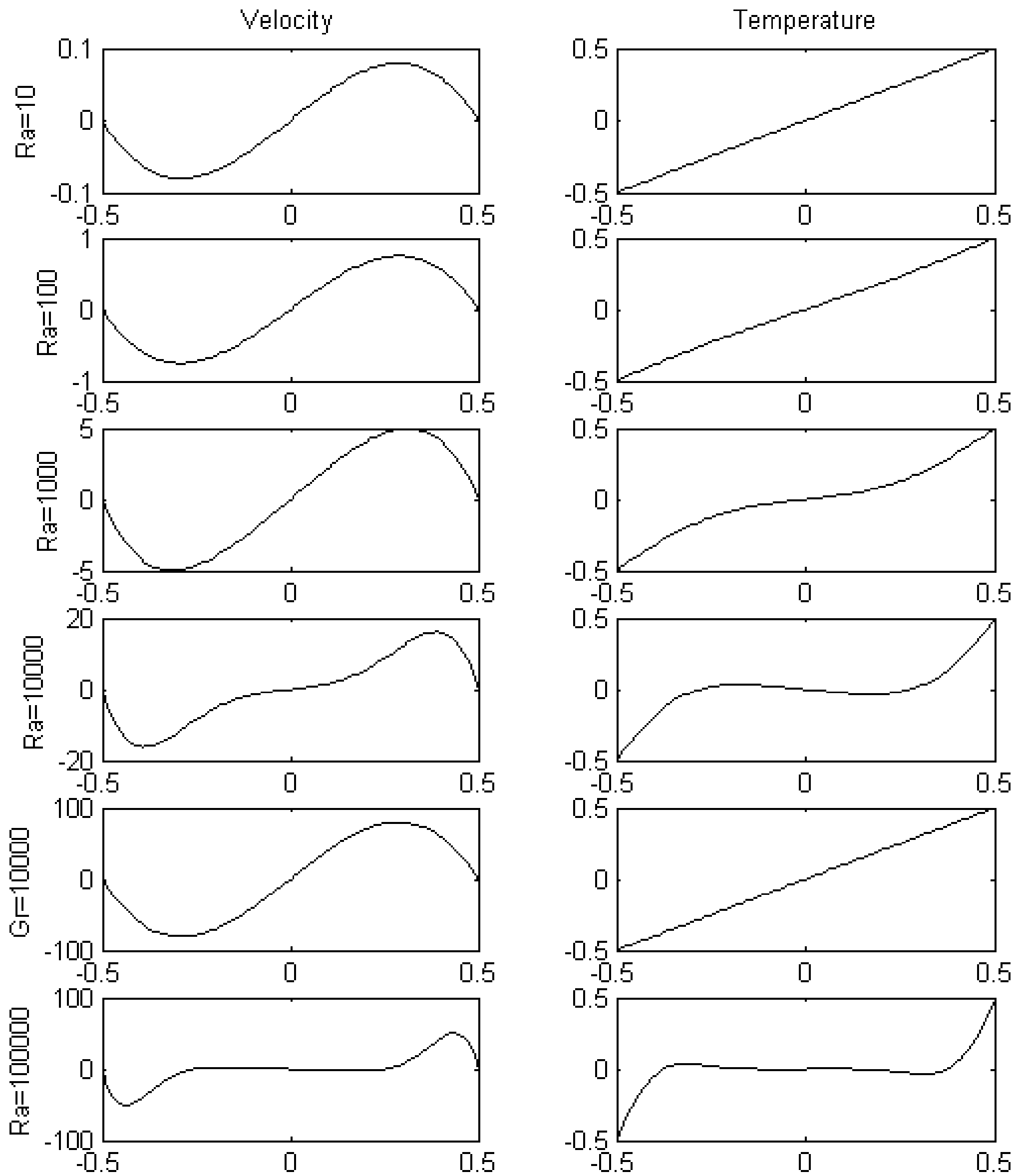
$$-\frac{1}{2}\alpha^2 = -2\alpha^2 A\sin\left(\frac{\alpha}{2}\right) \Rightarrow A = \frac{1}{4\sin\left(\frac{\alpha}{2}\right)}$$

$$T = \frac{1}{4}\mathrm{Re}\left[\frac{\sin\alpha y}{\sin\left(\frac{\alpha}{2}\right)} + \frac{\sinh\alpha y}{\sinh\left(\frac{\alpha}{2}\right)}\right] + \gamma x$$

$$\mathrm{u} = \frac{\alpha^2}{4\gamma\ \mathrm{Pr}}\mathrm{Re}\left[-\frac{\sin\alpha y}{\sin\left(\frac{\alpha}{2}\right)} + \frac{\sinh\alpha y}{\sinh\left(\frac{\alpha}{2}\right)}\right]$$

This solution appears to be quite different than that of the first case. But it can be confirmed that in the limit as γ (and thus also α) goes to zero, it reduces to the same equations. We can that the shapes of both T and u depend only on α. If α is kept constant, only the magnitude of the velocity flow is changed by the Prandtl number and γ. Keeping Pr and γ constant = 1, we look at the flow patterns for various α, dependent only on Ra.

The first thing to notice is that for the smallest Rayleigh number of 10, it obeys the very nearly the cubic velocity profile, and linear temperature distribution. As the Rayleigh number is increased, the temperature distribution begins to slowly invert itself. As it reaches 10000, the velocity and temperature distribution tend to level out near the middle of the flow, and as the Rayleigh number increases further, this effect is taken to the extreme, and it appears to be nearly zero as seen in the last set of graphs. But in fact, the profiles are oscillatory (with very small amplitude) in this region. The bulk of the flow and temperature variation are confined to the regions immediately near the wall. Also the magnitude of these flows is much less than the case with constant temperature walls. The fifth row of graphs shows this for Gr = Ra / Pr = 10000. The maximum velocity is about 5 times higher than the case of increasing wall temperature.


Velocity
Temperature
Ra=10
Ra=100
Ra=1000
Ra=10000
Gr=10000
Ra=100000
0.1
0
-0.1
1
-1
5
-5
20
-20
100
-100
0.5
-0.5

We return now to the case of $\gamma = 0$, but now we introduce gravity oscillations so that $g = g_0 + g_1 \cos \omega t$.

$$0 = \frac{d^2 T}{dy^2}$$

$$0 = \text{Gr(t)} \;\; x + \frac{\partial^2 u}{\partial y^2}$$

$$0 = \text{Gr}_0\, y + \text{Gr}_1\, y \cos \omega t + \frac{\partial^2 u}{\partial y^2}$$

The temperature equation in unchanged, but now the Grashof number is a function of time, and so momentum equation has an added sinusoidal term. We have immediately dropped the pressure term as it can be taken as zero.

Though not derived here, the solution in the form given by W.Y. Chen & C.F. Chen (1999) is

$$u(y,t) = \frac{\text{Gr}_0}{6}\left(\frac{y}{4} - y^3\right) + \frac{\text{Gr}_1}{\omega}\left\{ y \sin\omega t - \frac{N_1(y)\sin\omega t + N_2(y)\cos\omega t}{2\sinh^2\lambda\cos^2\lambda + 2\sin^2\lambda\cosh^2\lambda}\right\}$$

$$N_1(y) = (\cos\lambda\sinh\lambda)\cos 2\lambda y \sinh 2\lambda y + (\sin\lambda\cosh\lambda)\sin 2\lambda y\cosh 2\lambda y$$

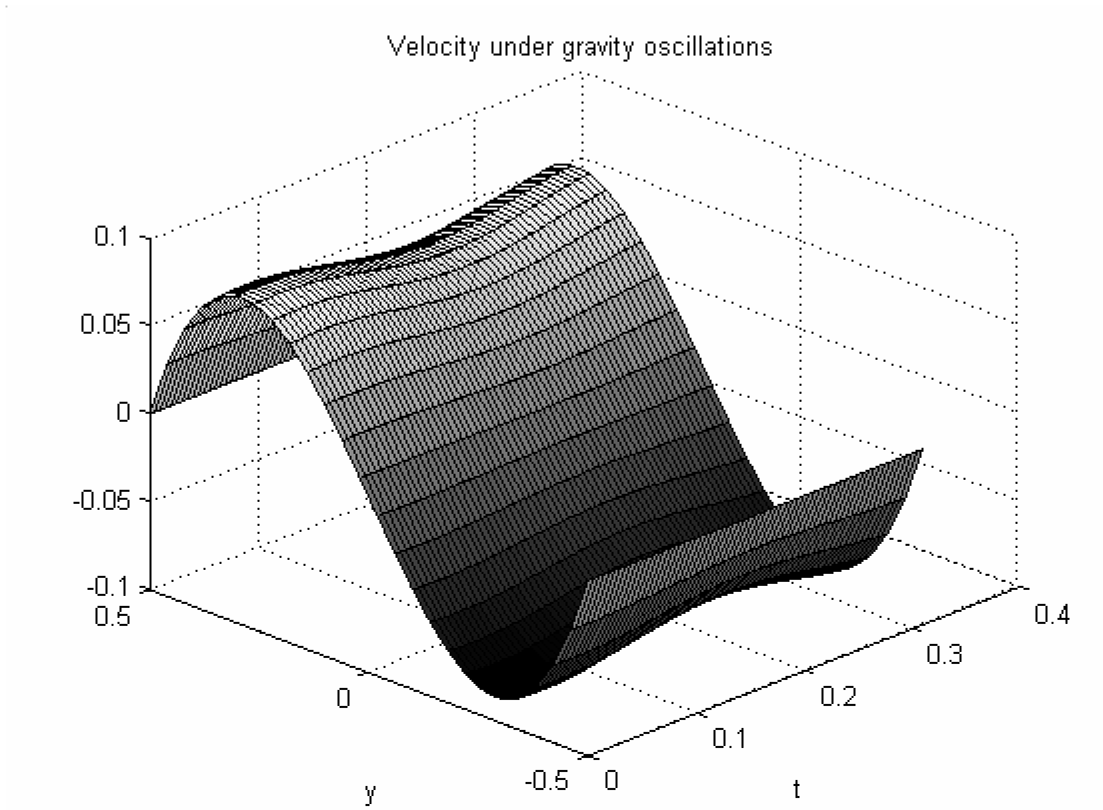
$$N_2(y) = (\cos\lambda\sinh\lambda)\sin 2\lambda y \cosh 2\lambda y - (\sin\lambda\cosh\lambda)\cos 2\lambda y\sinh 2\lambda y$$

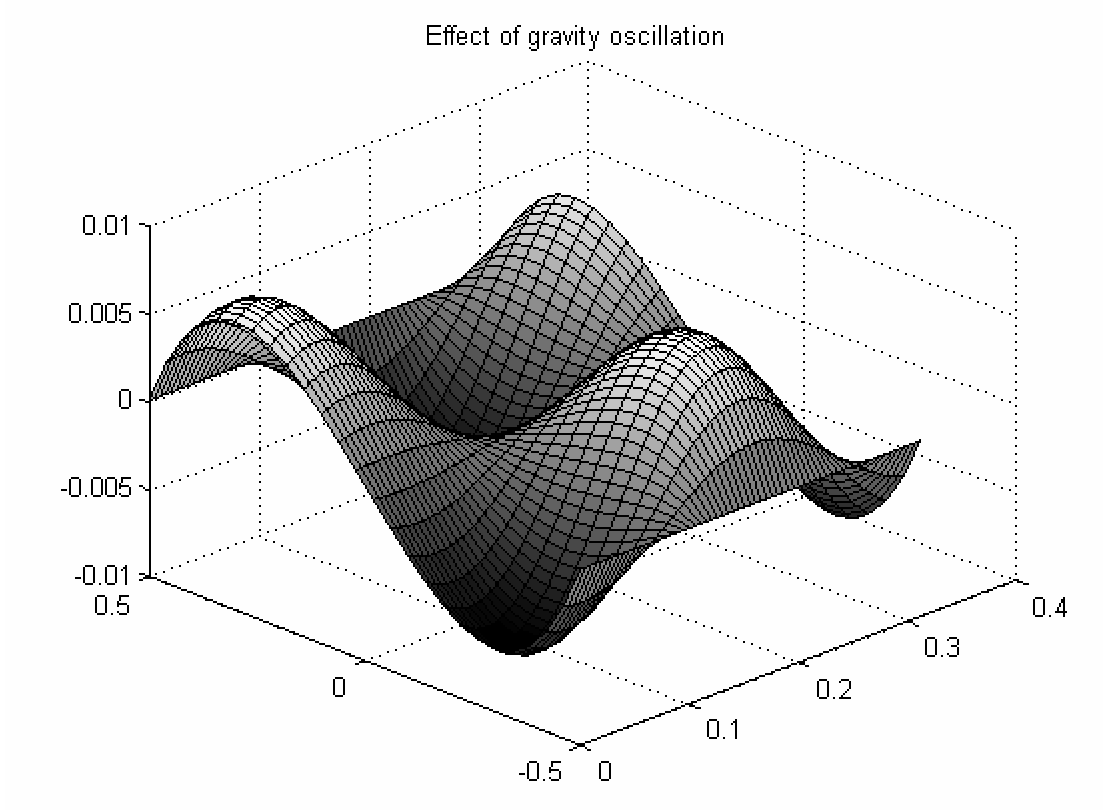
$$\lambda = \left(\frac{\omega}{8}\right)^{1/2}$$

$$T = y$$

For the case of $\text{Gr}_1 = 0$, we see the steady state we found earlier. We choose $\omega = 20$, $\text{Gr}_0 = 10$, and $\text{Gr}_1 = 1$. Then in addition to plotting u, we also plot the time-dependent part separately to see the relative effect this has.

The undisturbed velocity has a maximum of about 8.0e-2, while the magnitude of the oscillation is about 7.0e-3, showing that the effect of gravity modulation is not in the same proportion as the Grashof numbers. As the oscillatory Grashof number approaches the constant one in magnitude, and as the frequency of modulation becomes very small, the natural convection patterns become unstable. We will return to the stability of these flow later.


Velocity under gravity oscillations
0.1
0.05
0
-0.05
-0.1
0.5
0
-0.5
y
0
0.1
0.2
0.3
0.4
t


Effect of gravity oscillation
0.01
0.005
0
-0.005
-0.01
0.5
0
-0.5
0
0.1
0.2
0.3
0.4

The logical next step might be to introduce nonzero γ to the previous system. But before we do that, we will examine these flows under transient behavior. With the temperature equal to the mean temperature of the walls, and no velocity flows, we model how these evolve into the steady state solutions. Again we begin with the simplest case.

$$\vec{V} = \vec{V}(x,y) = (u,v,w) = (u,0,0)$$
$$T(x,y) = S_T(y) + \Theta(y,t)$$
$$u(x,y) = S_u(y) + U(y,t)$$
$$\frac{\partial U}{\partial t} = \mathrm{Gr}_1\Theta(y,t) + \frac{\partial^2 U}{\partial y^2}$$
$$\frac{\partial \Theta}{\partial t} = \frac{1}{\mathrm{Pr}}\frac{\partial^2 \Theta}{\partial y^2}$$

We already know the steady state solutions, so we ignore them for the time being, and just write our equations for the transient part. The temperature is solved then with separation of variables. Then replacing back and solving for U will be done using eigenfunction expansion.

The solution for theta is obvious, but to make the algebra a little easier, we will leave the parameters as letters.

$$\Theta = \sum_{n=1}^{\infty} A_n e^{-\frac{n^2\pi^2}{\mathrm{Pr}\,L^2}t} \sin\frac{n\pi y}{L} = \sum_{n=1}^{\infty} A_n(t)\sin\frac{n\pi y}{L}$$
$$\Theta(0,y) + S_T(y) = 0 \rightarrow \Theta(0,y) = -y$$
$$\Theta(0,y) = \sum_{n=1}^{\infty} A_n \sin\frac{n\pi y}{L} = -y \rightarrow A_n = \frac{2}{L}\int_0^L -y\sin\frac{n\pi y}{L}dy$$
$$U(x,t) = \sum_{n=1}^{\infty} B_n(t)\sin\frac{n\pi y}{L}$$
$$\sum_{n=1}^{\infty} B_n'(t)\sin\frac{n\pi y}{L} = \sum_{n=1}^{\infty} \mathrm{Gr}_1 A_n(t)\sin\frac{n\pi y}{L} + \sum_{n=1}^{\infty} -\frac{n^2\pi^2}{L^2}B_n(t)\sin\frac{n\pi y}{L}$$
$$B_n'(t) + \frac{n^2\pi^2}{L^2}B_n(t) = \mathrm{Gr}_1 A_n(t) = \mathrm{Gr}_1 A_n e^{-\frac{n^2\pi^2}{\mathrm{Pr}\,L^2}t}$$
$$B_n(t) = C_n e^{-\frac{n^2\pi^2}{L^2}t} + \frac{\mathrm{Ra}\ A_n L^2}{n^2\pi^2(\mathrm{Pr}-1)}e^{-\frac{n^2\pi^2}{\mathrm{Pr}\,L^2}t}$$
$$U(y,t) = \sum_{n=1}^{\infty}\left(C_n e^{-\frac{n^2\pi^2}{L^2}t} + \frac{\mathrm{Ra}\ A_n L^2}{n^2\pi^2(\mathrm{Pr}-1)}e^{-\frac{n^2\pi^2}{\mathrm{Pr}\,L^2}t}\right)\sin\frac{n\pi y}{L}$$
$$U(y,0) = 0 = \sum_{n=1}^{\infty}\left(C_n + \frac{\mathrm{Ra}\ A_n L^2}{n^2\pi^2(\mathrm{Pr}-1)}\right)\sin\frac{n\pi y}{L} = -S_u = \frac{\mathrm{Gr}_1}{6}\left(y^3 - \frac{y}{4}\right)$$
$$C_n = -\frac{\mathrm{Ra}\ A_n L^2}{n^2\pi^2(\mathrm{Pr}-1)} + \frac{2}{L}\int_0^L \frac{\mathrm{Gr}_1}{6}\left(y^3 - \frac{y}{4}\right)\sin\frac{n\pi y}{L}dy$$

No plot for this. But we see that the transient solution just exponentially “decays” into the steady state. Although it is notable that for the velocity, there are actually two rates of decay. One is dependent on the Prandtl number like for the temperature, and the other is not.

Solving all the other cases involves proceeding in the same way, so let's go ahead and jump to the most difficult problem, of both gravity oscillation and non constant walls, and see what we can do.

$$T(x,y,t) = S_T(x,y) + \Theta(y,t)$$
$$u(y,t) = S_u(y) + U(y,t)$$
$$\mathrm{Gr} = \mathrm{Gr}_0 + \mathrm{Gr}_1 \cos \omega t$$
$$\frac{\partial \Theta}{\partial t} + \gamma U = \frac{1}{\Pr}\frac{\partial^2 \Theta}{\partial y^2} \rightarrow U = \frac{1}{\gamma}\left(\frac{1}{\Pr}\frac{\partial^2 \Theta}{\partial y^2} - \frac{\partial \Theta}{\partial t}\right)$$
$$\frac{\partial U}{\partial t} = \mathrm{Gr}\,\Theta(y,t) + \frac{\partial^2 U}{\partial y^2}$$
$$\frac{1}{\gamma}\left(\frac{1}{\Pr}\frac{\partial^3 \Theta}{\partial t \partial y^2} - \frac{\partial^2 \Theta}{\partial t^2}\right) = \mathrm{Gr(t)}\ \Theta(y,t) + \frac{1}{\gamma}\left(\frac{1}{\Pr}\frac{\partial^4 \Theta}{\partial y^4} - \frac{\partial^3 \Theta}{\partial y^2 \partial t}\right)$$

There doesn't appear to be much we can do with these mixed partial derivatives. We can sine transform in y to get rid of the y derivatives.

$$\frac{1}{\gamma}\left(-n^2\frac{1}{\Pr}\frac{d\hat{\Theta}}{dt} - \frac{d^2\hat{\Theta}}{dt^2}\right) = \mathrm{Gr(t)}\ \hat{\Theta} + \frac{1}{\gamma}\left(\frac{n^4}{\Pr}\hat{\Theta} + n^2\frac{d\hat{\Theta}}{dt}\right)$$
$$-n^2\hat{\Theta}' - \Pr\hat{\Theta}'' = \gamma \mathrm{Ra}(t)\hat{\Theta} + n^4\hat{\Theta} - \Pr n^2\hat{\Theta}'$$
$$\Pr\hat{\Theta}'' + (1-\Pr)n^2\hat{\Theta}' + \left(\gamma \mathrm{Ra}(t) + n^4\right)\hat{\Theta} = 0$$

And so we are left with Mathieu's equation, an ordinary differential equation with periodic coefficients. There are numerical techniques to work from here, but they will not be pursued. We will use another method entirely.

We will try to use the method of Chebyshev polynomials as described by Sinha and Wu (1991). Chebyshev polynomials are an orthogonal set of polynomials which have unique properties in their integration and multiplication against each other. They are particularly well suited to solving systems with periodic forcing.

Define the Chebyshev polynomials as

$$T_0(t) = 1$$
$$T_1(t) = t$$
$$T_{n+1} = 2tT_n(t) - T_{n-1}(t)$$

Then define the shifted Chebyshev polynomials as then defined as T'(t)=T(2t-1). These obey the orthogonality relationship

$$\int_0^7 T'_r(t')T'_k(t')w(t')dt' = \begin{matrix} 0, r \neq k \\ \pi/2, r = k \neq 0 \\ \pi, r = k = 0 \end{matrix}$$
$$w(t') = (t' - t'^2)^{-1/2}$$

They also satisfy the following relationships

$$\int_0^{t'} T_r'(s)ds = \frac{1}{4}\left(\frac{T_{r+1}'(t')}{r+1} - \frac{T_{r-1}'(t')}{r-1}\right) - \frac{(-1)^r}{2(r^2-1)}, r = 0, 2, 3, 4, \ldots$$

$$\int_0^{t'} T_1'(s)ds = \frac{1}{8}\left(T_2'(t') - T_0'(t')\right)$$

$$0 \le t' \le 1$$

If you define a row vector $\{T'(t')\} = \{T_0'(t'), \quad T_1'(t'), \ldots, \quad T_{m-1}'(t')\}$, then

$\int_0^{t'} \{T'(s)\}ds = \{T'(t')\}\mathbf{G}^T$, where **G** is a constant matrix defined be the above integration relations to be:

$$\mathbf{G} = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 & \ldots & \ldots & 0 \\ -1/8 & 0 & 1/8 & 0 & \ldots & \ldots & 0 \\ -1/6 & -1/4 & 0 & 1/12 & \ldots & \ldots & 0 \\ 1/16 & 0 & -1/8 & 0 & \ddots & \ldots & \vdots \\ \vdots & \vdots & \ldots & \ddots & \ddots & 1/4(m-2) & 0 \\ 0 & 0 & \ldots & \ldots & -1/4(m-3) & 0 & 1/4(m-1) \\ \frac{-(-1)^{m-1}}{2m(m-2)} & 0 & \ldots & \ldots & 0 & -1/4(m-2) & 0 \end{bmatrix}$$

There is also a multiplication formula that states

$$T_r'(t')T_k'(t') = \frac{1}{2}\left(T_{r+k}'(t') + T_{|r-k|}'(t')\right)$$

so for two functions

$$F(t) = \{\mathbf{f}\}\{T'(t')\}^T = \{f_0 \quad f_1 \quad \ldots \quad f_{m-1}\}\begin{Bmatrix} T'_0(t') \\ T'_1(t') \\ \vdots \\ T'_{m-1}(t') \end{Bmatrix}$$

$$G(t) = \{\mathbf{g}\}\{T'(t')\}^T$$

The product of F and G is given by $F(t')G(t') = \{\mathbf{f}\}\{T'(t')\}^T\{T'(t')\}\{\mathbf{g}\}^T = \{T'(t')\}\mathbf{Q}\{\mathbf{g}\}^T$ with **Q** as a matrix consisting of the coefficients of **g**:

$$\mathbf{Q} = \begin{bmatrix} g_0 & g_1/2 & g_2/2 & \ldots & \ldots & g_{m-1}/2 \\ g_1 & g_0 + \frac{g_2}{2} & \frac{1}{2}(g_1+g_3) & \frac{1}{2}(g_2+g_4) & \ldots & \frac{1}{2}(g_{m-2}+g_m) \\ g_2 & \frac{1}{2}(g_1+g_3) & g_0 + \frac{g_4}{2} & \frac{1}{2}(g_1+g_5) & \ldots & \frac{1}{2}(g_{m-3}+g_{m+1}) \\ \vdots & \frac{1}{2}(g_2+g_4) & \frac{1}{2}(g_1+g_5) & \ldots & \ldots & \vdots \\ \vdots & \vdots & \ldots & \ldots & \ldots & \frac{1}{2}(g_1+g_{2m-3}) \\ g_{m-1} & \frac{1}{2}(g_{m-2}+g_m) & \frac{1}{2}(g_{m-3}+g_{m+1}) & \ldots & \frac{1}{2}(g_1+g_{2m-3}) & g_0 + \frac{g_{2m-2}}{2} \end{bmatrix}$$

With this information, we can solve the first order linear ODE in the following way:

$$y' = f(t)y + g(t)$$

$$y = y_0 + \int_0^t f(t)y \ dt + \int_0^t g(t) \ dt$$

$$\{T'(t')\}\{\mathbf{y}\}^T = \{T'(t')\}\begin{Bmatrix} y_0 \\ 0 \\ \vdots \\ 0 \end{Bmatrix} + \int_0^t \{T'(t)\}\mathbf{Q}_f\{\mathbf{y}\}^T \ dt + \int_0^t \{T'(t)\}\{\mathbf{g}\}^T \ dt$$

$$\{T'(t')\}\{\mathbf{y}\}^T = \{T'(t')\}\begin{Bmatrix} y_0 \\ 0 \\ \vdots \\ 0 \end{Bmatrix} + \{T'(t')\}\mathbf{G}^T\mathbf{Q}_f\{\mathbf{y}\}^T + \{T'(t')\}\mathbf{G}^T\{\mathbf{g}\}^T$$

Now it is very simple to solve for the coefficients of y as:

$$\{\mathbf{y}\}^T = (\mathbf{I} - \mathbf{G}^T\mathbf{Q}_f)^{-1}\left(\begin{Bmatrix} y_0 \\ 0 \\ \vdots \\ 0 \end{Bmatrix} + \mathbf{G}^T\{\mathbf{g}\}^T\right)$$

provided the solution exists.

Notice that the formula for the Q matrix involves coefficients that run well past m-1, which is the highest term in our approximation of y. Numerically, we will for now simply omit these terms as zero.

The coefficients of g are obtained simply by the orthogonality relationship.

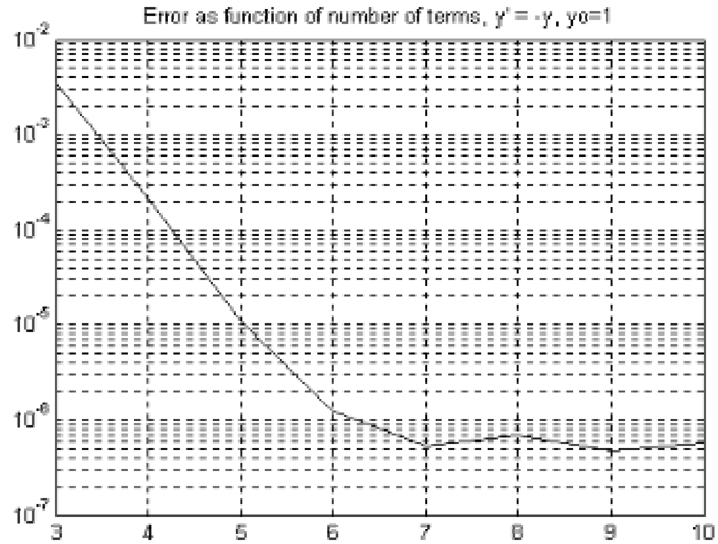
$$g_0 = \frac{1}{\pi}\int_0^1 g(t)T_0'(t)w'(t) \ dt$$

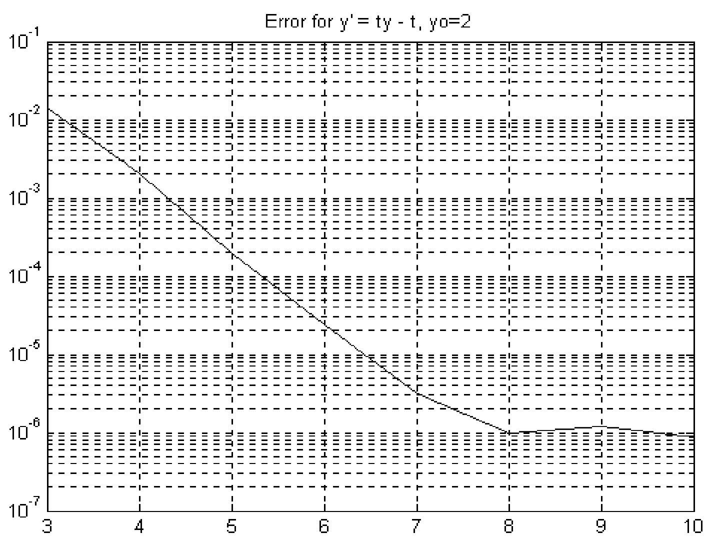
$$g_{m\neq 0} = \frac{1}{2\pi}\int_0^1 g(t)T_m'(t)w'(t) \ dt$$

To illustrate how this works, we show the results of two examples on the next page.

The error decreases geometrically with the number of terms, in the same way it does for example with a spectral method. In calculating the f and g coefficients, the error tolerance for the integrations was set to 10^{-6} and as such, this is where the number of terms stops making a difference.

There are some instances in which this method fails to converge to the right answer, for example, with f(t) = 5 cos 5t, g(t)=5 sin (5t). Although f(t) = cos(5t), g(t) = sin(5t) works fine. As of yet I have been unable to find a specific criteria that will determine true convergence or not. But the most likely reason is that for certain f(t)'s, the omission of higher order terms in the Q matrix in fact cannot be safely done. This however seems to be the exception rather than the norm, and for now, we will continue to use this method as is until it seems this is really the problem and needs correcting.





Before we can use this numerical method to solve our PDE's, we must reduce it to a system of ODE's. We can accomplish this using Galerkin methods. We modify our dimensions slightly, to the following

$$T(x, y, t) = y + \Theta(y, t) + \gamma x$$

$$\Theta(-1) = -1, \Theta(1) = 1$$

$$u(-1, t) = u(1, t) = 0$$

$$\mathrm{Gr} = \mathrm{Gr}(t)$$

$$\frac{\partial U}{\partial t} = \mathrm{Gr}\,\Theta + \mathrm{Gr}\ y + \frac{\partial^2 U}{\partial y^2}$$

$$\frac{\partial \Theta}{\partial t} + \gamma U = \frac{1}{\mathrm{Pr}} \frac{\partial^2 \Theta}{\partial y^2}$$

so now we are working over the interval [-1,1].

After several unsuccessful trial functions, we finally came to choose

$$u = \sum_{i=0}^{M-1} A_i(t)(1-y^2)(T_i(y))$$
$$\Theta = \sum_{i=0}^{M-1} B_i(t)(1-y^2)(T_i(y))$$

The factor $1-y^2$ is added to ensure zero at the boundary, and these T's are unshifted Chebyshev polynomials. Now, we replace into the equations, and proceed.

$$\sum_{i=0}^{M-1} A_i'(t)(1-y^2)(T_i(y)) = \mathrm{Gr}\sum_{i=0}^{M-1} B_i(t)(1-y^2)(T_i(y)) + Gr\cdot y + \frac{\partial^2}{\partial y^2}\left(\sum_{i=0}^{M-1} A_i(t)(1-y^2)(T_i(y))\right)$$
$$\sum_{i=0}^{M-1} B_i'(t)(1-y^2)(T_i(y)) + \gamma\sum_{i=0}^{M-1} A_i(t)(1-y^2)(T_i(y)) = \frac{1}{\mathrm{Pr}}\frac{\partial^2}{\partial y^2}\left(\sum_{i=0}^{M-1} B_i(t)(1-y^2)(T_i(y))\right)$$

But now instead of multiplying by our same trial function and integrating, we multiply by T(y).

$$\int_0^1 \sum_{j=0}^{M-1} T_j(y)\times\left[\sum_{i=0}^{M-1} A_i'(t)(1-y^2)(T_i(y)) = \mathrm{Gr}\sum_{i=0}^{M-1} B_i(t)(1-y^2)(T_i(y)) + Gr\cdot y + \frac{\partial^2}{\partial y^2}\left(\sum_{i=0}^{M-1} A_i(t)(1-y^2)(T_i(y))\right)\right] w(y)\ dy$$
$$\int_0^1 \sum_{j=0}^{M-1} T_j(y)\times\left[\sum_{i=0}^{M-1} B_i'(t)(1-y^2)(T_i(y)) = -\gamma\sum_{i=0}^{M-1} A_i(t)(1-y^2)(T_i(y)) \frac{1}{\mathrm{Pr}}\frac{\partial^2}{\partial y^2}\left(\sum_{i=0}^{M-1} B_i(t)(1-y^2)(T_i(y))\right)\right] w(y)\ dy$$

This requires that we have some expression for the first and second derivatives of the Chebyshev polynomials. The first derivative can be given as

$$T_n' = \frac{-nxT_n + nT_{n-1}}{(1-x^2)}$$

and reiterating yields

$$T_n'' = \frac{1}{(1-x^2)^2}\left[T_n((n^2-n)x^2-n) + T_{n-1}((3n-2n^2)x) + T_{n-2}(n^2-n)\right]$$

These formulas can be easily implemented using MATLAB's polynomial multiplication and division. From the Galerkin method we get a system of 2 x M equations.

$$\begin{bmatrix} A_0' \\ A_1' \\ \vdots \\ A_m' \end{bmatrix} = [\mathbf{L}]^{-1}[\mathbf{R}]\begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_m \end{bmatrix} + Gr\begin{bmatrix} B_0 \\ B_1 \\ \vdots \\ B_m \end{bmatrix} + Gr[\mathbf{L}]^{-1}\begin{bmatrix} 0 \\ 1/2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} B_0' \\ B_1' \\ \vdots \\ B_m' \end{bmatrix} = -\gamma\begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_m \end{bmatrix} + \frac{1}{\mathrm{Pr}}[\mathbf{L}]^{-1}[\mathbf{R}]\begin{bmatrix} B_0 \\ B_1 \\ \vdots \\ B_m \end{bmatrix}$$

L and R are the matrices derived from integration:

$$[\mathbf{L}]=\begin{bmatrix} 1/2 & 0 & -1/4 & 0 & \ldots & 0 & 0 \\ 0 & 1/8 & 0 & -1/8 & 0 & \vdots & 0 \\ -1/4 & 0 & 1/4 & 0 & -1/8 & \vdots & 0 \\ 0 & -1/8 & 0 & 1/4 & 0 & \ddots & \vdots \\ \vdots & 0 & -1/8 & 0 & 1/4 & \ddots & -1/8 \\ 0 & \ldots & \ldots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \ldots & -1/8 & 0 & 1/4 \end{bmatrix} \quad [\mathbf{R}]=\begin{bmatrix} -2 & 0 & -6 & 0 & -12 & \ldots & -3(M-1) \\ 0 & -3 & 0 & -9 & 0 & \ldots & 0 \\ 0 & 0 & -6 & 0 & -12 & \ldots & -3(M-1) \\ 0 & 0 & 0 & -10 & 0 & \ldots & \vdots \\ 0 & 0 & 0 & 0 & -15 & \ddots & -3(M-1) \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -M(M+1)/2 \end{bmatrix}$$

We can write all this as one single vector equation of 2M elements.

$$\begin{bmatrix} A_0' \\ \vdots \\ A_m' \\ B_0' \\ \vdots \\ B_m' \end{bmatrix} = \begin{bmatrix} [\mathbf{L}]^{-1}[\mathbf{R}] & Gr\mathbf{I} \\ -\gamma\mathbf{I} & \frac{1}{\Pr}[\mathbf{L}]^{-1}[\mathbf{R}] \end{bmatrix} \begin{bmatrix} A_0 \\ \vdots \\ A_m \\ B_0 \\ \vdots \\ B_m \end{bmatrix} + \begin{bmatrix} [\mathbf{L}]^{-1} & \mathbf{0} \\ \mathbf{0} & [\mathbf{L}]^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ \frac{1}{2}Gr \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Now this is in a form that can be used by the Chebyshev expansion method outlined earlier. Each coefficient $A_0(t)$, $A_1(t)\ldots B_0(t), B_1(t)\ldots$ is itself expanded into shifted Chebyshev polynomials as a function of time. We must first extend the method to higher dimensions. Using m terms to expand each of n equations, it works as follows:

$$\int_0^t \begin{pmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_n(t) \end{pmatrix}' dt = \int_0^t \begin{bmatrix} F_{11}(t) & F_{12}(t) & .. & F_{1n}(t) \\ F_{21}(t) & \ldots & \ldots & \ldots \\ \vdots & \ldots & \ldots & \ldots \\ F_{n1}(t) & \ldots & \ldots & F_{nn}(t) \end{bmatrix} \begin{pmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_n(t) \end{pmatrix} + \begin{pmatrix} G_1(t) \\ G_2(t) \\ \vdots \\ G_n(t) \end{pmatrix} dt$$

$$\{T'_0 \ \ldots \ T'_{m-1} \ \ldots \ \ldots \ \ldots \ T'_0 \ \ldots \ T'_{m-1}\} \begin{pmatrix} \mathbf{y_1} \\ \mathbf{y_2} \\ \vdots \\ \mathbf{y_n} \end{pmatrix} = \{T'_0 \ \cdots \ T'_{m-1}\} \begin{pmatrix} y_1(0) \\ 0 \\ \vdots \\ y_2(0) \\ \vdots \\ \vdots \end{pmatrix} +$$

$$\{T'_0 \ \cdots \ T'_{m-1}\} \begin{bmatrix} \mathbf{G}^T\mathbf{Q}_{F11} & \mathbf{G}^T\mathbf{Q}_{F12} & \ldots & \mathbf{G}^T\mathbf{Q}_{F1n} \\ \mathbf{G}^T\mathbf{Q}_{F21} & \ldots & \ldots & \ldots \\ \vdots & \ldots & \ldots & \ldots \\ \mathbf{G}^T\mathbf{Q}_{Fn1} & \ldots & \ldots & \mathbf{G}^T\mathbf{Q}_{Fnn} \end{bmatrix} + \{T'_0 \ \cdots \ T'_{m-1}\} \begin{bmatrix} \mathbf{G}^T & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^T & \ldots & \ldots \\ \vdots & \ldots & \ddots & \mathbf{0} \\ \mathbf{0} & \ldots & \mathbf{0} & \mathbf{G}^T \end{bmatrix} \begin{pmatrix} \mathbf{g_1} \\ \mathbf{g_2} \\ \vdots \\ \mathbf{g_n} \end{pmatrix}$$

Every **G** and **Q** is of size m x m, and the **g**'s and **y**'s are column vectors of size m x 1, so we now have a linear algebra problem of size n x m.

$$\begin{pmatrix} \mathbf{y_1} \\ \mathbf{y_2} \\ \vdots \\ \mathbf{y_n} \end{pmatrix} = \begin{pmatrix} y_{1,1} \\ \vdots \\ y_{1,m} \\ y_{2,1} \\ \vdots \\ y_{2,m} \\ \vdots \\ \vdots \\ y_{n,m} \end{pmatrix} = \left\{ \mathbf{I}_{nm} - \begin{bmatrix} \mathbf{G}^T & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^T & \dots & \dots \\ \vdots & \dots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{G}^T \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{F11} & \mathbf{Q}_{F12} & \dots & \mathbf{Q}_{F1n} \\ \mathbf{Q}_{F21} & \dots & \dots & \dots \\ \vdots & \dots & \dots & \dots \\ \mathbf{Q}_{Fn1} & \dots & \dots & \mathbf{Q}_{Fnn} \end{bmatrix} \right\}^{-1} \left\{ \begin{pmatrix} y_1(0) \\ 0 \\ \vdots \\ y_2(0) \\ \vdots \\ \vdots \end{pmatrix} + \begin{bmatrix} \mathbf{G}^T & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^T & \dots & \dots \\ \vdots & \dots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{G}^T \end{bmatrix} \begin{pmatrix} \mathbf{g_1} \\ \mathbf{g_2} \\ \vdots \\ \mathbf{g_n} \end{pmatrix} \right\}$$

$$\begin{pmatrix} \mathbf{y_1} \\ \mathbf{y_2} \\ \vdots \\ \mathbf{y_n} \end{pmatrix} = \left\{ \mathbf{I}_{nm} - \left[Big\mathbf{G}^T \right] \left[Big\mathbf{Q} \right] \right\}^{-1} \left\{ \begin{pmatrix} y_1(0) \\ 0 \\ \vdots \\ y_2(0) \\ \vdots \\ \vdots \end{pmatrix} + \left[Big\mathbf{G}^T \right] \begin{pmatrix} \mathbf{g_1} \\ \mathbf{g_2} \\ \vdots \\ \mathbf{g_n} \end{pmatrix} \right\}$$

Fortunately, for our equations Big**Q** will be relatively simple to calculate. Since their elements are constant, the $[\mathbf{L}]^{-1}[\mathbf{R}]$'s and $\gamma\mathbf{I}$'s expand their **Q** simply as **I** times each element. The upper right hand side of Big**Q** matrix will also just contain a diagonal sequence of Gr(t)'s **Q**. We also have all the **g**'s except for $\mathbf{g}_2$ equal to zero. $\mathbf{g}_2$ is again obtained by using the orthogononality relationships and then multiplying the resulting vector by $[\mathbf{L}]^{-1}$. And of course, our initial conditions are whatever we want them to be, zero working well.

The only drawback is that our shifted Chebyshev polynomials will only work from a time range of zero to one. This is unacceptable, and if we want any meaningful results from our calculations we have to make this work for as long as we need.

The most obvious way to do it is to run it once, and use the final result at t=1as your initial condition and then run it again, and repeat as many times as necessary. This method is very easy to do, since at t=1, all the shifted Chebyshev polynomials are equal to one, and your new initial conditions end up being exactly what you got for the coefficients in the first place. A better idea though, is use the result at t=1/2 as your initial condition, since the error generally grows as t nears 1. At the midway point, half of the polynomials are zero, the other half are either +1 or -1, and again the new initial conditions are very easy to determine.

In fact, this doesn't work. The problem is, that unlike many numerical methods, the initial conditions you put in are not in fact the initial conditions that you get from the algorithm's solution. The Chebyshev expansion method not only approximates the solution over the time interval, it also approximates the initial condition. For example, using Y' = -Y, with Yo = 1. The approximate solution will not start at exactly 1.0, but will be off slightly by an amount that diminishes as the number of terms increase. This is hardly noticeable and not such a big deal for single equation systems, but for a large system, this error can be quite crippling when trying to append each solution smoothly one after the other.

Fortunately, there is one very simple way to do the job. Instead of approximating as $y = \sum T'(t)$, make the simple adjustment to $y = \sum T'\left(\frac{t}{d}\right)$. Now this is valid for all time from zero to d. A couple things change now.

The orthogonality relationships change:

$$g_0 = \frac{1}{d\pi}\int_0^d g(t)T_0'(t/d)w'(t/d)\ dt$$

$$g_{m\neq 0} = \frac{1}{2d\pi}\int_0^d g(t)T_m'(t/d)w'(t/d)\ dt$$

Similarly, since $\int_0^d T_a'(t/d)\cdot T_b'(t/d)\ dt = d\int_0^d T_a'(t)\cdot T_b'(t)\ dt$

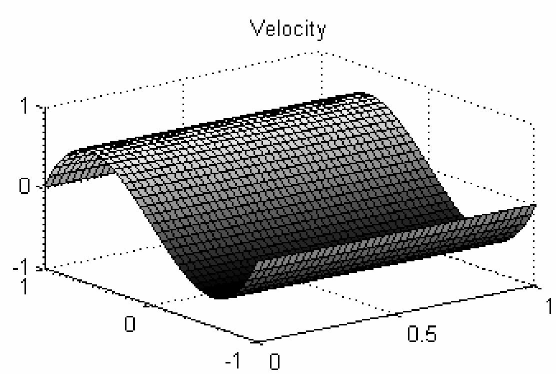
the **G** integration matrix will be increased by a factor of d.
The product matrix **Q** will remain unchanged.

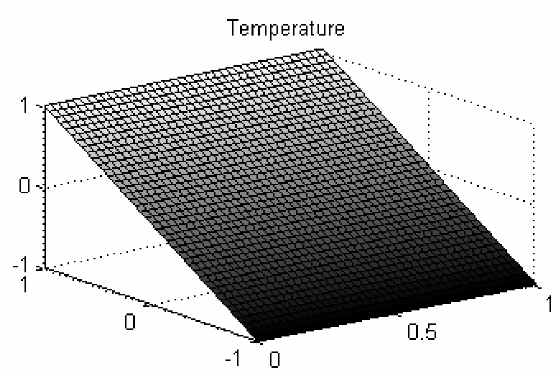
Of course, as d gets larger, more terms will be needed to maintain accuracy over the entire interval.

This algorithm was written up using MATLAB, and the results were very good compared to the analytic results obtained earlier.

For the cases where there is constant gravity, and you expect some solution that eventually becomes constant in time, the best approach is to prime the algorithm several steps by using the repeated initial condition method described earlier. Each pass will bring it closer to the final solution, and will minimize the error over the plotted distance. But for time-dependent gravity, this method introduces error at both the initial and final regions of the plot, and thus it is best to simply start with zero initial conditions and do a single pass.

The results of several calculations are shown.



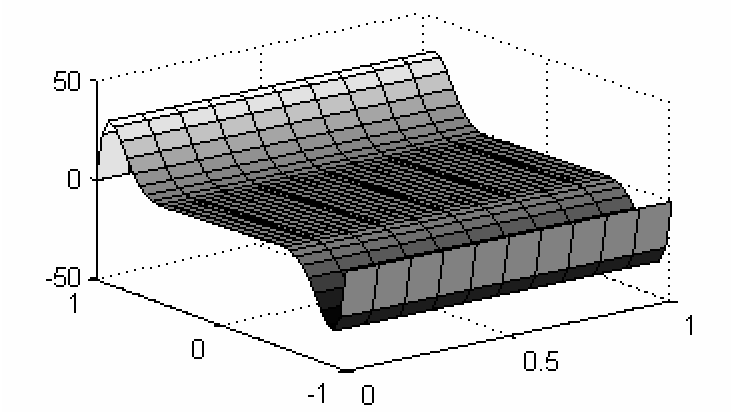


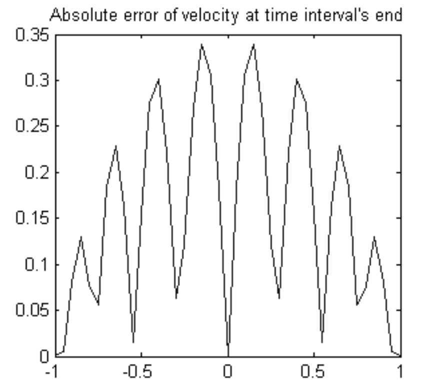
Gr = 10
Gamma = 0

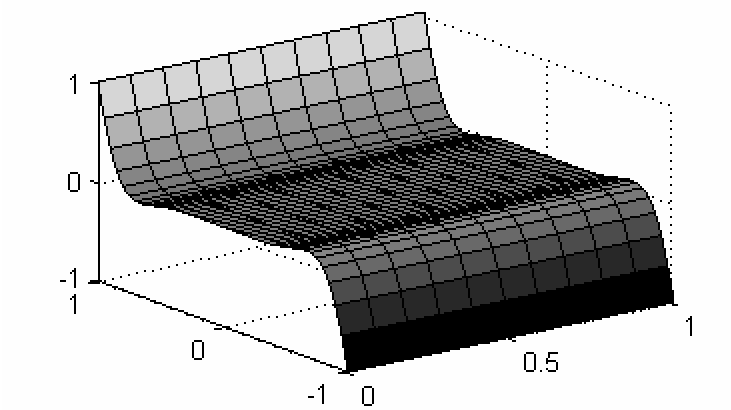
Space Terms = 3
Time Terms = 5
Grid size = 41 x 41
Primed = 5 half steps
Maximum Relative Error = 5.15e-3
Approximate Time = 5 min. (at 400MHz)

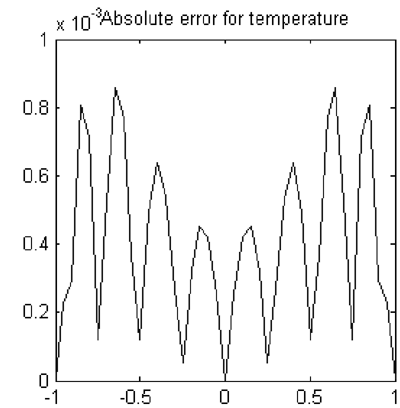
Gr = 10000
Gamma = 1

Terms = 3 time, 11 space
Grid = 11 x 41
Primed: 5 half steps
Time = 4 min.

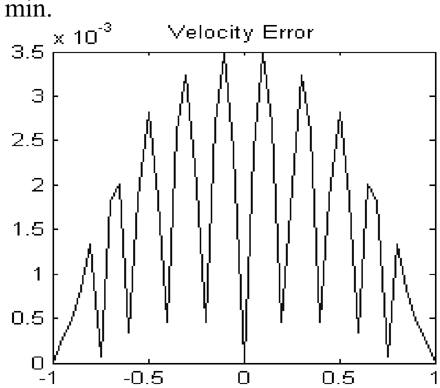


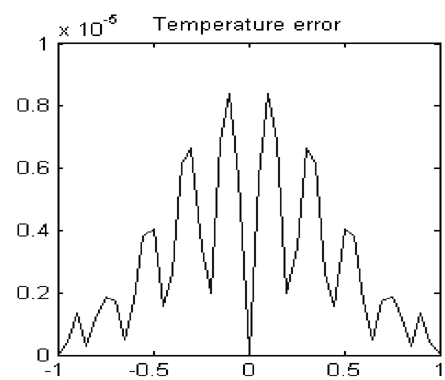


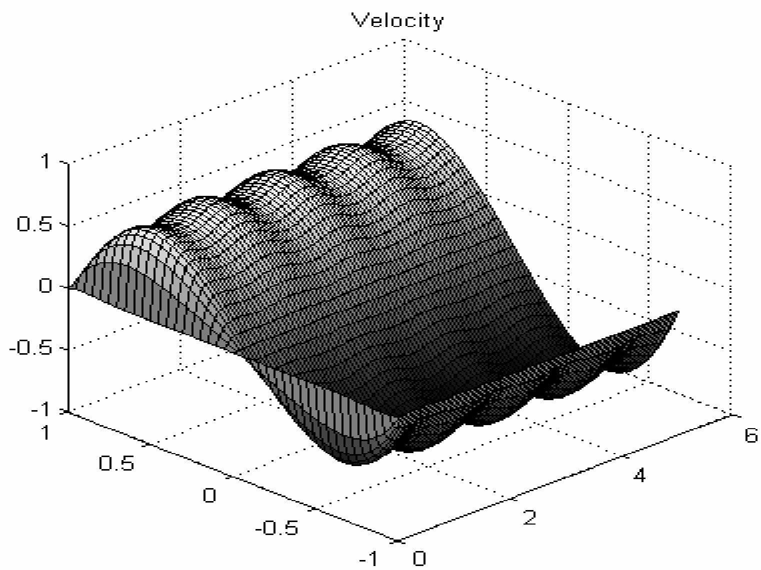




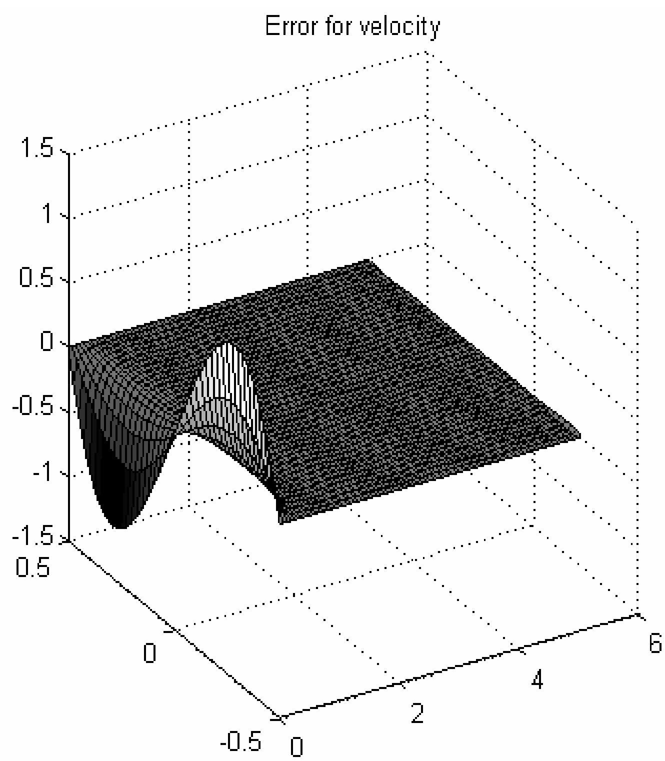
Terms = 3 time, 15 space
Primed: 7 half steps
Time = 6 min.







Gr = 10 + cos 5t
Terms = 21 time 21 space
Grid = 41 x 81
Not primed
Distance = 8π/5
Time = ~6 hours



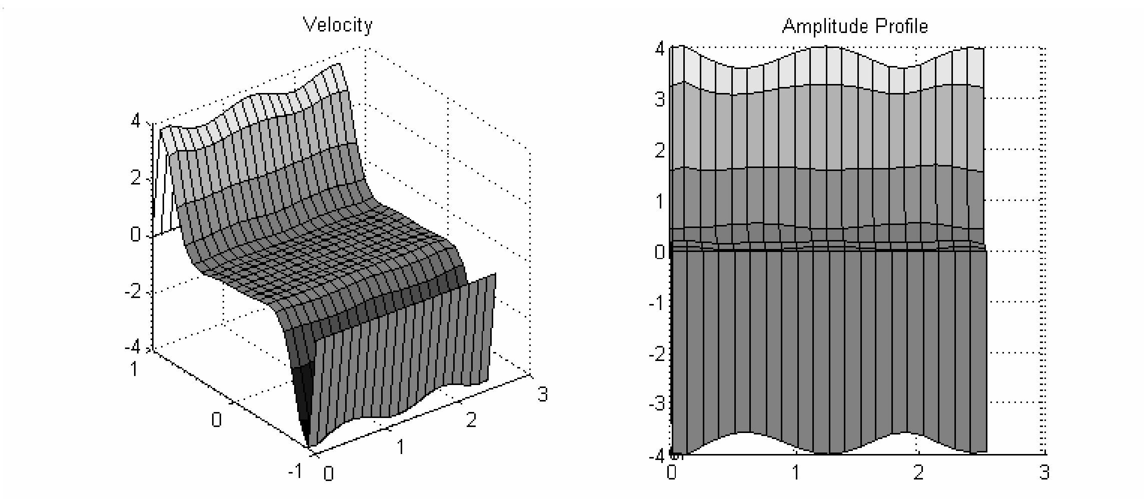
Error:
At 21 x 21 terms, when comparing to the analytical solution over a time period after the initial transient behavior, the maximum relative error is < 1%.

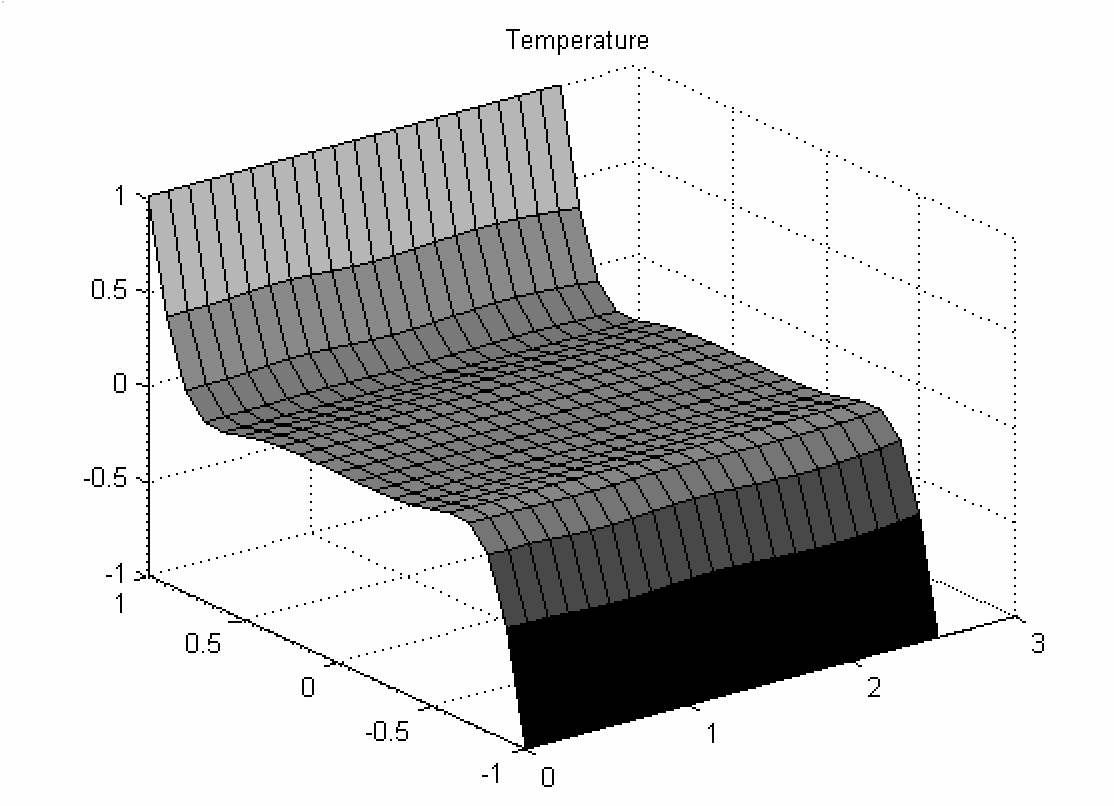
When repeating this with 13 time x 13 space terms, the maximum relative error is < 5%, and the time elapsed is ~2.5 hours.

At this point, the program was then run with time-dependent gravity and nonzero gamma. There is no analytical solution to compare to, but we can expect that it be periodic, and have shape somewhat similar to the nonzero gamma cases we know. When running however, the algorithm produced some rather odd results. For example, even though the initial conditions were set to zero, the initial values put out were nothing close to zero. Recall, we chose to omit the higher order coefficients in the Q matrix. It turns out that this cannot be done here. And even though it worked before, including these higher order terms increases the rate of convergence. After going back and making this adjustment, the program yielded some more useful looking results.

Ra = 1000 + 100 cos (5t)

Gamma = 1
Pr = 7
Terms = 13 time, 19 space
Grid size = 21 x 21
Distance = 4π/5
Time = ~20 minutes





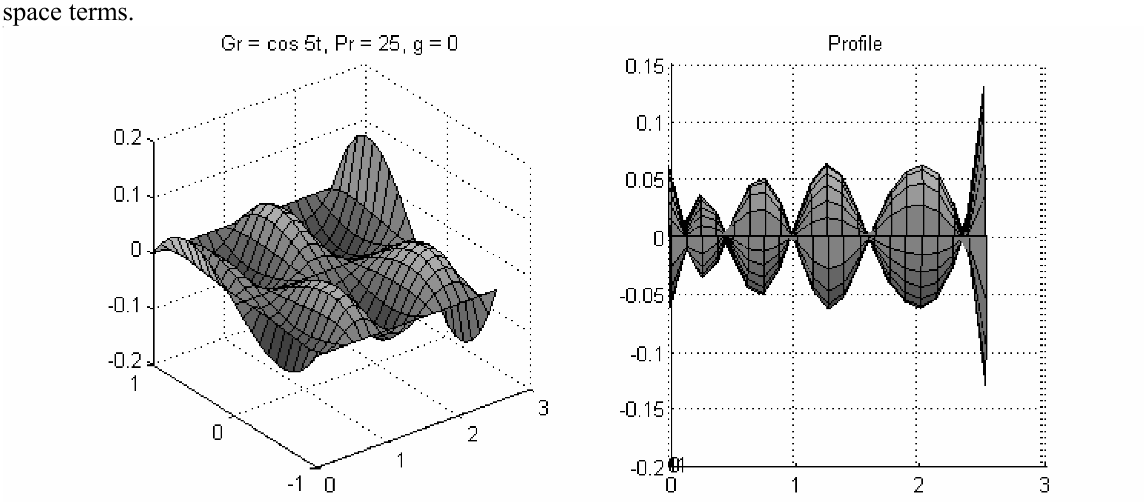
The results are similar to the case of zero gamma, where the time dependent solution seemed to be very close to the steady solution with a sinusoidal pattern in amplitude. Also similar to the steady stade, the solutions seem to be dependent on only two parameters. The amplitude is scaled by γPr, and the shape is determined by (γPr)Gr.

To give us more confidence that this method is giving us reasonable results for the solutions to these equations, we continued to run a great number of cases, under all sorts of different parameters. This data is all available in MATLAB file format.

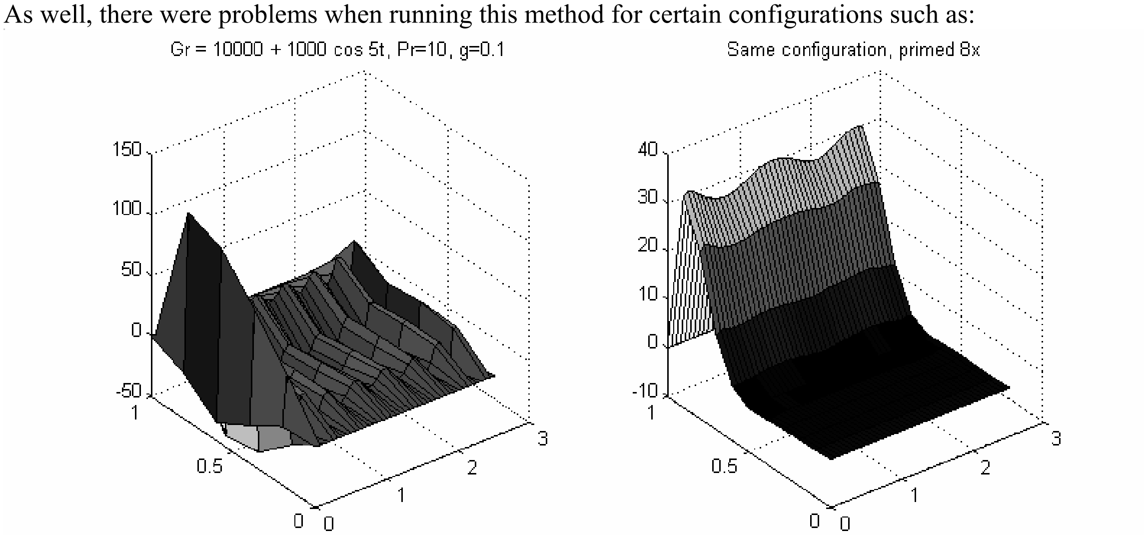
While the majority of cases looked to be accurate, there were some cases which gave pause.

When running Gr with a high relative amplitude of modulation, or with no constant term and only modulation, we received poor results. Recall $Gr = Gr_0 + Gr_1 \cos \omega t$. Define $h = Gr_1/Gr_0$. Running at zero gamma over 3 periods, no matter how much the terms were increased, the errors were no better than:
h = 0.1 → ~1% Error
h = 0.2 → ~2.3% Error
h = 0.5 → ~8.7% Error

This is really not acceptable. When h equals infinity ($Gr_0 = 0$), the results are quite poor at 13 time and 19 space terms.



As well, there were problems when running this method for certain configurations such as:



Note that only y from 0 → 1 is being plotted. Since these functions are odd in space, it occurred to me that there is no need to plot or calculate the other half.
The first graph was run with the initial conditions set to zero, the second was primed 8 times. It is not inconceivable that as appears in the first case, the solution oscillates at a higher frequency than the frequency of the forcing. But as we will show later, there should be no reason that the initial conditions would alter the final behavior.

A library of most all the test cases' MATLAB data files is included with this report.

In light of these problems, we attempted to make some minor modifications to the Chebyshev expansion algorithm, including expanding the Q matrix to include more terms, and running higher accuracy in the integration to obtain the coefficients of the Grashof expansion, however this made little difference, and we were unable to single out the source of the problems just mentioned.

After the Galerkin method step, we chose to use Chebyshev expansion to solve our resulting set of differential equations. At this point however, it seemed best if we went back and just used a simple initial value solver, such as Runge-Kutte 4, to deal with these simple nonlinear equations. So this is in fact what we did, and we did in fact get very nice results with none of the problems that we had earlier. Moreover, the IVP solver, while slightly slower in the calculation phase, since we are only dealing with one set of polynomials (in space), rather than both space and time as we were previously, the plotting time is reduced tremendously. What previously took anywhere from 15 minutes to 3 hours is now being run in about one to five minutes. And taking a look at how backwards Euler would solve this, we can see that there is no reason that the initial value would affect the long term behavior.

Recall backwards Euler as $y_n = hf(y_n, t_n) + y_{n-1}$

For our equations, this becomes

$$\vec{\mathbf{y}}_n = h\begin{bmatrix} [\mathbf{L}]^{-1}[\mathbf{R}] & Gr(t_n)\mathbf{I} \\ -\gamma\mathbf{I} & \frac{1}{\Pr}[\mathbf{L}]^{-1}[\mathbf{R}] \end{bmatrix}\vec{\mathbf{y}}_n + h\begin{bmatrix} [\mathbf{L}]^{-1} & \mathbf{0} \\ \mathbf{0} & [\mathbf{L}]^{-1} \end{bmatrix}\begin{bmatrix} 0 \\ \frac{1}{2}Gr(t_n) \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \vec{\mathbf{y}}_{n-1}$$

Solving for y_n

$$\vec{\mathbf{y}}_n = \left(\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} - h\begin{bmatrix} [\mathbf{L}]^{-1}[\mathbf{R}] & Gr(t_n)\mathbf{I} \\ -\gamma\mathbf{I} & \frac{1}{\Pr}[\mathbf{L}]^{-1}[\mathbf{R}] \end{bmatrix}\right)^{-1}\left(h\begin{bmatrix} [\mathbf{L}]^{-1} & \mathbf{0} \\ \mathbf{0} & [\mathbf{L}]^{-1} \end{bmatrix}\begin{bmatrix} 0 \\ \frac{1}{2}Gr(t_n) \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \vec{\mathbf{y}}_{n-1} \right)$$

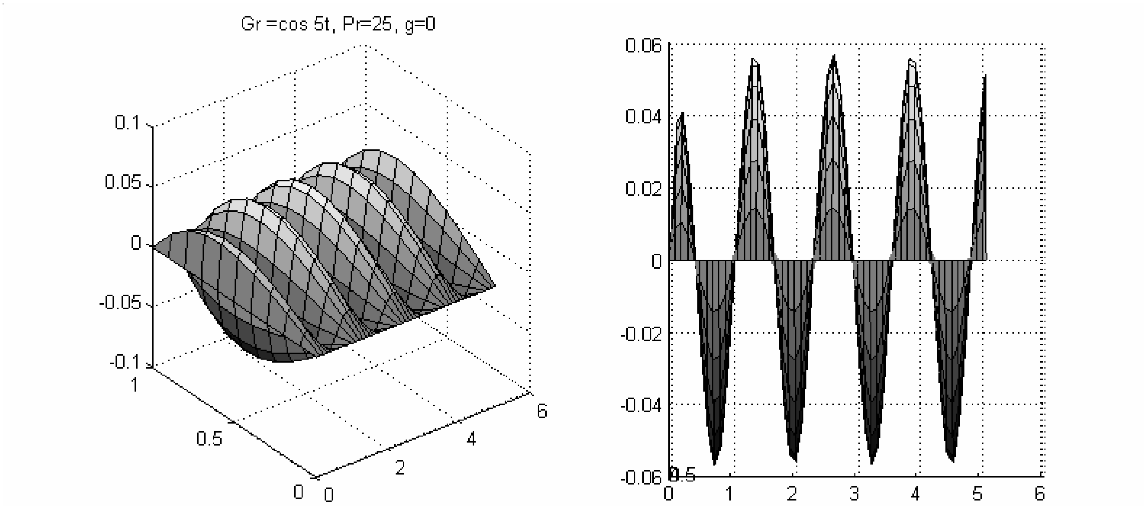
$$\vec{\mathbf{y}}_n = \left(\mathbf{I} - h\mathbf{A}_n\right)^{-1}\left(h\vec{\mathbf{e}}_n + \vec{\mathbf{y}}_{n-1}\right) = \mathbf{K}_n\left(h\vec{\mathbf{e}}_n + \vec{\mathbf{y}}_{n-1}\right)$$

The recursion is quite apparent, and the explicit solution can be obtained as

$$\vec{\mathbf{y}}_1 = \mathbf{K}_1\left(h\vec{\mathbf{e}}_1 + \vec{\mathbf{y}}_0\right)$$
$$\vec{\mathbf{y}}_2 = \mathbf{K}_2\left(h\vec{\mathbf{e}}_2 + \vec{\mathbf{y}}_1\right) = \mathbf{K}_2\left(h\vec{\mathbf{e}}_2 + \mathbf{K}_1\left(h\vec{\mathbf{e}}_1 + \vec{\mathbf{y}}_0\right)\right) = \mathbf{K}_2\mathbf{K}_1\left(h\vec{\mathbf{e}}_1 + \vec{\mathbf{y}}_0\right) + \mathbf{K}_2 h\vec{\mathbf{e}}_2$$
$$\vec{\mathbf{y}}_3 = \mathbf{K}_3\left(h\vec{\mathbf{e}}_3 + \vec{\mathbf{y}}_2\right) = \mathbf{K}_3\left(h\vec{\mathbf{e}}_3 + \mathbf{K}_2\mathbf{K}_1\left(h\vec{\mathbf{e}}_1 + \vec{\mathbf{y}}_0\right) + \mathbf{K}_2 h\vec{\mathbf{e}}_2\right) = \mathbf{K}_3\mathbf{K}_2\mathbf{K}_1\left(h\vec{\mathbf{e}}_1 + \vec{\mathbf{y}}_0\right) + \mathbf{K}_3\mathbf{K}_2 h\vec{\mathbf{e}}_2 + \mathbf{K}_3 h\vec{\mathbf{e}}_3$$
$$\vec{\mathbf{y}}_n = \left(\prod_{i=1}^{n}\mathbf{K}_i\right)\vec{\mathbf{y}}_0 + h\sum_{i=1}^{n}\left(\prod_{j=i}^{n}\mathbf{K}_j\right)\vec{\mathbf{e}}_i$$

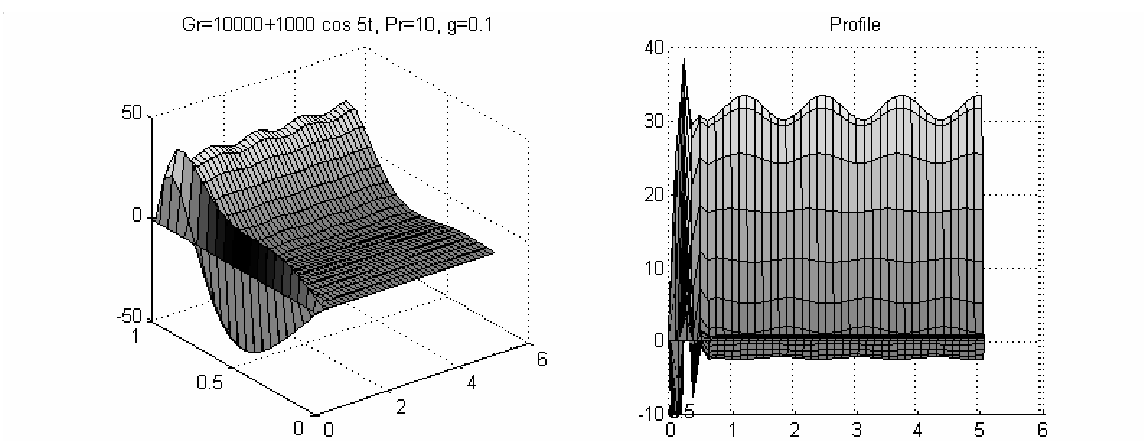
This is guaranteed to converge if the spectral radius of all the K's is less than one. In this case, as n goes to infinity, the contribution of the initial condition disappears and the behavior is dominated only by the periodic Grashof function.

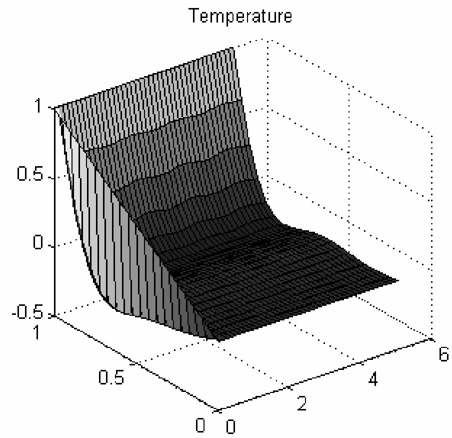
We plot again those two functions which showed problems, and see that we have better results with backwards Euler.



Starting from zero initial conditions, the function quickly reaches its periodic solution and does not stray from it. RK4 may as well have been used, but it requires a smaller step size to converge. And since these all approach a stable periodic solution, the error will remain quite minimal even with a first order solver.

Our other problematic case is shown here to have no problems at all when starting with zero initial conditions. After a very brief initial transient state in which in jumps around wildly, it falls into the periodical solution very nicely.





As well, the temperature goes from linearly distributed to its periodic solution very nicely.

The success of this plain numerical method raises one obvious question. Why didn't we do this in the first place? The thought was that when running over long time periods, the IVP solvers would require a lot of steps and would take far too long. But after having run it, it seems this is not true.

However let's not jump on the bandwagon just yet. We must be absolutely sure of our numerical scheme's accuracy and robustness before we can continue with the linear theory and stability part of this research. We put the Chebyshev method to countless tests, and we must do the same to this method before we can go on. And this is where the current progress of this research is, experimenting with all sorts of cases, comparing errors, and such.

There were a great many MATLAB files of significant use in this project, the important ones have been given here, with a brief description of each of these files below.

be_conv.m
The backwards Euler routine, Pr and g are edited inside this file, changing Gr requires the editing of both ypconv_e.m and yp_extra.m

chebgenf.m
A function which takes in an integer n and returns in a vector the coefficients for T_n.

chebmntp.m
The Chebyshev expansion algorithm. The plotting parameters are set at line 170 and 172. Gr is given in function qw2.m. All other setting are prompted by the program.

cp.m
The Chebyshev polynomial evaluator. cp(n,b) = T_n(b). Requires chebgenf.m.

genlhs.m / genlhsm.m
These generate the L and R matrices needed. They are identical, except in genlhs the matrix size variable is n, while in genlhsm, it is m. This allowed for more freedom in working with programs that already had one of those letters in use as something else.

gjit.m
The analytic solution for zero gamma, and oscillatory gravity, as calculated by W.Y Chen and C.F. Chen. All parameters are edited inside the file. This was written for y ranging -½ → ½, and so the Grashof number must be increased by a factor of 8 to yield consistent results with y = -1 → 1.

plotscp.m / plotscpd.m
These two files are scripts which use Chebyshev coefficients given in a vector b, recreates the function using scp.m, and then plots them over a distance of 1 or d.

qw2.m
The Grashof function used by chebmntp.m, multiplied by the weight function and the Chebyshev polynomial.

rk4conv.m
The RK4 routine, Pr and g are edited inside this file, changing Gr requires the editing of ypconv.m

scp.m
The shifted Chebyshev polynomial evaluator. scp(n,b) = T'_n(b). Requires chebgenf.m.

vel2.m / temp2.m
Functions called by vt_plot2_nf.m, to evaluate the velocity and temperature.

vt_plot2_nf.m
Plots the analytic solution derived in this paper for nonzero gamma, and constant Gr. All parameters are edited in the file. y = -1 → 1. Requires vel2.m and temp2.m.

ypconv.m
Is the function for the derivative used in rk4conv.m

ypconv_e.m
Generates the matrix **A** used by backwards Euler in the step of $(\mathbf{I}-h\mathbf{A})^{-1}$

ypextra.m
Generates the vector **e** used by backwards Euler

Reading Saved Data Files
The file names were saved as:
(*time terms*)T(*space terms*) Yg(*gamma*)G(G_0_ G_1_ω)_Pr(*Prandtl*)_*distance_priming*_[*L3*]_[*y grid size*]_[*y range*]_[*UT*].m

The bracketed expressions may not necessarily appear in the filename.

For example

13T19Yg0p01G1K_100_5_Pr100_2per_p8_L3_41y_0to1_UT.m

13T → **13** time terms
19Y → **19** space terms
g**0p01** → gamma = **0.01**
G**1K_100_5_** → Gr(t) = **1000** + **100** cos (**5**t)
Pr**100_** → Pr = **100**
2per_ → distance plotted was two periods = **2** * (2π) / w
p**8_** → primed **8** half-steps
L3 → indicates that only 3 time steps were plotted: at the top, bottom, and middle of the periodic cycle
41y → indicates that the y dimension was plotted with **41** points
0to1 → indicates that the y range was 0 to 1, rather that -1 to 1 as in many earlier plots
UT → indicates that this m file only contains the values that are necessary to plot the answer, and none of the intermediate variables, thereby reducing the file size significantly. This was not included in the newer plots' file names, although they too only save the useful information.

To view these files, simply open them and use the MATLAB surf command.
surf(t,y,U) for velocity or surf(t,y,T) for temperature.

Saved figures
In the early trials, figures were saved instead of data files. The filename scheme is similar to what was mentioned above, although there is usually a T or a V at the beginning or end which makes it obvious as to whether this figure is plotting temperature of velocity.

The figure zero_prime_overlapping.fig is included to demonstrate how the priming procedure works.