SCALING OF MATLAB ON HIGH PERFORMANCE COMPUTING SYSTEMS

Ms. Lindsey Blair[1], Dr. Jason Cassibry[2]

University of Alabama in Huntsville MAE Department, Huntsville, Alabama, 35899

MATLAB is a high-level language and interactive environment for numerical computation, visualization, and programming. It is an excellent tool for prototyping codes for solving engineering problems and allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages. However, code performance is limited when utilized on desktop and laptop platforms. These limitations are what I am seeking to explore. The limitations will be found by determining what the scaling (computational time, problem size) is for running test cases using a three dimensional fluid code on MATLAB as a function of platform (laptop, workstation, supercomputer). A new three dimensional code developed by Dr. Jason Cassibry and his graduate students will be utilized to run classic test cases. Resolution will be increased to study convergence, wall clock computational time, and problem size limitations on several computing platforms. Results will be compared against number of processors, RAM and similar figures of merit. In this process, the validity of the code itself could also be explored.

Nomenclature

UAH	=	University of Alabama in Huntsville
MAE	=	Mechanical and Aerospace Engineering
SPH	=	Smoothed Particle Hydrodynamics
kg	=	kilogram
m	=	meter
s	=	second
N	=	number of particles
x	=	position
v	=	velocity
ρ	=	density
T	=	temperature
ms	=	millisecond
K	=	Kelvin
$A(r)$	=	trivial identity of a particle field
r	=	position of the particles
$\boldsymbol{r}'$	=	velocity of the particles
$w(u,h)$	=	interpolating kernel
h	=	smoothed length
k	=	volume element
r_k	=	position of the center of mass of the given particle

[1] Undergraduate Student, UAH Mechanical and Aerospace Department, 5000 Technology Drive, Huntsville, AL 35899
[2] Professor, UAH Mechanical and Aerospace Department, 5000 Technology Drive, Huntsville, AL 35899

m_k	=	mass
i	=	particle
$\frac{dv_i}{dt}$	=	momentum equation
P	=	pressure
u	=	unit mass
$\frac{du_i}{dt}$	=	rate of change of the thermal energy
α, β	=	arbitrary integers
$\bar{c}_{ik}$	=	average speed sound
η	=	clipping function
$\bar{\rho}_k$	=	average density of the given particle

I. Introduction

MATLAB (**mat**rix **lab**oratory) is a numerical computing environment that is widely used in academic and research institutions, such as UAH. Dr. Jason Cassibry and his graduate students at UAH have developed a new three dimensional fluid code using the smooth particle hydrodynamic method that pushes the limitations of MATLAB on some machines. This new code has created a need to take an in-depth look at the limitations of MATLAB as they vary from system to system and have never been firmly established. In order to do this, a Square Wave Test case problem was run on a Dell Precision M6500 laptop, Dell Alienware Aurora ALX, and Dell Precision T7600 workstation to compare wall clock computational time.

II. Numerical Model

Smoothed particle hydrodynamics (SPH) is a computational method that was originally developed to deal with problems in astrophysics involving fluid masses moving arbitrarily in three dimensions in the absence of boundaries. SPH involves the motion of a set of points at which velocity and thermal energy are known at any point. These points are also assigned a mass and are therefore referred to as particles. In order for the particles to move correctly through a time step it is necessary to construct the forces which an element of fluid would experience. These forces are constructed from the information carried by the particles and begin with the approximation

$$\langle A(r)\rangle = \int A(\mathbf{r}')w(r-\mathbf{r}',h)d\mathbf{r}' \quad \text{(Equation 1)}$$

where w(u,h) is an interpolating kernel with the properties

$$\int w(u,h)d\boldsymbol{u} = \mathbf{1} \qquad \text{(Equation 2)}$$

and $w(u,h) \xrightarrow{yields} \delta(u)$. Assuming we have a fluid of density ρ(r), Equation 1 becomes

$$\langle A(r)\rangle = \int [\frac{A(\mathbf{r}')}{\rho(\mathbf{r}')}](w(r-\mathbf{r}',h)\rho(\mathbf{r}')d\mathbf{r}' \qquad \text{(Equation 3)}$$

The contribution to the integral from the volume element k then leads to an approximation given by

$$\langle A(r)\rangle = \sum_{k=1}^{N} m_k \frac{A_k}{\rho_k}(w(r-r_k,h) \qquad \text{(Equation 4)}$$

where $A_k = A(r_k)$. If the particles are equi-separated and their masses are equal then Equation 4 is a simple Riemann sum. Using Equation 4 we can approximate any field A by an analytical function $\langle A(r)\rangle$. The density estimate, which is sometimes interpreted as the smoothing of the particle's point mass by the kernel so as to obtain a continuous density field from a set of particles, is given by

$$\langle p(r)\rangle = \sum_{k=1}^{N} m_k w(r-r_k,h) \qquad \text{(Equation 5)}$$

To follow the motion of an arbitrary particle i we need an estimate of the change in pressure over density. Combining this estimate with Equation 4, we can then write the momentum equation for particle i as

$$\frac{dv_i}{dt} = -\sum_k m_k(\frac{P_k}{\rho_k{}^2} + \frac{P_i}{\rho_i{}^2})\nabla_i w_{ik} \qquad \text{(Equation 6)}$$

where ∇_i means take the gradient with respect to the coordinates of particle i and $w_{ik} = w(r_i - r_k, h)$. The rate of change of the thermal energy per unit mass u can be written in many ways to be suitable for computation. One of these ways is to use Equation 4 to estimate ρv and ρ. The rate of change of the thermal energy per unit mass u for a particle i can then be written as

$$\frac{du_i}{dt} = \frac{P_i}{\rho_i{}^2}\sum_{k=1}^{N} m_k(v_i - v_k)\cdot\nabla_i w_{ik} \qquad \text{(Equation 7)}$$

In order to estimate Equations 6 and 7, ρ must first be calculated using Equation 5. This process can be lengthy and it is sometimes useful to find ρ from matter conservation in the form

$$\frac{d\rho_i}{dt} = -(\rho\nabla \cdot v)_i = -\sum_k m_k(v_i - v_k) \cdot \nabla_i w_{ik} \qquad \text{(Equation 8)}$$

At this point, the right hand sides of Equations 6, 7, and 8 can be calculated simultaneously.

However, these equations do not correctly simulate clouds of gas interacting supersonically. This is because the particles from one cloud penetrate the other and then stream through each other. This difficulty can be removed with the introduction of an artificial viscosity. For instance, Equation 6 can be replaced by

$$\frac{dv_i}{dt} = -\sum_k m_k[\frac{P_k}{{\rho_k}^2} + \frac{P_i}{{\rho_i}^2} + \frac{1}{\bar{\rho}_k k}(-\alpha\mu_{ik}\bar{c}_{ik} + \beta{\mu_{ik}}^2)]\nabla_i w_{ik} \qquad \text{(Equation 9)}$$

where the notation $\bar{A}_{ik} = \frac{1}{2}(A_i + A_k), A_{ik} = A_i - A_k$, C is the sound speed if $v_{ik}r_{ik} < 0$

$$\mu_{ik} = hv_{ik} \cdot \frac{r_{ik}}{({r_{ik}}^2 + \eta^2)} \qquad \text{(Equation 10)}$$

Otherwise, $\mu_{ik} = 0$. The energy equation consistent with Equation 9 is

$$\frac{du_i}{dt} = \frac{1}{2}\sum_k m_k[\frac{P_i}{{\rho_k}^2} + \frac{P_i}{{\rho_i}^2} + \frac{1}{\bar{\rho}_k}(-\alpha\mu_{ik}\bar{c}_{ik} + \beta{\mu_{ik}}^2)]v_{ik} \cdot \nabla_i w_{ik} \qquad \text{(Equation 11)}$$

Once Equations 5, 8, 9, and 11 are known, we may begin to solve the problems of galaxy and star formation, binary star interactions, comet and asteroid impacts and self-gravitating disks (Monaghan).

III. Technical Approach

The square wave test is a problem with a square wave in density initialized in a gas moving with uniform velocity bounded by walls, in which the two wall faces with unit normals parallel to the flow direction move with the flow. The temperature is initialized to be reciprocal to the density (i.e. the temperature dips when the density spikes) such that the pressure is constant. In a typical fluid solver, the sharp discontinuity in density and temperature will cause local oscillations or smearing of the wave (numerical dispersion and diffusion errors), so the sharp boundaries are not maintained as the wave propagates (Cassibry). Below is a graph of the initial density (left) and temperature (right) for the square wave problem. The initial density is 1 or 2 kg/m^3, temperature is 300 or 150 K, and the velocity is a constant 1000 m/s. The run stops at 10^{-4} s, after the wave has propagated 1 unit length of the wave.

For this problem, the wave should remain a square with errors around 10^{-8}. This code was written such that advection should be accurate to floating point precision, which it appears to do. The case has been run for 343, 1728, and 10,648 particles representing the gas on a Dell Precision M6500 laptop, Dell Alienware Aurora ALX, and Dell Precision T7600 workstation.

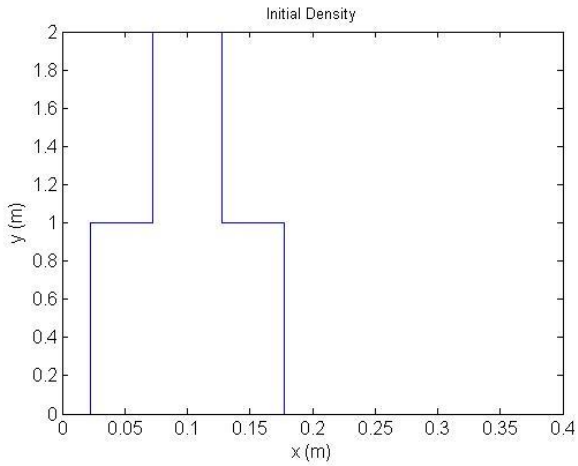


Figure 1: Initial Density of a Square Wave

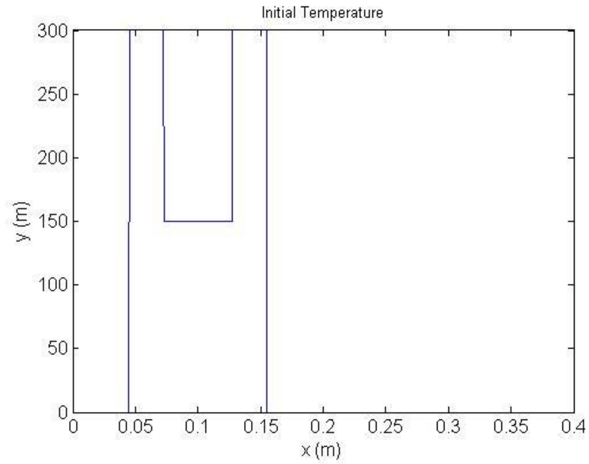


Figure 2: Initial Temperature of a Square Wave

IV. Results and Discussion

Table 1: Wall Clock Computational Time

Machine	N	Time (s)
M6500 Laptop	343	14.9
	1,728	57.1
	10,648	504.2
Alienware	343	9.77
	1,728	32.6
	10,648	259.8
T7600 workstation	343	10.2
	1,728	37.7
	10,648	273.1

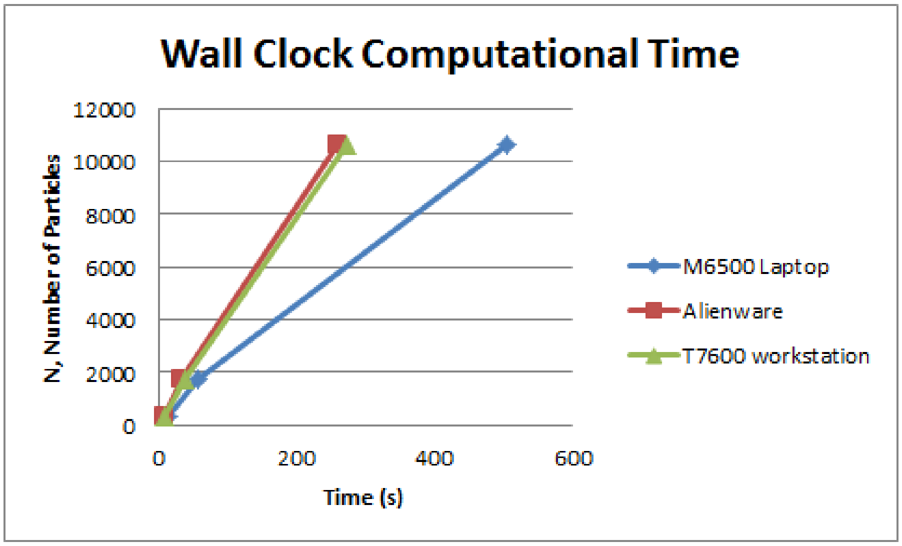


Figure 3: Graph comparing the computational time per number of particles for each machine.

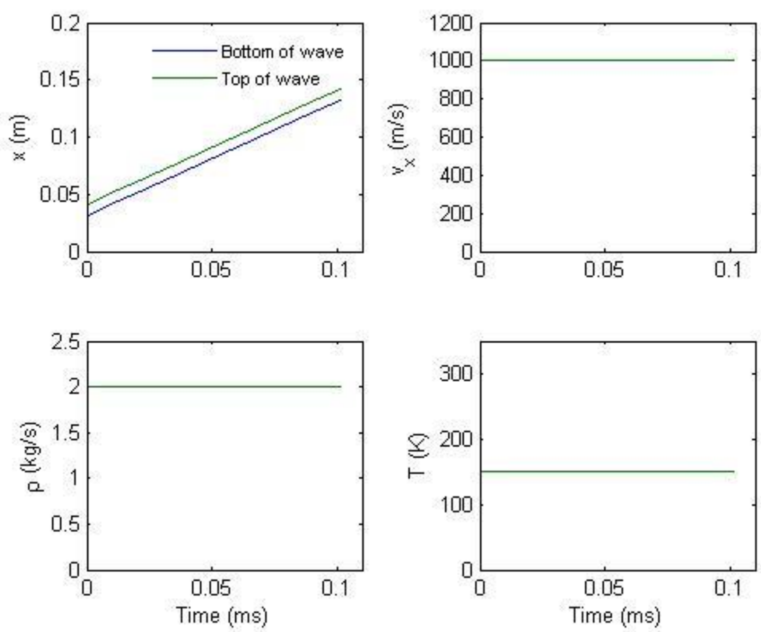


Figure 4: Various properties of the Square Wave Test with 343 particles observed on the M6500 Laptop

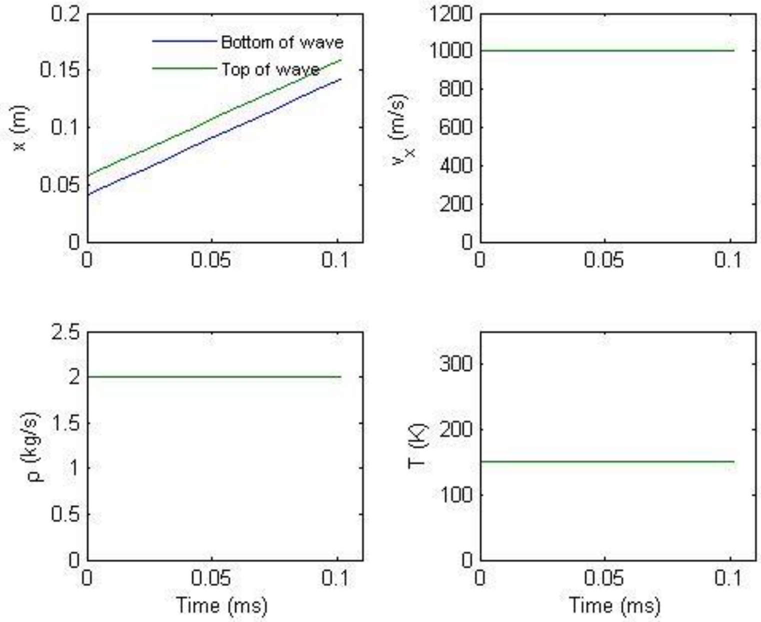


Figure 5: Various properties of the Square Wave Test
with 1728 particles observed on the M6500 Laptop

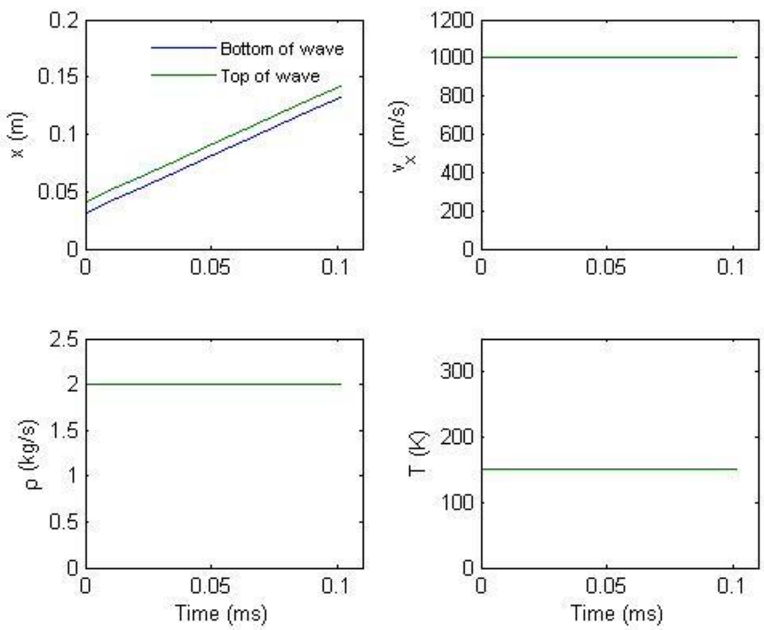


Figure 6: Various properties of the Square Wave Test
with 10648 particles observed on the M6500 Laptop

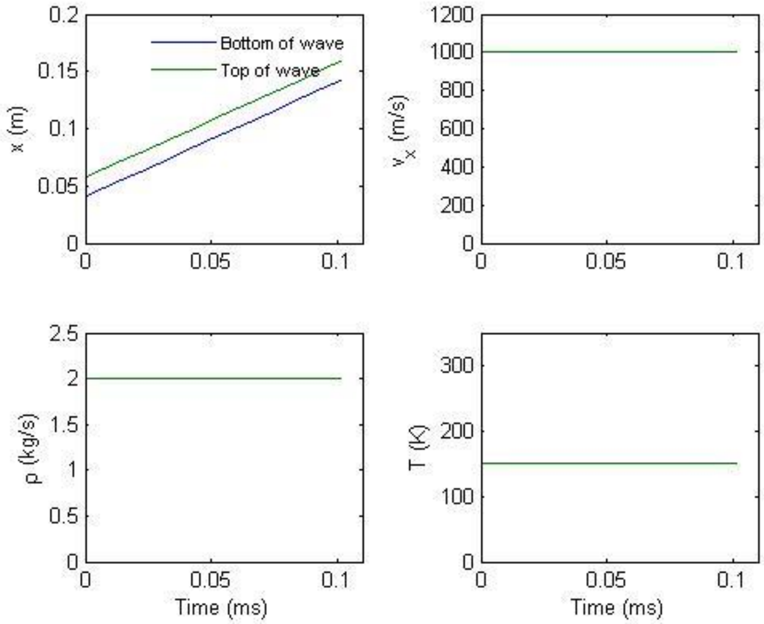
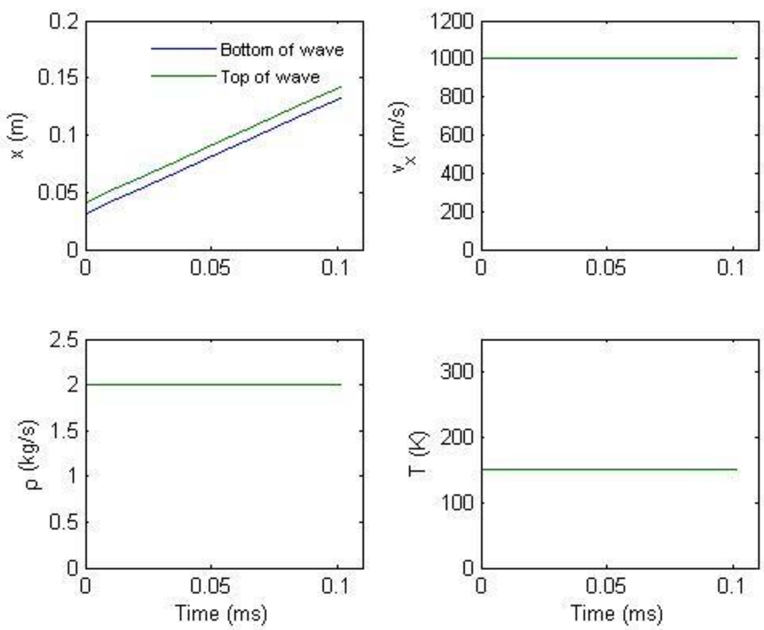


Figure 7: Various properties of the Square Wave Test
with 343 particles observed on Alienware



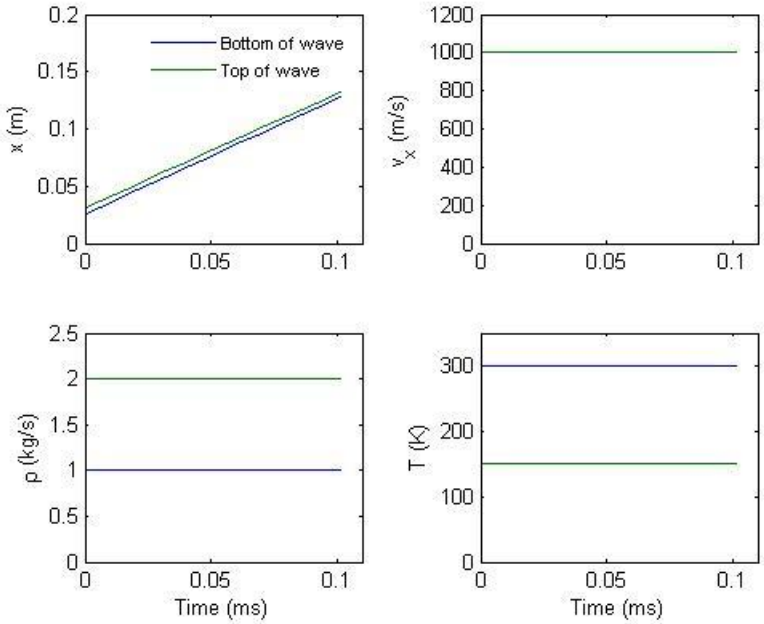


Figure 9: Various properties of the Square Wave Test
with 10648 particles observed on Alienware

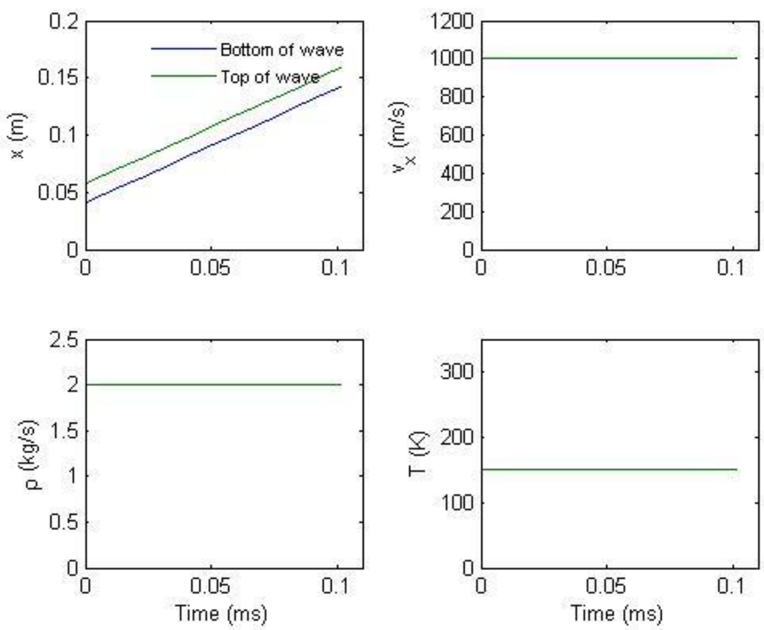


Figure 10: Various properties of the Square Wave Test with 343 particles observed on the T7600 Workstation

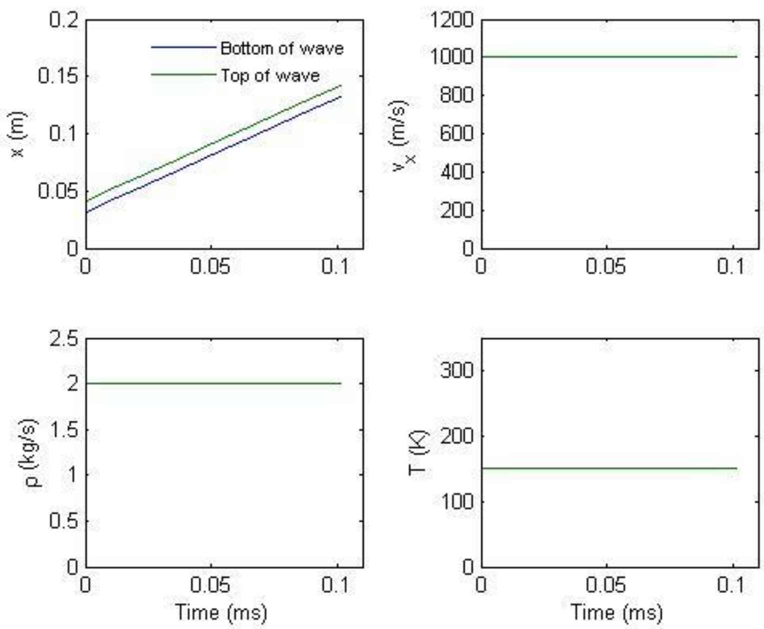


Figure 11: Various properties of the Square Wave Test with 1728 particles observed on the T7600 Workstation

Both Table 1 and Figure 3 show the wall clock computational time in regards to the number of particles for each machine. The time in seconds for the M6500 Laptop to run the square wave test was 14.9, 57.1, and 504.2 for 343 particles, 1,728 particles, and 10,648 particles, respectively. The time in seconds for the Alienware to run the square wave test was 9.77, 32.6, and 259.8 for 343 particles, 1,728 particles, and 10,648 particles, respectively. The time in seconds for the T7600 Workstation to run the square wave test was 10.2, 37.7, and 273.1 for 343 particles, 1,728 particles, and 10,648 particles, respectively. The machine that ran the fastest in all three cases was the Alienware. The Alienware was far faster than the M6500 Laptop while only barely beating out the T7600 workstation. When a smaller number of particles are run in the Square Wave Test, the differences between the computational times are not that large. Where the machines really show what they are capable of is when the number of particles increase immensely.

Figures 4 through 12 show the position in the x direction, velocity in the x direction, density, and temperature of the Square Wave as it moves through time. The results for the M6500 Laptop are shown in Figures 4, 5, and 6 for 343 particles, 1,728 particles, and 10,648 particles, respectively. The results for the Alienware are shown in Figures 7, 8, and 9 for 343 particles, 1,728 particles, and 10,648 particles, respectively. The results for the T7600 workstation are shown in Figures 10, 11, and 12 for 343 particles, 1,728 particles, and 10,648 particles, respectively. For every case, the velocity in the x direction, density, and temperature remained constant throughout time at 1000 m/s, 1 to 2 kg/s, and 150 to 300 K, respectively. These do not visibly change because the SPH code advects square waves accurately. Also, the bottom of the wave and the top of the wave became closer as the number of particles increased. I think it is interesting to note that in every case, the graphs look relatively the same. This shows that the code is accurate no matter how long it takes to run or how many particles it is evaluating.

V. Conclusion

MATLAB is a tool used by many institutions to simulate a variety of test cases that may not be easily observable. The code developed by Dr. Cassibry and his graduate students is meant to take a look at how a gas behaves in space. By using the square wave test case, this code can be tested for accuracy as well as testing the limitations that may be encountered on certain machines. The square wave test was run on each of the three machines for three different cases which were determined by the number of particles. The results of these tests were all very similar with the velocity in the x direction, density, and temperature remaining constant throughout time at 1000 m/s, 1 to 2 kg/s, and 150 to 300 K, respectively as shown in Figures 4 through 12. The results were all constant because the SPH code advects square waves quite accurately. In terms of computational time, the Alienware machine performed the best with 9.77

seconds, 32.6 seconds, and 259.8 seconds for 343 particles, 1,728 particles, and 10,648 particles, respectively. As the number of particles increased so did the time difference between the three machines. The final observation that can be made is that the SPH code is very precise as the results were similar in all 9 cases no matter the number of particles or computational time.

VI. References

Cassibry, Jason. *SPFMax Manual*. N.p.: n.p., n.d. Print.

Monaghan, J. J. "An Introduction to SPH." *Computer Physics Communications* 48 (1988): 89-96. Print.