MLS Meshless Method for Wave and Advection Equation

Artur Gower 06/06/2012

1 Abstract

We present the moving finite element method together with a brief on the origins of the idea together with an overview and some simple deductions of the method. We then present the general 1D elastic wave equation, reveal it's domain of dependence and show how to formulate the equation in terms of an MLS meshless method. Upwind schemes are discussed and numerical results are presented for the advection equation.

2 Meshless Methods

To introduce meshless methods here is an extract from [1] "*Meshless Methods: an Overview and Recent Developments*":

"The problems of computational mechanics grow ever more challenging. For example, in the simulation of manufacturing processes such as extrusion and molding, it is necessary to deal with extremely large deformations of the mesh while in computations of castings the propagation of interfaces between solids and liquids is crucial. In simulations of failure processes, we need to model the propagation of cracks with arbitrary and complex paths. In the development of advanced materials, methods which can track the growth of phase boundaries and extensive microcracking are required.

These problems are not well suited to conventional computational methods such as nite element, nite volume or nite di erence methods. The underlying structure of these methods which originates from their reliance on a mesh is not well suited to the treatment of discontinuities which do not coincide with the original mesh lines. Thus the most viable strategy for dealing with moving discontinuities in methods based on meshes is to remesh in each step of the evolution so that mesh lines remain coincident with the discontinuities throughout the evolution of the problem. This can, of course, introduce numerous di culties such as the need to project between meshes in successive stages of the problem, which lead to degradation of accuracy and complexity in the computer program, not to mention the burden associated with a large number of remeshings."

The first steps to develop such methods were based on mollifiers, or a

kernal approximation,

$$u^h(x) = \int_\Omega \omega(x-y,h)u(y)dV, \tag{1}$$

where $u^h(x)$ is the approximation, $\omega(x-y,h)$ is a kernel or weight function, and h is a measure of the size of the support. Where ω was required to satisfy certain properties, such as

- $\int_\Omega \omega(x-y,h)dV = 1$
- $\omega(x,h) \to \delta(s)$ as $h \to 0$, where $\delta(x)$ is the delta dirac function.

A simple and much used choice for ω is radially symmetric functions, for example the exponential:

$$\omega(x,h) = \begin{cases} \exp^{-\beta x^2/h^2} & x/h \leq 1, \\ 0 & x/h > 0. \end{cases}$$

For the discrete version of the Mollifier equation (1) we wish to obtain a formula in terms of nodal values $u_j = u(x_j)$ for $j = 1$ to N. This would lead to formulas of the type:

$$u^h(x) = \sum_j \omega(x-x_j,h)u_j\Delta V_j = \sum_j \Phi_j^h(x)u_j.$$

We call Φ_j^h the shape functions where in most cases $u_j \neq u^h(x_j)$ for the Φ_j^h's are not true interpolants. The problem with these methods, most commonly known as "Smooth Particle Hydrodynamics" is two fold: to develop robust techniques for assigining ΔV_j's for each node and more crucially the discrete form above does not lead to consistent methods, in other words for reasonable choices of ω and ΔV_j it can be shown that for $u(x) = x$,

$$u^h(x_i) = \sum_j \Phi_j^h(x_i)x_j \neq x_i,$$

for a nonuniform arrangment of nodes and boundaries. See [5] and [1]. Now we present a most successful method for meshless methods that is flexible and consistent.

2.1 Moving Least Squares - MLS

Following the idea of mollifiers we wish to weight nodal values of $u(x)$ on the nodes $x_I = [x_1, x_2, \ldots, x_N]$ then we denote $U_I = [u(x_1), u(x_2), \ldots, u(x_N)]^T$. For the weight functions let the matrix $W_I(x)$ be a diagnonal matrix of the vector $[\omega((x-x_1)/r_1), \omega((x-x_2)/r_2), \ldots, \omega((x-x_N)/r_N)]$, each node can have a different range of influence. First we wish to approximate $u(x)$ giving a weight to each node,

$$\sum_j w(x/r_j - x_j/r_j)u_j = W_I(x)U_I,$$

now the major obstacle is consistency, i.e. we want the method to be exact for $u(x) = \alpha_K x^K + \alpha_{K-1}x^{K-1} + \ldots \alpha_0$. To make this so let

$$U_I^k = [x_1^k, x_2^k, \ldots, x_N^k]^T, \tag{2}$$

$$P_I = \left[U_I^1 | U_I^2 | \cdots | U_I^K\right], \tag{3}$$

$$\boldsymbol{\alpha} = [\alpha_0, \alpha_1, \ldots, \alpha_K]^T, \tag{4}$$

then consistency means we wish to recover α_j's from $W_I(x)P_I\boldsymbol{\alpha}$ for every x. By construction $W_I(x)P_I\boldsymbol{\alpha}$ should be one-to-one, to restrict it's image to it's codomain we multiply by $(W_I(x)P_I)^T$ on the left resulting in $P_I^T {W_I}^2(x)P_I\boldsymbol{\alpha}$, this can also be seen as removing everything orthogonal to the space generated by $W_I(x)P_I$. Now we may invert to obtain

$$(P_I^T {W_I}^2(x)P_I)^{-1}P_I^T {W_I}^2(x)P_I\boldsymbol{\alpha} = \boldsymbol{\alpha},$$

now to finish we denote $p(x) = [1, x, \ldots, x^K]$ and use W_I in place of ${W_I}^2$ consistency follows from the fact that

$$p(x)(P_I^T W_I(x)P_I)^{-1}P_I^T W_I(x)U_I^k = x^k,$$

Hence the so called shape functions $\Phi_j(x)$ and method are given by

$$\Phi_j(x) = p(x)(P_I^T W_I(x)P_I)^{-1}P_I^T W_I(x)e_j, \tag{5}$$

$$u(x) = \sum_j \Phi_j(x)u_j = \Phi_I(x)U_I. \tag{6}$$

In the literature the MLS method is deduced as a weighted minimization: to represent $u(x)$ approximately as $u(x) = p(x)\boldsymbol{\alpha}$ where

$$\min_\alpha \sum_j \omega((x-x_j)/r_j)\|p(x_j)\boldsymbol{\alpha} - u(x_j)\|^2,$$

which results in the same method.

Let us investigate some of the Φ_j shape functions. We shall use the exponential weight function:

$$\omega(x) = \begin{cases} \exp^{-3x^2} & x \leq 1, \\ 0 & x > 0. \end{cases} \tag{7}$$

The resulting shape functions for

$$x_I = [0, 0.2, 0.4, 0.6, 0.8, 1], \tag{8}$$

$$p(x) = [1, x], \tag{9}$$

$$\omega_j(x) = \omega(x/0.3 - x_j/0.3), \tag{10}$$

are shown in Figure 1. Note that during the proof of consistency of the MLS method there is considerable flexability in the choice of the ω_j's. One interesting choise is to use $\omega_j(x) = \omega(x/r_j - x_j/r_j - b_j)$ so as to slant the domain of influence of the node. This strategy can be usefull to help acheive an upwind behaviour for hyperbolic equations. See Figure 2 for an example with the above data and $b_j = 1/4$ which shifts the domain of influence roughly r_j to the right.

The power of approximation of MLS is near-best in the sense that the local error is bounded in terms of the error of a local best polynomial approximation given some regularity constraints on the data and weight functions. See [4].

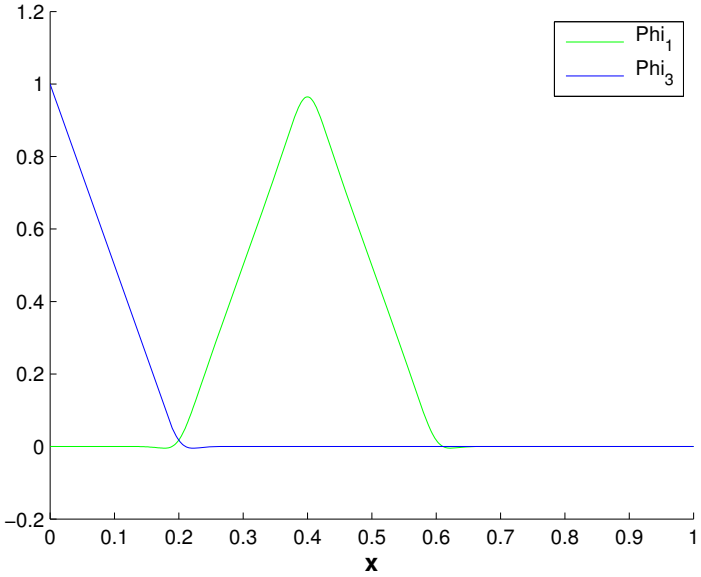

Figure 1: The shape functions Φ_1, Φ_3 for $x(1) = 0$, $x(3) = 0.4$.

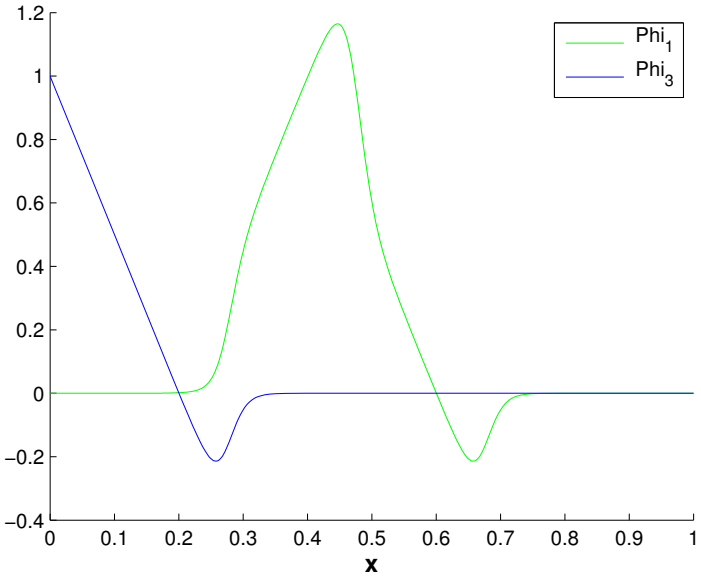

Figure 2: The slanted shape functions for $x(1) = 0$, $x(3) = 0.4$ and $\omega_j = \omega(x/0.3 - x_j/0.3 - 1.4)$.

3 Wave equations

We will briefly illustrate the domain of dependence of the 1D-elastic equations, which is fundamental for designing numerical schemes.

$$\rho(x)v_t(x,t) = \sigma_x(x,t) + f(x,t)\ ,\ \sigma(x,t) = \rho(x)\frac{\partial \Psi}{\partial u_x}(u_x,x), \tag{11}$$

$$u_t(x,t) = v(x,t). \tag{12}$$

We shall formulate these equation as a Hyperbolic system,

$$\begin{bmatrix} u \\ v \\ \sigma \end{bmatrix}_t = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1/\rho \\ 0 & \rho\psi & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ \sigma \end{bmatrix}_x + \begin{bmatrix} v \\ f \\ 0 \end{bmatrix} \tag{13}$$

or equivalently

$$Q_t = A(Q,x)Q_x + F(Q,x) \tag{14}$$

where

$$\psi(u_x,x) = \frac{\partial^2 \Psi}{\partial u_x^2}(u_x,x) \text{ and } Q = [u,v,\sigma]^T.$$

Then by multiplying both sides of equation (14) on the left by the inverse eigenvector matrix of A we get

$$\begin{aligned} \rho\sqrt{\psi}\frac{d}{dt}v(\gamma_1(t),t) + \frac{d}{dt}\sigma(\gamma_1(t),t) &= f\rho\sqrt{\psi}, \\ -\rho\sqrt{\psi}\frac{d}{dt}v(\gamma_2(t),t) + \frac{d}{dt}\sigma(\gamma_2(t),t) &= -f\rho\sqrt{\psi}, \\ \frac{d}{dt}u(\gamma_3(t),t) &= v, \end{aligned} \tag{15}$$

where

$$\begin{aligned} &\partial_t\gamma_1(\tau) = -\sqrt{\psi(\gamma_1(\tau),\tau)},\ \gamma_1(t) = x, \\ &\partial_t\gamma_2(\tau) = \sqrt{\psi(\gamma_2(\tau),\tau)},\ \gamma_2(t) = x, \\ &\partial_t\gamma_3(\tau) = 0,\ \gamma_3(t) = x. \end{aligned}$$

These equations together with causality[1] indicate that the boundary of the domain of dependence for each point x is determined by the curves $(\gamma_1(\tau),\tau)$ and $(\gamma_2(\tau),\tau)$ for $\tau < t$.

[1] that information from the future can not effect the present.

If ψ only depends on x, which can be considered as a linearized material close to some state, we can manipulate the characteristic equations (15) further to obtain:

$$\begin{aligned}
\frac{d}{dt}\left[\rho(\gamma_1(t))\sqrt{\psi(\gamma_1(t))}v(\gamma_1(t),t)+\sigma(\gamma_1(t),t)\right] &= f(\gamma_1(t),t)\rho(\gamma_1(t))\sqrt{\psi(\gamma_1(t))} \\
&\quad + \frac{d}{dt}\rho(\gamma_1(t))\psi(\gamma_1(t)), \\
\frac{d}{dt}\left[-\rho(\gamma_2(t))\sqrt{\psi(\gamma_2(t))}v(\gamma_2(t),t)+\sigma(\gamma_2(t),t)\right] &= -f(\gamma_2(t),t)\rho(\gamma_2(t))\sqrt{\psi(\gamma_2(t))} \\
&\quad - \frac{d}{dt}\rho(\gamma_2(t))\psi(\gamma_2(t)), \\
\frac{d}{dt}u(\gamma_3(t),t) &= v(\gamma_3(t),t),
\end{aligned}$$

which in turn implies that

$$\begin{aligned}
2\rho(x)\psi(x)u_x(x,t) &= \int_{t_0}^{t} f(\gamma_1(\tau),\tau)\sqrt{\psi(\gamma_1(\tau))} - f(\gamma_2(\tau),\tau)\sqrt{\psi(\gamma_2(\tau))}d\tau \\
&\quad + \rho(\gamma_2(t_0))\psi(\gamma_2(t_0)) - \rho(\gamma_1(t_0))\psi(\gamma_1(t_0)), \\
2\rho(x)\sqrt{\psi(x)}u_t(x,t) &= \int_{t_0}^{t} f(\gamma_1(\tau),\tau)\sqrt{\psi(\gamma_1(\tau))} + f(\gamma_2(\tau),\tau)\sqrt{\psi(\gamma_2(\tau))}d\tau \\
&\quad + 2\rho(x)\psi(x) - \rho(\gamma_1(0))\psi(\gamma_1(0)) - \rho(\gamma_2(0))\psi(\gamma_2(0)),
\end{aligned} \tag{16}$$

where in this case $\sigma = \rho\psi u_x$.

The domain of dependence of each point must be taken into account when designing the numerical scheme. For the MLS method, if we fix a constant time step then for every point we can calculate the radius of it's corresponding weight function to cover the point's domain of dependence.

3.1 Variation Formulation and MLS Method

The convergence of polynomial finite element methods for the wave equation [2] and a certain class of nonlinear wave equation [3] has been be proven. Adapting these proofs for the meshless MLS should be viable but is outside the scope of this project. Also, to the authors knowledge there exist no proof of convergence of the general elastic wave equation, presented bellow, by using finite elements.

The elastic wave equation (12) can be formulated in the variation form

$$< \rho v_t, \delta u > + < \sigma, \delta u_x > = < f, \delta u > + \sigma \delta u \Big|_{x=a}^{x=b}. \tag{17}$$

A nice property about this method is that it potentially conserves a discrete energy. Let

$$E(t) = \int_a^b \rho \frac{v^2(x,t)}{2} + \rho \Psi(x,t) dx,$$

then if for our numerical scheme we have that

$$< \rho(\cdot) v_t(\cdot,t), v(\cdot,t) > + < \sigma(\cdot,t), v_x(\cdot,t) > = \frac{E(t_{n+1}) - E(t_n)}{t_{n+1} - t_n}, \tag{18}$$

for $t_n \leq t \leq t_{n+1}$ then choosing $\delta u = v$ equation (17) becomes

$$E(t_{n+1}) - E(t_n) \;=\; (t_{n+1} - t_n) \left(< f(\cdot,t), v(\cdot,t) > + \sigma(b,t) v(b,t) - \sigma(a,t) v(a,t) \right), \tag{19}$$

which is a discrete energy conservation.

For the Meshless MLS method we approximate

$$u(x,t) \approx u_I(x,t) = \Phi^T(x) U(t), \tag{20}$$

$$\delta u(x) = \Phi^T(x) \delta U, \tag{21}$$

where $U(t), \delta U \in \mathbb{R}^n$, $\Phi(x) \in \mathbb{R}^n$. Though we are not free to choose any U and δU for we want to satisfy the boundary conditions exaclty $\delta_x u_I(b,t) = \delta_x \Phi^T(b) U(t) = \delta_x \Phi^T(b) \delta U = g_2(t)$ and $\delta_x u_I(a,t) = \delta_x \Phi^T(a) U(t) = \delta_x \Phi^T(a) \delta U = g_1(t)$. To do so, let

$$A = \begin{pmatrix} \partial_x \Phi^T(a) \\ \partial_x \Phi^T(b) \end{pmatrix} \text{ and } A b_j = 0 \text{ , for } j = 1 \text{ to } n-2$$

where the b_j's form the basis of kernel(A). Let

$$B = (b_1 | b_2 | \dots | b_{n-2}) \text{ and } A y(t) = \begin{pmatrix} g_1(t) \\ g_2(t) \end{pmatrix},$$

then every $U(t)$ and δU can be written as

$$U(t) = B \tilde{U}(t) + y(t), \quad \delta U = B \delta \tilde{U} + y(t) \tag{22}$$

and

$$u_I(x,t) = \Phi^T(x)B\tilde{U}(t) + \Phi^T(x)y(t), \;\; \delta u(x) = \Phi^T(x)B\delta\tilde{U} + \Phi^T(x)y(t) \tag{23}$$

where $\tilde{U}(t), \delta\tilde{U} \in \mathbb{R}^{n-2}$. To simplify notation let $\Phi_B(x) = B^T\Phi(x)$, we denote $\tilde{U}(t)$ as simply $U(t)$ and will work with $g_2 = g_1 = 0$ for what follows. Substituting

$$u_I(x,t) = \Phi_B(x) \cdot U(t), \tag{24}$$

$$\delta u(x) = \Phi_B(x), \tag{25}$$

in the variation form for the linearized material we get

$$\int_a^b \rho(x)\Phi_B(x) \otimes \Phi_B(x)dx \cdot U_{tt}(t) + \int_a^b \rho(x)\psi(x)\partial_x\Phi_B(x) \otimes \partial_x\Phi_B(x)dx \cdot U(t) = \int_a^b f(x,t)\Phi_B(x)dx,$$

where $\delta u = \Phi_B(x)$ so that solving the above system is to solve for every $\delta u = \Phi_B^T(x)\delta U$.

We will also formulate the advection equation (3.1) in the same manner.

$$u_t = u_x + f, \; u(0,t) = u_0(t), \; u(b,t) = 0,$$

whose variation form, with the same analysis as before. leads to,

$$\int_a^b \Phi_B(x) \otimes \Phi_B(x)dx \cdot U_t(t) = \int_a^b \Phi_B(x) \otimes \partial_x\Phi_B(x)dx \cdot U(t) + \int_a^b f(x,t)\Phi_B(x)dx, \tag{26}$$

where B's columns generate the nullspace of

$$A = \left(\Phi^T(b)\right).$$

3.2 Numerical Schemes

We shall discuss a scheme for the advection equation (3.1) first. For the scheme to be stable our approximation to $u(x, t+h)$ must take information

only from the domain of dependence, which for each point (x,t) is determined by the line $(x+\tau, t-\tau)$ for $\tau > 0$. The only form to approximate u_t from along this line is to extract it's contribution to $(u(x,t+h) - u(x+h,t))/h$, which by doing a taylor expansion on this discrete difference would tell us that

$$(u(x,t+h) - u(x+h,t))/h = u_t - u_x + O(h^3) = f(x,t) + O(h^3).$$

As this involves a discrete difference for the space derivative it does not translate exactly to the variational form (26), i.e. one form would be to use the scheme

$$\begin{aligned} U^j(t+h) &= U^{j+1}(t) \\ U(t+h) &= U(t+h) + < (\Phi_B \otimes \Phi_B)^{-1}, f(\cdot, t+h/2)\Phi_B >, \end{aligned} \tag{27}$$

where $U^j(t+h)$ corresponds to a node on $x = jh$, this method gives an almost exact answer, has the same order of approximation as MLS "interpolation". A simulation result is shown in Figure (3). We have used $r_j = dt$ for all nodes and simulations.. So to use equation (26) we must either introduce dissipation or instability. One manner to do so is the following,

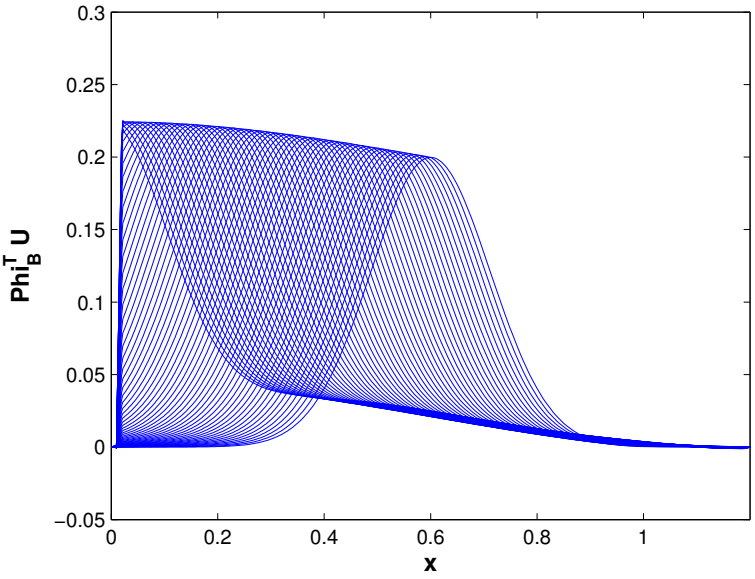

Figure 3: The evolution in time of the initial condition being advected to the left with $f(x) = 0.2\cos(x - x(N) + \pi/2)\sin(x)$.

$u_t(x, t + h/2) = \Phi_B^T(x)U_t(t + h/2)$ and $u_x(x, t + h/2) = \partial_x\Phi_B^T(x)U(t + h/2)$,

then the most evident would be to adopt

$$\Phi_B^T(x)U_t(t + h/2) = \frac{\Phi_B^T(x)}{h}(U(t + h) - U(t)),$$

note that $\Phi_B^T(x)U(t)$ is outside $\Phi_B^T(x)U(t + h)$ domain of dependence, thus instability may appear and convergence is unlikely. Let us try to recover by moving $u_x(x, t + h/2)$ to within the domain of dependence,

$$u_x(x, t+h/2) = u_x(x+h/2, t)-u_{xx}(x+h/2, t)h/2+u_{xt}(x+h/2, t)h/2+O(h^2),$$

where by the advection equation gives us that $u_{xx} = u_{xt}$, hence up to order-2 $u_x(x, t + h/2) = u_x(x + h/2, t) + f_x(x + h/2, t)h/2$. To implment this we substitute in the variation formulation of the advection equation (26),

$$\begin{aligned}\Phi_B^T(x)U_t(t + h/2) &= \frac{\Phi_B^T(x)}{h}(U(t + h) - U(t)),\\ \partial_x\Phi_B^T(x)U(t + h/2) &= \partial_x\Phi_B^T(x + h/2)U(t) + f_x(x + h/2, t)h/2.\end{aligned}$$

Leading to

$$\begin{aligned}U(t + h) = U(t)+&\\ &h\left(\int_a^b \Phi_B(x) \otimes \Phi_B(x)dx\right)^{-1} \int_a^b \Phi_B(x) \otimes \partial_x\Phi_B(x + h/2)dx \cdot U(t)+\\ &h\left(\int_a^b \Phi_B(x) \otimes \Phi_B(x)dx\right)^{-1} \int_a^b (f(x, t) + f_x(x + h/2, t)h/2)\Phi_B(x)dx.\end{aligned} \tag{28}$$

where the domain of influence for each node, r_j, should be around h, for when taking a h time step back, i.e. $U(t + h)$ depending on $U(t)$, the domain of dependence is h distant from $U(t)$ in the x axis. Hence we have used $r_j = 1.2 * h$ for all the following simluations.

The result for this simulation with $h = 0.02$, $N = 60$, $f = 0$ and x_I evenly distributed are shown in Figure (4). Initially the the solution is advected exactly, but the price for conservating energy and taking from outside of the domain of dependence soon is paid with the onset of complete instability. This behaviour does not change by decreasing h. To attempt to bring

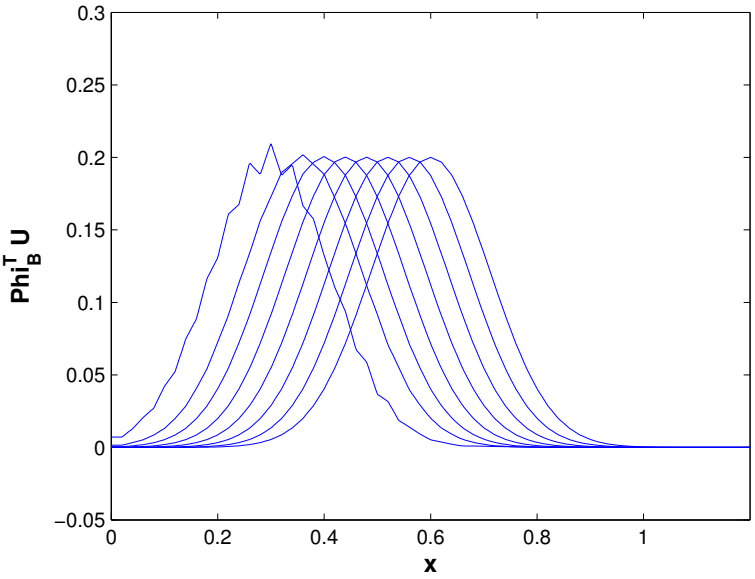

Figure 4: The evolution in time during 0.4s of the initial condition by scheme (28).

back some stability we can add some numerical dissipation. One way is to shift $u_x(x, t + h/2)$ to $u_x(x, t + h/2 + 0.3h)$. The results are shown in Figure 5 with $h = 0.1$. The good news is that the smaller h is the closer the approximation gets to the real solution. To demonstrate we compare the results of this dissipative scheme with the exact solution, which we take to be the result of the scheme given by equations (27). The result is shown in Figure 6 for $f(x) = 0.2\cos(x - x(N) + \pi/2)\sin(x)$ and $h = 0.005$.

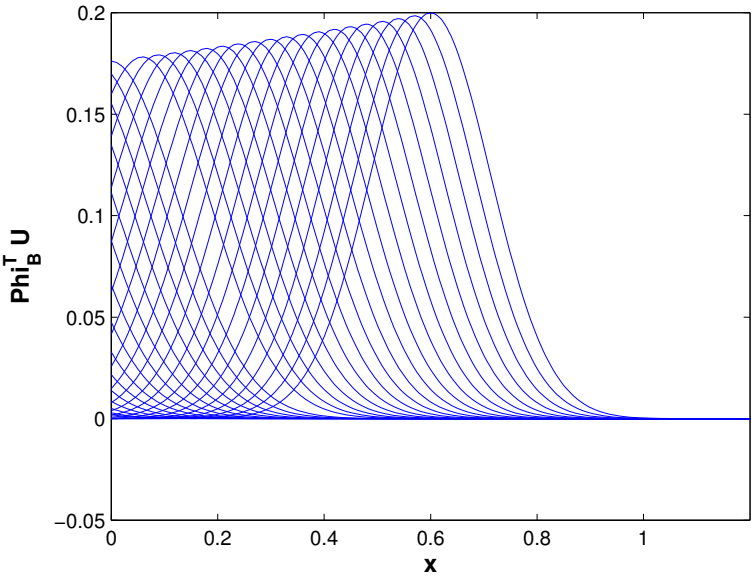

Figure 5: The evolution in time during 1s of the initial condition by adding dissipation to scheme (28).

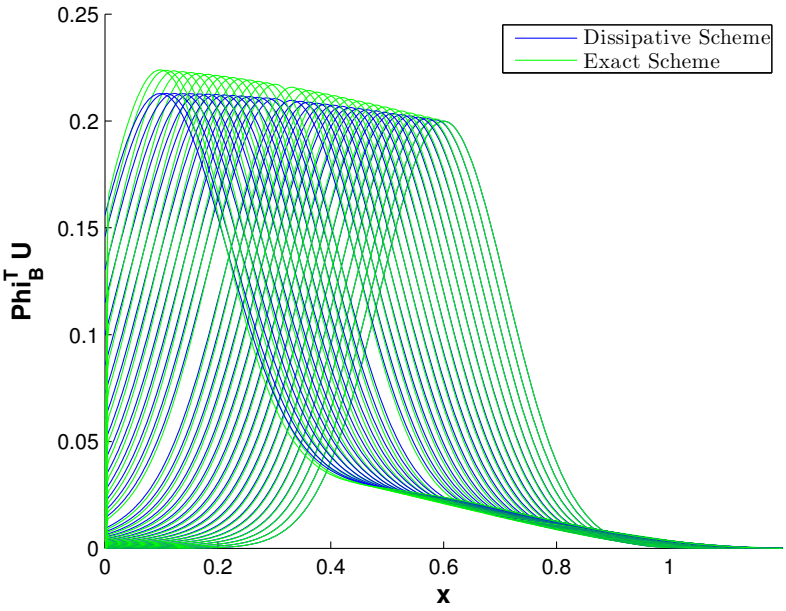

Figure 6: The evolution in time of the comparison of the exact solution and dissipative scheme during 0.5s.

References

[1] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, 139(14):3 – 47, 1996.

[2] Donald A. French and Todd E. Peterson. A continuous space-time finite element method for the wave equation. *Comput. Methods Appl. Mech. Engrg*, 107:145–157, 1996.

[3] Ohannes Karakashian and Charalambos Makridakis. Convergence of a continuous galerkin method with mesh modification for nonlinear wave equations. *Math. Comp.*

[4] David Levin. The approximation power of moving least-squares. Technical report, Math. Comp, 1998.

[5] M.B. Liu and G.R. Liu. Restoring particle consistency in smoothed particle hydrodynamics. *Applied Numerical Mathematics*, 56(1):19 – 36, 2006.