

SYMPLECTIC TIME INTEGRATION METHODS FOR THE MATERIAL POINT METHOD, EXPERIMENTS, ANALYSIS AND ORDER REDUCTION.

MARTIN BERZINS ¹

¹ University of Utah
Salt Lake City, UT 84112
mb@sci.utah.edu

Key words: Material Point Method Symplectic Integration Stage Error

Abstract. The provision of appropriate time integration methods for the Material Point Method (MPM) involves considering stability, accuracy and energy conservation. A class of methods that addresses many of these issues are the widely-used symplectic time integration methods. Such methods have good conservation properties and have the potential to achieve high accuracy. In this work we build on the work in [5] and consider high order methods for the time integration of the Material Point Method. The results of practical experiments show that while high order methods in both space and time have good accuracy initially, unless the problem has relatively little particle movement then the accuracy of the methods for later time is closer to that of low order methods. A theoretical analysis explains these results as being similar to the stage error found in Runge Kutta methods, though in this case the stage error arises from the MPM differentiations and interpolations from particles to grid and back again, particularly in cases in which there are many grid crossings.

1 INTRODUCTION

The Material Point Method (MPM) is often described as a solid mechanics method that is derived [16] from the fluid implicit particle, FLIP and PIC methods. MPM has been very successful when applied to very many large deformation problems. However some of the properties of the method are still not as well-understood as they might be in areas such as time integration. MPM time integration is usually undertaken with a Symplectic Euler (Euler-Cromer) method or some variant of Stormer-Verlet type methods. The relationship between MPM time integration and symplectic time integration methods is considered by [6]. Stormer-Verlet type methods are used by [17], [19] and many others. The energy conservation properties of Stormer-Verlet methods for MPM are considered by [5] and a more general survey is [10]. Overall the Stormer-Verlet method has third order accuracy locally, has good conservation properties, is very widely used in many applications such as molecular dynamics and planetary orbits and dates back to Newton as was demonstrated by Feynman see [10].

While such methods are sufficient for low-order versions of MPM, the use of higher order spline type methods in MPM is now increasingly popular following [15]. This naturally suggests the use of higher-order time integration methods such as the symplectic methods based upon Runge-Kutta-Nystrom methods e.g [8]. Some methods of this type were considered by [19] before being rejected in favor of a lower order method with the comment "A number of different time integration schemes have been investigated

for use with GIMP including Runge-Kutta, Runge Kutta Nystrom, Newmark methods and multistep methods. In the authors' experience few of these methods have been able to achieve their formal level of accuracy." Similarly [2] found no real improvements in using higher order spline methods in space.

In order to investigate this further a simple model vibrating bar problem used by [9], is considered with different time integration schemes and similar behavior to that seen by [19] found. However when a simple three component ordinary differential equations model is derived that has some of the structure of MPM, then the level of accuracy expected is seen. Adding sufficiently large errors consistent with the differentiation and interpolation errors found in MPM degrades the performance of all the methods to that of the low-order methods for this model problem.

The issue of whether or not it is the low-order accuracy in space that is problematic is considered by using high-order spline methods based upon the Matlab spline toolbox [12]. Again while the initial accuracy is good, longer time accuracy is not as good. Finally a simple theoretical analysis explains this reduced accuracy. While order-reduction in Runge-Kutta methods for partial differential equations is well understood going back to [18] and also applies to Runge-Kutta Nystrom Methods [1], the phenomenon observed here is not related to the boundary conditions (as zero Dirichlet conditions are used), but arises from the differentiation and interpolation errors that occur when moving from particles to the grid and back again. This also matches the results seen for displacement error by the author [5] where the Stormer-Verlet Method provides much better energy conservation the Symplectic Euler method but the displacement errors are very similar. The overall conclusion is that there is scope for using high-order methods in space and time with MPM but further investigation is needed in how to do so and when.

2 MPM MODEL PROBLEM

The now standard MPM model used here follows [9] in being a pair of equations connecting velocity v , displacement u , stress σ and density ρ (here assumed constant):

$$\frac{Du}{Dt} = v, \quad (1)$$

$$\rho \frac{Dv}{Dt} = \frac{\partial \sigma}{\partial x} + b(x, t), \quad (2)$$

with a linear stress model $\sigma = E \frac{\partial u}{\partial x}$ for which Young's modulus, E , is constant, a body force b , which is initially assumed to be zero, and with appropriate boundary and initial conditions. For convenience a mesh of equally spaced $N + 1$ fixed nodes X_i with intervals $I_i = [X_i, X_{i+1}]$, on the interval $[a, b]$ is used where

$$a = X_0 < X_1 < \dots < X_N = b, \quad (3)$$

$$h = X_i - X_{i-1}. \quad (4)$$

It will also be assumed that periodic boundary conditions exist, together with appropriate initial conditions.

3 THE MPM METHOD

In considering time integration methods for MPM, the model problem above will be used as all the analysis and derivation does apply more generally and in multiple space dimensions with obvious modi-

fications. It will also be assumed that are n_p particles initially between each pair of nodes, situated at x_p^n points where at each time step, $t^n = \delta t * n$, where n is the n th time step, and the computed displacement at the p th particles will be written as $u_p^n = u(x_p^n, t^n)$. The calculation of the internal forces in MPM at the nodes requires the calculation of the volume integral of the divergence of the stress [19] using

$$f_i^{int}(t) = - \sum_p D_{pi}(x_p^n(t)) \sigma_p(t) V_p(t) \quad (5)$$

In this case the subscript pi represents a mapping from particles p to node i while the subscript ip would represent a mapping from nodes i to particles p . The negative sign arises as a result of using integration by parts [9]. The mass at node i is defined by

$$m_i(t) = \sum_p m_p S_{pi}(x_p^n(t)) \quad (6)$$

The initial volume of the particles is uniform for the n_p particles in an interval. The particle volumes are defined using the deformation gradient, F_p^n , and the initial particle volume, V_p^0 ,

$$V_p^n = F_p^n V_p^0, \text{ where } V_p^0 = \frac{h}{n_p}, \text{ where } F_p^0 = 1 \quad (7)$$

The deformation update equation is

$$\dot{F}_p(t) = \frac{\partial v}{\partial x}(x_p(t), t) F_p \quad (8)$$

where the spatial velocity derivatives at particles are calculated using nodal velocities as

$$\frac{\partial v}{\partial x}(x_p(t)) = \sum_i D_{ip}(x_p(t)) v_i(t) \quad (9)$$

The equation for the particle position update is

$$\dot{x}_p(t) = v_p(t) \quad (10)$$

While the stress update equation is, using the appropriate constitutive model and Young's Modulus, E ,

$$\dot{\sigma}_p(t) = E \frac{\partial v}{\partial x}(x_p(t)) \quad (11)$$

From (5,6) the acceleration equation in the MPM method in this simple case is

$$a_i(t) = \frac{-1}{m_i} \sum_p D_{pi}(x_p(t)) \sigma_p(t) F_p(t) V_p^0 \quad (12)$$

The equation to update velocity at the nodes, as denoted by v_i^n is then given by

$$\dot{v}_i(t) = a_i(t) \quad (13)$$

The equation for the update of the particle velocity is then

$$\dot{v}_p(t) = a_p(t) \quad (14)$$

where the value of the acceleration at a point x_p^n is given by interpolation based upon nodal values of acceleration

$$a_p(t) = \sum_i S_{ip}(x_p(t)) a_i(t) \quad (15)$$

4 THE SYMPLECTIC EULER TIME INTEGRATION METHOD FOR MPM

In solving the system of equations defined above by equations (5) to (15) one approach used is to order the equations in a certain order (stress first) and then to solve them in turn using explicit methods. Differences in how the equations are solved corresponds to whether or not the stress is updated first or last in a timestep, a choice that is discussed at length by [3] and [7]. These two different choices are related to the use of the semi-implicit Euler A or B method, [6]. The nodal velocity is initially given by

$$v_i^n = \sum_p S_{pi}(x_p^n) \frac{m_p}{m_i} v_p^n \quad (16)$$

At the beginning of each subsequent step it is assumed that particle positions and velocities x_p^n and v_p^n , stresses and deformation gradients σ_p^n and F_p^n and nodal velocities v_i^n exist. While perhaps not optimal, the stress first approach is used as this fits with the form of symplectic higher order time integration methods such as [8]. The nodal values of velocity are used to calculate a velocity gradient that is used to update the stress and deformation gradients at particles

$$\frac{\partial v}{\partial x}(x_p^n) = \sum_i D_{i,p}(x_p^n) v_i^n \quad (17)$$

$$F_p^{n+1} = F_p^n + dt \frac{\partial v}{\partial x}(x_p^n) F_p^n \quad (18)$$

$$\sigma_p^{n+1} = \sigma_p^n + dt E \frac{\partial v}{\partial x}(x_p^n) F_p^n \quad (19)$$

These values of the stress and the deformation gradient are then used to define nodal accelerations using equations (12). The equation to update the velocities at the nodes is then given by

$$v_i^{n+1} = v_i^n + dt a_i^n \quad (20)$$

The value of the acceleration at a point x_p^n is given by interpolation based upon nodal accelerations as

$$a_p^n = \sum_i S_{ip}(x_p^n) a_i^n \quad (21)$$

The equations for the update of the particle velocities and positions are then

$$v_p^{n+1} = v_p^n + dt \sum_i S_{ip}(x_p^n) a_i^n \quad (22)$$

$$x_p^{n+1} = x_p^n + dt v_p^n \quad (23)$$

The Stormer Verlet time integration method is an alternative approach to time integration (See Hairer et al.[10]). Applying the method of p. 407 of [10] requires one extra step over the Symplectic Euler stress last or stress first methods [6, 3] to define both nodal and particle velocities, and solution values at particle points at half time step values, but has better energy conservation properties [5].

5 RUNGE-KUTTA NYSTROM METHODS

The form of these methods at each of s stages is similar to that of that Symplectic Euler A method or MPM stress-first approach. Let the values of particle velocity, position, stress and deformation gradient at the s th stage of the step to t_{n+1} from t_n be given by $v_p^{n+1,s}, x_p^{n+1,s}, \sigma_p^{n+1,s}, F_p^{n+1,s}$ and let the nodal velocity be $V_i^{n+1,s}$. Furthermore let the values at stage 0 be the values at the time step t_n .

For each of the stages we perform the following steps. Updated particle positions are calculated using

$$x_p^{n+1,s} = x_p^{n+1,s-1} + (c_j - c_{j-1}) dt v_p^{n+1,s-1} \quad (24)$$

The derivatives of velocity are used to calculate a velocity gradient that is used to update the stress and deformation gradients at particles

$$\frac{\partial v}{\partial x}(x_p^{n+1,s-1}) = \sum_i D_{i,p}(x_p^{n+1,s-1}) v_i^{n+1,s-1} \quad (25)$$

$$F_p^{n+1,s} = F_p^{n+1,s-1} + (c_j - c_{j-1}) dt \frac{\partial v}{\partial x}(x_p^{n+1,s-1}) F_p^{n+1,s-1} \quad (26)$$

$$\sigma_p^{n+1,s} = \sigma_p^{n+1,s-1} + (c_j - c_{j-1}) dt E \frac{\partial v}{\partial x}(x_p^{n+1,s-1}) F_p^{n+1,s-1} \quad (27)$$

These values are then used to calculate the nodal accelerations

$$a_i^{n+1,s} = \frac{-1}{m_i} \sum_p D_{pi}(x_p^{n+1,s}) \sigma_p^{n+1,s} F_p^{n+1,s} V_p^0 \quad (28)$$

The equation to update the velocities at the nodes is then given by

$$v_i^{n+1,s} = v_i^{n+1,s-1} + b_j dt a_i^{n+1,s} \quad (29)$$

The value of the acceleration at a point x_p^n is given by interpolation based upon nodal accelerations is

$$a_p^{n+1,s} = \sum_i S_{ip}(x_p^{n+1,s}) a_i^{n+1,s} \quad (30)$$

The equation for the update of the particle velocity is then

$$v_p^{n+1,s} = v_p^{n+1,s-1} + b_j dt \sum_i S_{ip}(x_p^{n+1,s}) a_i^{n+1,s} \quad (31)$$

A second order Runge-Kutta Nystrom method is given by $c_1 = 0, c_2 = 1.0, b_1 = 0.5$. The fourth order but five stage method of Calvo and Sanz Serna [8] is given by the coefficients in Table 1

Table 1: Coefficients of 4th Order RKN Method [8]

i	b_i	c_i
1	0.061758858135626325	0.0
2	0.338978026553643355	0.205177661542286386
3	0.614791307175577566	0.608198943146500973
4	-0.140548014659373380	0.487278066807586965
5	0.125019822794526133	1.0

6 COMPUTATIONAL EXPERIMENTS PROBLEM 1

In order to compare the time integration schemes the standard 1D bar problem used by [9] is employed, see [14], for further details. This problem is defined by:

$$\sigma = E \frac{\partial u}{\partial X} = E(F - 1), \quad (32)$$

where E is the Young's modulus. The analytic solution for displacement is:

$$u(X, t) = A \sin(2\pi X) \sin(c\pi t), \quad (33)$$

where $c = \sqrt{E/\rho_0}$, A is the maximum displacement While the body force normally used is,

$$b(X, t) = 3A(c\pi)^2 u(X, t) \quad (34)$$

here the value of E was multiplied by 0.25 instead of using the body force. The periodic nature of the analytic solution means that periodic zero Dirichlet boundary conditions are used. The initial conditions for the updated Lagrangian description of the particles are as in previous experiments [9]. The cell width is $h = 0.05$, the material density is $\rho_0 = 1$ and the time interval is from $t = 0$ to $t = 1$. The initial spatial discretization uses two evenly spaced particles per cell with the spatial domain being $[0, 1]$. The Young's modulus values are $E = 16, 256, 1000$ and $10,000$. The maximum displacement is $A = 0.05$ and the time step values used are $dt = 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}$. For the values given of A and E there are 2K, 8K, 16K and 57K particles crossing grid nodes. All the errors shown are calculated using the analytical solution and the calculated solution with an infinity norm being used.

Experiments were undertaken with the MPM GIMP method [4] The experiments used a fixed time step of $1e - 4$ as shown in Table 1 a value of * indicates that particles leave the domain and integration is halted. The time integration methods used were as follows

- GIMP Method with an equal number of steps to that of the sub-steps in RKN4 below
- RKN 4th order by Calvo and Sanz-Serna [8]

Experiments were also conducted with a number of other methods such as RKN 3rd order methods. From these results there does appear to be a slight advantage in using a higher order time integration method, until the last case where the integration blows up. Nevertheless the improvement is slight. In what follows this observation will be explained.

Table 2: Representative Results $dt = 1.0e-4$

A	E	EulerA U Error	Euler A σ Error	RKN4 U error	RKN4 σ Error
5e-2	16	8.0e-4	4.5e-1	1.7e-3	6.4-1
5e-2	256	1.0e-2	2.5e+1	1.5e-3	1.6e+2
5e-2	1000	6.2e-3	1.0e+2	5.3e-3	1.0e+2
5e-2	10000	1.5e-2	1.7e+3	*	*

7 MODEL PROBLEM 2

In order to understand and model the results in Table 2 a system of three odes that mimics the form of the MPM equations was constructed. The three equations are defined by:

$$\frac{dx}{dt} = v(t) \quad (35)$$

$$\frac{d\sigma}{dt} = 3E\pi v(t) \quad (36)$$

$$\frac{dv}{dt} = -\sigma(t)\pi + 2\pi^2 Ex(t) \quad (37)$$

The analytic solution to these equations is given by

$$x(t) = \sin(C\pi t), C = 8, 16 \text{ or } 32 \quad (38)$$

$$v(t) = C\pi \cos(C\pi t) \quad (39)$$

$$\sigma(t) = C^3 \pi^2 \sin(C\pi t) \quad (40)$$

The application of symplectic methods preserves the Hamiltonian given by

$$H(x, v, \sigma) = v^2 + \frac{\sigma}{3E} - 2E\pi^2 x^2 = E\pi^2 \quad (41)$$

For example, applying a symplectic Euler method gives the equation:

$$v^{n+1} = v^n - dt(\sigma^n - 2E\pi^2 x^n) \quad (42)$$

$$x^{n+1} = x^n + dt(v^{n+1} + IE) \quad (43)$$

$$\sigma^{n+1} = \sigma^n + dt3E\pi(v^{n+1} + IE) \quad (44)$$

In this case the extra term IE is used as a model for the error in velocity both from interpolation and differentiation in the full MPM Method and when $IE = 0$ the analytic solution is the exact solution to the ODE system. This method approximately preserves $H(x, y)$. The Stormer-Verlet method and the RKN method can be similarly perturbed for this model problem by adding IE to the velocity once created.

The results in Table 3 with $IE = 0$ show that the RKN4 method does indeed give superior results to the Stormer Verlet method in terms of both Velocity and Hamiltonian error. Table 4 however shows that as the model term, IE , for the velocity error is increased then when the error is $O(dt^2)$ only, the methods perform as expected. When the IE term is larger then the methods all perform identically. The conclusion perhaps is that when the velocity error is large, then it is not clear that higher order time integration is useful.

Table 3: Global Maximum Errors in MPM model IE = 0

Method dt/Errors	Stormer-Verlet Velocity	RKN4 Velocity	Stormer-Verlet Hamiltonian	RKN4 Hamiltonian
1.0e-2	1.57	1.7e-4	1.0	1.7e-3
1.0e-3	1.5e-2	1.7e-8	1.0e-1	1.87e-7
1.0e-4	1.6e-4	5.6e-11	1.0e-3	6.3e-11
1.0e-5	1.6e-6	1.0e-9	1.0e-5	1.8e-10
1.0e-6	1.2e-8	3.4e-9	1.0e-7	4.1e-10

Table 4: Global Maximum Errors in MPM model dt = 1.0e-4 with varying IE

Method Error	Euler A Velocity	StorVer Velocity	RKN4 Velocity	Euler A Hamiltonian	StorVer Hamiltonian	RKN4 Hamiltonian
IE=1.0e-8	3.2e-2	1.6e-4	2.0e-8	7.9e-1	4.0e-3	2.0e-8
IE=1.0e-4	3.2e-2	2.7e-4	2.0e-4	8.0e-1	1.0e-2	1.0e-1
IE=1.0e-2	4.3e-2	2.0e-2	2.0e-2	1.7	1.0	1.0

8 HIGH ORDER SPLINES MPM AND RESULTS ON PROBLEM 3

As the previous section shows that it is possible to achieve higher order accuracy by reducing velocity errors, one approach might be to use higher order methods spatially. While quadratic or cubic spline methods have been used with MPM, they do not always appear to deliver better accuracy, [2]. It is now easy to use the Matlab spline toolbox to introduce higher order splines still, albeit with a slightly different formulation when calculating the acceleration. The following steps are used in a prototype algorithm that here is described in a stress-last form.

1. The Matlab spline route *spapi* is used to create a high order spline through the stress values at the current particle points. This interpolant is differentiated using *fnder* and evaluated at nodes using *fnval* to calculate nodal accelerations in what amounts to a strong form of the differential equation as opposed to the usual weak form that uses integration by parts.
2. The nodal accelerations are used to increment nodal velocities and interpolated back to particles to create particle accelerations. These in turn are used to create updated particle velocities.
3. A spline based on the nodal velocities is differentiated using *fnder* to calculate velocity derivatives at particles. The velocity derivatives are used to increment the particle stresses and deformation gradients.

A stress-last algorithm is similarly and easily derived. While the use of the Matlab toolbox is clearly not particularly efficient it does allow ideas to be quickly evaluated.

The use of the higher order spline approach on the model MPM problem described in Section 6 but now with $A = .00725$ and $E = 64$ with $dt = 1.0e - 3$ or $dt = 1.0e - 4$ has almost no grid crossing and Stormer-Verlet time stepping yields the results in Table 5. With a spline approach, the improved Verlet scheme of [13] (eqn. 15) is used as it gives slight accuracy improvements. In this smooth case there is a clear accuracy advantage in using a fifth order spline method.

Table 5: Displacement Errors in MPM Problem , $np = 4$

Method	GIMP dt = 1e-3	GIMP dt=1e-4	Cubic Spline dt = 1e-3	Quintic Spline dt = 1.0e-3	Quintic dt = 1e-4
Time	U Error	U Errors	U Error	U Error	U Error
1e-3	4.5e-8	2.7e-8	2.6e-11	1.5e-13	9.6e-14
1e-2	1.7e-6	1.3e-7	8.6e-9	7.5e-10	8.8e-10
2e-2	5.7e-6	2.6e-7	2.1e-8	1.2e-8	1.8e-8
1e-1	4.3e-5	3.2e-6	3.7e-7	1.1e-6	1.0e-6
2e-1	1.2e-5	1.6e-6	3.8e-7	1.6e-6	1.6e-6
5e-1	5.4e-6	9.3e-7	1.3e-6	1.5e-7	2.0e-8
1.0	9.3e-6	1.8e-7	2.7e-6	3.2e-7	4.3e-8

9 PROTOTYPE HIGH ORDER SPLINES without MPM AND RESULTS ON PROBLEM 3

In order to investigate the effect of that part of the error error due to interpolation, a version of the spline approach without the MPM mapping to the nodes was used. As in the previous experiment, the following steps are used in a prototype algorithm that here is described in a stress-last form.

1. The Matlab spline route *spapi* is used to create a high order spline through the stress values at the current particle points. This interpolant is differentiated using *fnder* and evaluated at particles using *fnval* to calculate nodal accelerations in what amounts to a strong form of the differential equation as opposed to the usual weak form with integration by parts.
2. The particle accelerations are used to increment particle velocities.
3. A spline based on the particle velocities and zero velocities at the boundaries is differentiated using *fnder* to calculate velocity derivatives at particles. The velocity derivatives are used to increment the particle stresses and deformation gradients.
4. In this case the standard classic Runge Kutta fourth order method is used with cubic and quintic splines.

The results in Table 6 show that interpolation makes some difference but is not in this case significant when the results of Tables 5 and 6 are compared. A more challenging problem would seem to be the lack of smoothness on the solution whether or not an MPM formulation is used. This explored below by reconsidering the problem in Section 6 with a higher resolution grid and more particles per element.

10 RESULTS FOR THE PROBLEM IN SECTION 6

In order investigate if the errors in Table 2 are due to poor spatial resolution, four particles per interval $np = 4$ and a smaller spatial interval $h = 0.0125$ were used. The results in 8K, 32K, 64K and 220K grid crossings as E is increased. The results are shown in Table 7 and show that while the spline methods perform well for small values of E this does not happen with large values of E and a large number of "grid crossings" indicating considerable particle movement.

Table 6: Displacement Errors in Problem 3, No MPM nodes, np = 4

Method	Cubic Spline dt=1.0e-3	Quartic Spline dt=1e-4
Time	U Error	U Error
1e-3	9.3e-14	9.2e-14
1e-2	9.6e-10	9.0e-10
2e-2	1.4e-8	1.4e-8
1e-1	1.1e-6	1.0e-6
2e-1	1.6e-6	1.6e-6
5e-1	1.7e-7	1.6e-8
1.0	3.5e-7	3.2e-8

11 INTERNAL STAGE ERRORS IN TIME INTEGRATION

In order to understand how the displacement errors grow it is sufficient to examine what happens inside a single time step and to consider the three equations (20),(22)and (23). The error in the nodal velocity ev_i^{n+1} then depends on the error in the nodal acceleration ea_i^n as follows:

$$ev_i^{n+1} = dte a_i^n \quad (45)$$

There is then an error in the particle acceleration, ea_p^{n+1} , when the value of the acceleration at a particle point x_p^n is given by interpolation based upon nodal accelerations given by

$$ea_p^{n+1} = \sum_i S_{ip}(x_p^n) ea_i^n + IeA_{ip} \quad (46)$$

where the underlying interpolation error of the method using the coefficients S_{ip} is denoted by IeA_{ip} . The error in the update of the particle velocity and positions is then

$$ev_p^{n+1} = dt(\sum_i S_{ip}(x_p^n) a_i^n + IeA_{ip}) \quad (47)$$

and so the local displacement error over one time step ex_p^{n+1} is given by

$$ex_p^{n+1} = dte v_p^{n+1} \quad (48)$$

or as

$$ex_p^{n+1} = dt^2 ea_p^{n+1} \quad (49)$$

Which is $O(dt^2)$ regardless of the spatial accuracy in the terms ea_i^n and IeA_{ip} . Standard theory from the numerical integration of initial value problems tells us that the global displacement error must be $O(dt)$. This situation is similar to the order reduction seen when Runge-Kutta or Runge-Kutta Nystrom methods are applied to the solution of partial differential equations, see [1, 11, 18]. There is a fundamental difference in that in those cases the reduction in order depends on the boundary conditions and does not happen for zero Dirichlet boundary conditions such as those used here. In the case considered here the

Table 7: Representative Results $dt = 1.0e-4$

A	E	EulerA GIMP U Error	EulerA GIMP σ Error	RKN4 GIMP U error	RKN4 GIMP σ Error	RKnoMPM Cubic U error	RKnoMPM Cubic σ error
6.25e-3	16	3.1e-6	2.6e-3	1.7e-4	2.3e-2	3.5e-6	3.5e-4
6.25e-3	256	9.6e-6	5.8e-2	4.6e-5	1.4e-1	1.4e-5	3.8e-2
6.25e-3	1000	4.4e-5	7.2e+1	2.4e-4	2.0	1.4e-3	2.3e+1
6.25e-3	10000	8.6e-5	9.7	1.0e-5	8.2	*	*

overall error arises from interpolation and differentiation errors and the way that they are propagated through each time step or each stage of a time step. From [1, 18] the same analysis as shown above may be applied to the Stormer-Verlet method and to each stage of the Runge-Kutta Nystrom methods considered above. The obvious conclusion is that if the interpolation and differentiation errors are too large they will remove the need for accurate time integration methods, exactly as is shown in the simple example of Section 7. There is no real advantage in using the spline method particularly with $E = 10000$ when particles exit the mesh and integration is halted.

12 CONCLUSIONS

The results here show that it is more challenging to make use of high order methods with MPM than might be thought. For problems with comparatively little particle movement higher order methods work well and provide increased accuracy in space and time. In the case of problems with more grid movement performance is not so good however and needs to be investigated further. It is clear if there is a large spatial error then there is little point in using high-order time integration. While this observation is not new for partial differential equations, the additional complexity in MPM arising from moving particles, differentiation and interpolation from particles to nodes considerably complicates matters.

ACKNOWLEDGEMENTS

Chris Gritton is thanked for the use of his code for the model problem. Lisa Wobbes is thanked for supplying code which alas I wasn't able to use here. As usual, Jim Guilkey made some very helpful observations. This research was partially sponsored by the Army Research Laboratory under Cooperative Agreement Number W911NF-12-2-0023. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies of the Army Research Laboratory or the U.S. Government.

REFERENCES

- [1] Alonso-Mallo I. , Cano B., and Moreta M.J., Order reduction and how to avoid it when explicit Runge–Kutta–Nyström methods are used to solve linear partial differential equations. *J. of Computat. and Appl. Math.* (2005) **176**: 293-318.
- [2] Andersen S. and Andersen L. Analysis of spatial interpolation in the material-point method. *Computers & Structures* (2010) **88**: 7–8 506-518.
- [3] Bardenhagen S., Energy conservation error in the material point method for solid mechanics. *J. of*

- Computat. Phys.* (2002) **180**: 383-403.
- [4] Bardenhagen s. and Kober E., The generalized interpolation material point method. *CMES* (2004) **5**: 477-495.
 - [5] Berzins M., Time Integration Errors and Energy Conservation Properties of the Stormer Verlet Method Applied to MPM, *Proc. of VI Int. Conf. on Particle-based Methods– Fundamentals and Applications*, Barcelona, Eds E. Onate et al., PARTICLES 2019.
 - [6] Berzins M. “Nonlinear stability and time step selection for the MPM method.” *Computational Particle Mechanics* (2018) **5**: 455–466.
 - [7] Buzi O., Pedrosa D.M. and Giacomini A. Caveats on the Implementation of the Generalized Material Point Method. *CMES.* (2008) **31**: 85-106.
 - [8] Calvo M.P. and SanzSerna J.M., Symplectic numerical methods for Hamiltonian problems. *Int. J. Modern Physics C* (1993) **4** 385-392.
 - [9] Gritton C.E. and Berzins M., Improving Accuracy In the MPM Methods by Using a Null Space Filter, *Computational Particle Mechanics* (2017) **4**: 131-142.
 - [10] Hairer E., Lubich C., and Wanner G., Geometric numerical integration illustrated by the Stormer-Verlet Method. *Acta Numerica* (2003) 399-450, Cambridge University Press.
 - [11] Ketcheson D., Loczi L. and Parsani M. Internal Error Propagation in Explicit Runge-Kutta Methods. *SIAM J. Numer. Anal.* (2014) **52**: 2227-2249.
 - [12] <https://www.mathworks.com/help/curvefit/curve-fitting-toolbox-splines-and-matlab-splines.html>
 - [13] Omelyan I.P., Mryglod I.M., Folk R. Optimized Verlet-like algorithms for molecular dynamics simulation. *Phys. Rev. E.*(2002) **65** 056706 1-5.
 - [14] Steffen M., Wallstedt P.C., Guilkey J.E. , Kirby R.M. and Berzins M., Examination and analysis of implementation choices within the Material Point Method (MPM). *CMES* (2008) **31** 2 107-127.
 - [15] Steffen M., Kirby M., and Berzins M. Analysis and reduction of quadrature errors in the material point method (MPM). *Int. J. Numer. Meth. Engng* (2008) **76**:922–948
 - [16] Sulsky D., Chen Z. and Schreyer H.L. A particle method for history-dependent materials. *Computer Methods in Appl. Mech. and Engng.* (1994) **118**: 179-196.
 - [17] Sulsky D., Schreyer H., Peterson K., Kwok R. and Coon M. Using the material-point method to model sea ice dynamics. *J. of Geophys. Research*, (2007) **112** C02S90.
 - [18] Sanz-Serna J.M., Verwer J.G. and Hundsdorfer W.H. Convergence and order reduction of Runge-Kutta schemes applied to evolutionary problems in partial differential equations. *Numerische Mathematik* (1986) **50**: 405–418.
 - [19] Wallstedt P.C. and Guilkey J.E., An evaluation of explicit time integration schemes for use with the generalized interpolation material point method. *J.of Computat. Phys.* (2008) **227** 22 9628-9642.