Computational Particle Mechanics
https://doi.org/10.1007/s40571-021-00457-3

# Energy conservation and accuracy of some MPM formulations

**Martin Berzins**[1]

Received: 30 July 2021 / Revised: 20 November 2021 / Accepted: 28 December 2021


**Abstract**
The success of the material point method (MPM) in solving many challenging problems nevertheless raises some open questions regarding the fundamental properties of the method such as time integration accuracy and energy conservation. The traditional MPM time integration methods are often based upon the symplectic Euler method or staggered central differences. This raises the question of how to best ensure energy conservation in explicit time integration for MPM. Two approaches are used here, one is to extend the symplectic Euler method (Cromer Euler) to provide better energy conservation and the second is to use a potentially more accurate symplectic methods, namely the widely used Stormer–Verlet method. The Stormer–Verlet method is shown to have locally third-order time accuracy of energy conservation in time, in contrast to the second order accuracy in energy conservation of the symplectic Euler methods that are used in many MPM calculations. It is shown that there is an extension to the symplectic Euler stress-last method that provides better energy conservation that is comparable with the Stormer–Verlet method. This extension is referred to as TRGIMP and also has third-order accuracy in energy conservation. When the interactions between space and time errors are studied it is seen that spatial errors may dominate in computed quantities such as displacement and velocity. This connection between the local errors in space and time is made explicit mathematically and explains the observed results that displacement and velocity errors are very similar for both methods. The observed and theoretically predicted third-order energy conservation accuracy and computational costs are demonstrated on a standard MPM test example.

**Keywords** MPM · Time integration · Stormer–Verlet method · Energy conservation accuracy

## 1 Introduction

The material point method (MPM) is often described as a solid mechanics method that is derived [17,18] 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 and conservation of energy. For example, energy conservation is considered by Bardenhagen [3] and it is shown that the standard MPM formulation gives second-order energy conservation over a timestep or first-order overall. The analysis of Love and Sulsky [12] extends these results and shows that energy conservation is possible if a full mass matrix is used. The same authors also show that using a lumped mass matrix gives second-order energy conservation. Other similar improved time integration methods based upon a central difference approach are considered by [19,21]. The relationship between MPM time integration and symplectic time integration methods is considered by [5]. Such symplectic methods have good conservation properties [10]. Furthermore, the Stormer–Verlet [10] method has third-order accuracy locally. This method is symplectic and very widely used in many applications [11] such as molecular dynamics and planetary orbits and dates back to Newton as discussed by Feynman, see [10]. The intention here is to apply the Stormer–Verlet method to MPM and to determine its accuracy and conservation properties and to compare them with the stress-last MPM integration approach. This comparison motivates an extension to the stress-last method that uses an idea of Weiss and Guilkey [22] and improves the energy conservation properties of the method. This paper was motivated by a conference paper [6], which used one of the two possible forms of the Stormer–Verlet method and which is partly implicit. In this paper, another form of the Stormer–Verlet method is used which is completely explicit.

✉ Martin Berzins
mb@sci.utah.edu
http://www.sci.utah.edu

[1] SCI Institute, University of Utah, Salt Lake City, USA

Published online: 04 February 2022

The spatial and temporal sources of error for this Stormer–Verlet method and the stress-last method and its extension are compared in detail. It is well known that with MPM the spatial error often dominates time accuracy for a stable time step [16]. In order to understand the impact of this, when more accurate time integration methods are used, an analysis is presented of the local space and time errors. This analysis shows that while the energy conservation properties of the Stormer–Verlet scheme are superior to that of symplectic Euler, the spatial errors present may dominate in the errors in displacement and velocity, thus negating any possible improved accuracy in those variables.

Section 2 describes the MPM formulation and the model problem used, and describes the individual space and time errors observed. Sections 3 and 4 explain how the Stormer–Verlet method may be applied to MPM and provides an analysis of the time stepping errors of the Stormer–Verlet method. Section 5 uses one of the ideas from the Stormer–Verlet method that was also used by Guilkey and Weiss [22] to extend the stress-last method to have better energy conservation properties. This new method is referred to as TRGIMP in this paper on account of its trapezoidal-rule-like correction Sect. 6 introduces the central difference method used by [19,21] while Sect. 7 compares the spatial errors of the symplectic Euler and Stormer–Verlet method. In Sect. 8, the magnitudes of some of the spatial errors are shown and the spatial errors present in the different time integration methods compared. Sections 9 and 10 derive the energy conservation error of the Stormer–Verlet and TRGIMP methods. Although a model one-dimensional problem is considered here the theoretical results are more broadly applicable to higher dimensions. The challenge of extending these ideas to multiple dimensions and more general constitutive models is considered in Sect. 11. Finally, Sect. 12 describes the computational experiments undertaken to compare the approaches on a model problem used by [9] and compares the accuracies of the different approaches with their computational costs. These experiments show that although the Stormer–Verlet method and the TRGIMP method have better conservation properties than the method considered in [3], the errors in displacement are almost identical. The computational cost of achieving higher-order energy conservation is about a 50% increase in computer time.

## 2 MPM model problem and method

The description of MPM used here follows [9] in that the model problem used here is a pair of equations connecting velocity $v$, displacement $u$ and density $\rho$ (here assumed constant):

$$\frac{Du}{Dt} = v, \tag{1}$$

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

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. While a simple linear stress model is considered here, the extension to more general models is briefly considered in Sect. 11.

For convenience, a mesh of equally spaced $N+1$ fixed nodes $X_i$ with intervals $I_i = \left[X_i, X_{i+1}\right]$, on the interval $[a,b]$ is used where

$$a = X_0 < X_1 < \cdots < X_N = b, \tag{3}$$

$$h = X_i - X_{i-1}. \tag{4}$$

These fixed nodes are referred to as the $i$ points. It will also be assumed that periodic boundary conditions exist in that

$$\sigma(a)v(a) = \sigma(b)v(b) \tag{5}$$

together with appropriate initial conditions. While the analysis of MPM for time integration error and energy conservation uses the model problem above it does apply more generally and in multiple space dimensions with a few obvious modifications, as discussed in Sect. 11. The computed solution at the $p$th particles will be written as $u_p^n = u(x_p^n, t^n)$. Suppose that the particles in interval $i$ lie between $X_i$ and $X_{i+1}$ and have positions $x_{im+j}$, $j = 1,..,m$. The calculation of the internal forces in MPM at the nodes requires the calculation of the volume integral of the divergence of the stress [21] using

$$f_i^{int} = -\sum_p D_{pi}(x_p^n)\sigma_p^n V_p^n \tag{6}$$

The subscript $pi$ represents a mapping from particles $p$ to node $i$, while the subscript $ip$ represents 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

$$m_i = \sum_p m_p S_{pi}(x_p^n) \tag{7}$$

It is important to note that the coefficients $D_{pi}(x_p^n)$ and $S_{pi}(x_p^n)$ (which here will be abbreviated to $D_{pi}^n$ and $S_{pi}^n$) depend explicitly on the background mesh and the particle positions and that they also may be chosen to reproduce derivatives of constant and linear functions exactly [9]. 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, $V_p^0$,

$$V_p^n = F_p^n V_p^0, \quad \text{where} \quad V_p^0, = \frac{h}{n_p}, \text{ and } F_p^0 = 1 \tag{8}$$

From (6) and (8), the continuous form of the acceleration equation in the MPM method in this simple case is

$$a_i(t) = \frac{-1}{m_i} \sum_p D_{pi}(x_p(t)) \sigma_p(t) F_p(t) V_p^0 \tag{9}$$

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

$$\dot{v}_i = a_i \tag{10}$$

The equation for the update of the particle velocity is

$$\dot{v}_p = a_p \tag{11}$$

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

$$a_p = \sum_i S_{ip}(x_p(t)) a_i \tag{12}$$

The equation for the particle position update is

$$\dot{x}_p = v_p \tag{13}$$

The update of the deformation gradients is given using,

$$\frac{\partial v}{\partial x}(x_p(t)) = \sum_i D_{ip}(x_p(t)) v_i \tag{14}$$

The deformation update equation is

$$\dot{F}_p = \frac{\partial v}{\partial x}(x_p(t), t) F_p \tag{15}$$

Finally, the stress update equation is using the appropriate constitutive model and Young's modulus, $E$,

$$\dot{\sigma}_p = E \frac{\partial v}{\partial x}(x_p(t)) \tag{16}$$

## 3 Stress-last GIMP MPM and local errors in space and time

In solving the system of equations defined above by Eqs. (6) to (16), one standard approach used is to order the equations in a certain order and then to solve them in turn using explicit methods [4]. 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 [8]. These two different choices are related to the use of the semi-implicit Euler A or B method [11], [5]. Following Bardenhagen [3], it is preferable to increment stress last. In this case, it is assumed that at time $t^n$ a consistent set of particle positions $x_p^n$, velocities $v_p$, stresses $\sigma_p^n$ and deformation gradients $F_p^n$ are available. The nodal velocity $v_i$ is calculated using

$$v_i^n = \sum_p S_{pi}^n \frac{m_p}{m_i} v_p^n \tag{17}$$

The nodal acceleration is updated by using the values of the stresses and deformation gradients at the current particle positions

$$a_i^n = \frac{-1}{m_i} \sum_p D S_{pi}^n \sigma_p^n F_p^n V_p^0, \tag{18}$$

where the nodal mass is defined by Eq. (7). The equation to update velocity at the nodes is then given by

$$v_i^{n+1} = v_i^n + dt a_i^n. \tag{19}$$

The local error in this forward Euler step is given by

$$lev_i^{n+1} = \frac{dt^2}{2} \frac{da_i^n}{dt} + dt E a_i^n \tag{20}$$

where $\frac{dt^2}{2} \frac{da_i^n}{dt}$ is the standard local error defined as the difference between the true local solution and the computed solution [11] and $Ea_i^n$ is the spatial error from using the approximation in Eq. (18), see Sect. 8.

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

$$a_p^n = \sum_i S_{ip}^n a_i^n \tag{21}$$

While there is no new temporal error here the error in acceleration $Ea_i^n$ is interpolated too and an additional interpolation error associated with the coefficients $S_{ip}^n$ used in Eq. (21) is introduced, as denoted by $EI_{ip}^n$, see Sect. 8. The equation for the update of the particle velocity is then:

$$v_p^{n+1} = v_p^n + dt a_p^n. \tag{22}$$

The associated local error is then given by

$$lev_p^{n+1} = \frac{dt^2}{2} \frac{d^2 v_p}{dt^2} + dt E I_{ip}^n + dt \sum_i S_{ip}^n E a_i^n \tag{23}$$

The velocity gradients at particles are calculated using the formula

$$\frac{\partial v^{n+1}}{\partial x}(x_p) = \sum_i D^n_{ip} v^{n+1}_i \tag{24}$$

with an associated differentiation error denoted by $Ev^{i+1}_{xp}$, see Sect. 8. These velocity gradients are used to update the stress and deformation gradients at particles

$$F^{n+1}_p = F^n_p + dt \frac{\partial v^{n+1}}{\partial x}(x^n_p, t_n) F^n_p dt \tag{25}$$

For the deformation gradient, $leF^{n+1}_p$ is the local time and space error given by

$$leF^{n+1}_p = \frac{dt^2}{2}\frac{d^2F}{dt^2} + dt F^n_p \left( Ev^n_{xp} + \sum_i D^n_{ip} lev^{n+1}_i \right) \tag{26}$$

While stress is updated using the appropriate constitutive model and Young's modulus, $E$,

$$\sigma^{n+1}_p = \sigma^n_p + dt E \frac{\partial v^{n+1}}{\partial x}(x^n_p, t_n) \tag{27}$$

In this case $le\sigma^{n+1}_p$ is the stress local time and space error given by

$$le\sigma^{n+1}_p = \frac{Edt^2}{2}\frac{d^2\sigma}{dt^2} + dtE \left( Ev^n_{xp} + \sum_i D^n_{ip} lev^{n+1}_i \right) \tag{28}$$

The equation for the particle position update is

$$x^{n+1}_p = x^n_p + dt v^{n+1}_p. \tag{29}$$

For this particle update, the local error, as denoted by $lex^{n+1}_p$, is given by

$$lex^{n+1}_p = \frac{dt^2}{2}\frac{d^2x_p}{dt^2} + dtlev^{n+1}_p \tag{30}$$

This analysis while deliberately not yet making explicit the magnitude of the spatial errors (see Sect. 8) shows how the spatial and temporal errors interact over the course of a time step.

## 4 The Stormer–Verlet time integration method

The Stormer–Verlet method is an alternative time integration approach to the symplectic Euler method used above, see Hairer et al. [10]. There are two possible Stormer–Verlet methods to use. The focus here is on the method of p. 407, equations (1.24) of [10] that requires one extra step over the Symplectic Euler stress last or stress first methods. For the system of equations defined by the o.d.e.s

$$\frac{dv}{dt} = f(v, q) \tag{31}$$

$$\frac{dq}{dt} = g(v, q) \tag{32}$$

this method is given by

$$v^{n+1/2} = v^n + \frac{dt}{2} f(v^{n+1/2}, q^n) \tag{33}$$

$$q^{n+1} = q^n + \frac{dt}{2}(g(v^{n+1/2}, q^n) + g(v^{n+1/2}, q^{n+1})) \tag{34}$$

$$v^{n+1} = v^{n+1/2} + \frac{dt}{2} f(v^{n+1/2}, q^{n+1}) \tag{35}$$

The challenge then is how to assign the MPM variables to the Stormer–Verlet variables $v$ and $q$. There are at least two obvious possible mappings. In [6] $v$ in [10] corresponds to $v_i, v_p, x_p$ and $\partial v_p/\partial x$ while the quantity $q$ in [10] corresponds to $\sigma_p, F_p$. This gives an implicit formulation that is linearized in [6]. The alternative is to define $v$ in [10] corresponding to $v_i, v_p$ and $\partial v_p/\partial x$ while the quantity $q$ corresponds to $x_p, \sigma_p, F_p$. This gives a fully explicit method and is the approach used here. It is also worth noting that the same method is called a generalized leapfrog method on page 156 of [11].

## 5 Applying the Stormer–Verlet time integration method to MPM

The Stormer–Verlet approach described in the previous section will now be applied to MPM. On the very first step, the nodal accelerations and velocities have to be calculated using

$$v^n_i = \sum_p S^n_{pi} \frac{m_p}{m_i} v^n_p \tag{36}$$

$$a^n_i = \frac{-1}{m_i} \sum_p D^n_{pi} \sigma^n_p F^n_p V^0_p \tag{37}$$

where the nodal mass values are calculated as in Eq. (7). The equation to update velocity at the nodes is then given by

$$v_i^{n+1/2} = v_i^n + \frac{dt}{2} a_i^n \tag{38}$$

In this case, if this error is denoted as $LEv_i^{n+1/2}$ then its value is given by the standard Euler's method local error with time step $dt/2$

$$LEv_i^{n+1/2} = \frac{dt^2}{8}\frac{d^2V_p}{dt^2} + \frac{dt}{2} Ea_i^n \tag{39}$$

where $Ea_i^n$ is the approximation error in the acceleration formula defined below in Sect. 8. The value of the acceleration at a point $x_p^n$ is given by interpolation based upon nodal accelerations as

$$a_p^n = \sum_i S_{ip}^n a_i^n \tag{40}$$

The equation for the update of the particle velocity to halfway across the step is then

$$v_p^{n+1/2} = v_p^n + \frac{dt}{2} a_p^n \tag{41}$$

The local error in this velocity approximation at the particles is

$$LEv_p^{n+1/2} = \frac{dt^2}{8}\frac{d^2a_p}{dt^2} + \frac{dt}{2} EI_{ip}^n + \frac{dt}{2}\sum_i S_{ip}^n Ea_i^n \tag{42}$$

The particle position values at the end of step are given by

$$x_p^{n+1} = x_p^n + dt\, v_p^{n+1/2} \tag{43}$$

For this particle update, the local error as denoted by $LEx_p^{n+1}$ is given by

$$LEx_p^{n+1} = \frac{dt^3}{24}\frac{d^3x_p}{dt^3} + dt\, LEv_p^{n+1/2} \tag{44}$$

where the first term corresponds to the local error of the midpoint method. The velocity gradients at particles using the differentiation matrix at the start and end of the step used below are calculated using

$$\frac{\partial v}{\partial x}(x_p^n) = \sum_i D_{ip}^n v_i^{n+1/2} \tag{45}$$

$$\frac{\partial v}{\partial x}(x_p^{n+1}) = \sum_i D_{ip}^{n+1} v_i^{n+1/2} \tag{46}$$

again with approximation errors $Ev_{xp}^n$ and $Ev_{xp}^{n+1}$as defined in Sect. 8. These velocity gradients are used to update the stress and deformation gradients at particles

$$F_p^{n+1} = F_p^n + \frac{dt}{2}\left(\frac{\partial v}{\partial x}(x_p^n)F_p^n + \frac{\partial v}{\partial x}(x_p^{n+1})F_p^{n+1}\right) \tag{47}$$

The local error in this update of the deformation gradient is

$$LEF_p^{n+1} = \frac{dt^3}{12}\frac{d^3F}{dt^3} + \frac{dt}{2}\left(F_p^n Ev_{xp}^n + F_p^{n+1} Ev_{xp}^{n+1} + \sum_i (F_p^n D_{ip}^n + F_p^{n+1} D_{ip}^{n+1}) LEv_i^{n+1/2}\right) \tag{48}$$

where the first term is the time local error of the trapezoidal rule. The stress is updated using the appropriate constitutive model and Young's modulus, $E$

$$\sigma_p^{n+1} = \sigma_p^n + \frac{dt}{2} E\left(\frac{\partial v}{\partial x}(x_p^{n+1}) + \frac{\partial v}{\partial x}(x_p^n)\right) \tag{49}$$

Similarly $LE\sigma_p^{n+1}$ is the stress local time and space error given by

$$LE\sigma_p^{n+1} = \frac{Edt^3}{12}\frac{d^3\sigma}{dt^3} + \frac{Edt}{2}\left(Ev_{xp}^n + Ev_{xp}^{n+1} + \sum_i (D_{ip}^n + D_{ip}^{n+1}) LEv_i^{n+1/2}\right) \tag{50}$$

The acceleration is updated with these two values which are the updated stresses and deformation gradients at the current grid points, but at the next time level $t^{n+1}$

$$a_i^{n+1} = \frac{-1}{m_i}\sum_p D_{ip}^{n+1}\sigma_p^{n+1} F_p^{n+1} V_p^0 \tag{51}$$

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

$$a_p^{n+1} = \sum_i S_{ip}(x_p^{n+1}) a_i^{n+1} \tag{52}$$

The equation for the final update of the particle velocity is then

$$v_p^{n+1} = v_p^n + \frac{dt}{2}\left(a_p^{n+1} + a_p^n\right) \tag{53}$$

with an associated local space and time error of

$$LEv_p^{n+1} = \frac{dt^3}{12}\frac{d^3v_p}{dt^3} + \frac{dt}{2}\left(\sum_i S_{ip}^n (lea_i^{n+1} + Ea_i^n + Ea_i^{n+1}) + EI_{ip}^n + EI_{ip}^{n+1}\right) \tag{54}$$

where the propagated error from the stress and deformation gradients in the acceleration is (ignoring products of these errors) given by

$$lea_i^{n+1} = \frac{-1}{m_i}\Big[\sum_p D_{ip}(x_p^{n+1} + LEx_p^{n+1})(\sigma_p^{n+1} + LE\sigma_p^{n+1}) \times (F_p^{n+1} + LEF_p^{n+1})V_p^0 - \sum_p D_{ip}^{n+1}\sigma_p^{n+1}F_p^{n+1}V_p^0\Big] \quad (55)$$

$$lea_i^{n+1} \approx \frac{-1}{m_i}[\sum_p D_{ip}^{n+1}(LE\sigma_p^{n+1}F_p^{n+1} + LEF_p^{n+1}\sigma_p^{n+1}) + \sum_p \frac{\partial D_{ip}^{n+1}}{\partial x_p} LEx_p^{n+1}F_p^{n+1}\sigma_p^{n+1})]V_p^0 \quad (56)$$

The equation for the final update of the particle position is then

$$x_p^{n+1} = x_p^n + \frac{dt}{2}\left(v_p^{n+1} + v_p^n\right) \quad (57)$$

with an associated local error of

$$LEx_p^{n+1} = \frac{dt^3}{12}\frac{d^3x_p}{dt^3} + \frac{dt}{2}LEv_p^{n+1} \quad (58)$$

In summary, even after taking into account the propagated time integration errors from the different MPM stages, the local time error appears to be third order, providing that the coefficients of the MPM $S_{ip}(x(t))$ and $D_{ip}(x(t))$ are sufficiently differentiable, which is the case for most MPM formulations except the original one.

## 6 TRGIMP method

The additional acceleration update and subsequent updates defined by Eqs. (51) to (57) may be used the update the velocities and displacements of the standard stress-last MPM method. In this case, the particle displacements and velocity at particles update given by Eqs. (53) and (57) are also consistent with the trapezoidal update given by Weiss and Guilkey [22]. For this reason, the method will be called TRGIMP in the remainder of this paper. The only difference between this and the Stormer–Verlet method lies in how the deformation gradients and stresses are updated. It will be seen that while the Stormer–Verlet method has theoretically better time accuracy in these variables, this may not translate into better accuracy overall because of spatial errors associated with interpolation and differentiation in MPM.

## 7 Comparison of Stormer–Verlet against staggered central difference time stepping

The main difference between the Stormer–Verlet scheme and the staggered central difference scheme used by and Sulsky [19] and Wallstedt and Guilkey [21] is that the latter schemes both use a staggered time step for velocity. Instead of Eqs. (41) and (38), we have the equations

$$v_p^{n+1/2} = v_p^{n-1/2} + dta_p^n \quad (59)$$

and

$$v_i^{n+1/2} = v_i^{n-1/2} + dta_i^n \quad (60)$$

as well as some small changes to the stress and deformation updates in which velocity particle derivatives at the time step start and end points are replaced by the particle velocity derivative at the midpoint e.g.

$$F_p^{n+1} = F_p^n + dt\frac{\partial v^{n+1/2}}{\partial x}(x_p^n)F_p^n \quad (61)$$

The use of Eqs. (59) and (60) gives a more accurate third-order value of $v_p^{n+1/2}$, as opposed to the second-order value. However, as both of these errors are multiplied by $dt$, this is not such a difference and does not appear to have an immediate impact on the computed accuracy. Furthermore, if the impact of the spatial error is considered, as in the next section, then it will be shown that the error seen with the different time integration approaches may be dominated by this spatial error and so will not differ from scheme to scheme for a time step size that is sufficiently small.

## 8 MPM GIMP and Stormer-Verlet spatial errors

The above equations illustrate how the spatial and temporal errors associated with MPM combine to give the overall local error in a step. Steffen et al. [16] studied these errors and arrived at the conclusion that for a stable time step temporal errors are dominated by spatial errors and that there was no point in decreasing the time step further. The error framework presented in the previous two sections makes it possible to be precise about how this also occurs when higher accuracy time integration methods are used and to be precise about this, once the magnitudes of the spatial errors are known.

In the cases of interest, here it is possible to measure the different spatial truncation errors for the analytic solutions in the experiments described below. Let the exact velocities ( accelerations) at the nodes be denoted by $\tilde{v}_i$, $(\tilde{a}_i)$ or at the particles by $\tilde{v}_p$, $(\tilde{a}_p)$. There are three main errors that are

undefined in the above analysis. These errors are estimated in recent work [7] for the GIMP basis functions [4] used with a simple model problem such as that used in the numerical experiments here. The maximum norms of these errors over the particle positions are given by

$$||Ea_i||_\infty = ||\tilde{a}_i - \frac{-1}{\tilde{m}_i}\sum_p \tilde{DS}^n_{pi}\tilde{\sigma}^n_p\tilde{F}^n_p V^0_p||_\infty = O(h) \tag{62}$$

$$||EI^n_{ip}i||_\infty = ||\tilde{v}_p - \sum_p S_{ip}\tilde{v}_i||_\infty = O(h^2) \tag{63}$$

$$||Ev^i_{xp}||_\infty = ||\tilde{v_{px}} - \sum_p D_{ip}\tilde{v}_i||_\infty = O(h) \tag{64}$$

In rare cases, such as when particles are symmetric about nodes, this last error may be $O(h^2)$ due to cancellation effects [7]. These results will also hold for the local solutions in every time step.

The importance of these results is that if these spatial errors dominate the temporal ones then the precise time accuracy of the method may not be of such importance as long as it is dominated by the spatial errors. In order to investigate this, the spatial error contributions to the local errors given above are defined by subtracting the temporal errors to get the following equation for the MPM stress-last GIMP method local spatial errors, as denoted by $le_s$;
For Eq. (20)

$$le_s v^n_i = dtEa^n_i \tag{65}$$

while for Eq. (23)

$$le_s v^{n+1}_p = dtEI^n_{ip} + dt\sum_i S^n_{ip}Ea^n_i \tag{66}$$

for Eq. (26)

$$le_s F^{n+1}_p = dtF^n_p\left(Ev^n_{xp} + \sum_i F^n_p D^n_{ip} LEv^{n+1}_i\right) \tag{67}$$

for Eq. (28)

$$le_s\sigma^{n+1}_p = dtE\left(Ev^n_{xp} + \sum_i D^n_{ip}LEv^{n+1}_i\right) \tag{68}$$

and finally for Eq. (30)

$$le_s x^{n+1}_p = dtle_s v^{n+1}_p \tag{69}$$

Similarly for the Stormer–Verlet method, the local spatial errors are denoted by $LE_s$. For Eq. (39)

$$LE_s v^{n+1/2}_i = \frac{dt}{2}Ea^n_i \tag{70}$$

while for Eq. (42)

$$LE_s v^{n+1/2}_p = \frac{dt}{2}EI^n_{ip} + \frac{dt}{2}\sum_i S^n_{ip}Ea^n_i \tag{71}$$

and for Eq. (44)

$$LE_s x^{n+1}_p = dtLE_s v^{n+1/2}_p \tag{72}$$

Similarly for Eq. (48)

$$LE_s F^{n+1}_p = \frac{dt}{2}\left(F^n_p Ev^n_{xp} + F^{n+1}_p Ev^{i+1}_{xp} + \sum_i (F^n_p D^n_{ip} + F^{n+1}_p D^{n+1}_{ip})LEv^{n+1/2}_i\right) \tag{73}$$

while for Eq. (50)

$$LE_s\sigma^{n+1}_p = \frac{dt}{2}E\left(Ev^{n+1}_{xp} + Ev^n_{xp} + \sum_i (D^n_{ip} + D^{n+1}_{ip})LEv^{n+1/2}_i\right) \tag{74}$$

and for Eq. (54)

$$LE_s v^{n+1}_p = \frac{dt}{2}\left(\sum_i S^n_{ip}(lea^{n+1}_i + Ea^n_i + Ea^{n+1}_i) + EI^n_{ip} + EI^{n+1}_{ip}\right) \tag{75}$$

On comparing these errors, we see that the spatial parts of the errors $le_s F_p, le_s\sigma_p, le_s v_p$ for MPM GIMP correspond to the errors $LE_s F_p, LE_s\sigma_p, LE_s v_p$ with the MPM Stormer–Verlet method. While the local errors in displacement $le_s x^{n+1}_p$ and $LE_s x^{n+1}_p$ differ by a factor of 2, the propagated errors in particle velocity $le_s v^{n+1}_p$ and $LE_s v^{n+1}_p$ are similar for the two methods. Hence, the error propagated from one step to the next will be similar and will ensure that the displacement errors are also similar. Overall, the conclusion to be drawn is that the individual errors for the two methods can be expected to be very similar if the spatial error dominates. The next issue to consider is the accuracy of the energy conservation properties of the two methods.

## 9 Energy conservation of Stormer–Verlet and TRGIMP MPM methods

The focus here is on the energy of the particles as this corresponds to the points moved in computation and solution values that used and displayed. The starting point is, for the moment, to ignore the body forces and then to use the approach of [3].

### 9.1 Grid point kinetic energy of Stormer–Verlet MPM

The change in kinetic energy as denoted by $\Delta K\mathscr{E}_{grid}$ on the points is given by [3]

$$\Delta K\mathscr{E}_{pts} = \frac{1}{2}\sum_p m_p (v_p^{n+1})^2 - \frac{1}{2}\sum_p m_p (v_p^n)^2 \tag{76}$$

where $m_p$ is the mass at particle point $x_p$. Hence, the grid kinetic energy is also given by

$$\Delta K\mathscr{E}_{pts} = \frac{1}{2}\sum_p m_p (v_p^{n+1} - v_p^n)(v_p^{n+1} + v_p^n) \tag{77}$$

This may be written in terms of the acceleration as

$$\Delta K\mathscr{E}_{grid} = \frac{dt}{4}\sum_p m_p (a_p^{n+1} + a_p^n)(v_p^{n+1} + v_p^n) \tag{78}$$

Substituting for the acceleration at a point using (12) gives

$$\Delta K\mathscr{E}_{grid} = \frac{dt}{4}\left[\sum_p m_p (v_p^{n+1} + v_p^n) \left(\sum_i S_{ip}^{n+1} a_i^{n+1} + \sum_i S_{ip}^n a_i^n\right)\right] \tag{79}$$

and again for the nodal acceleration using (9) gives

$$\Delta K\mathscr{E}_{grid} = \frac{-dt}{4}\sum_p (v_p^{n+1} + v_p^n) \left[\sum_i S_{ip}^{n+1}\frac{m_p}{m_i}\sum_q D_{qi}^{n+1}\sigma_q^{n+1}F_q^{n+1}V_q^0 + \sum_i S_{ip}^n \frac{m_p}{m_i}\sum_q D_{qi}^n \sigma_q^n F_q^n V_q^0\right] \tag{80}$$

### 9.2 Strain energy of Stormer–Verlet and TRGIMP MPM methods

The rate of change of strain energy is given by (50) in [3]. Hence, integrating this equation from $t^n$ to $t^{n+1}$ and using the trapezoidal rule gives

$$\Delta S\mathscr{E} = \frac{dt}{2}\sum_p \left(\sigma_p^{n+1}F_p^{n+1}\frac{\partial v_p^{n+1}}{\partial x} + \sigma_p^n F_p^n \frac{\partial v_p^n}{\partial x}\right) V_p^0 + O(dt^3) \tag{81}$$

This expression is different from that derived by Bardenhagen [3] using piecewise linear approximations for $\sigma$ and for $F$. The error in Eq. (81) is $\frac{dt^3}{12}\frac{\partial t^2}{\partial t^2}(\sigma F \frac{\partial V_p}{\partial x})$, while the error in Bardenhagen's expression is about a factor of three larger. It is worth noting that the accuracy of this equation is for the exact stress and if the computed stresses are used then their errors must be included too. Substituting for the spatial derivatives in Eq. (81) gives

$$\Delta S\mathscr{E} = \frac{dt}{2}\sum_p \left(\sigma_p^{n+1}F_p^{n+1}\sum_i D_{ip}^{n+1}\sum_q S_{qi}^{n+1}v_q^{n+1}\frac{m_q}{m_i} + \sigma_p^n F_p^n \sum_i D_{ip}^n \sum_q S_{qi}^n v_q^n \frac{m_q}{m_i}\right) V_p^0 \tag{82}$$

Changing the order of summation gives

$$\Delta S\mathscr{E} = \frac{dt}{2}\Big(\sum_q v_q^{n+1}\sum_i S_{qi}^{n+1}\frac{m_q}{m_Z i} \times \sum_p D_{ip}^{n+1}\sigma_p^{n+1}F_p^{n+1}V_p^0 + \sum_q v_q^n \sum_i S_{qi}^n \frac{m_q}{m_i}\sum_p D_{ip}^n \sigma_p^n F_p^n V_p^0\Big) \tag{83}$$

The two symmetry relations

$$S_{qi}^n = S_{iq}^n \tag{84}$$

and

$$D_{qi}^n = D_{iq}^n \tag{85}$$

are needed to be able to write Eq. (83) as

$$\Delta S\mathscr{E} = \frac{dt}{2}\left(\sum_q v_q^{n+1}\sum_i S_{iq}^{n+1}\frac{m_q}{m_i}\sum_p D_{pi}^{n+1}\sigma_p^{n+1}F_p^{n+1}V_p^0 + \sum_q v_q^n \sum_i S_{iq}^n \frac{m_q}{m_i}\sum_p D_{pi}^n \sigma_p^n F_p^n V_p^0\right)$$

## 10 Energy conservation errors of the Stormer-Verlet and TRGIMP scheme

Combining the kinetic and strain energy expressions gives the energy conservation error

$$\begin{aligned}\Delta\mathscr{E}\mathscr{E}_{err} = &\frac{-dt}{2}\Big(\sum_p \frac{(v_p^{n+1}+v_p^n)}{2}\Big)\\ &\Big[\sum_i S_{ip}^{n+1})\frac{m_p}{m_i}\sum_q D_q^{n+1}\sigma_q^{n+1}F_q^{n+1}V_q^0\\ &+\sum_i S_{ip}^n\frac{m_p}{m_i}\sum_q D_{qi}^n\sigma_q^n F_q^n V_q^0\Big]-\\ &\Big(\sum_q v_q^{n+1}\sum_i S_{iq}^{n+1}\frac{m_q}{m_i}\sum_p D_{pi}^{n+1}\sigma_p^{n+1}F_p^{n+1}V_p^0\\ &+\sum_q v_q^n\sum_i S_{iq}^n\frac{m_q}{m_i}\sum_p D_{pi}^n\sigma_p^n F_p^n V_p^0\Big)\end{aligned}$$

This equation may now be simplified to be

$$\begin{aligned}\Delta\mathscr{E}\mathscr{E}_{err} = &\frac{dt}{2}\Big(\Big(\sum_q(v_q^{n+1}\\ &-\frac{(v_q^{n+1}+v_q^n)}{2})\sum_i S_{iq}^{n+1}\frac{m_q}{m_i}\sum_p D_{pi}^{n+1}\sigma_p^{n+1}F_p^{n+1}V_p^0\\ &+\sum_q(v_q^n-\frac{(v_q^{n+1}+v_q^n)}{2})\sum_i S_{iq}^n\frac{m_q}{m_i}\\ &\sum_p D_{pi}^n\sigma_p^n F_p^n V_p^0\Big)\Big)\end{aligned}\tag{86}$$

and again to

$$\begin{aligned}\Delta\mathscr{E}\mathscr{E}_{err} = &\frac{dt}{2}\sum_q(\frac{(v_q^{n+1}-v_q^n)}{2})\\ &\Big(\sum_i S_{iq}^{n+1}\frac{m_q}{m_i}\sum_p D_{pi}^{n+1}\sigma_p^{n+1}F_p^{n+1}V_p^0\\ &-\sum_i S_{iq}^n\frac{m_q}{m_i}\sum_p D_{pi}^n\sigma_p^n F_p^n V_p^0\Big)\end{aligned}\tag{87}$$

Using (37, 40, 49) and (52), this may now be written as

$$\Delta\mathscr{E}\mathscr{E}_{err} = \frac{dt}{2}\sum_q(\frac{(v_q^{n+1}-v_q^n)}{2})m_q\left(a_q^{n+1}-a_q^n\right)\tag{88}$$

It follows that if all the components of the above equation are differentiable then there is third-order energy conservation

$$\Delta\mathscr{E}\mathscr{E}_{err} = O(dt^3)\tag{89}$$

### 10.1 Contribution from the body forces

The body forces may be divided by nodal mass to get body accelerations at the nodes, denoted here by $\hat{b}_i$. The body accelerations at the particles are denoted by $\hat{b}_p$. It is then routine to show that the energy error due to the body forces, $\Delta\mathscr{E}\mathscr{E}_{err}^B$ is given by

$$\begin{aligned}\Delta\mathscr{E}\mathscr{E}_{err}^B = &\frac{-dt}{2}\Big(\sum_p\frac{(v_p^{n+1}+v_p^n)}{2})\Big[\sum_i S_{ip}^{n+1}\hat{b}_i^{n+1}\\ &+\sum_i S_{ip}n\hat{b}_i\Big]-\sum_q v_q^{n+1}\hat{b}_q^{n+1}-\sum_q v_q^n\hat{b}_q^n\Big)\end{aligned}$$

Using the expression

$$\hat{b}_p^n = \sum_i S_{ip}^n\hat{b}_i^n + (\hat{b}_p^n - \sum_i S_{ip}^n)\hat{b}_i^n))\tag{90}$$

allows the above equation to be written as

$$\begin{aligned}\Delta\mathscr{E}\mathscr{E}_{err}^B = &\frac{dt}{2}\Big(\sum_q(v_q^{n+1}-\frac{(v_q^{n+1}+v_q^n)}{2})\hat{b}_q^{n+1}\\ &+\sum_q(v_q^n-\frac{(v_q^{n+1}+v_q^n)}{2})\hat{b}_q^n+ERR_b\Big)\end{aligned}\tag{91}$$

and as

$$\Delta\mathscr{E}\mathscr{E}_{err}^B = \frac{dt}{2}\Big(\sum_q(\frac{(v_q^{n+1}-v_q^n)}{2})(\hat{b}_q^{n+1}-\hat{b}_q^n)+ERR_b\Big)\tag{92}$$

where the term $ERR_B$ is given by

$$\begin{aligned}ERR_B = &\frac{dt}{2}\Big(\sum_p\frac{(v_q^{n+1}-v_q^n)}{2}(b_p^n-b_p^{n+1}\\ &+\sum_i S_{ip}^{n+1}b_i^{n+1}-\sum_i S_{ip}^n b_i^n)\Big)\end{aligned}\tag{93}$$

Again all these terms are $O(dt^3)$ if there is sufficient smoothness in time.

### 10.2 Energy conservation errors of the stress first/last schemes

The above analysis may easily be extended to the stress-first and stress-last schemes discussed by [3]. In the case of stress-last this gives, in terms of the particular stress-first

**Table 1** Maximum energy and maximum displacement errors over timesteps (*=failure)

| | $A = 1.5e{-}2$ | $E = 10{,}000$ | | | $A = 5e{-}2$ | $E = 10{,}000$ | |
|---|---|---|---|---|---|---|---|
| *dt* | Method | Max energy error | Max Displ. error | *dt* | Method | Max energy error | Max Displ. error |
| 1e−4 | GIMP | 3.3e−1 | 1.3e−4 | 1e−4 | GIMP | * | * |
| | TRGP | 7.5e−5 | 1.8e−4 | | TRGP | 1.3e−2 | 3.2e−2 |
| | SVGP | 1.3e−4 | 3.0e−4 | | SVGP | 1.3e−2 | 6.0e−3 |
| 1e−5 | GIMP | 3.8e−3 | 1.8e−4 | 1e−5 | GIMP | * | * |
| | TRGP | 9.7e−8 | 1.6e−4 | | TRGP | 1.9e−5 | 2.1e−3 |
| | SVGP | 1.0e−8 | 1.8e−4 | | SVGP | * | * |
| 1e−6 | GIMP | 3.9e−5 | 1.8e−4 | 1e−6 | GIMP | * | * |
| | TRGP | 1.3e−10 | 1.8e−4 | | TRGP | 4.1e−8* | 1.8e−3 |
| | SVGP | 1.1e−10 | 1.8e−4 | | SVGP | * | * |
| | $A = 1.5e{-}2$ | $E = 1000$ | | | $A = 5e{-}2$ | $E = 1000$ | |
| *dt* | Method | Max energy error | Max Displ. error | *dt* | Method | Max energy error | Max Displ. error |
| 1e−3 | GIMP | 4.2e−1 | 2.0e−4 | 1e−4 | GIMP | * | * |
| | TRGP | 2.8e−4 | 2.4e−4 | | TRGP | 1.3e−2 | 2.7e−3 |
| | SVGP | 3.8e−4 | 2.8e−4 | | SVGP | 8.7e−3 | 2.4e−3 |
| 1e−4 | GIMP | 3.6e−3 | 1.6e−4 | 1e−5 | GIMP | 1.4e−1 | 2.3e−3 |
| | TRGP | 2.7e−7 | 1.3e−4 | | TRGP | 2.5e−5 | 1.2e−3 |
| | SVGP | 3.2e−7 | 1.7e−4 | | SVGP | 5.0e−5 | 2.8e−3 |
| 1e−5 | GIMP | 3.8e−5 | 1.8e−4 | 1e−6 | GIMP | 1.5e−3 | 4.2e−3 |
| | TRGP | 3.6e−10 | 1.8e−4 | | TRGP | 1.2e−7 | 1.5e−3 |
| | SVGP | 3.4e−10 | 1.8e−4 | | SVGP | 1.9e−7 | 1.6e−3 |
| 1e−6 | GIMP | 3.9e−5 | 1.8e−4 | 1e−7 | GIMP | 9.2e−5 | 3.5e−3 |
| | SVGP | 4.6e−13 | 1.9e−4 | | SVGP | 1.3e−10 | 1.6e−3 |
| | SVGP | 3.4e−13 | 1.8e−4 | | SVGP | 4.5e−10 | 4.8e−3 |
| | $A = 1.5e{-}2$ | $E = 256$ | | | $A = 5e{-}2$ | $E = 256$ | |
| *dt* | Method | Max energy error | Max Displ. error | *dt* | Method | Max energy error | Max Displ. error |
| 1e−3 | GIMP | 2.2e−2 | 1.3e−4 | 1e−3 | GIMP | 3.7e−1 | 8.7e−4 |
| | TRGP | 7.3e−6 | 1.4e−4 | | TRGP | 1.6e−4 | 8.1e−4 |
| | SVGP | 7.5e−6 | 1.2e−4 | | SVGP | 3.8e−4 | 1.3e−3 |
| 1e−4 | GIMP | 1.1e−4 | 2.3e−4 | 1e−4 | GIMP | 6.6e−3 | 8.7e−4 |
| | TRGP | 7.6e−9 | 1.1e−4 | | TRGP | 7.6e−6 | 1.3e−3 |
| | SVGP | 7.7e−9 | 1.1e−4 | | SVGP | 9.8e−7 | 1.0e−3 |
| 1e−5 | GIMP | 2.4e−6 | 1.2e−4 | 1e−5 | GIMP | 8.8e−5 | 1.0e−3 |
| | TRGP | 8.3e−12 | 1.2e−4 | | TRGP | 1.3e−9 | 1.3e−3 |
| | SVGP | 7.7e−12 | 1.2e−4 | | SVGP | 2.0e−9 | 1.0e−3 |
| 1e−6 | GIMP | 2.4e−8 | 1.2e−4 | 1e−6 | GIMP | 9.8e−7 | 1.0e−3 |
| | TRGP | 8.9e−15 | 1.2e−5 | | TRGP | 1.6e−12 | 1.3e−3 |
| | SVGP | 8.0e−15 | 1.2e−4 | | SVGP | 2.1e−12 | 1.0e−3 |
| | $A = 1.5e{-}2$ | $E = 64$ | | | $A = 5e{-}2$ | $E = 64$ | |
| *dt* | Method | Max energy error | Max Displ. error | *dt* | Method | Max energy error | Max Displ. error |
| 1e−3 | GIMP | 1.2e−3 | 9.4e−5 | 1.e−3 | GIMP | 1.9e−2 | 5.7e−4 |
| | TRGP | 2.0e−7 | 9.3e−5 | | TRGP | 5.7e−6 | 6.2e−4 |

**Table 1** continued

| | $A = 1.5e{-}2$ | $E = 64$ | | | $A = 5e{-}2$ | $E = 64$ | |
|---|---|---|---|---|---|---|---|
| $dt$ | Method | Max energy error | Max Displ. error | $dt$ | Method | Max energy error | Max Displ. error |
| | SVGP | 1.8e−7 | 6.5e−5 | | SVGP | 1.1e−5 | 5.2e−4 |
| 1e−4 | GIMP | 1.2e−5 | 6.8e−5 | 1e−4 | GIMP | 2.6e−4 | 5.4e−4 |
| | TRGP | 1.9e−10 | 6.7e−5 | | TRGP | 1.5e−8 | 6.7e−4 |
| | SVGP | 2.0e−10 | 7.1e−5 | | SVGP | 2.7e−8 | 5.4e−4 |
| 1e−5 | GIMP | 1.3e−7 | 7.2e−5 | 1e−5 | GIMP | 2.8e−6 | 5.4e−4 |
| | TRGP | 2.0e−13 | 6.8e−5 | | TRGP | 1.7e−11 | 6.7e−4 |
| | SVGP | 2.0e−13 | 7.2e−5 | | SVGP | 3.2e−11 | 5.4e−4 |
| 1e−6 | GIMP | 1.3e−9 | 7.2e−5 | 1e−6 | GIMP | 2.9e−8 | 5.5e−4 |
| | TRGP | 2.0e−16 | 6.8e−5 | | TRGP | 1.7e−14 | 6.7e−4 |
| | SVGP | 2.0e−16 | 7.2e−5 | | SVGP | 3.3e−14 | 5.4e−4 |

accelerations and velocities the energy conservation error

$$\Delta \mathscr{E} \mathscr{E}_{err}^{first} = \frac{dt}{2} \sum_q v_q^n m_q \left( a_q^{n+1} - a_q^n \right) \tag{94}$$

While in the case of stress first, the sign is reversed. It follows that both these errors are $O(dt^2)$.

## 11 Dimensional and material model extensions

The motivation behind this work was to address energy conservation in these schemes in as simple a way as possible through the use of a model problem. The main proof ideas may be extended by using the approaches in [3] to derive multi-space dimensional results. In the case of much more complex constitutive models such as elasto-plastic cases and damage, the challenge lies in the sheer complexity of these models. For example, early work by Banerjee shows energy conservation for collisions [2] and then describes more complex elasto-plastic cases in detail [1]. Many other papers such as [13] consider such cases as do two recent detailed surveys of MPM research [14] and [20]. The proofs presented earlier, following those in [3], rely on eventually representing all the energy terms using a single variable so as to make cancellation possible. Providing that this can be done for more complex cases then it will be possible to extend the ideas presented here to more challenging cases. It is not at all clear that this is straightforward however.

## 12 Computational experiments

The two main aspects of the above methods to considered here are the computational accuracy and the cost.

### 12.1 Computational accuracy

In order to illustrate the above results, the model 1D bar problem used by [9] and [5] is used.

The cell width is $h = 10^{-2}$, the material density is $\rho_0 = 1$, and the time interval is [0,1]. The initial spatial discretization uses two evenly spaced particles per cell with the spatial domain being [0, 1]. The Young's modulus values are $E = 10,000$, $E = 1000$, $E = 256$ and $E = 64$. The maximum displacement is $A = 0.015$, and $A = 0.05$ and the time step values used are $dt = 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}$. In both these cases for the values given of $A$, it should be noted that with the use of the above parameters particles will cross from one cell to another.

Experiments were undertaken with the MPM GIMP method [15], the TRGIMP method of Sect. 6 and the Stormer–Verlet method of Sect. 5 SVGMP. These experiments were run with fixed time steps as shown in Table 1 in which the time step, $dt$ is varied appropriately. The number of grid crossings varies greatly, see [5]. Roughly speaking the case with $A = 1.5e{-}2$ has about half the grid crossings of the case when $A = 5.0e{-}2$. The results in Table 1 show for the maximum over the time steps of the vector L2 norms of energy and displacement errors averaged over the number of particles that the Stormer–Verlet and the TRGIMP methods are much better at conserving energy that a standard MPM approach, as indeed the theoretical results suggest.

### 12.2 Computational cost

The performance of the above methods was measured on an Apple imac with an Intel core i5-4570 using MATLAB code in its native form without, say, the MATLAB optimizations used by [23]. A comparison between the steps taken by the methods shows that the Stormer–Verlet method has a number of extra steps given by Eqs. (51), (52), (53), (57). Roughly

**Table 2** Comparison of computational costs

| $dt$ | $A = 1.5e{-}2$ Method | $E = 64$ N=Measured CPU time in seconds |
|---|---|---|
| 1e−3 | GIMP | 0.07 |
| | TRGP | 0.1 |
| | SVGP | 0.17 |
| 1e−4 | GIMP | 0.29 |
| | TRGP | 0.48 |
| | SVGP | 0.44 |
| 1e−5 | GIMP | 1.93 |
| | TRGP | 3.32 |
| | SVGP | 2.98 |
| 1e−6 | GIMP | 17.7 |
| | TRGP | 31.4 |
| | SVGP | 28.7 |

speaking, this corresponds to an extra 50% increase in computational steps over the standard symplectic MPM Euler method. Table 2 shows the cpu times for one of the example cases in Table 1 and confirms this observation. Results are presented for only one case from the 8 shown as the actual arithmetic operations count is independent of the case, though of course the results are different in each case.

## 13 Conclusions

In comparing the standard MPM stress-last method against the Stormer–Verlet and TRGIMP methods, it is seen that the Stormer–Verlet and TRGIMP methods have much better energy conservation properties. However, it is also proved and shown computationally that when the spatial errors dominate the errors in velocity and displacement are very similar for the three methods. It is also shown that the cost of better energy conservation is an increase of 50% in the computational cost.

**Acknowledgements** Chris Gritton is thanked for the use of his MPM GIMP code for the model problem used to obtain the results shown in Sect. 6 and extended to the Stormer–Verlet and TRGIMP methods in this work. 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, either expressed or implied, of the Army Research Laboratory or the U.S. Government.

## Declarations

**Conflict of interest** The author states that there is no conflict of interest.

## References

1. Banerjee B (2004) MPM validation: a myriad of Taylor impact tests, C-SAFE Internal Report, C-SAFE-CD-IR-04-004. University of Utah, Department of Mechanical Engineering
2. Banerjee B (2004) MPM validation: sphere-cylinder impact: medium resolution simulations C-SAFE Internal Report, C-SAFE-CD-IR-04-003. University of Utah, Department of Mechanical Engineering
3. Bardenhagen S (2002) Energy conservation error in the material point method for solid mechanics. J Comput Phys 180:383–403
4. Bardenhagen S, Kober E (2004) The generalized interpolation material point method. Comput Model Eng Sci 5:477–495
5. Berzins M (2018) Nonlinear stability and time step selection for the MPM method. Computational Particle Mechanics
6. Berzins M (2019) Time integration errors and energy conservation properties of the Stormer–Verlet method applied to MPM. In: Oñate E, Bischoff M, Owen DRJ, Wriggers P, Zohdi T (eds) Proceedings of VI international conference on particle-based methods—fundamentals and applications, Barcelona, PARTICLES 2019, pp 555–566. ISBN: 978-84-121101-1-1
7. Berzins M (2021) Time stepping with space and time errors and stability of the material point method. In: Wriggers P, Bischoff M, Oñate E, Duster A, Zohdi T (eds) Proceedings of VII international conference on particle-based methods, Hamburg, PARTICLES 2021, ISBN: 978-84-121101-1-1 (to appear)
8. Buzi O, Pedroso DM, Giacomini A (2008) Caveats on the implementation of the generalized material point method. CMES 31(2):85–106
9. Gritton CE, Berzins M (2017) Improving accuracy in the MPM methods by using a null space filter. Comput Part Mech 4:131–142
10. Hairer E, Lubich C, Wanner G (2003) Geometric numerical integration illustrated by the Stormer–Verlet method. Acta Numerica. Cambridge University Press, pp 399–450
11. Leimkuhler B, Reich S (2004) Simulating Hamiltonian dynamics. Cambridge Monographs on Applied and Computational Science. Cambridge University Press, Cambridge
12. Love E, Sulsky D (2006) An energy consistent material point method for dynamic finite deformation plasticity. Int J Numer Methods Eng 65:1608–1638
13. Raymond SJ, Jones BD, Williams JR (2019) Modeling damage and plasticity in aggregates with the material point method (MPM). Comput Part Mech 6:371–382
14. Solowski WJ, Berzins M, Coombs WM, Guilkey JE, Moller M, Tran QA, Adibaskoro T, Seyedan SMJ, Tielen, R Soga K (2021) Material point method: overview and challenges ahead. Advances in Applied Mechanics 9
15. Steffen M, Wallstedt PC, Guilkey JE, Kirby RM, Berzins M (2008) Examination and analysis of implementation choices within the Material Point Method (MPM). Comput Model Eng Sci 31(2):107–127
16. Steffen M, Kirby RM, Berzins M (2010) Decoupling and balancing of space and time errors in the material point method (MPM). Int J Numer Methods Eng 82(10):1207–1243
17. Sulsky D, Chen Z, Schreyer HL (1994) A particle method for history-dependent materials. Comput Methods Appl Mech Eng 118:179–196
18. Sulsky D, Zhou S-J, Schreyer HL (1995) Application of a particle-in-cell method to solid mechanics. Comput Phys Commun 87:236–252
19. Sulsky D, Schreyer H, Peterson K, Kwok R, Coon M (2007) Using the material-point method to model sea ice dynamics. J Geophys Res 112:C02S90. https://doi.org/10.1029/2005JC003329

20. de Vaucorbeil A, Nguyen VP, Sinaie S, Wu JY (2020) Chapter two-material point method after 25 years: theory, implementation, and applications. In: Bordas SPA, Balint DS (eds) Advances in applied mechanics, vol 53. Elsevier, Amsterdam, pp 185–398
21. Wallstedt PC, Guilkey JE (2008) An evaluation of explicit time integration schemes for use with the generalized interpolation material point method. J Comput Phys 227(22):9628–9642
22. Guilkey JE, Weiss JA (2003) Implicit time integration for the material point method: quantitative and algorithmic comparisons with the Finite element method. Int J Numer Methods Eng 57:1323-1338
23. Wyser E, Alkhimenov E, Jaboyedoff M, Podladchikov YV (2020) A fast and efficient MATLAB-based MPM solver: fMPM-solver v1.1. Geosci Model Dev 13:6265–6284