CPSC 520 - Numerical Solution of Differential Equations
ADI Cloth

Eddy Boxerman

June 1, 2003

Abstract

The ideal cloth simulation technique would be general and provide accurate and stable results in $O(n)$ time. This paper investigates the viability of using an ADI (Alternating Direction Implicit) method to achieve this ideal. I investigate and apply several ADI techniques to a mass-spring formulation for cloth. En route, I investigate and expose some relationships between mass-spring systems and the PDEs of elasticity theory, opening the door to the rich set of PDE analysis tools for mass-spring systems.

1 Introduction

Predicting the motion, or static drape, of cloth is of interest in garment design, textile engineering and computer graphics. Computer simulation can be of use in solving this complex problem. It would alleviate the need for manufacturing fabric and garment prototypes. It is already of use in relieving computer animators from the burden of "hand" animating the motion of cloth. It is also of use in virtual reality settings, such as computer games.

This problem has already received considerable attention from mechanical and textile engineers, as well as computer scientists. Whereas engineers and garment designers are interested in the realistic modeling of fabrics, computer scientists tend to place more emphasis on computational speed - as long as it looks good.

There are two dominant techniques for modeling cloth:

- Finite element models, which divide the body into a set of elements, and seek to find approximations to functions which satisfy deformation equilibrium equations between the elements; continuity of the function is enforced between elements.

- Mass-spring models, which divide the body into a set of point masses connected by springs which resist deformation of the structure; deformation continuity is not enforced.

There are however many approaches within each technique (and many opinions as to which is better).

Finite element models are generally more accurate, and computationally more expensive, than mass-spring models (due to the finer mesh sizes required). As a result, they see more use in the engineering literature than in computer graphics. Various element types have been employed, including plates, shells and beams. Collier et al. [5] present a GNFEM (geometric nonlinear finite element method) approach using plate elements. Ascough et al. [1] employ beam elements to improve processing time at the cost of accuracy. Tan et al. [14] introduce geometric constraints to a thin plate element model, assuming that the lengths of the threads in a fabric remain unchanged after deformation. Eishen and Bigliani (chapter 4 of [7]) use a "geometrically exact resultant shell theory", including a nonlinear stress-strain relationship based on real fabric measurements.

A benefit of mass-spring techniques is they do not require as fine a mesh resolution as finite element techniques to exhibit folding and wrinkling behaviour (a drawback of FEM's continuity enforcement). Another benefit is their ease of implementation. Mass-spring techniques were proposed for cloth simulation by Breen et al [3]. Since then, others in the computer graphics community have followed suit [10] [6] [4]. The main issues involved in this technique are numerical stability - ie. how large a time-step can one take - and accuracy, how well does this simulate a real fabric. More on this in section 2.

There are, of course, important approches which do not fall into either category. Terzopoulos et al. [13] uses an "approximate continuum model", which applies simplifying assumptions (or analogous formulations) to elasticity theory and proceeds to numerically solve the discretized equations. Baraff and Witkin [2], who introduced implicit techniques for cloth, used a formulation that has elements from both continuum *and* mass-spring systems.

Despite all these approches, be they FEM or mass-spring, explicit or implicit, the "holy grail" of cloth simulation has yet to be achieved: namely, an accurate, general, stable technique that can be computed in $O(n)$ time (where n is the number of nodes, or particles, in the mesh). It must be said however, that various authors have achieved three of these four requirements.

The motivating force behind this paper is to attain that holy grail (although I haven't achieved it, yet! ;-). The idea behind this paper is the following: cloth is a 2D surface moving in 3D space; there exist techniques to solve 2D PDEs in $O(n)$ time that are (numerically) unconditionally stable; perhaps they can be applied to cloth. Towards this end, I investigate the relationship between elasticity theory PDEs and mass-spring systems; I then formulate, analyze and experiment with various ADI schemes.

2 Cloth as a Mass-Spring System

The dominant method for modeling cloth in the computer graphics community is to represent it as a system of masses connected by springs. The material's structure is discretized into a mesh of point masses; the material's resistance to deformation is modeled by springs connecting masses to their neighbours in the mesh. The dynamics of the system is then simulated in time according to Newton's second law, $f = ma$.

The question arises: is a mass-spring system a reasonable model for cloth? ie. Can we simulate realistic cloth behaviour with it? Considerable thought and effort has gone into answering this question. Breen et al. [3] argued that cloth is not a continuous substance; its macroscopic behaviour results from small-scale yarn interactions. Conceptually, the yarn crossing points can be modeled as particles, and the mechanical interactions between yarns can be modeled using springs. Making use of this model, along with experimental data obtained from the Kawabata Evalutation System (standardized fabric measurement equipment and tests), they were able to reproduce the drape of specific materials accurately. Fortuitously, the mesh resolution required to reasonably simulate cloth is much coarser (several orders of magnitude) than the actual scale of typical cloth weaves.

2.1 Mass-spring structure

Figure 1 shows a typical connectivity structure of a mass-spring system for modeling cloth. Here we see only the springs connected to the centre mass. The springs which connect the mass to its nearest horizontal and vertical neighbours (densest lines) are *stretch* springs. As the name implies, these resist in-plane (along the surface) stretching of the material. The diagonal springs are *shear* springs; these resist in-plane shearing. The longer springs (dashed lines) connect the mass to its two-away neighbours; these resist bending out of the plane (wrinkling, folding, waving). There are variations in the literature on this model, but the fundamentals remain the same.

Most fabrics have a high resistance to stretch as compared to bending. As a result, the stiffness of the stretch springs must be much larger than the stiffness of the bend springs (generally from five to seven orders of magnitude!). The stiffness of the shear springs lies somewhere in between the two (but generally closer to the stretch springs than the bending springs). A fabric's resistance to shear and bending are the primary deciding factors in its drape characteristics. Stretch and shear are usually modeled using linear springs (constant stiffness); however, due to the large bending deformation that occurs, bend springs are often modeled as nonlinear. Second to fifth order polynomials fitted to experimental data are typically used.

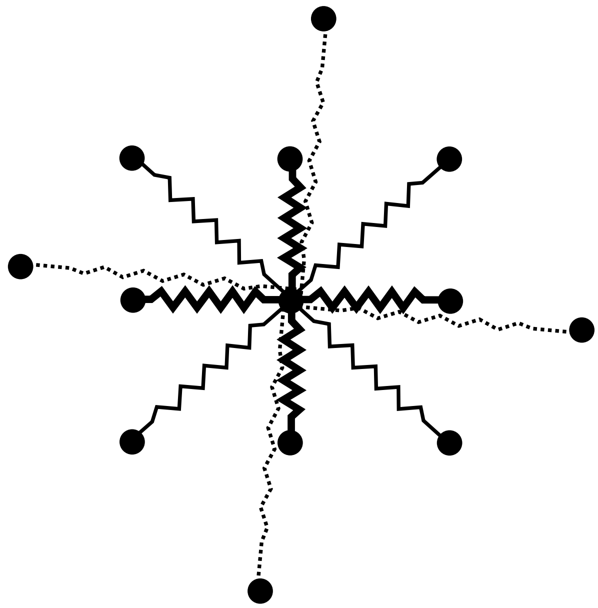

Figure 1: Typical connectivity of a cloth mass-spring system

2.2 Numerical Simulation

The dynamics of the cloth's motion is goverend by Newton's second law $f = ma$. For the collection of particles in our cloth mesh, this can be formulated as a system of ODEs:

$$\ddot{x} = \mathbf{M^{-1}f(x, \dot{x})} \tag{1}$$

Here, x is a vector representing the current position of the various particles (in say, lexicographic ordering) of the mesh; M is the square, (normally) diagonal mass matrix of the system; and f is the vector of forces acting on the particles. For a piece of fabric modeled as a collection of n particles, the size of the vectors is $3n$. It is convenient to think of $f(x, \dot{x})$ as the sum of two types of forces: external forces (gravity, friction, collision response) and internal forces (due to deformation).

When simulating Newtonian mechanics, it is convenient to define $v = \dot{x}$. This allows us to transform (1) into a first order system

$$\frac{d}{dt}\begin{bmatrix} x \\ v \end{bmatrix} = \begin{bmatrix} v \\ M^{-1}f(x, v) \end{bmatrix}$$

Discretizing in time, the simplest choice is to use an explicit technique such as forward Euler, which gives, for the current time-step n,

$$\left[\begin{array}{c} \Delta x \\ \Delta v \end{array}\right]_n = \left[\begin{array}{c} x^{n+1} - x^n \\ v^{n+1} - v^n \end{array}\right] = k \left[\begin{array}{c} v^n \\ M^{-1} f(x^n, v^n) \end{array}\right] \tag{2}$$

where x^n and v^n are the system's position and velocity vectors at time n (this is not a power), and k is the time-step.

However, this is not an ideal scheme for the current problem. There is a huge difference between the stiffnesses of the stretch and the bending springs in the cloth model. This results in a "stiff" differential equation. Moreover, we are not interested in visualising the high-frequency, in-plane oscillations of the particles, but rather the out-of-plane, low-frequency, bending behaviour. Forward Euler, and explicit schemes in general, require excessively small time-steps when faced with these types of problems.

In 1998, Baraff and Witkin [2] published a paper using an implicit formulation for cloth simulation.[1] This technique allowed for the stable use of large time-steps in the simulation. (2) becomes

$$\left[\begin{array}{c} \Delta x \\ \Delta v \end{array}\right]^n = k \left[\begin{array}{c} v^n + \Delta v^n \\ M^{-1} f(x^n + \Delta x^n, v^n + \Delta v^n) \end{array}\right] \tag{3}$$

which is a nonlinear equation in Δx and Δv. They approximately solve this equation using a Taylor series expansion of f

$$f(x^n + \Delta x^n, v^n + \Delta v^n) = f^n + \frac{\partial f}{\partial x}\Delta x + \frac{\partial f}{\partial v}\Delta v.$$

where $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial v}$ are the Jacobians of the particle forces with respect to position and velocity respectively. Baraff and Witkin develop expressions for the Jacobians of the various internal spring forces in their paper. Due to the local connectivity structure of the mesh, these are sparse matrices. Substituting this in (3) and rearranging,

$$(I - kM^{-1}\frac{\partial f}{\partial v} - k^2 M^{-1}\frac{\partial f}{\partial x})\Delta v = kM^{-1}(f^n + k\frac{\partial f}{\partial x}v^n) \tag{4}$$

[1]The model presented here has more in common with Choi and Ko's [4] formulation than Baraff and Witkin's, who used more of a continuum model than a mass-spring one.

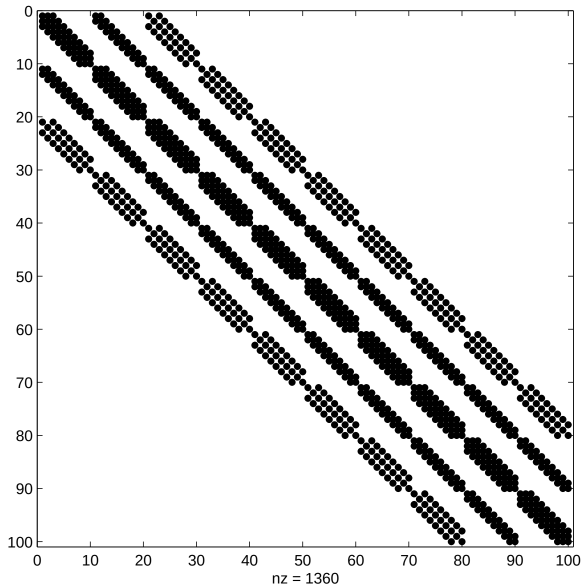

Figure 2: Sparseness structure of LHS of (4)

The matrix on the left-hand side of this equation is sparse and (via some simplifying assumptions on the authors' part), positive definite. The sparsity structure of this matrix can be seen in figure 2 for a 10 by 10 mesh; each point represents a 3x3 matrix. They then proceed to solve this equation at each time-step using a modified conjugate-gradient algorithm. The reported cost of solving this equation is $O(n^{1.5})$, where n is the number of particles in the system.

Since Baraff and Witkin's paper, others have followed this approach. Choi and Ko [4] used a different compression and bending formulation to overcome "post-buckling instability" - popping of the membrane due to excessive compression resitance (as seen in sheet-metal or paper). Desbrun et al. [6] make use of an approximate implicit formulation to achieve an $O(n)$, "unconditionally stable" scheme; they pre-invert the matrix (for the cloth's rest configuration) and use this solution at every time-step, applying a post-correction factor for excessive deformation and global rotational momentum. Their technique, however, is neither general (as they admit) nor accurate.

2.3 IMEX Schemes

How can we improve upon existing methods? Let's begin by looking at the implicit formulation. IMEX schemes combine implicit and explicit schemes together, treating stiff components of the formualtion implicitly and non-stiff ones explicitly. Applying this idea to cloth simulation, let's

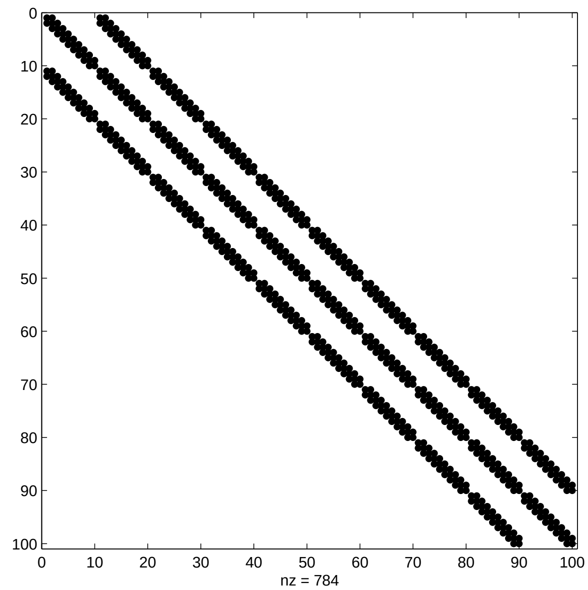

Figure 3: Sparseness structure of LHS of (4), without bend springs

divide the force terms into two categories

$$f = f_{stiff} + f_{non-stiff}$$

Instead grouping all spring forces into the first category, let's move the bend springs to the second. This eliminates the need to calculate their contribution to the jacobians, and also increases the sparsity of the matrix as seen in figure 3. Having made this simple change to a java implementation of the solver, performance more than doubled. For all practical purposes, this method is also stable for large time-steps (we are, after all, interested in the frequency of the solution of the cloth bending).

We can do better if the shear springs are also non-stiff (a valid assumption for some fabrics). In this case the matrix becomes even sparser, as in figure 4, and performance further improves. The method now has a time-step restriction; but this may not be overly significant for low-shear fabrics, especially if relatively small time-steps are already required (say, for collision handling).

At this point, the matrix is nearly (block) tridiagonal. If it were, this could be solved in $O(n)$ time. To make that final leap, while still preserving stability, we may try to use a technique from the PDE community known as an *ADI - alternating direction implicit* method.

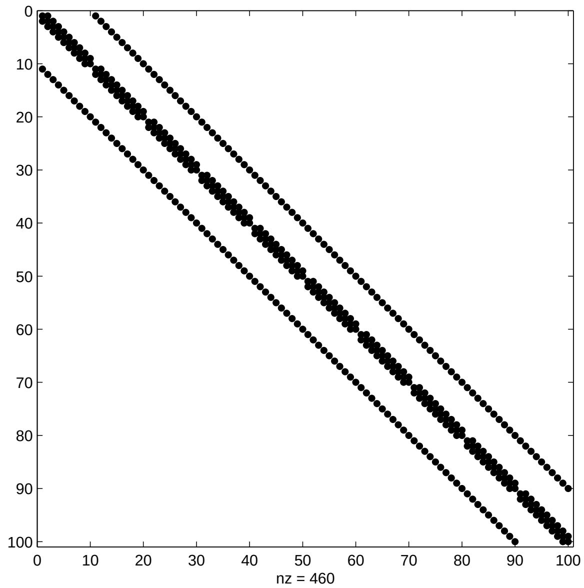

Figure 4: Sparseness structure of LHS of (4), without shear or bend springs

3 Mass-spring vs. PDE formulations

Before continuing, let's take a brief look at PDE formulations for cloth modeling.

3.1 Elasticity Theory

Elasticity theory is the study of the deformation of elastic continua [11]; it has a long and rich history. An investigation into the details of how cloth can be modeled using these formulations is beyond the scope of this project. For now, I will content myself with simply investigating the *nature* of the formulations.

The most general equation (in Euclidean coordinates) for the dynamic behaviour of a three-dimensional elastic solid (from [8]) is

$$L^T c L u + b = \rho \ddot{u} \tag{5}$$

where u is the three-component displacement vector, b is the vector of body forces, L is the differential operator matrix

$$L = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{bmatrix}$$

and c is a symmetric 6x6 matrix of material coefficients.

Various simplified formulations of this equation exist for 2D (plane stress, plane strain) and 1D (truss, beam) problems. For cloth, however, curvilinear coordinates are required.

Researchers who simulate cloth using finite element techniques *must* be using an underlying PDE for their solution, but I have been as yet unable to find (or understand) one. Possibly the most general approach in this category is that by Eischen and Bigliani (chapter 4 of [7]). Though they don't explicitly formulate the PDEs, they do mention that five nonlinear partial differential equations must be solved at each point of the midsurface of the fabric. Fourth order tensors are also involved. In any case, from various articles, it seems these finite element techniques are generally expressed in the following form:

$$[M]\ddot{u} + [C]\dot{u} + [K_E + K_G(P)]u = \{P\}$$

where $[M]$ and $[C]$ are the mass and damping matrices, u are the nodal displacements, $\{P\}$ is the vector of external forces, and K_E and $K_G(P)$ are the elastic and geometric stiffness matrices.

Researchers who use mass-spring formulations usually dismiss discussion of the underlying PDE by stating that the particle system can be seen as a semi-discretization in space of the original problem.

3.2 Elasticity theory and mass-spring systems

What more can be said about the relationship between the PDEs of elasticity theory and mass-spring systems? Well, looking at (5), we have a second order dervative in time and various mixed second order derivatives in space: it's a wave equation in three unknowns.

Approaching the cloth at the fibre level, we see the material as a woven structure of (nearly) one-dimensional objects; (at rest) these are a set of (parallel) warp and a set of (parallel) weft fibres which have been woven (perpendicualrly) together. Ignoring twist, each of these fibres may undergo one dimension of axial deformation and two dimensions of bending. The positions

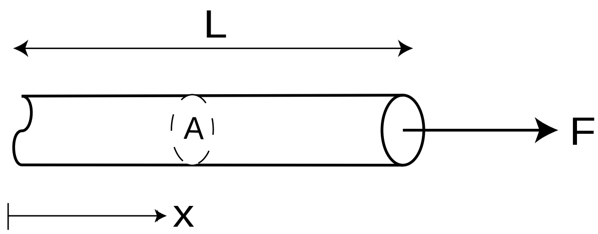

Figure 5: A member undergoing axial deformation

of these fibres are coupled together at the various crossing points of the weave. As a starting point, let's isolate the axial deformation of a single thread.

The physical scenario is depicted in 5. We have a truss (or fibre) which is constrained to deform in the axial direction. It has the following material properties:

- constant cross-sectional area A
- Young's modulus E
- mass per unit length ρ
- length L

It is also acted upon by gravity g in the positive x direction and a force F at the right boundary.

From [8], the PDE describing the deformation of the truss is

$$\frac{\partial}{\partial x}(EA\frac{\partial u}{\partial x}) + b_x = \rho\ddot{u} \tag{6}$$

Discretizing the second derivatives in space and time using second order difference operators (with mesh spacing h and k respectively), we obtain the explicit scheme

$$v_j^{n+1} = 2v_j^n - v_j^{n-1} + \alpha^2\mu^2 D_+D_-v_j^n + k^2 g \tag{7}$$

where $\alpha^2 = \frac{EA}{\rho}$, $\mu = \frac{k}{h}$, and $g = \frac{b_x}{\rho}$ is the acceleration due to gravity.

A reasonable mass-spring analog to this discretization (see figure 6) would be to use:

- the same mesh spacing; the rest length of the springs should be h
- the same density; each mass "represents" its neighbourhood, so $m = \rho h$ ($\frac{\rho h}{2}$ at the boundaries)

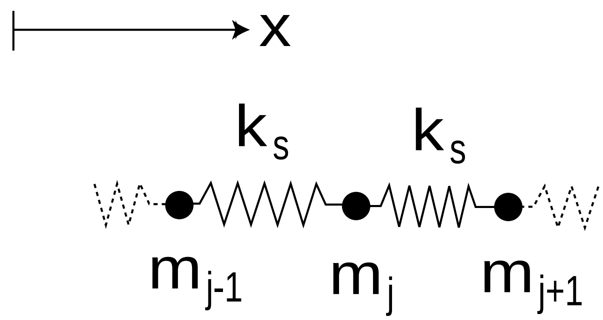

Figure 6: Zoomed in on the corresponding 1D mass-spring scenario

- spring constant k_s; it is not yet clear how to set this

Now a change of variables is in order for (7). For this simple case, the deformation u (or v) is the difference between the mesh point's current and original positions, given in body coordinates. Taking the truss member's left boundary as the body's origin, the relationship between the position x of a mesh point and its displacement u is given by

$$x_j = \sum_{i=1}^{j} h + v_j = jh + v_j \tag{8}$$

Applying this change of variables (from v to x) to (7) and canceling jh terms, we obtain

$$x_j^{n+1} = 2x_j^n - x_j^{n-1} + \alpha^2\mu^2(x_j^{n+1} - 2x_j^n + x_j^{n-1}) + k^2 g \tag{9}$$

We have yet to make this look like (2). How can $f(x_n, v_n)$ be expressed in terms of x_j? Looking at the mass-spring analog (figure 6), what is the force applied to m_j due to its neighbours? The standard equation relating displacement to force for a linear spring is $f = k_s \Delta x$. Using the mesh spacing h as the rest length between springs, the force on m_j at time-step n is

$$\begin{aligned} f_j^n &= k_s(x_j^{n+1} - x_j^n - h - (x_j^n - x_j^{n-1} - h)) \\ \frac{f_j^n}{k_s} &= (x_j^{n+1} - 2x_j^n + x_j^{n-1}) \end{aligned}$$

Substituting this into (9), we obtain

$$x_j^{n+1} = 2x_j^n - x_j^{n-1} + \alpha^2\mu^2\frac{f_j^n}{k_s} + k^2 g \tag{10}$$

Defining $(x_{n+1} - x_n) = kv_{n+1}$ and dividing through by k, we can write this as a first order system similar to (2)

$$\begin{bmatrix} x^{n+1} - x^n \\ v^{n+1} - v^n \end{bmatrix} = k \begin{bmatrix} v^{n+1} \\ \alpha^2\frac{f^n}{h^2 k_s} \end{bmatrix} \tag{11}$$

(Note that the $k^2 g$ term in (10) is simply another acceleration term and is already included in the $M^{-1}f^n$ in (2).) Comparing (2) and (11), we see a difference in the update formula; in the latter, x^n is updated using v^{n+1} instead of v^n. This is known as a *Symplectic* Euler scheme. This is still an explicit scheme; we can first solve for v^{n+1} and then for x^{n+1}. Symplectic schemes have "energy-preserving" properties and are appropriate for modeling oscillatory solutions (such as for the wave equation!).

So we see that a 2nd order discretization of the wave equation gives a formulation identical to a symplectic Euler method applied to a mass-spring system. We can now equate constants between the two problems. From (11) and (2), we have (for a single particle)

$$m^{-1} = \frac{\alpha^2}{h^2 k_s} = \frac{EA}{\rho h^2 k_s}$$

Using the relation $m = \rho h$, this reduces to

$$k_s = \frac{EA}{h} \tag{12}$$

We now have a formula for k_s that allows for a consistent modeling of the physical problem.

Finally, boundary conditions must be considered. For our example, let's apply a Dirichlet condition ($x_0 = v_0 = 0$) at the left boundary, and a von Neumann condition (which amounts to specifying the force) at the right boundary. The dirichlet condition is trivial to apply; more is required for the von Neumann condition. For the PDE, we have (from [8]) the relation $EAu_x(L) = F$, which after discretization becomes $\frac{EA}{2h}(v_{L+1} - v_{L-1}) = F$. The update scheme for v_L^{n+1} thus becomes

$$v_L^{n+1} = 2v_L^n - v_L^{n-1} + \mu^2\alpha^2(-2v_L^n + 2v_{L-1}^n) + \frac{2k^2F}{h\rho}.$$

For the mass-spring system, the external force F is simply added to the appropriate term in the force vector.

I have also verified a similar transformation (from PDE-discretization to mass-spring system) for the implicit scheme. Details are not included here. The implicit formulations are also identical. I have not yet undertaken this anaylsis in two dimensions. However, it should be noted that the two dimensional wave equation

$$u_{tt} = \alpha^2(u_{xx} + u_{yy})$$

will not suffice; we must deal with two dimensions of displacement (u and v) which are coupled and must be analyzed in curvilinear coordinates (due to shearing). Analysis of the one dimensional fibre with bending (in two or three dimensional space) could prove enlightening as well.

4 PDE Stability

Now that we have a relationship (albeit a primitive one) between mass-spring systems and PDE discretizations, we can begin applying the powerful stability anylsis tools from numerical PDE theory.

4.1 Stability of the explicit scheme

What is the stability condition for the explicit scheme (7), rewritten in slightly different form here

$$v_j^{n+1} = (2 + \alpha^2\mu^2 D_+ D_-)v_j^n - v_j^{n-1}$$

(Note: the gk^2 has been dropped for the stability analysis. It does not affect the result so long as we are willing to accept a growth on the order of $1 + O(k^2)$. This is polynomial as opposed to exponential growth, which is the real concern.) Applying a Fourier transform in space (x) to this equation, this becomes

$$\hat{v}(t + k, \omega) = (2 - \beta^2)\hat{v}(t, \omega) - \hat{v}(t - k, \omega)$$

where ω is a frequency, or mode, in the data, $\beta^2 = 4\alpha^2\mu^2 \sin^2 \frac{\zeta}{2}$ and $\zeta = \omega h$. Following Strikwerda's notation [12], the amplification polynomial for this scheme is

$$\Phi(g, \zeta) = g^2 + (\beta^2 - 2)g + 1$$

and we demand that the magnitude of all roots of this polynomial satisfy $|g_{root_i}| \leq 1$. Applying the quadratic equation, the roots are

$$g_{\pm} = \frac{1}{2}(2 - \beta^2 \pm \sqrt{\beta^4 - 4\beta^2})$$

requiring that $\beta^4 - 4\beta^2 < 0$ (the reverse leads to roots that violate our criterion), this becomes

$$g_{\pm} = \frac{1}{2}(2 - \beta^2 \pm \imath\beta\sqrt{4 - \beta^2})$$

the magnitude of which is

$$|g| = \sqrt{\frac{1}{4}(4 - 4\beta^2 + \beta^4 + 4\beta^2 - \beta^4)} = 1$$

And so the criteria for stability is $\beta^2 < 4$ or $\alpha\mu < 1$.

This result can now be applied to the symplectic mass-spring formulation. Making use of the definitions of our various constants, along with (12), we have

$$\alpha\mu = \sqrt{\frac{EAk^2}{\rho h^2}} = \sqrt{\frac{k_s k^2}{\rho h}} = \sqrt{\frac{k_s k^2}{m}} < 1$$

This gives us the stability requirement for a 1D mass-spring system

$$k < \sqrt{\frac{m}{k_s}}$$

There are a few points of note here. At first glance, the time-step does not appear to be restricted by the mesh spacing h. But k_s *is* dependent on h. From (12), we have $k_s h = EA$. If we wish to halve the mesh spacing of a simulation ($h = \frac{h}{2}$), we must double the value of k_s and halve the value of m in order to have a solution consistent with the physical problem. This in turn reduces our maximum time-step ($k = \frac{k}{2}$).

4.2 Stability of the implicit scheme

The implicit scheme

$$(1 - \alpha^2 \mu^2 D_+ D_-) v_j^{n+1} = 2v_j^n - v_j^{n-1} \tag{13}$$

is unconditionally stable. The proof is similar to the explicit one. And so the 1D implicit mass-spring formulation is unconditionally stable as well.

5 The ADI Method and Cloth

In one dimension, an implicit scheme such as (13) results in a tridiagonal system that can be solved in $O(n)$ time. In two dimensions, matters are not so simple; a sparse linear system must be solved at each time-step. However, in the 1950's, ADI (alternating direction implicit) methods were proposed by Peaceman and Rachford [9] that allowed for the solution of PDEs such as

$$u_t = \alpha^2 (u_{xx} + u_{yy}) \tag{14}$$

in $O(n)$ time with unconditional stability. The basic idea is to divide the implicit formulation into two stages: handling only one dimension implicitly at a time.

The Crank-Nicholson scheme applied to (14) is

$$(I - \frac{\mu}{2}(D_{x+}D_{x-} + D_{y+}D_{y-})v^{n+1} = (I + \frac{\mu}{2}(D_{x+}D_{x-} + D_{y+}D_{y-})v^n \tag{15}$$

Applying the ADI idea to (15), we obtain the following two step scheme

$$\begin{aligned}
(I - \frac{\mu}{2}D_{x+}D_{x-})w &= (I + \frac{\mu}{2}D_{y+}D_{y-})v^n \\
(I - \frac{\mu}{2}D_{y+}D_{y-})v^{n+1} &= (I + \frac{\mu}{2}D_{x+}D_{x-})w
\end{aligned}$$

This is an $O(2,2)$ accurate scheme with unconditional stability. A reordering of the unknown vector into row-major or column-major order (wrt the mesh) is required at each step to maintain the tridiagonal nature of the left hand side operator.

Note that this scheme can also be obtained by *splitting* the operators in (15), as in

$$(I - \frac{\mu}{2}D_{x+}D_{x-})(I - \frac{\mu}{2}D_{y+}D_{y-})v^{n+1} = (I + \frac{\mu}{2}D_{x+}D_{x-})(I + \frac{\mu}{2}D_{y+}D_{y-})v^n \tag{16}$$

Now, how can this idea be applied to cloth simulation? At the end of 2.3, the left-hand-side matrix had been reduced to a (stiff) operator that looks similar to the operator in (14). We can further divide the stretch spring into two types, "x" and "y", or warp and weft. The ADI idea is to treat only one of these types implicitly at a time. The question is, how do we treat a fibre when it is *not* being treated implicitly? The simplest way to analyze this approach is to isolate a single fibre. We are searching for a consistent, stable, two step formulation, where one of the steps involves a purely explicit formulation. There are many possibilities. I consider a few in the remainder of this section, restricting the analysis to the one dimensional wave equation (no bending).

5.1 Simplest (but useless) ADI scheme

Imagine a scheme where the explicit step is simply

$$v_j^{n+1} = v_j^n.$$

This is unusable. It is impossible to decouple the motion of a warp thread from the motion of a weft thread - they share the same particles! No, the dynamics of the two steps must be identical, only the treatment of the forces, the u_{xx} term, can vary.

5.2 Crank-Nicholson ADI scheme

For the heat equation, the C-N ADI scheme gave ideal results. Let's examine how it fares for the wave equation. The scheme is a combination of (7) and (13)

$$\begin{aligned} Explicit\ step &: \quad v_j^{n+1} = (2 + \alpha^2\mu^2 D_+ D_-)v_j^n - v_j^{n-1} \\ Implicit\ step &: \quad (1 - \alpha^2\mu^2 D_+ D_-)v_j^{n+1} = 2v_j^n - v_j^{n-1} \end{aligned}$$

giving a combined scheme of

$$(1 - \alpha^2\mu^2 D_+ D_-)v_j^{n+2} = (3 + 2\alpha^2\mu^2 D_+ D_-)v_j^n - 2v_j^{n-1}$$

This is an $O(k^2, h^2)$ accurate scheme. To analyze the stability of this scheme, we convert it into a first order system. Define $w^n = v^{n-1}$ (so $w^{n+2} = v^{n+1}$). We can now rewrite this as

$$\begin{bmatrix} 1 - \alpha^2\mu^2 D_+ D_- & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}_j^{n+2} = \begin{bmatrix} 3 + 2\alpha^2\mu^2 D_+ D_- & -2 \\ 2 + \alpha^2\mu^2 D_+ D_- & -1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}_j^{n}$$

Applying a Fourier transform in space, the amplification matrix (squared) becomes

$$g^2(\zeta) = \begin{bmatrix} 1+\beta^2 & 0 \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 3-2\beta^2 & -2 \\ 2-\beta^2 & -1 \end{bmatrix}$$

where $\beta^2 = 4\alpha^2\mu^2 \sin^2 \frac{\zeta}{2}$. The von Neumann condition for stability of this system requires that the eigenvalues of the amplification matrix satisfy $|\lambda| \leq 1$. Its eigenvalues are

$$\lambda_\pm = \frac{1}{2(1+\beta^2)} \left[2 - 3\beta^2 \pm \sqrt{9\beta^4 - 16\beta^2}\right]$$

Requiring that $9\beta^4 - 16\beta^2 < 0$ (the reverse leads to roots that violate our criterion), this becomes

$$\lambda_\pm = \frac{1}{2(1+\beta^2)} \left[2 - 3\beta^2 \pm \imath\sqrt{16\beta^2 - 9\beta^4}\right]$$

the magnitude of which is

$$\begin{aligned} |\lambda| &= \frac{1}{4(1+\beta^2)^2} \left[4 - 12\beta^2 + 9\beta^4 + 16\beta^2 - 9\beta^4\right] \\ |\lambda| &= \frac{4(1+\beta^2)}{4(1+\beta^2)^2} = \frac{1}{1+\beta^2} \leq 1 \end{aligned}$$

And so, from the requirement mentioned above, the criteria for stability is $9\beta^2 - 16 < 0$, or

$$\alpha\mu < \frac{2}{3}$$

A poor result given the nature of the two component steps that comprise this scheme. Let's see if we can do better.

5.3 On-Off ADI scheme

In this scheme, the u_{xx} term is ignored (off) in the explicit step, and then applied (on) with (roughly) double the "intensity" during the implicit step. With respect to a cloth mass-spring system, this is akin to simulating a series of independent fibres in the warp direction with an implicit scheme for one time step, and then doing the same in the weft direction at the next time step. Connections between the (more or less locally parallel) fibres are ignored for one time-step in alternating directions. The hope is that the implicit formulation at every second step will “recover” from the drift incurred during the off-step.

The scheme is a composed of two steps

$$\begin{aligned} Off\ (explicit)\ step &: \quad v_j^{n+1} = 2v_j^n - v_j^{n-1} \\ On\ (implicit)\ step &: \quad (1 - c_1\alpha^2\mu^2 D_+ D_-)v_j^{n+1} = 2v_j^n - v_j^{n-1} \end{aligned}$$

giving a combined scheme of

$$(1 - c_1\alpha^2\mu^2 D_+ D_-)v_j^{n+2} = 3v_j^n - 2v_j^{n-1} \tag{17}$$

Note the c_1 constant in (17). My first instinct was to set this to 2. I then (more reasonably) decided to examine the truncation error of this scheme via Taylor series expansion to determine what value of c_1 was required to obtain a consistent scheme. As it turns out,

$$\frac{1}{k^2}(v_j^{n+2} - 3v_j^n + 2v_j^{n-1}) = 3u_{tt} + O(k)$$

And $c_1 = 3$. This is an $O(k, h^2)$ accurate scheme. As in 5.2, we convert to a first order system. (Note: the following proof contains $c_1 = 2$; the same result holds for $c_1 = 3$.)

$$\begin{bmatrix} 1 - 2\alpha^2\mu^2 D_+ D_- & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}_j^{n+2} = \begin{bmatrix} 3 & -2 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}_j^{n}$$

Applying a Fourier transform in space, the amplification matrix (squared) becomes

$$g^2(\zeta) = \begin{bmatrix} 1 + 2\beta^2 & 0 \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 3 & -2 \\ 2 & -1 \end{bmatrix}$$

Its eigenvalues are

$$\lambda_\pm = \frac{1}{2(1+\beta^2)}\left[1 - \beta^2 \pm \sqrt{\beta^4 - 4\beta^2}\right]$$

Requiring that $|\lambda| \le 1$, we have two cases which must be satisfied:

$$\frac{1}{2(1+\beta^2)}\left[1-\beta^2 \pm \sqrt{\beta^4-4\beta^2}\right] < 1$$

and

$$\frac{1}{2(1+\beta^2)}\left[1-\beta^2 \pm \sqrt{\beta^4-4\beta^2}\right] > -1.$$

Both cases give us the condition $\beta^2 \geq -\frac{1}{2}$; this is an unconditionally stable scheme.

5.4 F-M ADI scheme

An unconditionally stable, $O(k^2, h^2)$ accurate ADI scheme for the 2D wave equation, attributed to Fairweather and Mitchell, is (taken from [12])

$$\begin{aligned}
(1-\frac{1}{4}\alpha^2\mu^2 D_{x+}D_{x-})\tilde{v}_{l,m}^{n+\frac{1}{2}} &= (1+\frac{1}{4}\alpha^2\mu^2 D_{y+}D_{y-})v_{l,m}^{n} \\
(1-\frac{1}{4}\alpha^2\mu^2 D_{y+}D_{y-})\tilde{v}_{l,m}^{n+1} &= (1+\frac{1}{4}\alpha^2\mu^2 D_{x+}D_{x-})\tilde{v}_{l,m}^{n+\frac{1}{2}} \\
v_{l,m}^{n+1} &= 2\tilde{v}_{l,m}^{n+1} - v_{l,m}^{n}
\end{aligned}$$

Adapting this to our 1D problem, this translates to the following two-step scheme

$$\begin{aligned}
Explicit\ step &: \quad v_j^{n+1} = (1+\alpha^2\mu^2 D_+D_-)v_j^n \\
Implicit\ step &: \quad v_j^{n+1} = 2(1-\alpha^2\mu^2 D_+D_-)^{-1}v_j^n - v_j^{n-1}
\end{aligned}$$

While this is an interesting and attractive scheme, it unfortunately possesses the same problem as 5.1; it does not include the dynamics (the u_{tt} term) in the first step. I'm not sure how (or if) it can be reformulated for the problem at hand.

5.5 Splitting the mass-spring formula

Ignoring the details of the relationship between PDE and mass-spring formulations for a moment, we may ask: can the same type of splitting (or factoring) that transformed (15) to (16) also be applied to the mass-spring formulation? The implicit formulation for the mass-spring system can be written as

$$(I - kM^{-1}\frac{df}{dv} - k^2M^{-1}\frac{df}{dx})dx = kv^n + k^2M^{-1}(f^n - \frac{df}{dv}v^n) \tag{18}$$

Topologically, this appears "similar" to (15). Before proceeding, a little notation is needed; warp (p) and weft (t) versions of various vectors and matrices are

- x_p is the position vector of the system using *warp*-major ordering; similarly for v_p.
- x_t is the position vector of the system using *weft*-major ordering; similarly for v_t.
- define the *flip* operation x^F which indicates a reordering of the vector from warp to weft major ordering (or vice verca). $x_p^F = x_t$, and $(x_p^F)^F = x_p$.
- M_p and M_t are similarly defined versions of the mass matrix.
- the jacobians ($\frac{df}{dx_p}$ for instance) have a similar notation. In addition to the ordering implied by this notation, a jacobian of this type only includes terms for the springs operating in that direction. So $\frac{df}{dx_p}$ and $\frac{df}{dx_t}$ are both (block) tridiagonal. In addition, $\frac{df}{dx_p} + (\frac{df}{dx_t})^F = \frac{df}{dx}$

Using the above notation and identities, and taking care to not spoil the tridiagonal property during the calculation, we can split (18) in the following way

$$\begin{aligned}(I - kM_p^{-1}\frac{df}{dv_p} - k^2M_p^{-1}\frac{df}{dx_p})\left[(I - kM_t^{-1}\frac{df}{dv_t} - k^2M_t^{-1}\frac{df}{dx_t})dx_t\right]^F = \\ kv_p^n + k^2M_p^{-1}(f^n - \frac{df}{dv_p}v_p^n - (\frac{df}{dv_t}v_t^n)^F)\end{aligned} \tag{19}$$

The right hand sides of (18) and (19) are identical. The same cannot be said for the left hand sides. Multiplying out the left hand side of (19), we obtain

$$\begin{aligned}I - kM_p^{-1}(\frac{df}{dv_p} + \frac{df}{dv_t}^F) - k^2M_p^{-1}(\frac{df}{dx_p} + \frac{df}{dx_t}^F) \\ +k^2M_p^{-2}\frac{df}{dv_p}\frac{df}{dv_t}^F + k^3M^{-2}(_p\frac{df}{dv_p}\frac{df}{dx_t}^F + \frac{df}{dx_p}\frac{df}{dv_t}^F) + k^4M_p^{-2}\frac{df}{dx_p}\frac{df}{dx_t}^F\end{aligned} \tag{20}$$

The first three terms of (20) are equivalent to the left hand side of (18). The remaining three are error terms. They are $O(k^2)$ to $O(k^4)$. Unfortunately, one of the correct terms is also $O(k^2)$. In addition, M^{-2} appears; since the masses of the particles in the simulation are typically small, this matrix will contain large terms. On the bright side, two of the error terms contain $\frac{df}{dv}$; in the case of low damping, these may become negligible; the other term is $O(k^4)$. It is difficult to predict how these error terms will affect the simulation quality; so I went ahead and implemented the scheme.

6 Experiments

I have conducted various experiments using PDE and mass-spring formulations. The PDE formulations were implemented in Matlab; the mass-spring simulator was implemented in Java. Some experiments were conducted to verify consistency between the two formulations, while others were conducted to verify or determine the stability properties of the various schemes.

6.1 PDE consistency tests - 1D

This series of tests was intended to verify the consistency/accuracy of the various PDE schemes I implemented in Matlab.

For these tests, I defined a simple 1D physical scenario to model with the wave equation. The scenario is that described in section 3.2, with the left boundary fixed in space (dirichlet) and a force applied at the right boundary (von Neumann). A rod with the following properties was simulated

- L = 5 metres
- EA = 2 Newtons
- density = 0.25 $\frac{kg}{m}$
- gravity = 9.80 $\frac{m}{s^2}$
- force = 1 Newton

Empricial results matched theory. As a sample for the explicit scheme (7), figure 7 shows how the error decreases with subsequent halvings of the mesh and time-step size. Here, accuracy is $O(h^2, k^2)$. Parameters for these tests can be found in table 1.

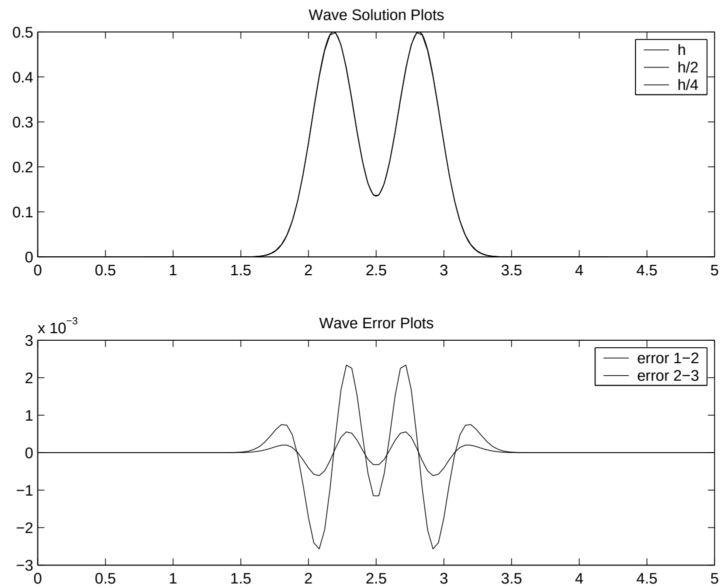

Figure 7: Solutions and errors for the wave equation. Explicit scheme, gaussian initial data, at $t = 1s$.

EA (N)	ρ (kg/m)	gravity (m/s^2)	force (N)	mesh spacing (m)	time-step (s)
2	0.25	9.8	1	0.04	0.001
”	”	”	”	0.02	0.0005
”	”	”	”	0.01	0.00025

Table 1: Parameters for 3 PDE self-consistency tests; explicit scheme

PDE parameters		Mass-spring parameters		Common parameters			
EA (N)	ρ (kg/m)	k_s (N/M)	m (kg)	gravity (m/s^2)	force (N)	mesh spacing (m)	time-step (s)
2	0.25	50	0.01	9.8	1	0.04	0.001
2	0.25	100	0.005	9.8	1	0.02	0.0005
2	0.25	200	0.0025	9.8	1	0.01	0.00025

Table 2: Parameters for 3 consistency tests (PDE vs. mass-spring formulations); explicit scheme

6.2 PDE / mass-spring consistency tests - 1D

This series of tests was intended to verify the consistency between the PDE and mass-spring formulations. For these purposes, I used the same physical scenario as in 6.1 - with a few exceptions; Here, the length $L = 1$ metre, and the initial conditions are "at rest", ie. $u = u_t = 0$ at $t = 0$ (or equivalently $u^0 = u^{-1} = 0$ for all x).

First, I compared the explicit PDE formulation against the symplectic Euler mass-spring formulation. Similarly to 6.1, I ran three tests, halving the mesh size and time-step each time. Parameters for these tests can be found in Table 2.

I found correspondence to be excellent (as expected). For "snapshots" taken at $t = 1s$, differences between the PDE and mass-spring solutions were on the order of $10^{-15}m$, where the magnitude of the solution was on the order of $1m$.

I ran identical tests for the implicit case. Here results differed on the order of $10^{-5}m$. The solutions and differences between them are plotted in figure 8. The reason for this increased difference is the solution method used on the mass-springs simulator: I was using a conguate-gradient solver to solve the tridiagonal system! (I didn't write code specifically for this 1D case, and there wasn't enough time to do so for this submission.) Still, the correspondence is very good.

I ran identical tests for the On-Off ADI case. Here the story is more involved. First, I performed a paper and pen analysis of the mapping from the PDE to the mass-spring formulation, as in 3.2. Details are omitted here, but the schemes *appear* to be inconsistent with one another. Further analysis is required. Experimentally, I found the PDE implementation of the On-Off ADI scheme to give considerably different results from the other schemes (on the same order as the solution itself). Thinking "why not?", I tried to set c_1 in (17) to 2 instead of 3. Strangely, it gave results consistent with the other schemes. Again, further analysis is required. In any case, I ran the experimental comparison betweem the PDE and mass-spring formulations using $c_1 = 2$. (k_s is also doubled for the *On* step in the mass-spring formulation). Results differed on the order of $10^{-3}m$. The solution and differences between them are plotted in figure 9. (Looking at the error

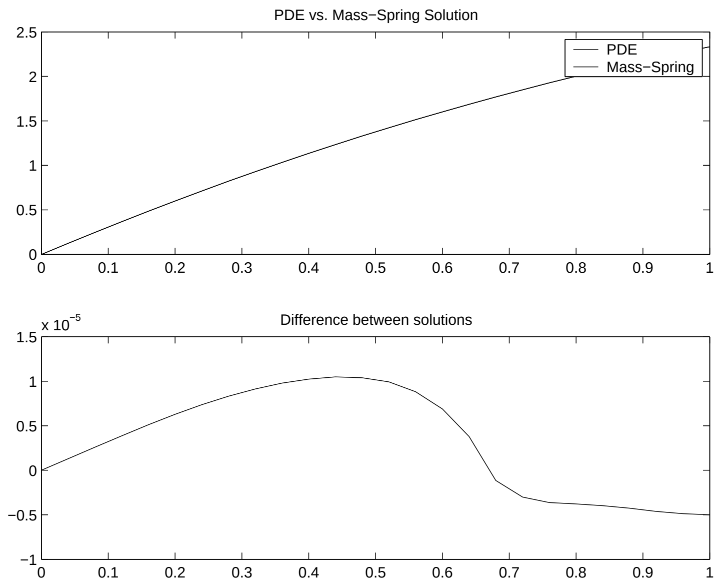

Figure 8: Solutions and differences for the wave equation. PDE vs. mass-spring formulations. Implicit scheme, "at rest" initial data, $t = 1s$.

plot, it appears there is a problem with my von Neumann BC handling.)

6.3 1D Stability Tests

This series of tests was intended to verify the theoretical stability results from sections 4 and 5. Tests were run for both the PDE and the mass-spring formulations in 1D. These were done for the explicit/symplectic, implicit, Crank-Nicholson ADI, and On-Off ADI schemes. Experimental results matched well with theory, namely

- explicit/symplectic: $\alpha\mu < 1$
- implicit: unconditional
- Crank-Nicholson ADI: $\alpha\mu < \frac{2}{3}$
- On-Off ADI: unconditional
- Split mass-spring: (EB: I should include results for this here)

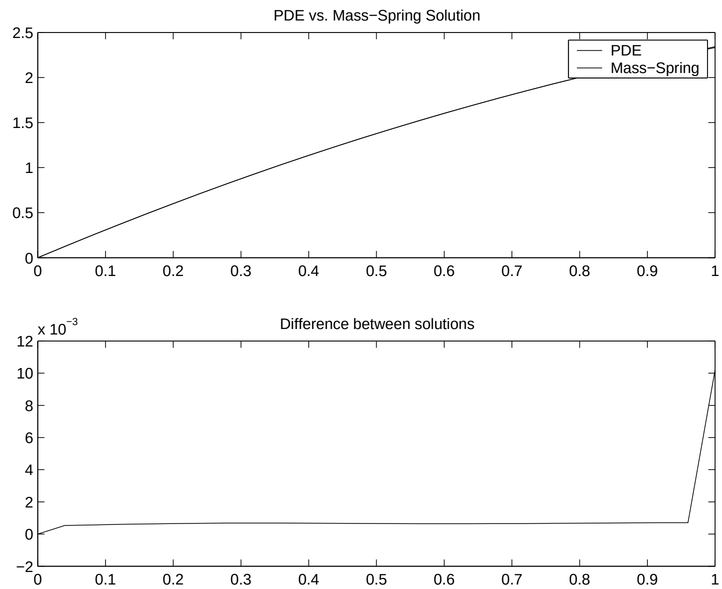

Figure 9: Solutions and differences for the wave equation. PDE vs. mass-spring formulations. On-Off scheme, "at rest" initial data, at $t = 1s$.

As predicted, varying other parameters in the simulation, such as the force F or gravity g had no effect on stability. Varying h in the mass-spring formulations seemed to have a *tiny* (perhaps 1%) effect. It should be noted that stability doesn't necessarily result in clean animations. Even the implicit scheme can give wildly chaotic changes between time-steps when velocities (u_t) are high. This usually occurs for very low density (particle mass) models and when large external forces are present.

6.4 2D Stability Tests

This series of tests was run strictly for the mass-spring formulation in 2D (all motion and forces in the plane). The same "fabric type" (same physical parameters) was used as in the 1D tests. The entire left side of the cloth was constrained (Dirichlet condition), while the force F (von Neumann condition) was applied to *one* corner on the right side - resulting in asymmetry and shear deformation. The objective was to explore experimental stability regions for the various schemes. Shear and bending stiffnesses were both set to zero.

- For the symplectic scheme, the stability region appeared to be $k < 0.63\sqrt{\frac{m}{k_s}}$ (as opposed to $k < \sqrt{\frac{m}{k_s}}$ in 1D).

- For the implicit scheme, unconditional stability was maintained.
- For the On-Off ADI scheme, the stability region appeared to be $k < 1.2\sqrt{\frac{m}{k_s}}$ (as opposed to uncondtional in 1D!).
- For the Split scheme (from 5.5), the stability region appeared to be $k < 2.3\sqrt{\frac{m}{k_s}}$ (EB: Check this in 1D).

These results are not surprising (except those for the On-Off ADI scheme, which is a bit discouraging). Again, stability doesn't neccessarily result in clean animations - but for "average" parameters, the Split scheme gives fairly reasonable (by eye) and efficient results.

6.5 3D Stability Tests

This series of tests was identical to the 2D tests - except external forces (gravity and the boundary force) were allowed to act out of the plane, resulting in full 3D motion (folding, etc). Stability regions appeared to be identical to the 2D case.

7 Conclusions and Future Work

This paper investigates the feasibility of using an ADI scheme for cloth simulation. It contains the preliminary work neccessary for this investigation, answers some questions, and raises many others. (At least in the author's mind!)

7.1 results

The results of this paper can be summarized as follows:

- I have shown the equivalence between the PDE and mass-spring formulations (at least for 1D) for the elasticity problem. Specifically,

  1. the identity between an explicit, 2nd order discretization of the PDE and a symplectic Euler time-stepping scheme applied to a mass-spring system,
  2. the same for an implicit discretization of the PDE and mass-spring system.

  This allows us to (more soundly) use the powerful tools and ideas associated with numerical PDE theory in the application of mass-spring systems.

- I have shown that applying an IMEX scheme to cloth mass-spring systems - primarily, handling the weak bending springs explicitly - gives improved computational efficiency while still maintaining desirable stability properties. This also reduces the implicit part of the formulation to a 2D surface (all forces *in* the plane) moving in 3D space.

- Given this "reduction to 2D", I have proposed the idea of using an ADI scheme for simulating cloth, along with several concrete examples. Using PDE stability techniques, I have begun analyzing the viability of these schemes.

- Finally, I have implemented several of these ADI schemes in a cloth mass-spring simulator and analyzed their consistency and stability. As yet, no ADI scheme has proven to have ideal stability properties when applied to the full problem.

7.2 Future work

Though I have tried to address the fundamental issues required to develop an ADI scheme for cloth, much remains to be done.

- First and foremost, develop an unconditionally stable, ADI scheme ($O(n)$ computation) for cloth!

- Analyze the On-Off scheme further. I have yet to show the identity between the PDE and mass-spring formulation for this scheme. I have yet to answer the "c_1 question"; does it equal 2 or 3? Hopefully this investigation will reveal more about the scheme's stability behaviour, and perhaps ways of improving it.

- In order to fully verify my formulations/implementations, I should determine the exact solution for given initial data and compare with the numerical solutions.

- Add damping. ie. consider the PDE $u_{tt} = \alpha^2 u_{xx} - \gamma u_t$. I have already done this for most of the analyses and implementations, but due to time constraints and incomplete results, I have omitted this from the report.

- Add collision detection and response. How will this affect the formulation, stability, etc.

- Include performance measures. Although I have done some tests to ensure the ADI solvers I've implemented are running in $O(n)$ time, some numbers would be nice.

- First order physics. If we consider a first order physical model (where damping dominates), the PDE becomes $u_t = \alpha^2 u_{xx}$ (for a different α). Schemes for PDEs of this type have very different stability properties (Crank-Nicholson ADI for instance). If our goal is simply to find the drape configuration of a piece of fabric - a common one in the fashion (and computer science) industry - we may have better success.

- Extend the analysis to 2D. This is not simply a matter of analyzing the 2D wave equation; it implies a second dimension in the solution. It would be very instructive to identify the system of PDEs that define this problem (from elasticity theory) and analyze it.
- Given the previous point, to extend the PDE implmentations to 2D.
- Apparently, ADI schemes exist which incorporate cross derivative terms, such as u_{xy}. I should investigate these.
- Perform self-consistency tests for the various solvers for the 3D mass-spring system. ie. Vary the mesh size for the 3D problem and verify if the solver gives consistent solutions (even if we don't know what PDE its *trying* to solve).
- Same as the previous point, but specifically for the Split scheme; I have little analysis and *no* PDE solver to compare it against.
- Experimentally investigate the role of shear, mainly on stability. Ie. perform 2D (and 3D) mass-spring tests with shear included.

8 Acknowledgments

I would like to thank Chen Greif for his time and input (and patience with my many questions) regarding this project.

References

[1] H. E. Ascough, J. Bez and A. M. Bricis. A simple beam element large displacement model for the finite element simulation of cloth drape. *Journal for the Textile Institute*, 87(1).

[2] David Baraff and Andrew Witkin. Large steps in cloth simulation. *Computer Graphics*, 32(Annual Conference Series):43–54, 1998.

[3] David E. Breen, Donald H. House, and Michael J. Wozny. Predicting the drape of woven cloth using interacting particles. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 365–372. ACM Press, 1994.

[4] Kwang-Jin Choi and Hyeong-Seok Ko. Stable but responsive cloth. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 604–611. ACM Press, 2002.

[5] J. R. Collier, B. J. Collier, O'Toole G., and S.M. Sargand. Drape prediction by means of finite-element analysis. *Journal for the Textile Institute*, 82(1).

[6] Mathieu Desbrun, Peter Schröder, and Alan Barr. Interactive animation of structured deformable objects. In *Graphics Interface*, pages 1–8, 1999.

[7] Donald H. House and David E. Breen. *Cloth modeling and animation*. A. K. Peters, Ltd., 2000.

[8] Gui-Rong Liu. *Mesh Free Methods*. CRC Press, 2003.

[9] D. W. Peaceman and H. H. Rachford. The numerical solution of parabolic and elliptic differential equations. *Journal for the Society of Industrial and Applied Mathematics*, 3:28–41, 1955.

[10] Xavier Provot. Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In *Proc. Graphics Interface*, pages 147–154, 1995.

[11] J. D. Renton. *Applied Elasticity*. Ellis Horwood, Ltd., 1987.

[12] John C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. Wadsworth & Brooks/Cole, 1989.

[13] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 205–214. ACM Press, 1987.

[14] S.T. Tan T.N. Wong Y.F. Zhao and W.J. Chen. A constrained finite element method for modeling cloth deformation. *The Visual Computer*, (15):90–99, 1999.