

Introduction to Compressible Computational Fluid Dynamics

James S. Sochacki
Department of Mathematics
James Madison University
jim@math.jmu.edu

Abstract

This document is intended as an introduction to computational fluid dynamics at the upper undergraduate level. It is assumed that the student has had courses through three dimensional calculus and some computer programming experience with numerical algorithms. A course in differential equations is recommended. This document is intended to be used by undergraduate instructors and students to gain an understanding of computational fluid dynamics. The document can be used in a classroom or research environment at the undergraduate level. The idea of this work is to have the students use the modules to discover properties of the equations and then relate this to the physics of fluid dynamics. Many issues, such as rarefactions and shocks are left out of the discussion because the intent is to have the students discover these concepts and then study them with the instructor. The document is used in part of the undergraduate MATH 365 - Computation Fluid Dynamics course at James Madison University (JMU) and is part of the joint NSF Grant between JMU and North Carolina Central University (NCCU): A Collaborative Computational Sciences Program.

This document introduces the full three-dimensional Navier Stokes equations. Assumptions to these equations are made to derive equations that are accessible to undergraduates with the above prerequisites. These equations are approximated using finite difference methods. The development of the equations and finite difference methods are contained in this document. Software modules and their corresponding documentation in Fortran 90, Maple and Matlab can be downloaded from the website: <http://www.math.jmu.edu/~jim/compressible.html>. The software modules and their corresponding documentation are mentioned in this document.

Introduction

Computational fluid dynamics (CFD) usually involves working with some form of the Navier-Stokes equations. These forms are usually obtained by making some assumptions that simplify the equations. The next step is to develop a numerical algorithm for solving these equations.

The Navier-Stokes equations are commonly expressed in one of two forms. One form is known as the incompressible flow equations and the other is known as the compressible flow equations. The incompressible flow equations model fluids whose density does not change over time. The compressible flow equations allow the density of the fluid to change with the flow.

This document will consider the compressible flow equations. The compressible Navier-Stokes equations are intimidating partial differential equations (PDE's) and rightfully so. They are extremely difficult mathematical equations that describe the motion of a fluid in three dimensions (3D). The equations allow for velocity changes, density changes, energy changes and viscous effects. To appreciate the complexity of these equations think about the flow of water down a rocky incline. These equations can be simplified by making some assumptions about the flow of the fluid that turns the problem into a two dimensional (2D) or sometimes a one dimensional (1D) problem.

The discussions in this document are meant to help you understand some of the complexities of the Navier-Stokes equations. Of course, there are whole industries working on these equations and no industry understands them completely. There are, of course, many mathematical questions that have not been answered regarding the full 3D Navier-Stokes equations.

If assumptions can be made about the fluid being studied the Navier-Stokes equations can be converted into 1D and 2D problems. Many mathematical properties are known about some of these 1D and 2D PDE's. You can study some of these PDE's numerically in the equation documents described in this work.

We now present the full 3D Navier-Stokes equations. The equations are obtained by applying Newton's force laws, the conservation of mass principle and conservation of energy principle to the motion of a fluid. In this document we will assume that you have seen this development in a physics course or have considered the modules at the James Madison University Computational Science site that develop these equations.

The (3D) Navier-Stokes equations are

[1] The Momentum Equation or The Momentum Balance Equation

$$\rho\left(\frac{\partial}{\partial t}U + (U \cdot \nabla)U\right) + \nabla p - \delta\Delta U - \left(\eta + \frac{1}{3}\mu\right)\nabla\nabla \cdot U = G$$

[2] The Continuity Equation or The Mass Balance Equation

$$\frac{\partial}{\partial t}\rho + \nabla \cdot (\rho U) = 0$$

[3] The Energy Equations or The Energy Balance Equations

$$(E1) \quad \frac{\partial}{\partial t}\rho\left(\frac{1}{2}U \cdot U + \epsilon\right) + \nabla \cdot (\rho U\left(\frac{1}{2}U \cdot U + \omega\right)) = 0$$

$$(E2) \quad \frac{\partial}{\partial t}E + \nabla \cdot ((E + p)U) = 0$$

[4] The Equations of State

$$(S1) \quad p = c^2\rho \quad (\text{Acoustics - sound waves})$$

$$(S2) \quad p = (\gamma - 1)\left(E - \frac{1}{2}\rho U \cdot U\right) \quad (\text{Gas Dynamics})$$

$$(S3) \quad p = \alpha\rho^k \quad (\text{Gas Dynamics})$$

The momentum equation is really 3 equations since

$$U = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} u_1(x_1, x_2, x_3, t) \\ u_2(x_1, x_2, x_3, t) \\ u_3(x_1, x_2, x_3, t) \end{pmatrix}$$

or

$$U = \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} u(x, y, z, t) \\ v(x, y, z, t) \\ w(x, y, z, t) \end{pmatrix}.$$

Newton's force law is applied in all three dimensions. Since $U = (u_1, u_2, u_3)$ or $U = (u, v, w)$, the u_1 (or u) component is the velocity of the fluid in the x_1 (or x) direction, the u_2 (or v) component is the velocity of the fluid in the x_2 (or y) direction and the u_3 (or w) component is the velocity of the fluid in the x_3 (or z) direction. (We will use both of these notations for our space dimensions.) Under the energy equations we have listed two energy equations because both forms of these equations are commonly used. We have also listed some equations of state. The equations of state are experimentally derived equations and allow us to eliminate one of the unknowns so that we have 5 PDE's and 5 unknowns. That is, the (3D) Navier-Stokes equations are a system of 5 PDE's. In this document, the documents describing the software modules and the software modules will only use the energy equation (E2)

and one of the two state equations (S1) or (S2). (In (E1) ϵ is the internal energy and ω is the enthalpy.)

We now explain the notation used in the Navier-Stoke's equations; $\rho = \rho(x, y, z, t)$ is the density of the fluid at a point (x, y, z) at time t , $p = p(x, y, z, t)$ is the pressure on the fluid at a point (x, y, z) at time t . The symbol ∇ represents the gradient (column) vector $(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3})$ or $(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$. By ∇p we mean the (column) vector (p_x, p_y, p_z) , where the subscript indicates partial differentiation with respect to that variable. The symbol Δ represents the Laplacian and is given by $\nabla \cdot \nabla$ or $\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ and is applied to each component of $U = (u, v, w)$. The parameters δ, η and μ are functions of (x, y, z) that describe the viscosity of the fluid. The function G is the (column) vector $(g_1(x, y, z, t), g_2(x, y, z, t), g_3(x, y, z, t))$ and models external forces acting on the fluid. In the modules we set $G = 0$. You can easily modify the codes to see what effects external forces have on the fluid flow and the mathematical equations. The internal energy of the fluid is modeled by $\epsilon = \epsilon(x, y, z, t)$ and the total energy of the fluid is modeled by $E = E(x, y, z, t)$. The speed of sound in the fluid is given by $c = c(x, y, z)$. The functions c, α, k and γ are determined experimentally. In the software modules c, α, k and γ are constant.

THE EULER EQUATIONS

The Euler equations are obtained from the Navier-Stokes equations by assuming the fluid is viscous-free. Dropping these terms from the Navier-Stokes equations gives us the (3D) Euler equations

[1] Momentum Equation or Momentum Balance

$$\rho\left(\frac{\partial}{\partial t}U + (U \cdot \nabla)U\right) + \nabla p = G$$

[2] Continuity equation or Mass Balance

$$\frac{\partial}{\partial t}\rho + \nabla \cdot (\rho U) = 0$$

[3] The Energy Equations or The Energy Balance Equations

$$(E1) \quad \frac{\partial}{\partial t}\rho\left(\frac{1}{2}U \cdot U + \epsilon\right) + \nabla \cdot (\rho U\left(\frac{1}{2}U \cdot U + w\right)) = 0$$

$$(E2) \quad \frac{\partial}{\partial t}E + \nabla \cdot ((E + p)U) = 0$$

[4] The Equations of State

$$(S1) \quad p = c^2\rho \quad (\text{Acoustics - sound waves})$$

$$(S2) \quad p = (\gamma - 1)\left(E - \frac{1}{2}\rho U \cdot U\right) \quad (\text{Gas Dynamics})$$

$$(S3) \quad p = \alpha\rho^k \quad (\text{Gas Dynamics})$$

As an exercise you should carry out the differentiation and vector operations in the Navier-Stokes and Euler equations and express the PDE's for these equations in terms of the components of U using either the

$$U = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} u_1(x_1, x_2, x_3, t) \\ u_2(x_1, x_2, x_3, t) \\ u_3(x_1, x_2, x_3, t) \end{pmatrix}$$

or

$$U = \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} u(x, y, z, t) \\ v(x, y, z, t) \\ w(x, y, z, t) \end{pmatrix}$$

notation.

THE 1D NAVIER-STOKES AND EULER EQUATIONS

In the modules we will only consider the (1D) equations. A study of the (1D) equations will give you an insight into the difficulties of studying fluid mechanics and of simulating fluid flow on a computer. Many of the problems arising in (1D) have to be considered in both (2D) and (3D). Of course, more problems arise in (2D) than (1D) and in (3D) than (2D). The modules are meant to give you an understanding of the complexities of computational science and of what computational science is and how it is used in CFD. Although, we are studying fluid flow the ideas of developing a system of equations that describe a phenomena, numerically approximating these equations and their solutions with algorithms, using computers to run the algorithms and using computers to visualize the results make up computational science. Of course, studying each of these steps and the difficulties of each step also make up computational science.

A physical interpretation of the 1D equations is to think of the flow of a fluid in a thin tube or a narrow channel and how the properties of the fluid; velocity, density and energy change only along the tube or the channel. Of course, problems like this occur all the time. Drinking straws, ditches and blood veins are some examples.

We will write the (1D) Navier-Stokes equations as

- [1] The Momentum Equation or The Momentum Balance Equation

$$\rho\left(\frac{\partial}{\partial t}u + u\frac{\partial}{\partial x}u\right) + \frac{\partial}{\partial x}p - \delta\frac{\partial^2}{\partial x^2}u - \left(\eta + \frac{1}{3}\mu\right)\frac{\partial^2}{\partial x^2}u = g(x, t)$$

- [2] The Continuity Equation or The Mass Balance Equation

$$\frac{\partial}{\partial t}\rho + \frac{\partial}{\partial x}(\rho u) = 0$$

- [3] The Energy Equation or The Energy Balance Equation

$$\frac{\partial}{\partial t}E + \frac{\partial}{\partial x}((E + p)u) = 0$$

- [4] The Equations of State

$$(S1) \quad p = c^2\rho \quad (\text{Acoustics - sound waves})$$

$$(S2) \quad p = (\gamma - 1)\left(E - \frac{1}{2}\rho u^2\right) \quad (\text{Gas Dynamics})$$

Note that the momentum equation can be written as

$$\rho u_t + \rho u u_x + p_x - \alpha u_{xx} = g$$

or

$$\rho(u_t + (\frac{1}{2}u^2)_x) + p_x - \alpha u_{xx} = g$$

where $\alpha = \delta + \eta + \frac{1}{3}\mu$. If the pressure does not depend on x this equation reduces to

$$\rho(u_t + uu_x) - \alpha u_{xx} = g$$

or

$$\rho(u_t + (\frac{1}{2}u^2)_x) - \alpha u_{xx} = g.$$

Now suppose the fluid density, ρ is a constant. Dividing by ρ gives us

$$(B1) \quad u_t + uu_x - \nu u_{xx} = G$$

or

$$(B2) \quad u_t + (\frac{1}{2}u^2)_x - \nu u_{xx} = G.$$

where $\nu = \frac{\alpha}{\rho}$ and $G(x, t) = \frac{g(x, t)}{\rho}$. These PDE's are known as Burger's equation with viscosity. Equation (B1) is known as the non-conservation form of Burger's equation and equation (B2) is known as the conservation form of Burger's equation. A study of these equations will help you to understand fluid velocity in the Navier-Stokes equations. You can click on the Burger's Modules Description Document link to learn how to do a numerical study of these equations through the software modules. A good discovery project is to modify the modules to include viscosity and then study what happens as $\nu \rightarrow 0$ and compare this with the numerical solution to Burger's equation with $\nu = 0$. After this study return to equation (B2) and see how you could verify your numerical results analytically.

We now consider the Continuity Equation PDE. If the fluid velocity u is a constant this PDE reduces to

$$(W) \quad \frac{\partial}{\partial t}\rho + u \frac{\partial}{\partial x}(\rho) = 0.$$

We leave it as an exercise for you to use the chain rule of differentiation to show that $\rho(x, t) = f(x - ut)$ is a general solution to this PDE for an arbitrary differentiable function f and a constant fluid velocity u . Note that $\rho(x, 0) = f(x)$. Therefore, physically f is the fluid density at time 0 or the initial fluid density. Also, note that the graph of $f(x - ut)$ is a translation of the graph of $f(x)$ to the right or left by a distance $|ut|$ depending on the sign of u . Therefore, the general solution to equation (W) is a translation of the initial density by $|ut|$. We see that over time the initial density moves to the right or left by an amount $|ut|$. That is, as the time t increases the initial density moves further to the right or to the left. This is known as a wave. Since $|ut|$ can be thought of as the distance the initial density moves, we will call $|u|$ the speed of the wave. To do a numerical study of advection diffusion equation click on the Advection Diffusion Modules Description Document link. This document will describe the software modules for this equation. You will observe the phenomena described above in these software modules.

Note that equation (B1) and equation (W) are similar to the PDE

$$(AD) \quad w_t + cw_x + dw_{xx} = h(x, t).$$

This PDE is known as the advection-diffusion equation. In order to get (B1) from (AD) we replace u by w in the partial derivative expressions, u with no derivatives by c and α by d . In order to get (W) from (AD) we replace ρ by w , u by c and d would equal 0. In the software modules you will see that c is the speed (or advection) of the wave and d is the diffusion of the system.

We now consider the (1D) Euler equations. An understanding of these equations is essential to understanding the (1D) Navier-Stokes equations. After completing the modules associated with the (1D) Euler equations you should be able to write your own codes for the (1D) Navier-Stokes equations.

The (1D) Euler equations which we will denote as (E1) are

- [1] The Momentum Equation or The Momentum Balance Equation

$$\rho\left(\frac{\partial}{\partial t}u + u\frac{\partial}{\partial x}u\right) + \frac{\partial}{\partial x}p = g(x, t)$$

- [2] The Continuity Equation or The Mass Balance Equation

$$\frac{\partial}{\partial t}\rho + \frac{\partial}{\partial x}(\rho u) = 0$$

- [3] The Energy Equation or The Energy Balance Equation

$$\frac{\partial}{\partial t}E + \frac{\partial}{\partial x}((E + p)u) = 0$$

- [4] The Equations of State

$$(S1) \quad p = c^2\rho \quad (\text{Acoustics - sound waves})$$

$$(S2) \quad p = (\gamma - 1)(E - \frac{1}{2}\rho u^2) \quad (\text{Gas Dynamics})$$

We now consider another form of the (1D) Euler equations. From the product rule we have that

$$(\rho u)_t = \rho_t u + \rho u_t$$

and that

$$\rho u_t = (\rho u)_t - \rho_t u.$$

From the continuity equation we have that $-\rho_t = \frac{\partial}{\partial x}(\rho u)$. Using this in the last equation we obtain

$$\rho u_t = (\rho u)_t + u\frac{\partial}{\partial x}(\rho u).$$

Substituting this into the momentum equation we see that

$$\begin{aligned} \rho u_t + \rho u \frac{\partial}{\partial x} u + \frac{\partial}{\partial x} p &= \\ (\rho u)_t + u \frac{\partial}{\partial x} (\rho u) + \rho u \frac{\partial}{\partial x} u + \frac{\partial}{\partial x} p &= \\ (\rho u)_t + \frac{\partial}{\partial x} (u \rho u) + \frac{\partial}{\partial x} p &= g \end{aligned}$$

where in the last equality we used the product rule. From the last line we now obtain the following form for the momentum equation

$$(\rho u)_t + \frac{\partial}{\partial x} (u^2 \rho + p) = g.$$

Using this form of the momentum equation the (1D) Euler equations (with this momentum equation) which we will denote as (E2) are

[1] The Momentum Equation or The Momentum Balance Equation

$$\frac{\partial}{\partial t} (\rho u) + \frac{\partial}{\partial x} (u^2 \rho + p) = g$$

[2] The Continuity Equation or The Mass Balance Equation

$$\frac{\partial}{\partial t} \rho + \frac{\partial}{\partial x} (\rho u) = 0$$

[3] The Energy Equation or The Energy Balance Equation

$$\frac{\partial}{\partial t} E + \frac{\partial}{\partial x} ((E + p)u) = 0$$

[4] The Equations of State

$$(S1) \quad p = c^2 \rho \quad (\text{Acoustics - sound waves})$$

$$(S2) \quad p = (\gamma - 1)(E - \frac{1}{2} \rho u^2) \quad (\text{Gas Dynamics})$$

If we use (S1) in the momentum equation and the continuity equation we obtain the following system of equations for u and ρ which we will denote as (E3)

[1] The Momentum Equation or The Momentum Balance Equation

$$\frac{\partial}{\partial t} (\rho u) + \frac{\partial}{\partial x} ((u^2 + c^2) \rho) = g$$

[2] The Continuity Equation or The Mass Balance Equation

$$\frac{\partial}{\partial t}\rho + \frac{\partial}{\partial x}(\rho u) = 0$$

or the following system of equations for u and p which we will denote as (E4)

[1] The Momentum Equation or The Momentum Balance Equation

$$\frac{\partial}{\partial t}\left(\frac{p}{c^2}u\right) + \frac{\partial}{\partial x}\left(\frac{u^2+c^2}{c^2}p\right) = g$$

[2] The Continuity Equation or The Mass Balance Equation

$$\frac{\partial}{\partial t}\frac{p}{c^2} + \frac{\partial}{\partial x}\left(\frac{p}{c^2}u\right) = 0$$

Since the speed of sound in the fluid is usually known, we see that both of these forms are two equations in two unknowns. We leave it as an exercise for you to show that these two equations can be written in the form

$$\begin{aligned}\frac{\partial}{\partial t}v_1 + \frac{\partial}{\partial x}f_1(v_1, v_2) &= g(x) \\ \frac{\partial}{\partial t}v_2 + \frac{\partial}{\partial x}f_2(v_1, v_2) &= 0.\end{aligned}$$

(You need to determine v_1, v_2, f_1 and f_2 .) Equations of this form are called conservation equations. You will find that the functions f_1 and f_2 are similar for (E3) and (E4). If you solve (E3) you can get the pressure from $p = c^2\rho$ and if you solve (E4) you can get the density from this same relationship. The software modules allow you to do a numerical study of equation (E3). To learn how to do this study click on the link [One Dimensional Density Velocity Equations Modules Description Document](#). This document will describe the software modules dealing with this equation. Later, we will use the acoustic condition S2 ($p = c^2\rho$) to generate a linear equation for pressure disturbances.

If we use (S2) in the energy equation we obtain the following system of equations for u, ρ and E which we will denote as (E5)

[1] The Momentum Equation or The Momentum Balance Equation

$$\frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}((\gamma - 1)(E - \frac{1}{2}\rho u^2)) = g$$

[2] The Continuity Equation or The Mass Balance Equation

$$\frac{\partial}{\partial t}\rho + \frac{\partial}{\partial x}(\rho u) = 0$$

[3] The Energy Equation or The Energy Balance Equation

$$\frac{\partial}{\partial t}E + \frac{\partial}{\partial x}((\gamma E - \frac{(1 - \gamma)}{2}\rho u^2)u) = 0$$

Note these equations can be put in the form

$$\frac{\partial}{\partial t}v_1 + \frac{\partial}{\partial x}f_1(v_1, v_2, v_3) = 0$$

$$\frac{\partial}{\partial t}v_2 + \frac{\partial}{\partial x}f_2(v_1, v_2, v_3) = 0$$

$$\frac{\partial}{\partial t}v_3 + \frac{\partial}{\partial x}f_3(v_1, v_2, v_3) = 0$$

where $v_1 = (\rho u), v_2 = \rho$ and $v_3 = E$ and where

$$\begin{aligned} f_1(v_1, v_2, v_3) &= \frac{v_1^2}{v_2} + (\gamma - 1)(v_3 - \frac{1}{2}\frac{v_1^2}{v_2}) \\ f_2(v_1, v_2, v_3) &= v_1 \\ f_3(v_1, v_2, v_3) &= (\gamma v_3 - \frac{(1-\gamma)}{2}\frac{v_1^2}{v_2})\frac{v_1}{v_2} \end{aligned} .$$

We leave it as an exercise for you to carry out the process we did to get these conservation equations from (E1) and (S2) to get the conservation equation form for the (3D) Euler equations. To do a numerical study of the (1D) Euler equations click on the link [One Dimensional Euler Equations Modules Description Document](#). This document will describe the software modules associated with this equation.

We could have substituted (S1) or (S2) directly into (E1) and used these equations in the modules. We leave this as an exercise for you. You can compare and contrast your codes for these systems with the codes in the modules.

The 1D Linearized Pressure Equation or Acoustic Equation

We will assume c is a constant in our derivation. To model the propagation of sound (acoustic or pressure) waves we will linearize the following two equations.

[1] The Momentum Equation or The Momentum Balance Equation

$$\frac{\partial}{\partial t} \left(\frac{p}{c^2} u \right) + \frac{\partial}{\partial x} \left(u^2 \frac{p}{c^2} + p \right) = 0$$

[2] The Continuity Equation or The Mass Balance Equation

$$\frac{\partial}{\partial t} \frac{p}{c^2} + \frac{\partial}{\partial x} \left(\frac{p}{c^2} u \right) = 0.$$

In acoustic (sound) waves the particle velocity u is small. That is, $|u| \ll 1$ implying $u^2 \ll u$. Therefore, we linearize the Momentum Equation by dropping the nonlinear terms containing u^2 . This gives us

[1] The Momentum Equation or The Momentum Balance Equation

$$\frac{\partial}{\partial t} \left(\frac{p}{c^2} u \right) + \frac{\partial}{\partial x} p = 0$$

[2] The Continuity Equation or The Mass Balance Equation

$$\frac{\partial}{\partial t} \frac{p}{c^2} + \frac{\partial}{\partial x} \left(\frac{p}{c^2} u \right) = 0.$$

We differentiate the Momentum Equation with respect to x and the Continuity Equation with respect to t and equate the mixed partial derivative term $\left(\frac{p}{c^2} u \right)_{xt}$ to get

$$(LP) \quad p_{tt} - c^2 p_{xx} = 0.$$

We leave it for you to show that

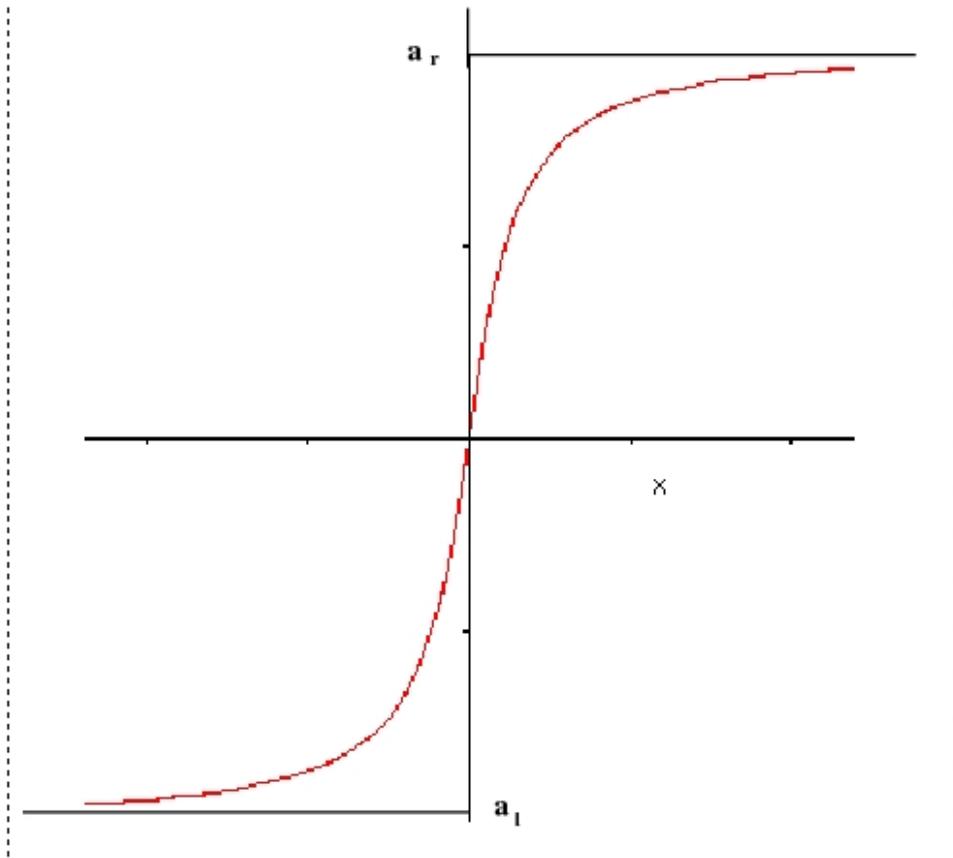
$$p(x, t) = f(x - ct) + g(x + ct)$$

for arbitrary functions f and g is a solution to (LP). That is, the linearized pressure equation has solutions made up waves traveling to the left and to the right with speed c .

THE NUMERICAL SOLUTIONS

Introduction

In order to generate solutions to any of the PDE's (W),(AD),(B1),(B2),(E3),(E4),(E5) or (LP) we have to give initial values to each of the functions in these PDE's. For example, in (AD) we must give a value to $w(x, t)$ at $t = 0$. In the software modules we will use the function $q \arctan(sx) + r$ for the initial value. The graph of this function looks like

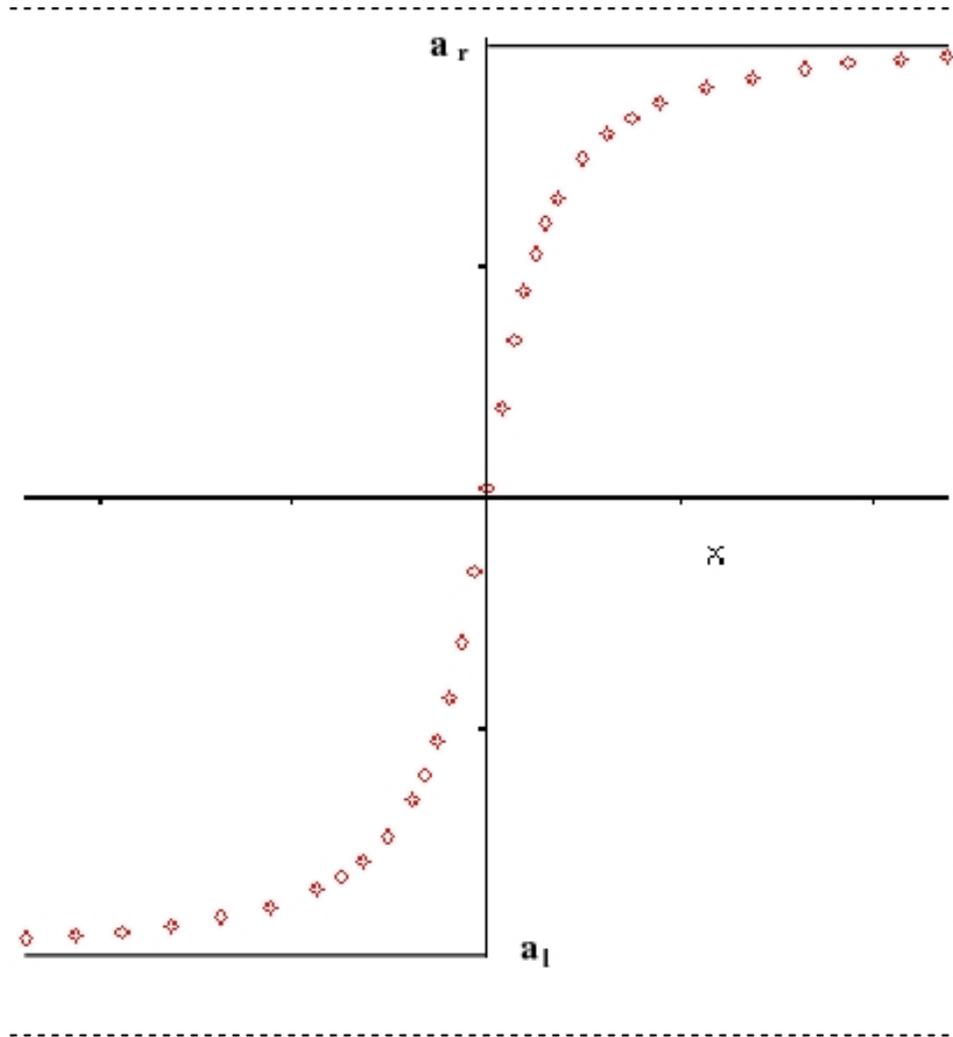


The Initial Condition for the Functions

We use this initial condition because it models a sudden change or front in the fluid properties. The values a_r and a_l are inputted by the user and from these values we obtain q and r . The value s is inputted by the user and gives the steepness of the front. The larger s is the more vertical the front will be. If the front is vertical the abrupt change from a_l to a_r is called a shock. We will study how this front changes or propagates over time t as determined by the PDE we are considering. Physically,

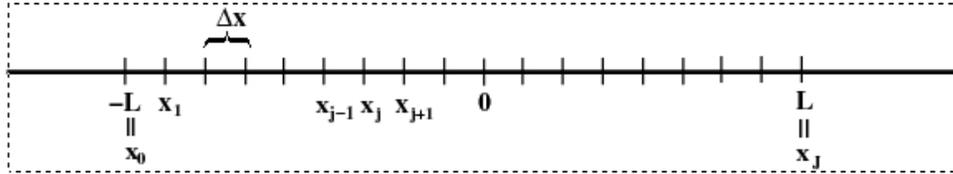
one can think of having a glass tube of length 20 meters and at the midpoint there is a filament that blocks air from traveling down the tube. However, this filament can be removed to allow the air to flow down the tube. Smoke filled air is put in the glass tube on the left. The right is filled with clean air. The filament is then removed. Intuitively, one would think the smoke would move to the right into the clean air. We will look to see if this happens in our modules that simulate fluid flow.

Since we will be approximating the solutions to the PDE's using a numerical algorithm on the computer, we really will only be using the initial profile at a set of discrete x values called grid points. Therefore our initial value really looks like



The Initial Condition for the Functions at the Grid Points

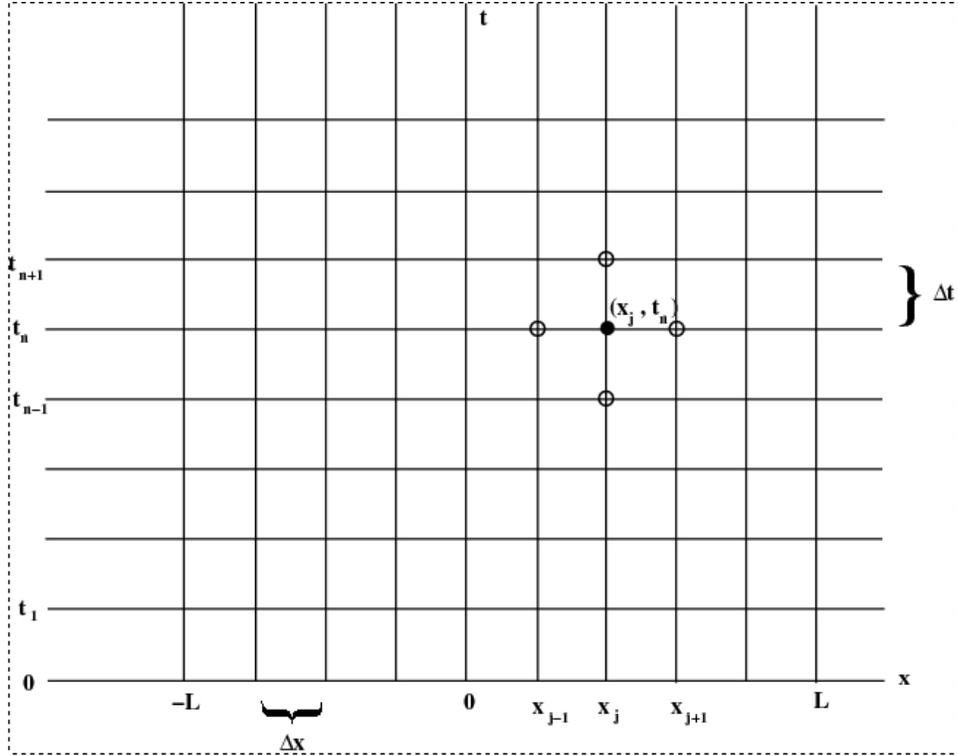
The term for approximating the solutions to the PDE's using a numerical algorithm is **numerical solutions**. We want to determine numerical solutions to our PDE's at some grid points. We will always use grid points that are equally spaced. That is, if our grid points are $x_0, x_1, \dots, x_{j-1}, x_j, x_{j+1}, \dots, x_J$ then $\Delta x = x_{j+1} - x_j = x_j - x_{j-1}$ for $j = 1, \dots, J$ as shown in the following figure. (You should consider how the algorithms presented should be adjusted to handle variable grid sizes.)



The Grid Points for our Numerical Solutions to the PDE's

Note that $x_0 = -L$ and that $x_j = x_0 + j\Delta x$ for $j = 1, \dots, J$ and that we are studying our fluid over a length $2L$.

We will have to determine our solutions over time t also. We will therefore discretize time t as we did the distance x with grid points. We will choose equally spaced discrete times also. That is, we will determine our numerical solutions at times $t_0, t_1, \dots, t_{n-1}, t_n, t_{n+1}, \dots, t_N$ where $t_0 = 0$ and $\Delta t = t_{n+1} - t_n = t_n - t_{n-1}$ for $n = 1, \dots, N$. For example, in (AD) we want to determine $w(x_j, t_n)$ for $j = 1, \dots, J$ and $n = 1, \dots, N$. That is, we want to determine a numerical approximation to w at each of the points in the following (2D) lattice in the (x, t) plane.



The Lattice of Points in the (x, t) plane at which Numerical Solutions to the PDE's will be determined

We will use the same variable for both the solution and the numerical solution. The context will make it clear as to which form of the solution we are using.

We can plot our solutions for all x_j grid points at each time t_n . That is, we can plot $w(x_j, t_n)$ $j = 1, \dots, J$ for each n from 0 to N . This process will build $N + 1$ time frames of the fluid property we are studying. We can then animate this as a movie. We could also plot our solutions for all the t values at a single grid point. That is, we can look at the fluid property at a single grid point as time moves forward. In the modules we will do the former and leave the latter for you to do.

To generate the numerical solutions we will use Taylor series approximations to the solutions. Suppose our solution $w(x, t)$ has a Taylor series in x then we have

$$w(x + \Delta x, t) = w(x, t) + w_x(x, t)\Delta x + \frac{1}{2}w_{xx}(x, t)\Delta x^2 + \frac{1}{3!}w_{xxx}(x, t)\Delta x^3 + \frac{1}{4!}w_{xxxx}(x, t)\Delta x^4 + \dots$$

and

$$w(x - \Delta x, t) = w(x, t) - w_x(x, t)\Delta x + \frac{1}{2}w_{xx}(x, t)(-\Delta x)^2 +$$

$$\frac{1}{3!}w_{xxx}(x,t)(-\Delta x^3) + \frac{1}{4!}w_{xxxx}(x,t)(-\Delta x)^4 + \dots$$

We call the first Taylor series the forward series and the latter Taylor series the backward series. From the forward series we have

$$w_x(x,t) = \frac{w(x+\Delta x,t) - w(x,t)}{\Delta x} - \frac{1}{2}w_{xx}(x,t)\Delta x - \frac{1}{3!}w_{xxx}(x,t)\Delta x^2 - \frac{1}{4!}w_{xxxx}(x,t)\Delta x^3 - \dots$$

and from the backward series we have

$$w_x(x,t) = \frac{w(x,t) - w(x-\Delta x,t)}{\Delta x} + \frac{1}{2}w_{xx}(x,t)\Delta x - \frac{1}{3!}w_{xxx}(x,t)\Delta x^2 + \frac{1}{4!}w_{xxxx}(x,t)\Delta x^3 \dots$$

We will call the approximation

$$(FD) \quad \frac{w(x+\Delta x,t) - w(x,t)}{\Delta x}$$

to $w_x(x,t)$ the forward difference in space approximation to $w_x(x,t)$ and we will call

$$(BD) \quad \frac{w(x,t) - w(x-\Delta x,t)}{\Delta x}$$

the backward difference in space approximation to $w_x(x,t)$. The error in both of these approximations is seen to depend on Δx and w_{xx} . If we subtract the backward series from the forward series we have

$$w_x(x,t) = \frac{w(x+\Delta x,t) - w(x-\Delta x,t)}{2\Delta x} - \frac{1}{3!}w_{xxx}(x,t)\Delta x^2 - \dots$$

We call

$$(CD) \quad \frac{w(x+\Delta x,t) - w(x-\Delta x,t)}{2\Delta x}$$

the centered difference in space approximation to $w_x(x,t)$. We see that the error in this approximation depends on w_{xxx} and Δx^2 . If $\Delta x \ll 1$ then we would expect using (CD) to give us better results than using (FD) or (BD). In the modules you will learn that this is not always the case.

If we add the forward and backward series we find that

$$w_{xx}(x,t) = \frac{w(x+\Delta x,t) - 2w(x,t) + w(x-\Delta x,t)}{\Delta x^2} + \frac{1}{4!}w_{xxxx}(x,t)\Delta x^2 + \dots$$

We call

$$(SCD) \quad \frac{w(x+\Delta x,t) - 2w(x,t) + w(x-\Delta x,t)}{\Delta x^2}$$

the second centered difference in space approximation to $w_{xx}(x, t)$.

Advection Diffusion Equation

Consider the equation (AD) (with $h = 0$)

$$w_t(x_j, t) = w'(x_j, t) = -cw_x(x_j, t) - dw_{xx}(x_j, t) \text{ for } j = 1, \dots, J - 1$$

If we apply (FD),(BD) and (CD) to this equation we obtain the following numerical schemes

$$\begin{aligned} & \text{(ADF)} \\ w_t(x_j, t) = w'(x_j, t) &= -c\left(\frac{w(x_j+\Delta x, t) - w(x_j, t)}{\Delta x}\right) - d\left(\frac{w(x_j+\Delta x, t) - 2w(x_j, t) + w(x_j-\Delta x, t)}{\Delta x^2}\right) \\ & \text{for } j = 1, \dots, J - 1 \end{aligned}$$

$$\begin{aligned} & \text{(ADB)} \\ w_t(x_j, t) = w'(x_j, t) &= -c\left(\frac{w(x_j, t) - w(x_j-\Delta x, t)}{\Delta x}\right) - d\left(\frac{w(x_j+\Delta x, t) - 2w(x_j, t) + w(x_j-\Delta x, t)}{\Delta x^2}\right) \\ & \text{for } j = 1, \dots, J - 1 \end{aligned}$$

$$\begin{aligned} & \text{(ADC)} \\ w_t(x_j, t) = w'(x_j, t) &= -c\left(\frac{w(x_j+\Delta x, t) - w(x_j-\Delta x, t)}{2\Delta x}\right) - d\left(\frac{w(x_j+\Delta x, t) - 2w(x_j, t) + w(x_j-\Delta x, t)}{\Delta x^2}\right) \\ & \text{for } j = 1, \dots, J - 1 \end{aligned}$$

respectively. We can rewrite these equations as

$$\begin{aligned} \text{(ADF)} \quad w_t(x_j, t) = w'(x_j, t) &= -c\left(\frac{w(x_{j+1}, t) - w(x_j, t)}{\Delta x}\right) - d\left(\frac{w(x_{j+1}, t) - 2w(x_j, t) + w(x_{j-1}, t)}{\Delta x^2}\right) \\ & \text{for } j = 1, \dots, J - 1 \end{aligned}$$

$$\begin{aligned} \text{(ADB)} \quad w_t(x_j, t) = w'(x_j, t) &= -c\left(\frac{w(x_j, t) - w(x_{j-1}, t)}{\Delta x}\right) - d\left(\frac{w(x_{j+1}, t) - 2w(x_j, t) + w(x_{j-1}, t)}{\Delta x^2}\right) \\ & \text{for } j = 1, \dots, J - 1 \end{aligned}$$

$$\begin{aligned} \text{(ADC)} \quad w_t(x_j, t) = w'(x_j, t) &= -c\left(\frac{w(x_{j+1}, t) - w(x_{j-1}, t)}{2\Delta x}\right) - d\left(\frac{w(x_{j+1}, t) - 2w(x_j, t) + w(x_{j-1}, t)}{\Delta x^2}\right) \\ & \text{for } j = 1, \dots, J - 1 \end{aligned}$$

Note that each of these is a system of ordinary differential equations for $J - 1$ functions $w_j(t) = w(x_j, t)$. Note that in (ADF) when $J = 1$ we need $w(x_0, t)$ and that in (ADB) when $j = J - 1$ we need $w(x_J, t)$. In our modules we will always use

$$w(x_0, t) = \text{qarctan}(sx_0) + r.$$

and

$$w(x_J, t) = \text{qarctan}(sx_J) + r.$$

That is, we will keep $w(x_0, t)$ and $w(x_J, t)$ constant over time. We will not solve systems of ordinary differential equations in this document. The reader can solve these systems.

We now replace the $w_t(x_j, t) = w'(x_j, t)$ by forward, backward and centered differences in time. For example, consider (ADF) we have

$$\frac{w(x_j, t_{n+1}) - w(x_j, t_n)}{\Delta t} = -c \left(\frac{w(x_{j+1}, t_n) - w(x_j, t_n)}{\Delta x} \right) - d \left(\frac{w(x_{j+1}, t_n) - 2w(x_j, t_n) + w(x_{j-1}, t_n)}{\Delta x^2} \right) \quad (\text{ADFF})$$

for $j = 1, \dots, J - 1$

$$\frac{w(x_j, t_n) - w(x_j, t_{n-1})}{\Delta t} = -c \left(\frac{w(x_{j+1}, t_n) - w(x_j, t_n)}{\Delta x} \right) - d \left(\frac{w(x_{j+1}, t_n) - 2w(x_j, t_n) + w(x_{j-1}, t_n)}{\Delta x^2} \right) \quad (\text{ADFB})$$

for $j = 1, \dots, J - 1$

$$\frac{w(x_j, t_{n+1}) - w(x_j, t_{n-1})}{2\Delta t} = -c \left(\frac{w(x_{j+1}, t_n) - w(x_j, t_n)}{\Delta x} \right) - d \left(\frac{w(x_{j+1}, t_n) - 2w(x_j, t_n) + w(x_{j-1}, t_n)}{\Delta x^2} \right) \quad (\text{ADFC})$$

for $j = 1, \dots, J - 1$

as the forward in time, backward in time and centered in time approximations for (ADF), respectively. The schemes (ADFF) and (ADFC) are known as explicit schemes because we can solve explicitly for $w(x_j, t_{n+1})$ for $j = 1, \dots, J - 1$. The scheme (ADFB) is known as an implicit scheme because we have to solve all the equations at once for $w(x_j, t_n)$ for $j = 1, \dots, J - 1$. We have to write (ADFB) as a system of $J - 1$ equations and solve the resulting matrix equation. We will only consider explicit schemes in the modules. Solving for $w(x_j, t_{n+1})$ in (ADFF) and (ADFC) we obtain

$$w(x_j, t_{n+1}) = w(x_j, t_n) - \frac{c\Delta t}{\Delta x} (w(x_{j+1}, t_n) - w(x_j, t_n)) - \frac{d\Delta t}{\Delta x^2} (w(x_{j+1}, t_n) - 2w(x_j, t_n) + w(x_{j-1}, t_n)) \quad (\text{ADFF})$$

for $j = 1, \dots, J - 1$

$$w(x_j, t_{n+1}) = w(x_j, t_{n-1}) - \frac{2c\Delta t}{\Delta x} (w(x_{j+1}, t_n) - w(x_j, t_n)) - \frac{2d\Delta t}{\Delta x^2} (w(x_{j+1}, t_n) - 2w(x_j, t_n) + w(x_{j-1}, t_n)) \quad (\text{ADFC})$$

for $j = 1, \dots, J - 1$

Let's now consider (ADFF). Suppose $n = 0$ then we have

$$w(x_j, t_1) = w(x_j, 0) - \frac{c\Delta t}{\Delta x} (w(x_{j+1}, 0) - w(x_j, 0)) - \frac{d\Delta t}{\Delta x^2} (w(x_{j+1}, 0) - 2w(x_j, 0) + w(x_{j-1}, 0))$$

for $j = 1, \dots, J - 1$

since $t_0 = 0$. We can see that in order to find $w(x_j, t_1)$ for $j = 1, \dots, J - 1$ we need to know $w(x_j, 0)$ for $j = 1, \dots, J - 1$. Of course, we do know this since $w(x_j, 0)$ is given by our initial condition. We now have an approximation to w at the next time step. Letting $n = 1$ in (ADFF) gives us

$$w(x_j, t_2) - \frac{c\Delta t}{\Delta x}(w(x_{j+1}, t_1) - w(x_j, t_1)) - \frac{d\Delta t}{\Delta x^2}(w(x_{j+1}, t_1) - 2w(x_j, t_1) + w(x_{j-1}, t_1))$$

for $j = 1, \dots, J - 1$.

Since we solved for $w(x_j, t_1)$ for $j = 1, \dots, J - 1$ when $n = 0$, we can determine $w(x_j, t_2)$ for $j = 1, \dots, J - 1$. We proceed in this fashion.

Note that to get $w(x_j, t_2)$ for $j = 1, \dots, J - 1$ we do not need to know $w(x_j, 0)$. Therefore, in our codes we will write $w(x_j, t_0)$ for $j = 1, \dots, J - 1$ to a file and replace $w(x_j, 0)$ for $j = 1, \dots, J - 1$ by $w(x_j, t_1)$ for $j = 1, \dots, J - 1$. We will then solve for $w(x_j, t_2)$ for $j = 1, \dots, J - 1$. We then write $w(x_j, t_2)$ for $j = 1, \dots, J - 1$ to a file and replace $w(x_j, 0)$ for $j = 1, \dots, J - 1$ by $w(x_j, t_2)$ for $j = 1, \dots, J - 1$. We continue this process for each time step we solve for $w(x_j, t_n)$. This is a common practice since it reduces the amount of computer memory our program will need. Since we put the w values in a file, we can use the w values in these files for analysis and visualization.

In the module where we study (AD) we use

$$(ADFF) \quad w(x_j, t_{n+1}) = w(x_j, t_n) - \frac{c\Delta t}{\Delta x}(w(x_{j+1}, t_n) - w(x_j, t_n)) - \frac{d\Delta t}{\Delta x^2}(w(x_{j+1}, t_n) - 2w(x_j, t_n) + w(x_{j-1}, t_n))$$

for $j = 1, \dots, J - 1$,

$$(ADBF) \quad w(x_j, t_{n+1}) = w(x_j, t_n) - \frac{c\Delta t}{\Delta x}(w(x_j, t_n) - w(x_{j-1}, t_n)) - \frac{d\Delta t}{\Delta x^2}(w(x_{j+1}, t_n) - 2w(x_j, t_n) + w(x_{j-1}, t_n))$$

for $j = 1, \dots, J - 1$

and

$$(ADCC) \quad w(x_j, t_{n+1}) = w(x_j, t_{n-1}) - \frac{c\Delta t}{\Delta x}(w(x_{j+1}, t_n) - w(x_{j-1}, t_n)) - \frac{2d\Delta t}{\Delta x^2}(w(x_{j+1}, t_n) - 2w(x_j, t_n) + w(x_{j-1}, t_n))$$

for $j = 1, \dots, J - 1$.

Schemes (ADFF) and (ADBF) are called upwind schemes and (ADCC) is called a leap frog scheme. To see where these names come from we let $d = 0$. We then have

$$(ADFF) \quad w(x_j, t_{n+1}) = w(x_j, t_n) - \frac{c\Delta t}{\Delta x}(w(x_{j+1}, t_n) - w(x_j, t_n))$$

for $j = 1, \dots, J - 1$,

$$(ADBF) \quad w(x_j, t_{n+1}) = w(x_j, t_n) - \frac{c\Delta t}{\Delta x}(w(x_j, t_n) - w(x_{j-1}, t_n))$$

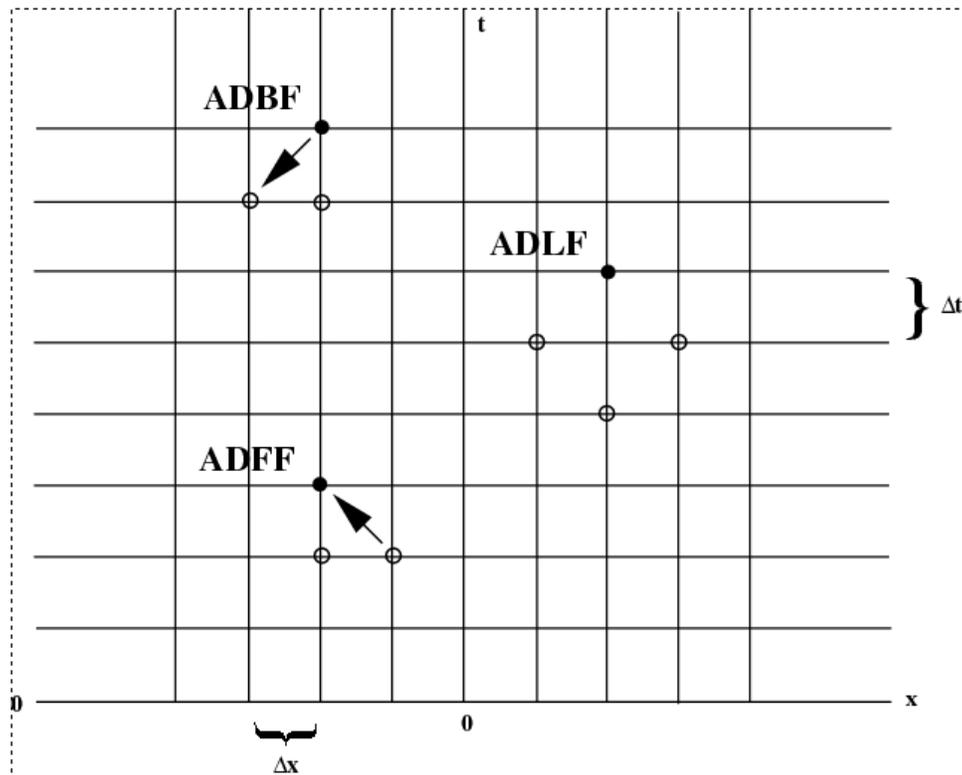
for $j = 1, \dots, J - 1$

and

$$(ADCC) \quad w(x_j, t_{n+1}) = w(x_j, t_{n-1}) - \frac{c\Delta t}{\Delta x}(w(x_{j+1}, t_n) - w(x_{j-1}, t_n))$$

for $j = 1, \dots, J - 1$.

In the next figure we show the grid points that are involved in the calculations of each of these equations. Recall that if $c > 0$ the advection is to the right and vice versa if $c < 0$. That is, the initial condition moves to the right for $c > 0$ and to the left for $c < 0$. We see that (ADFF) gives movement to the right and (ADBF) gives movement to the left. That is, these schemes move the initial condition up with the 'wind' speed; right if $c > 0$ and left if $c < 0$. We see that (ADCC) 'leap frogs' over the grid point (x_j, t_n) . We also note that in order to use (ADCC) we must start with $n = 1$. Starting at $n = 1$ requires that we know $w(x_j, t_1)$ for $j = 1, \dots, J - 1$. We obtain $w(x_j, t_1)$ for $j = 1, \dots, J - 1$ using either (ADFF) or (ADBF). Since in (ADCC) we see that we need both $w(x_j, 0)$ and $w(x_j, t_1)$ for $j = 1, \dots, J - 1$ in order to get $w(x_j, t_2)$ for $j = 1, \dots, J - 1$, we write $w(x_j, 0)$ and $w(x_j, t_1)$ for $j = 1, \dots, J - 1$ to a file we obtain the next time step by replacing $w(x_j, 0)$ with $w(x_j, t_1)$ for $j = 1, \dots, J - 1$ and $w(x_j, t_1)$ with $w(x_j, t_2)$ for $j = 1, \dots, J - 1$. We then continue this process in a manner similar as we did for (ADFF) and (ADBF).



The stencils for the Upwind and Leap Frog Schemes

Note that if $\frac{c\Delta t}{\Delta x} = 1$ and $d = 0$ in (ADBF) we have

$$w(x_j, t_{n+1}) = w(x_{j-1}, t_n).$$

That is, the value of w at the point (x_j, t_{n+1}) is equal to the value of w at a point Δx back in space and a point Δt back in time. Since $c = \frac{\Delta x}{\Delta t}$ in this case, we see that the values of w are moving forward in time at the speed c . Therefore, this scheme gives the exact solution since we know that this is what happens in the PDE when $d = 0$. We leave it for you to show that a similar phenomenon occurs in (ADFF) under these same conditions. You can test this in the modules. You should then test what happens if $\frac{c\Delta t}{\Delta x} < 1$ and then what happens if $\frac{c\Delta t}{\Delta x} > 1$. The phenomenon you will observe is called the Courant-Friedrichs-Lewy condition.

Burger's Equation

We now proceed to the viscous Burger equations (B1) and (B2). We first consider the leap frog schemes. That is, we use centered differences on the equations

$$(B1) \quad u_t + uu_x - \nu u_{xx} = G$$

and

$$(B2) \quad u_t + \left(\frac{1}{2}u^2\right)_x - \nu u_{xx} = G.$$

We will only do these equations for $\nu = 0$ and $G = 0$. We leave the viscous case for you. For (B1) we have

$$(B1LF) \quad u(x_j, t_{n+1}) = u(x_j, t_{n-1}) - \left(\frac{\Delta t}{\Delta x}\right)u(x_j, t_n)(u(x_{j+1}, t_n) - u(x_{j-1}, t_n))$$

$$\text{for } j = 1, \dots, J - 1.$$

$$(B2LF) \quad u(x_j, t_{n+1}) = u(x_j, t_{n-1}) - \left(\frac{\Delta t}{\Delta x}\right)(u(x_{j+1}, t_n)^2/2 - u(x_{j-1}, t_n)^2/2)$$

$$\text{for } j = 1, \dots, J - 1.$$

The scheme (B1LF) depends on the differentiated (non-conservation) form of the Burger's equation and (B2LF) depends on the undifferentiated (conservation) form of the Burger's equation. If our initial condition has a steep slope the derivative will be large and we expect larger errors. We should check for this in the Burger's equation modules.

The last numerical approximation we consider for the solutions to our PDE's is the Lax-Wendroff scheme. In the Lax-Wendroff scheme we look at the Taylor series of $w(x, t)$ in t . We have

$$w(x, t + \Delta t) = w(x, t) + \Delta t w_t(x, t) + \frac{1}{2} \Delta t^2 w_{tt}(x, t) +$$

$$\frac{1}{3!} \Delta t^3 w_{ttt}(x, t) + \frac{1}{4!} \Delta t^4 w_{tttt}(x, t) + \dots$$

If we truncate after the $w_t(x, t)$ term we call the approximation the first order Lax-Wendroff scheme, if we truncate after the $w_{tt}(x, t)$ term we call the approximation the second order Lax-Wendroff scheme and so on. We replace the t derivatives by the PDE. For example, if we apply the second order Lax-Wendroff scheme to (B2) (with $\nu = 0$ and $G = 0$) we have

$$u_t = -\left(\frac{1}{2}u^2\right)_x$$

and

$$u_{tt} = \left(-\left(\frac{1}{2}u^2\right)_x\right)_t = -\left(\left(\frac{1}{2}u^2\right)_t\right)_x.$$

In the last equation we assumed that we could switch the order of differentiation. Carrying out the differentiation we have

$$u_{tt} = -\left(\left(\frac{1}{2}u^2\right)_t\right)_x = -(uu_t)_x.$$

Now replacing u_t by the PDE we have

$$u_{tt} = -(uu_t)_x = \left(u\left(\frac{1}{2}u^2\right)_x\right)_x.$$

We now have our second order Lax-Wendroff approximation

$$\begin{aligned} u(x, t + \Delta t) &= u(x, t) + u_t(x, t)\Delta t + \frac{1}{2}u_{tt}(x, t)\Delta t^2 = \\ &= u(x, t) - \left(\frac{1}{2}u^2\right)_x\Delta t + \frac{1}{2}\left(u\left(\frac{1}{2}u^2\right)_x\right)_x\Delta t^2. \end{aligned}$$

We now replace the outside x derivatives by centered differences in the following way

$$\begin{aligned} \text{(B2LW)} \quad u(x_j, t + \Delta t) &= u(x_j, t) - \left(\frac{\Delta t}{2\Delta x}\right)(u(x_{j+1}, t)^2/2 - u(x_{j-1}, t)^2/2) + \\ &\quad \frac{1}{2}\frac{\Delta t^2}{\Delta x}(u(x_{j+1/2}, t)\left(\frac{1}{2}u(x_{j+1/2}, t)\right)_x - u(x_{j-1/2}, t)\left(\frac{1}{2}u(x_{j-1/2}, t)\right)_x). \end{aligned}$$

Note that we used centered differences on the last term with Δx at the points $(x_{j-1/2}, t)$ and $(x_{j+1/2}, t)$. We now replace

$$u(x_{j+1/2}, t) \text{ and } u(x_{j-1/2}, t)$$

by

$$u(x_{j-1/2}, t) \approx (u(x_j, t) + u(x_{j-1}, t))/2 \text{ and } u(x_{j+1/2}, t) \approx (u(x_{j+1}, t) + u(x_j, t))/2.$$

That is, we approximate these values by the average value of adjacent grid points. We then do centered differences on $u(x_{j-1/2}, t)_x^2$ and $u(x_{j+1/2}, t)_x^2$ to get

$$u(x_{j-1/2}, t)_x^2 \approx \frac{(u(x_j, t)^2 - u(x_{j-1}, t)^2)}{\Delta x} \text{ and } u(x_{j+1/2}, t)_x^2 \approx \frac{(u(x_{j+1}, t)^2 - u(x_j, t)^2)}{\Delta x}.$$

Substituting these into (B2LW) gives us

$$(B2LW)$$

$$u(x_j, t_{n+1}) = u(x_j, t_n) - \left(\frac{\Delta t}{\Delta x}\right)(u(x_{j+1}, t_n)^2/2 - u(x_{j-1}, t_n)^2/2) + \frac{1}{2} \frac{\Delta t^2}{\Delta x} [(u(x_{j+1}, t_n) + u(x_j, t_n))/2] \left(\frac{1}{2} \frac{(u(x_{j+1}, t_n)^2 - u(x_j, t_n)^2)}{\Delta x}\right) - [(u(x_j, t_n) + u(x_{j-1}, t_n))/2] \left(\frac{1}{2} \frac{(u(x_j, t_n)^2 - u(x_{j-1}, t_n)^2)}{\Delta x}\right).$$

We leave it as an exercise for you to apply Lax-Wendroff to (B1).

We use leap frog and Lax-Wendroff on (E3) or (E4) and (E5). These are done in the software for the 1D Density Velocity Equations and 1D Euler Equations, respectively. To apply leap frog or Lax-Wendroff to a system of PDE's you apply it to each equation in the system. For example, consider the numerical solutions to (E4) (with $g = 0$)

$$\frac{\partial}{\partial t} m + \frac{\partial}{\partial x} \left(\frac{m^2}{w} + c^2 w \right) = 0$$

$$\frac{\partial}{\partial t} w + \frac{\partial}{\partial x} m = 0$$

using leap frog. (We have made the substitutions $w = \frac{p}{c^2}$ and $m = uw$ in (E4).) Applying centered differences to the derivatives we obtain

$$\frac{m(x_j, t_{n+1}) - m(x_j, t_{n-1})}{2\Delta t} = - \frac{\frac{m(x_{j+1}, t_n)^2}{w(x_{j+1}, t_n)} - \frac{m(x_{j-1}, t_n)^2}{w(x_{j-1}, t_n)} + c(x_{j+1})^2 w(x_{j+1}, t_n) - c(x_{j-1})^2 w(x_{j-1}, t_n)}{2\Delta x}$$

$$\frac{w(x_j, t_{n+1}) - w(x_j, t_{n-1})}{2\Delta t} = - \frac{m(x_{j+1}, t_n) - m(x_{j-1}, t_n)}{2\Delta x}.$$

Of course, in the first equation we solve for $m(x_j, t_{n+1})$ and in the second equation we solve for $w(x_j, t_{n+1})$. Notice that we must solve for both of these for $j = 1, \dots, J - 1$ before we go on to solve for $m(x_j, t_{n+2})$ and $w(x_j, t_{n+2})$. To get the pressure we use $p(x_j, t_{n+1}) = c(x_j)^2 w(x_j, t_{n+1})$ and to get the fluid velocity we use $u(x_j, t_{n+1}) = m(x_j, t_{n+1})/w(x_j, t_{n+1})$.

The 1D Linear Pressure Equation

We now numerically solve the 1D linear pressure equation (LP). We use centered differences in time and space to get

$$\frac{p(x_j, t_{n+1}) - 2p(x_j, t_n) + p(x_j, t_{n-1}))}{\Delta t^2} = c^2 \frac{p(x_{j+1}, t_n) - 2p(x_j, t_n) - p(x_{j-1}, t_n)}{\Delta x^2}.$$

Solving this equation for $p(x_j, t_{n+1})$ we have

$$(NLP)$$

$$p(x_j, t_{n+1}) = 2p(x_j, t_n) - p(x_j, t_{n-1}) + c^2 \frac{\Delta t^2}{\Delta x^2} (p(x_{j+1}, t_n) - 2p(x_j, t_n) - p(x_{j-1}, t_n)).$$

Note that in order to obtain $p(x_j, t_2)$ for $j = 1, \dots, J - 1$ we must substitute $n = 1$ on the right hand side of the above equation. Doing this we see that we need both $p(x_j, t_1)$ and $p(x_j, t_0)$. Since $t_0 = 0$, we get $p(x_j, t_0)$ from the initial condition $p(x, 0)$. To get $p(x_j, t_1)$ we require that $p_t(x_j, 0)$ be given. Suppose $p(x, 0) = p_0(x)$ and $p_t(x, 0) = q_0(x)$ for given $p_0(x)$ and $q_0(x)$. Then we have $p(x_j, 0)$ from $p_0(x)$ and we obtain $p(x_j, t_1)$ from a forward difference in time approximation on $p_t(x_j, 0)$. That is,

$$\frac{p(x_j, t_1) - p(x_j, 0)}{\Delta t} = q_0(x_j)$$

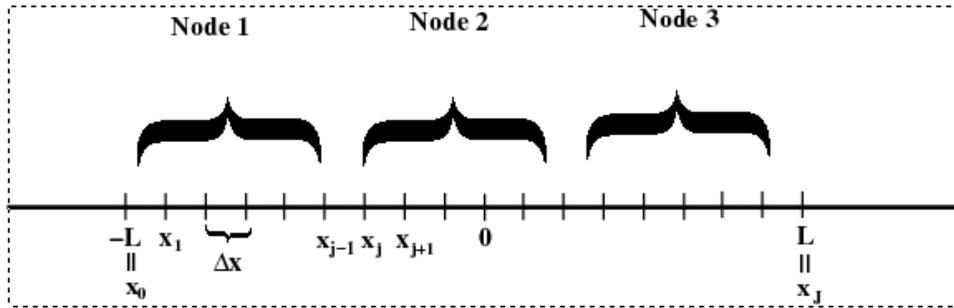
so that

$$p(x_j, t_1) = p(x_j, 0) + q_0(x_j)\Delta t.$$

Now that we have $p(x_j, 0)$ and a numerical approximation for $p(x_j, t_1)$ we can use (NLP) to get a numerical approximation for $p(x_j, t_2)$ for $j = 1, \dots, J - 1$. We then output $p(x_j, 0)$ for $j = 1, \dots, J - 1$ to a file and replace $p(x_j, 0)$ with $p(x_j, t_1)$ and replace $p(x_j, t_1)$ with $p(x_j, t_2)$ and then use (NLP) with $n = 1$ to get $p(x_j, t_3)$ for $j = 1, \dots, J - 1$. We output $p(x_j, t_1)$ for $j = 1, \dots, J - 1$ to a file and continue the process as usual.

Parallelization

The last topic we cover is that of parallelization. Note that since our schemes are explicit we can solve for $w(x_1, t_{n+1})$ and $w(x_2, t_{n+1})$ at the same time. Therefore, it makes sense to divide this work among several computers. The Center for Computational Mathematics and Modeling (CCMM) at JMU houses a 16 node computer for this purpose. One can divide the grid points among the 16 nodes as shown in the next figure. This process of dividing a program among several nodes of a computer is called parallelizing a code. Click on the link [Parallel Computing Document](#) to learn more about parallel computing. This document will also give information on two software modules that contain parallel codes for (AD) and (LP). We use MPI (Message Passing Interface) coupled with Fortran 90 in these parallel computing software modules. The software modules are for the (AD) equation with no diffusion using (ADFF) or (ADBF) and the linearized pressure wave equation (LP) using (NLP). We use the initial condition e^{-x^2} in these modules. This initial condition is a bell shaped curve and you will see the bell propagate in each module. You can experiment with Δt and Δx to test the stability of (ADFF) , (ADBF) and (NLP). We leave it as a team exercise for you to parallelize the leap frog and Lax-Wendroff schemes for (E3) after going through the parallel computing document and modules.



A schematic showing how to divide the grid points among the nodes of a parallel computer

We recommend that you have this document handy as you go through the modules so that you can refer back to the appropriate equations and numerical schemes for each module. The Maple routines have a lot of the mathematical manipulations to get the equations and schemes in them, but that they are slower computationally. The Matlab routines allow movie output choices for visualizing the fluid 'flow'. The Fortran 90 routines are faster computationally, but the data sets have to be loaded into Maple or Matlab for viewing. Maple and Matlab routines for viewing the data sets are with the Fortran software modules.