Enhanced Material Point Method to face dynamic problems:

Local-maximum entropy approximation and explicit predictor-corrector scheme

Miguel Molinos · **Pedro Navas** · **Manuel Pastor** · **Miguel Martn Stickle**

Abstract This document is devoted to describe the suitability of the Local *maximum-entropy* (LME) meshfree approximation technique under the framework of the Material Point Method for dynamic problems.

Keywords LME · MPM · Dynamic problems

1 Introduction

Since the proposal of the Material Point Method (**MPM**) by Sulsky *et al.* (1994)[**?**] as a generalization to solids of the fluid implicit particle (FLIP) method [**?**]. It popularity has increased due to its ability to deal with large strain regime without suffer mesh distortion inaccuracies. However, this method suffers other kind of instabilities, such those when material points crossing cell boundaries. This give rise to the development of other interpolation techniques to overcome this limitation such as the generalized interpolation material point method (GIMP) Bardenhagen & Kober (2004)[**?**], the dual domain material point method (DDMP) Zhang *et al.* (2011)[**?**], the B-Spline MPM Tran *et al.* (2019)[**?**] or the Conservative Taylor Least Squares reconstruction Wobbes *et al.* (2018)[**?**]. This techniques are devoted to mitigate the "grid crossing" error. Nevertheless, in the presence of shock waves spurious numerical noises appears despite of this using techniques [**?**]. These numerical inaccuracies also known as wiggles are due to the time discretization technique. A simple approach to face those spurious noises is to add a damping source to the equilibrium equations. To overcome this limitation, many researchers has proposed alternative time integration schemes which reduce them by filtering high-frequencies. Some of these attempts are the implicit GIMP

(iGIMP) Charlton *et al.* (2017) or the generalised-α scheme for the MPM proposed by Tran & Solowski (2019).

The aim of this paper is to mitigate this spurious oscillations by the employ of a suitable combination of the maximum-entropy (or local *max-ent*) γ family shape functions, and the proposal of a explicit predictor-corrector scheme. Local *max-ent* shape functions was first introduced by Arroyo & Ortiz (2006)[**?**], it belongs to the class of convex approximation schemes and provides a seamless transition between finite elements (**FE**) and mesh-free interpolations. The approximation scheme is based on a compromise between minimizing the width of the shape function support and maximizing the information entropy of the approximation. The local *max-ent* approximation may be regarded as a regularization, or *thermalization*, of Delaunay triangulation which effectively resolves the degenerate cases resulting from the lack of uniqueness of the triangulation. Local *max-ent* basis functions possess many desirable properties for mesh-free algorithms. First of all, they are entirely defined by the nodal set and the domain of analysis. They are also non-negative, satisfy the partition of unity property, and provide an exact approximation for affine functions [**?**]. Furthermore, as it is not based in a voxel discretization as the GIMP, it is no prone to suffer voxel domains overlap, which can introduce severe inaccuracies. This approximation scheme has been proof to have a good performance under the dynamic regime by other researchers like Navas *et al.* (2018)[**?**] and Li *et al.* (2012)[**?**] for Optimal Transportation Meshfree (**OTM**) method. And more recently under the MPM framework by Wobbes *et al.*(2020)[**?**] but limiting the γ parameter to zero. On the other hand, the time integration scheme here proposed is an accommodation of the Newmark a-form ($\gamma = 0.5$ and $\beta = 0$)[1] which is the central difference

Address(es) of author(s) should be given

---

[1] Notice that these γ and β parameters does not have the same meaning for the LME shape functions.

Table 1: Physical variables involved in the problem

ρ	Density field	Scalar
a	Acceleration field	First order tensor
v	Velocity field	First order tensor
u	Displacement field	Second order tensor
σ	Cauchy stress tensor	Second order tensor
ε	Cauchy strain tensor	Second order tensor
D	Constitutive tensor	Fourth order tensor

explicit (CD). We have choose this method among other suitable alternatives as Wilson *et al.* (1972) or Chung & Hulbert (1993) because it simplicity and it good performance dealing with solids dynamics problems like those presented in [**?**].

The article is organized as follows. Section 2 briefly reviews the notation here employed. Next, Section 3 is devoted to present briefly the governing equations of the elastic problem, the variational formulation and the Galerkin procedure. In Section 4 an explicit predictor-corrector time integration scheme for the MPM is proposed. Section 5 briefly reviews the local *max-ent* basis functions here employed. In Section **??** a comparison between both methods MPM and OTM employing *max-ent* shape function is performed as well a parametric study for the γ parameter. Both techniques will face to solve a classic dynamic benchmark proposed by Dyka & Ingel (1995)[**?**]. Finally, conclusions and future research topics are exposed in Section 7.

2 Brief note concerning the notation

In what follows, we will adopt the following convention. All the physical variables involved in this problem are collected in Table 1.

Three kind of subscript or superscript are used within paper. The subscript $\square_p$ is used to define a particle variable. While the subscript $\square_l$ is reserved in this notation for denoting nodal variables. And finally, the superscript $\square^\psi$ involves a virtual magnitude. For the operators, the convention is : $\dot{\square}$ and $\ddot{\square}$ for the first and second time derivative, $\otimes$ means the dyadic operator, $(\cdot)$ and $(:)$ means the first and second contraction of a tensor, $div(\square)$ denotes the divergence operator, and finally $grad(\square)$ and $grad^s(\square)$ denotes the gradient and its symmetric part. Einstain subscripts convention is adopted so repeated index means addition.

3 Derivation of the MPM

The aim of this section is to provide an overview of the standard explicit MPM algorithm [**?**]. Therefore, it is structured in the following shape: we will first give a introduction to

the governing equations 3.1, later the variational statement of the problem will be presented 3.2, and finally a discretization procedure thorough Galerkin is performed 3.3.

The method has three main steps: (i) a variational recovery process, where particle data is projected to the grid nodes, (ii) an Eulerian step, where balance of momentum equation is expressed as a nodal equilibrium equation thorough a FEM-like procedure, and finally (iii) a Lagrangian advection of the particles.

3.1 Governing equations

In the MPM approach a continuum is considered. Let define a Ω domain occupied by an elastic body, and $\partial\Omega$ the boundary of the domain, where $\partial\Omega = \Gamma_d \bigcup \Gamma_n$ and $\Gamma_d \cap \Gamma_n = \emptyset$. Note that Γ_d is reserved for Dirichlet boundary conditions, and Γ_n for the Neumann. In a inifinitesimal strain context, the behaviour of the continuum can be described by the following set of equations: (i) the balance of momentum equation

$$\rho a = div(\sigma) + \rho b, \tag{1}$$

(ii) the compatibility equation with ensure that the velocity field is compatible with the Cauchy strain field,

$$\dot{\varepsilon} = grad^s(v), \tag{2}$$

(iii) the constitutive equation, which relates the rate of Cauchy strain tensor with the rate of Cauchy stress tensor,

$$\dot{\sigma} = D : \dot{\varepsilon}, \tag{3}$$

and finally (iv) the mass conservation, which can be obtained by setting to zero the total derivative of the density field,

$$\frac{D\rho}{Dt} = \dot{\rho} + \rho \, div(v) = 0. \tag{4}$$

3.2 Variational formulation

To write the variational statement of the problem, let us define a virtual velocity field such that

$$v^\psi \in \mathscr{H}_0^1(\Omega) = \{v^\psi \in \mathscr{H}^1 \mid v^\psi = 0 \text{ on } \Gamma_d\}. \tag{5}$$

And which satisfies that the Cauchy sequences are convergent in Ω

$$\int_\Omega v^\psi \, d\Omega < \infty \quad \text{and} \quad \int_\Omega \varepsilon^\psi \, d\Omega < \infty \tag{6}$$

The principle of virtual work states that the equilibrium solution to the boundary value problem of elasticity is the function $v \in \mathscr{H}_0^1$ such that, for $v^\psi \in \mathscr{H}_0^1$, the following holds:

$$\int_\Omega \rho \left(\frac{\partial \mathbf{v}}{\partial t} - \mathbf{b} \right) \cdot \mathbf{v}^\psi \, d\Omega = \int_{\Gamma_d} \mathbf{t} \cdot \mathbf{v}^\psi \, d\Gamma - \int_\Omega \sigma : \varepsilon^\psi \, d\Omega,$$
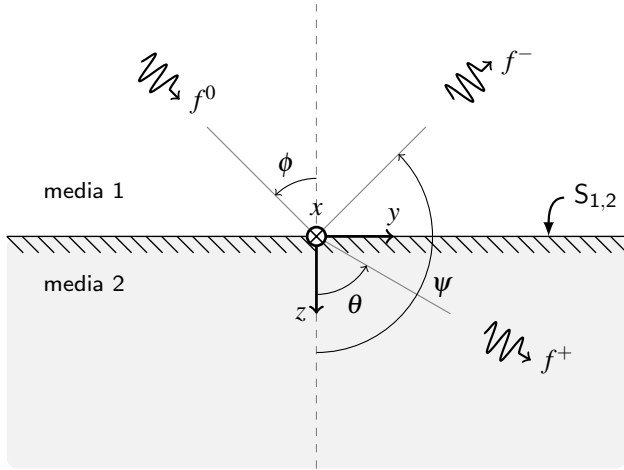
Fig. 1: MPM discretization.

$$(7)$$

therefore (7) is the balance of momentum formulated in its weak form.

3.3 Galerkin procedure

In order to arrive to a finite set of equations, in contrast with the FEM, in the MPM a double discretization procedure is performed as we will describe here below. First, the continuum domain Ω is discretized with a finite sum of material points (in the following particles), each one represent a part of the discretized domain $\Omega_p \subset \Omega$ with $p = 1, 2 \ldots, N_p$ where N_p is the number of particles. The material point $\mathbf{x}_p$ is defined at the centroid of each Ω_p, figure 1.

Each material point is assigned with initial values of position, velocity, mass, volume and stress denoted by $\mathbf{x}_p$, $\mathbf{v}_p$, m_p, V_p and σ_p, but also the virtual field v_p^{ψ}. So employing the definition of the material integral, where we recover the Riemann integral definition as an addition of a finite set of points. Therefore individual terms in (7) are solved as follows.

– Acceleration forces :

$$\int_{\Omega} \rho \, \mathbf{a} \cdot v^{\psi} \, d\Omega = \dot{\mathbf{v}}_p \cdot v_p^{\psi} \, m_p. \tag{8}$$

– Internal forces :

$$\int_{\Omega} \sigma : \varepsilon^{\psi} \, d\Omega = \sigma_p : \varepsilon_p^{\psi} \, V_p. \tag{9}$$

– Body forces :

$$\int_{\Omega} \rho \, \mathbf{b} \cdot v^{\psi} \, d\Omega = \mathbf{b}_p \cdot v_p^{\psi} \, m_p. \tag{10}$$

– Loads :

$$\int_{\Gamma_d} \mathbf{t} \, v^{\psi} \, d\Gamma = \int_{\Gamma_d} \rho \, \mathbf{t}^s \cdot v^{\psi} \, d\Gamma = \mathbf{t}_p^s \cdot v_p^{\psi} \, h^{-1} \, m_p, \tag{11}$$

where h is the thickness of the continuum in a 2D case. Here is where the second discretization procedure appears. A background mesh composed by a finite set of grid points with coordinates $\mathbf{x}_I$, $I = 1, 2 \ldots, N_n$, is generated. Where N_n is the number of grid nodes. This mesh is employed as a support to compute gradients and divergences.

Introducing (8), (9), (10), and (11) in (7), and approximating the real and virtual velocity field of the particle p as

$$\mathbf{v}_p = N_I(\mathbf{x}_p) v_I \quad \text{and} \quad \mathbf{v}_p^{\psi} = N_I(\mathbf{x}_p) v_I^{\psi}, \tag{12}$$

$$grad(v)_p = v_I N_I(\mathbf{x}_p) \quad \text{and} \quad \mathbf{v}_p^{\psi} = \sum_{I}^{Nn} N_I(\mathbf{x}_p) v_I^{\psi}, \tag{13}$$

we reach to the particle balance of forces of the continuum,

$$\dot{\mathbf{p}}_I = \mathsf{m}_{IJ} \dot{\mathbf{v}}_J = \mathbf{f}_I^{int} + \mathbf{f}_I^{ext}, \tag{14}$$

where $\dot{\mathbf{p}}_I$ is the rate of momentum at grid node I, the nodal mass matrix m_{IJ} is,

$$\mathsf{m}_{IJ} = \sum_{p}^{Np} N_{Ip} m_p N_{Jp}. \tag{15}$$

To improve the computational efficiency and stability, the nodal mass matrix (15) can be substituted by the lumped mass matrix m_{IJ}^{lumped}. Later, internal and external forces are computed as follows,

$$\mathbf{f}_I^{int} = -\sum_{p}^{N_p} \sigma_p \cdot grad(N_{Ip}) \frac{m_p}{\rho_p} \tag{16}$$

$$\mathbf{f}_I^{ext} = \sum_{p}^{N_p} N_{Ip} \mathbf{b}_p m_p + \sum_{p}^{N_p} N_{Ip} \mathbf{t}_p^s m_p h^{-1} \tag{17}$$

where $\sigma_p = \sigma_p(\varepsilon_p)$ is the particle p stress field, which can be integrated employing the suitable constitutive model. The strain tensor is updated employing the rate of stress tensor $\dot{\varepsilon}_p$ used to update the strain tensor is as follows (18).

$$\dot{\varepsilon}_p = \frac{\Delta \varepsilon_p}{\Delta t} = \frac{1}{2} \sum_{I}^{Nn} [grad(N_{Ip}) \otimes \mathbf{v}_I + \mathbf{v}_I \otimes grad(N_{Ip})]. \tag{18}$$

Next, imposing $\frac{D\rho}{Dt} = 0$, we ensures the mass conservation and update the density field.

$$\dot{\rho} = -\rho \, tra(\dot{\varepsilon}) \qquad (19)$$

Finally, to solve the equation (14), a second order temporal integration scheme is required. Therefore, time is discretized in to a finite set of time steps $k = 1 \dots, Nt$, where k is the current time step and N_t is the total number of time steps. Once the nodal equilibrium equation it is solved, the values in the nodes are interpolated back in to the particles and each particle it is advected to the new position,

$$\dot{\mathbf{v}}_p = \sum_I^{N_n} N_{Ip} \, \mathbf{a}_I, \quad and \quad \dot{\mathbf{x}}_p = \sum_I^{N_n} N_{Ip} \, \mathbf{v}_I \qquad (20)$$

In the MPM literature, this equations (14) and (20), are solved with an explicit forward Euler algorithm. The time integration scheme of the MPM has been described in detail by many researchers [?], [?], [?] and summarized in Figure 2.

Other authors have proposed many others time integration alternatives like [?], [?], [?]. In the first publication on the MPM [?], the nodal acceleration was employed to update the particles as

$$v_p^{k+1} = v_p^k + \Delta t \sum_I^{N_n} N_{Ip}^k \, \mathbf{a}_I^k \qquad (21)$$

$$x_p^{k+1} = x_p^k + \Delta t \sum_I^{N_n} N_{Ip}^k \, \mathbf{v}_I^k. \qquad (22)$$

However, as Andersen (2009)[?] point out, this algorithm has been shown to be numerically unstable due to that $f_I^{int,k}$ can be finite for an infinitesimal nodal mass m. This can lead to numerical issues when nodal acceleration is obtained for evaluating (22),(21). Hence, a corrected version of this algorithm in shown in Zhang *et al.* (2016)[?]

$$x_p^{k+1} = x_p^k + \Delta t \sum_I^{N_n} \frac{N_{Ip}^k \, \mathbf{p}_I^k}{\mathsf{m}_I}. \qquad (23)$$

$$v_p^{k+1} = v_p^k + \Delta t \sum_I^{N_n} \frac{N_{Ip}^k \, \mathbf{f}_I^k}{\mathsf{m}_I}, \qquad (24)$$

Later Tran & Solowski (2019)[?] presented a generalized-α scheme for the MPM inspired in the explicit time integration algorithm proposed by Chung & Hulbert (1993)[?], but with the particularity that the acceleration is evaluated both in the beginning and the end of the time step.

$$v_p^{k+1} = v_p^k + \Delta t \sum_I^{N_n} N_{Ip}^k \left[(1-\gamma) \, a_I^k + \gamma \, a_I^{k+1} \right], \qquad (25)$$

$$m_I^k = \sum_p^{N_p} m_p N_{Ip}^k$$
$$p_I^k = \sum_p^{N_p} m_p v_p^k N_{Ip}^k$$
$$v_I^k = \frac{p_I^k}{m_I^k}$$

$$\varepsilon_p = \tfrac{1}{2}(\nabla N_{Ip}^k v_I^k + v_I^k \nabla N_{Ip}^k)$$

$$f_I^k = f_I^{int,k}(\dot{\varepsilon}_p) + f_I^{ext,k}$$
$$p_I^{k+1} = p_I^k + f_I^k \Delta t$$

$$v_p^{k+1} = v_p^k + \sum_I^{N_e} \frac{f_I^k N_{Ip}^k}{m_I^k} \Delta t$$
$$x_p^{k+1} = x_p^k + \sum_I^{N_e} \frac{p_I^{k+1} N_{Ip}^k}{m_I^k} \Delta t$$

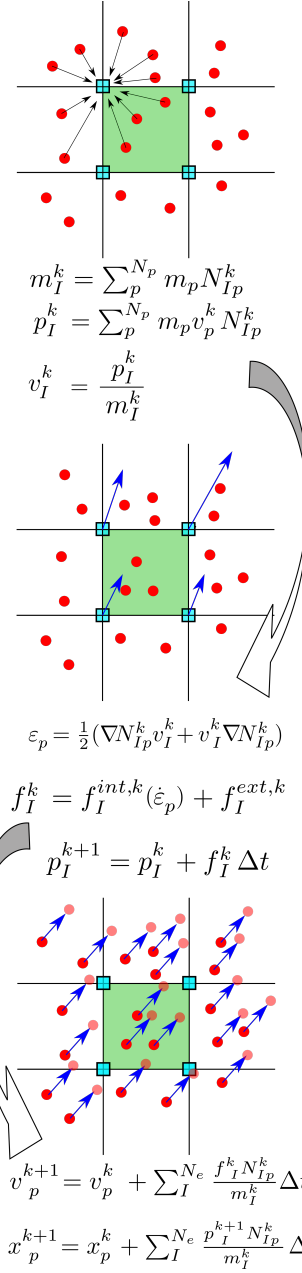**Fig. 2** MPM standard algorithm.

$$x_p^{k+1} = x_p^k + \sum_I^{N_n} N_{Ip}^k \left[\Delta t \, \mathbf{v}_I^k + \Delta t^2 \left((\tfrac{1}{2} - \beta) \, \mathbf{a}_I^k + \beta \, \mathbf{a}_I^{k+1} \right) \right] \qquad (26)$$

$$a_p^{k+1} = \sum_I^{N_n} N_{Ip}^k \, \mathbf{a}_I^{k+1}. \qquad (27)$$

This scheme has prof to damps out the higher frequency noises [?]. But it can present the same numerical instabilities as in (22),(21) when nodal masses become infinitesimal.

4 Explicit predictor-corrector scheme for MPM.

In this section, an explicit predictor-corrector time integration scheme is presented. It is based in the Newmark a-form $\gamma = 0.5$ and $\beta = 0$ which is the central difference explicit. This method is devoted to solve a system of equations of type

$$\mathbf{M}_{IJ}\ddot{\mathbf{d}}_J + \mathbf{C}_{IJ}\dot{\mathbf{d}}_J + \mathbf{K}_{IJ}\mathbf{d}_J = \mathbf{F}_I.$$

As the MPM has a nodal stage, it is possible to apply this methods successfully in the MPM framework as was proved by [?]. Taking the predictor definition from the classic literature [?] and calculating nodal velocity, and updating particles position employing nodal values of velocity and acceleration. Therefore we get the following expression for the **predictor** stage for both velocity and displacement.

$$\mathbf{x}_p^{k+1} = \mathbf{x}_p^n + \Delta t \underbrace{\sum_I^{N_n} N_{Ip}^k \, \mathbf{v}_I^k}_{\mathbf{v}_p^k} + \frac{1}{2}\Delta t^2 \underbrace{\sum_I^{N_n} \frac{N_{Ip}^k \, \mathbf{f}_I^k}{\mathsf{m}_I^k}}_{\mathbf{a}_p^k} \qquad (28)$$

$$\mathbf{v}_I^{k+1} = \mathbf{v}_I^n + (1-\gamma)\,\Delta t\,\mathbf{a}_I^k \quad , \quad \mathbf{a}_I^k = \frac{\mathbf{f}_I^k}{\mathsf{m}_I^k} \qquad (29)$$

Consequently, the **corrector** stage for the nodal velocity is in the following way

$$\mathbf{v}_I^{k+1} = \mathbf{v}_I^{k+1} + \gamma\,\Delta t\,\mathbf{a}_I^{k+1} \quad , \quad \mathbf{a}_I^{k+1} = \frac{\mathbf{f}_I^{k+1}}{\mathsf{m}_I^{k+1}} \qquad (30)$$

Notice that the predictor of the particle displacement is computed using information only from the predictor of the previous step. However, velocities are computed using half information of the past corrector and half of the past predictor. Therefore here we share similarities with the **leapfrog integration** which updates the position at full time step, but updates the velocity at half time steps. Notice also that with this approach the calculation of nodal momentum vales in not required, and particles velocities is just a post-process. So during the lagrangian step only the particles position is truly computed. This minimizes the information transference between the particles and nodes, and in consequence minimizes the associated error to this process.

Due to its simplicity allows be implemented with minor modifications over a standard forward Euler. It is summarized in shape of pseudo-algorithm **??**.

5 Local max-ent approximants

The popularity of the MPM has increase notoriously during the recent years due to its ability to deal with large strain problems without mesh distorsion issues inherent to mesh

---

Algorithm Explicit Predictor-Corrector scheme

1: **Explicit Newmark Predictor** ($\gamma = 0.5$):
Only in those active nodes:

$$\mathbf{v}_I^{k+1} = \mathbf{v}_I^n + (1-\gamma)\,\Delta t\,\mathbf{a}_I^k \quad , \quad \mathbf{a}_I^k = \frac{\mathbf{f}_I^k}{\mathsf{m}_I^k}$$

2: **Impose essential boundary conditions**:
At the fixed boundary, set $\mathbf{v}_I^{k+1} = 0$.

3: **Update mass matrix**:

$$\mathsf{m}_I = \sum_p^{N_p} N_{Ip}^{k+1}\, m_p,$$

4: **Deformation tensor increment calculation**.

$$\dot{\varepsilon}_p^{\,k+1} = \frac{1}{2}\sum_I^{N_n} grad(N_{Ip}^{k+1})\mathbf{v}_I^{k+1} + (grad(N_{Ip}^{k+1})\mathbf{v}_I^{k+1})^T$$

$$\Delta\varepsilon_p^{k+1} = \Delta t\,\dot{\varepsilon}_p^{\,k+1}$$

5: **Update the density field**:

$$\rho_p^{k+1} = \frac{\rho_p^k}{1 + tra\left[\Delta\varepsilon_p^{k+1}\right]}.$$

6: **Balance of forces calculation**:
Calculate the total grid nodal force $\mathbf{f}_I^{k+1} = \mathbf{f}_I^{int,k+1} + \mathbf{f}_I^{ext,k+1}$ evaluating (16) and (17) in the time step $k+1$. In the grid node I is fixed in one of the spatial dimensions, set it to zero to make the grid nodal acceleration zero in that direction.

7: **Explicit Newmark Corrector** ($\gamma = 0.5$):

$$\mathbf{v}_I^{k+1} = \mathbf{v}_I^{k+1} + \gamma\,\Delta t\,\mathbf{a}_I^{k+1} \quad , \quad \mathbf{a}_I^{k+1} = \frac{\mathbf{f}_I^{k+1}}{\mathsf{m}_I^{k+1}}$$

8: **Update particles position and velocity**:

$$\mathbf{x}_p^{k+1} = \mathbf{x}_p^n + \Delta t \sum_I^{N_n} N_{Ip}^k \, \mathbf{v}_I^k + \frac{1}{2}\Delta t^2 \sum_I^{N_n} \frac{N_{Ip}^k \, \mathbf{f}_I^k}{\mathsf{m}_I^k}$$

$$\mathbf{v}_p^{k+1} = \mathbf{v}_p^n + \Delta t \sum_I^{N_n} \frac{N_{Ip}^k \, \mathbf{f}_I^k}{\mathsf{m}_I^k}$$

---

based methods like FE, see Zdzislaw [?]. However, in the simulations made with the original MPM, there are numerical noises when particles crossing the cell boundaries. Recently the local maximum-entropy (or local *max-ent*) shape function first introduced by Arroyo & Ortiz (2006)[?] has been tested under the MPM framework by Wobbes *et al.* (2020)[?] where they prof that simulations performed with

the maxent basis functions show considerably more accurate stress approximations for MPM. Although, in [?] authors does not deep in how β affects to the accuracy and stability of the solution.

The basic idea of the shape functions based on such an estimate is to interpret the shape function $N_I(\mathbf{x})$ as the probability of $\mathbf{x}$ to obtain the value $\mathbf{x}_I$, $I = 1, \ldots, n$. Here n is the number of nodes in the domain.

Taking the definition of entropy as a measure of how uncertainty a random variable is averaged on all its possible outcomes.

Therefore following the principle of *maximum-entropy* stated by Jaynes [?], the least biased shape functions are those that maximizes the entropy of the scheme

estimate is defined as the type of statistical inference, which is the least biased possible on the given information.

Taking Shannon's entropy as the starting point:

$$H(p_1(\mathbf{x}), \ldots, p_n(\mathbf{x})) = -\sum_{I=1}^{N_n} p_I(\mathbf{x}) \log p_I \tag{31}$$

where $p_I(\mathbf{x})$ is the probability, equivalent to the mentioned shape function $N_I(\mathbf{x})$, satisfying both the zeroth and first-order consistency.

The least-biased approximation scheme is given by

(LME) Maximize $H(p) = -\sum_{I}^{N_n} p_I(\mathbf{x}) \log p_I$

subject to $p_I \geq 0$, I=1, ..., n

$$\sum_{I=1}^{N_n} p_I = 1$$

$$\sum_{I=1}^{N_n} p_I \mathbf{x}_I = \mathbf{x}$$

The local max-ent approximation schemes (LME) as a Pareto set, defined by [?] is as follows

(LME)$_\beta$ For fixed $\mathbf{x}$ minimise $f_\beta(\mathbf{x}, p) = \beta H(\mathbf{x}, p) - H(p)$

subject to $p_I \geq 0$, I=1, ..., n

$$\sum_{I=1}^{N_n} p_I = 1$$

$$\sum_{I=1}^{N_n} p_I \mathbf{x}_I = \mathbf{x}$$

for $\beta \in (0, \infty)$ is Pareto optimal. The unique solution of the local max-ent problem (LME)$_\beta$ is:

$$p(\mathbf{x}) = \frac{\exp \left[-\beta \, |\mathbf{x} - \mathbf{x}_I|^2 + \lambda \, (\mathbf{x} - \mathbf{x}_I) \right]}{Z(\mathbf{x}, \lambda^*(\mathbf{x}))} \tag{32}$$

where

$$Z(\mathbf{x}, \lambda) = \sum_{I=1}^{N_n} \exp \left[-\beta \, |\mathbf{x} - \mathbf{x}_I|^2 + \lambda \, (\mathbf{x} - \mathbf{x}_I) \right] \tag{33}$$

being $\lambda^*(\mathbf{x})$ the unique minimiser for $\log Z(\mathbf{x}, \lambda)$

In order to obtain the first derivatives of the shape function, it is also necessary to compute ∇p_I^*

$$\nabla p_I^* = p_I^* \left(\nabla f_I^* - \sum_{J}^{N_n} p_J^* \nabla f_J^* \right) \tag{34}$$

where

$$f_I^*(\mathbf{x}, \lambda, \beta) = -\beta \, |\mathbf{x} - \mathbf{x}_I|^2 + \lambda \, (\mathbf{x} - \mathbf{x}_I) \tag{35}$$

Employing the chain rule, rearranging and considering β as a constant, Arroyo and Ortiz [?] obtained the following expression:

$$\nabla p_I^* = -p_I^* (\mathsf{J}^*)^{-1} (\mathbf{x} - \mathbf{x}_I) \tag{36}$$

where J is the Hessian matrix, defined by:

$$\mathsf{J}(\mathbf{x}, \lambda, \beta) = \frac{\partial \mathbf{r}}{\partial \lambda} \tag{37}$$

$$\mathbf{r}(\mathbf{x}, \lambda, \beta) \equiv \partial_\lambda \log Z(\mathbf{x}, \lambda) = \sum_{I}^{N_n} p_I(\mathbf{x}, \lambda, \beta) (\mathbf{x} - \mathbf{x}_I) \tag{38}$$

Note that, the objective of the above procedure is to find the λ which minimises $\log Z(\mathbf{x}, \lambda)$. The traditional way to obtain such a minimiser is using Eq. (37) to calculate small increments of $\partial \lambda$ in a Newton-Raphson approach.

Similar to alternative non-polynomial meshfree basis functions, the LME approximation scheme requires more than $d + 1$ nodes to determine the values of the shape functions as well as their derivatives at any point in the convex hull of the nodal set, where d is the dimension of the problem. Due to the FE-compatibility, the LME shape function is degenerated to linear finite element shape function if $d + 1$ neighbouring nodes are chosen as the support. The support size of LME shape functions may be controlled by adjusting a dimensionless parameter, $\gamma = \beta h^2$ (e.g. two dimensional example shown in figure. 3)[?].

Since p_I is defined in the entire domain, in practice, the function $\exp(-\beta \mathbf{r})$ truncated by a given tolerance, 10^{-6}, for example, would ensure a reasonable range of neighbours, see [?] for details. This tolerance defines the limit values of the influence radius and is used thereafter to find the neighbour nodes of a given integration point. Regarding this aspect an important note in the implementation of this interpolation technique in the MPM is

In this research and in [?], is a scalar as the influence area of the shape function is controlled by the Euclidean norm, therefore the search area is geometrically a circle in 2D, or a sphere in 3D. Building upon the idea of anisotropic shape functions, [?] introduced an enhanced version of the original local max-ent scheme, which uses an anisotropic support to deal with tensile inestability. Nonetheless this is out of the scope of the present document but will be incorporated in future research.

(a) Piecewise linear (Q4).

(b) Local *max-ent* $\gamma = 17$.

(c) Uniform generalized interpolated material point.

(d) Local *max-ent* $\gamma = 10$.

(e) Local *max-ent* $\gamma = 7$.
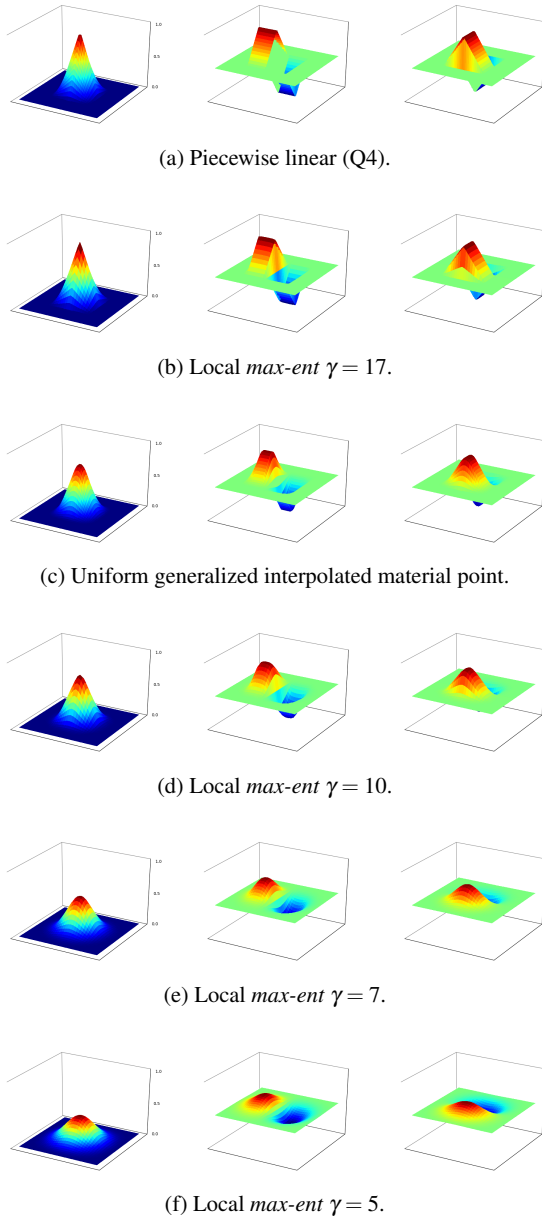

(f) Local *max-ent* $\gamma = 5$.

Fig. 3: Local max-ent shape functions for a two-dimensional arrangement of nodes, and spatial derivatives for several values of $\gamma = \beta/h^2$.

6 Application to linear elasticity dynamic problems.

This section is devoted to test the ability of both predictor-corrector time integration scheme and the local *max-ent* approximants to overcome spurious oscillations due to the grid crossing and high frequency loads under the context of the MPM.
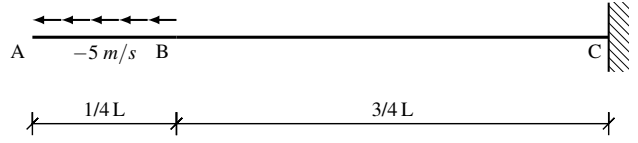
Fig. 4: Geometrical description of the Dyka [?] bar.

6.1 Dyka bar

High frequency , the benchmark proposed by Dyka & Ingel (1995)[?] is considered to illustrate the capacity of the proposed time integration algorithm to avoid velocity fields instabilities.

In the one-dimensional bar sketched in the figure 4, the left end of the bar is fixed and the right and an initial velocity $v_0 = 5\ m/s$ is given to the last quarter of it in the x positive direction. The length of it is 0.1333 meters with an unit section.

The elastic parameters consider for this test are:

– Density : 7833 $kg/m3$
– Poisson ratio : 0
– Elastic modulus : $200 \cdot 10^9\ Pa$

The boundary conditions are:

$$\sigma|_{x=L} = 0 \quad , \quad v|_{x=0} = 0 \tag{39}$$

The total simulation time is 0.0002 seconds, a direct consequence of that is we let the elastic wave generated travel thorough the bar (from A to C and back to A) at least four times. For the spatial discretization we will set a mesh size of 3.3325 millimeters. The initial mesh will be composed by a single particle in the center of the element.

Regarding this, here we give one consideration to properly modellize this benchmark under the MPM framework. Notice that the free border of the bar has a horizontal displacement of 0.033 millimeters, therefore a computational domain with an extra gap of 0.033 millimeters is required in order to accommodate the unconstrained displacement of the particles in the left border of the bar. Here below the time step is controlled by the courant number, in this case, we adopt a middle value of it (CFL=0.5) as a compromise between stability and numerical accuracy.

The analytical solution in terms of velocity 12, and stress 13 can be found in the appendix A given by the Method of characteristics.

For the convergence analysis, the root-mean-square (RMS) error in the velocity field is computed. RMS error is defined as follows

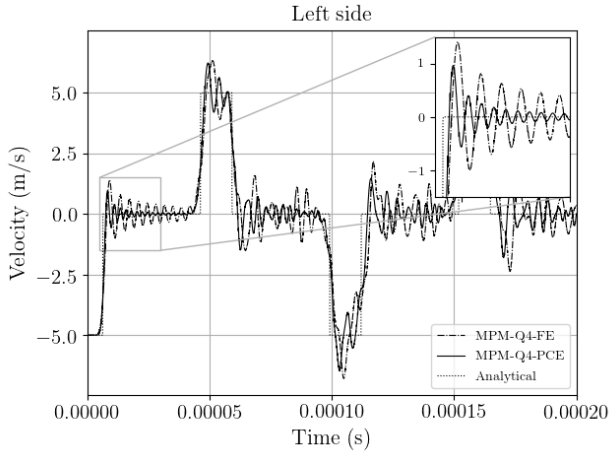$$RMS = \sqrt{\frac{1}{N}\sum_{p}^{N}(v_p - \hat{v}_p)^2}, \tag{40}$$

Fig. 5: Velocity evolution at the point in the bar left side.
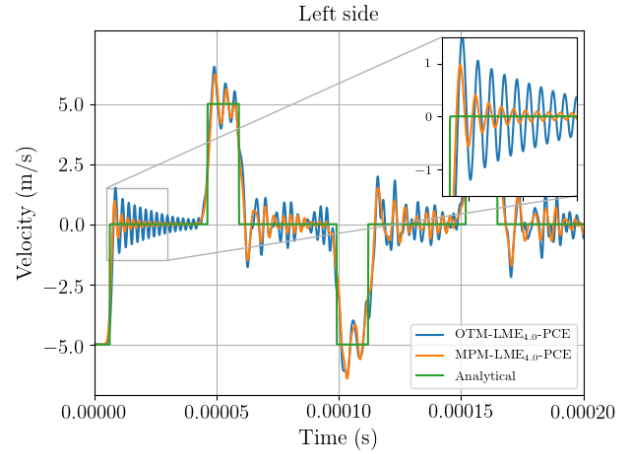

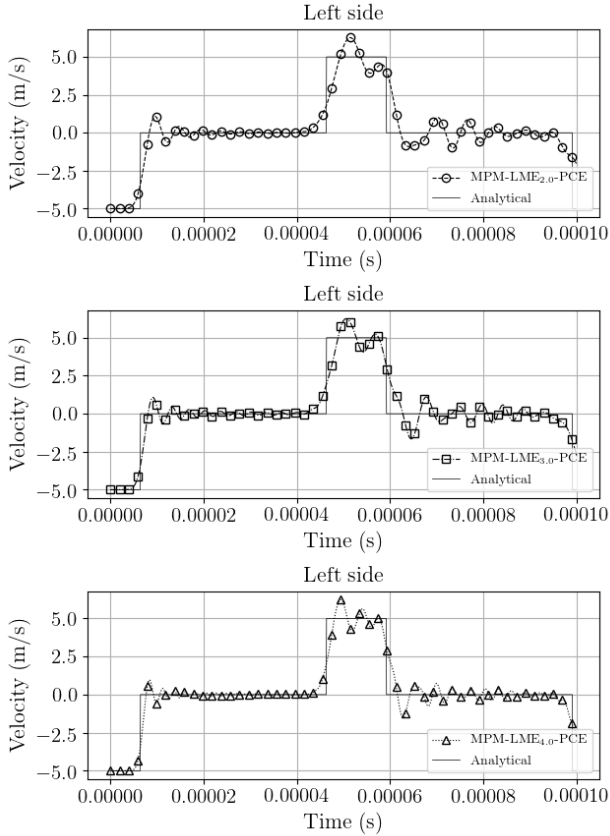Fig. 7: Velocity evolution at the point in the bar left side.

6.2 Andersen block

In order to test the ability of this interpolation technique to deal with grid crossing instabilities we will simulate the vertical compression of a square block (10 by 10 meters) of soft soil sketched in figure 6.2 and loaded using an incremental gravitation scheme.

This test was taken from Andersen (2009)[**?**]. The elastic parameters consider for this test are:

– Initial density : $6 \cdot 10^3 \ kg/m^3$
– Poisson ratio : 0
– Elastic modulus : 5 *MPa*

The gravity force is a apply as an external force according to the equations (10), (17). Using a total time period of T (20 seconds) to apply the gravity, it is increased from 0 to $9.81 m/s$ with a sinus function until T/2 and then maintained constant until T in order to arrive to a state of equilibrium,

$$\mathbf{g}(t) = \begin{cases} 0.5\mathbf{g}(\sin(\frac{2t\pi}{T} - \frac{\pi}{2}) + 1) & \text{if } t \leq T/2 \\ \mathbf{g} & \text{if } t > T/2 \end{cases} \qquad (41)$$

In order to get a stable solution, we will adopt a time step conducted by a Courant number of 0.1. On the other hand, the explicit predictor-corrector scheme is here employed looking forward getting better results. For the initial spatial discretization we will employ four particles per cell ($\Delta x = 2 \ m$). The initial setting of particles inside of the cell changes according to the interpolation technique adopted. For the bi-linear shape functions (Q4) and the LME shape functions, the initial position corresponds to the location of the gauss-points in a standard quadratic finite element. For the uniform-GIMP (uGIMP) shape function the initial position of each particle is located in the center of each voxel, due to the fact that in the initial situation, the voxel domain should not overlap each others, Steffen *et al.* (2008)[**?**]. Figure 10 shows the vertical displacement evolution of a point in the free surface of the block.

Fig. 6: Velocity evolution at the point in the bar left side.

where v_p and $\hat{v}_p$ are respectively the analytical and numerical solutions evaluated in the final time step in the position of each particle.
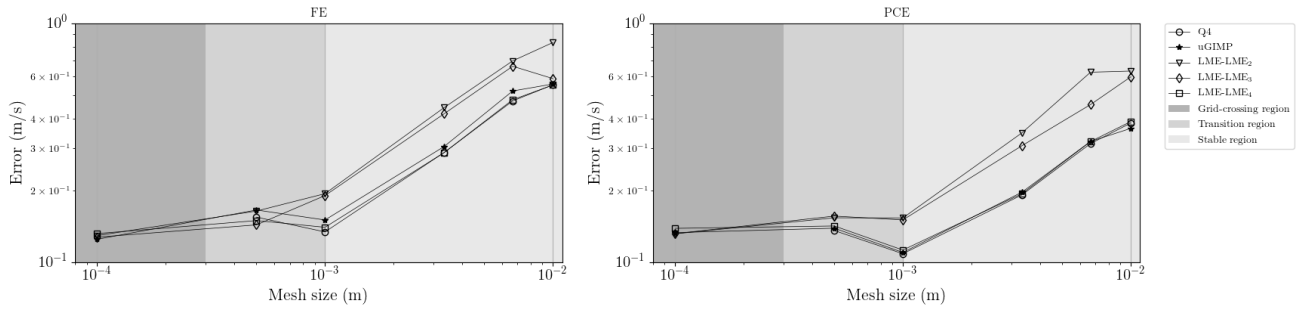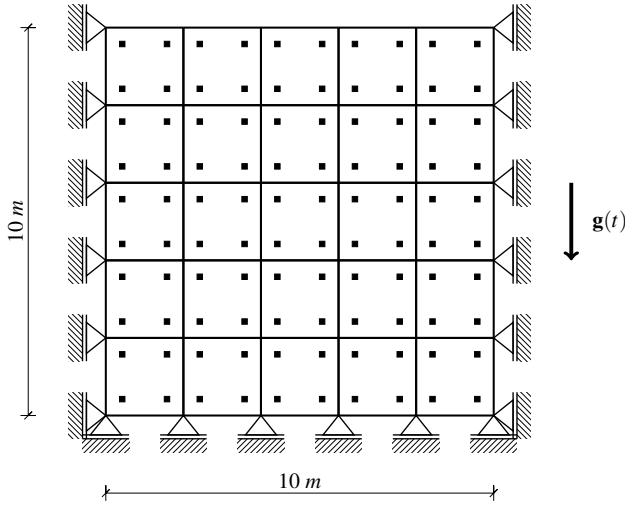
Fig. 8: Velocity evolution at the point in the bar left side.

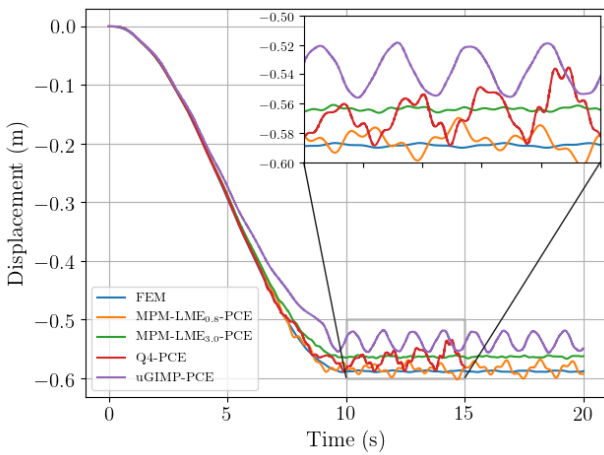Fig. 9: Geometrical description of a soil block

.

Fig. 10: Comparative of the vertical displacement evolution in a point located in the free surface employing different interpolation schemes and numerical techniques.

As we can see the simulation using the Q4 interpolation technique turns out to be unstable and in the second 15 it fails. The uGIMP simulation is more stable than the one performed by the Q4. Despite this is still unstable and could trigger severe oscillations if we pretend to simulate non-linear materials. The LME simulation was performed using two kinds of shape functions, one with a low value of the dimensionless parameter, $\gamma = 0.8$, and other with a larger value of it, $\gamma = 3.0$. Notice that the results are both stable, but the larger values of γ give us a very stable solution. This is due to the fact that with larger value of γ, the shape functions behaves in a similar way to the FEM, which performs very accurate in those cases with a reasonable mesh distorsion, and with a lower value it behaves in a similar way to the uGIMP. This behaviour was noticed previously by [**?**], were authors highlight how by adjusting the spatial variation of $\beta(x)$, it is possible to select regions of the domain of analysis which are treated by finite elements and regions that are treated in the style of meshfree methods, with seamless transitions between those regions.

Figure 11 shows the evolution of the vertical stress during the loading process. The result is physically realistic as stress increments linearly from the top to the bottom of the specimen, and the value of the vertical stress in a material point located in the bottom of the specimen oscillates centered in $5.2 MPa$, which is the analytic value given by $\sigma_{yy} = \rho g h_y$.

7 Conclusions

We have developed a new time integration scheme for the MPM, and proved how local *max-ent* arise as a promising tool to overcome the grid-crossing error.

adapt the value of β to solve the equations fem-like of meshfree-like depending of how behaves the region

This alternative technique overcome the constriction of the GIMP to regular mesh.

The time integration here proposed mitigates the presence of wiggles and overshot

future research : localization of the local max ent
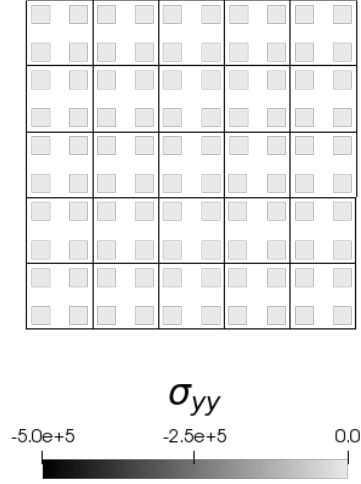
(a) t = 0 seconds.
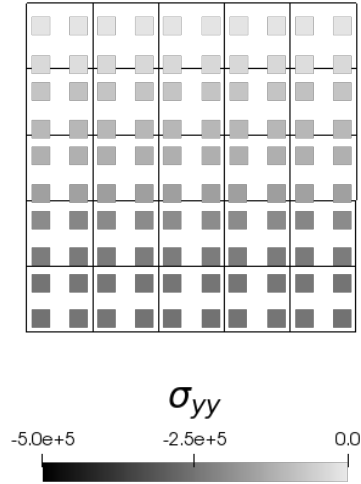
(b) t = 5 seconds.
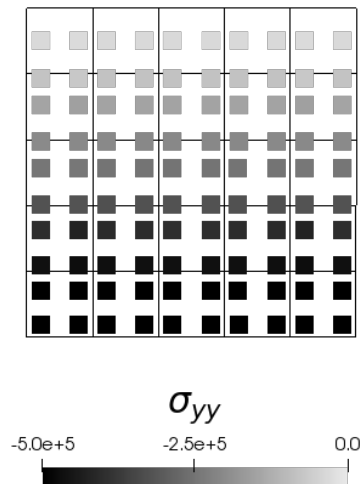
predictor-corrector for high dynamic problems like impact or collisions

Conflict of interest

The authors declare that they have no conflict of interest.

A The analytical solution of the 1D Dyka benchmark

For the derivation of this analytical solution we will consider the dynamic behaviour of a 1D elastic bar. The governing equations are the following: (i) The balance of linear momentum,

$$\rho \frac{\partial v}{\partial t} = \frac{\partial \sigma}{\partial x}, \tag{42}$$

where σ is the stress value, ρ is the density, and v is the velocity. (ii) The constitutive model, which for convenience of the following developments will be written in terms of displacement and velocities as,

$$\frac{\partial \sigma}{\partial t} = E \frac{\partial \varepsilon}{\partial t}, \tag{43}$$

where E is the elastic modulus. (iii) The compatibility equation also in terms of the velocity field,

$$\frac{\partial \varepsilon}{\partial t} = \frac{\partial v}{\partial x}. \tag{44}$$

Next for simplicity, we will introduce (44) in (43), so we get the following system of equations,

$$\frac{\partial v}{\partial t} = \frac{1}{\rho} \frac{\partial \sigma}{\partial x}, \tag{45}$$

$$\frac{\partial \sigma}{\partial t} = E \frac{\partial v}{\partial x}. \tag{46}$$

Introducing (46) in (45) and expressing the remaining equation in terms of the displacement, we reach the 1D wave equation for linear elastic materials,

$$\frac{\partial^2 u}{\partial t^2} = \frac{E}{\rho} \frac{\partial^2 u}{\partial x^2} = c^2 \frac{\partial^2 u}{\partial x^2} \tag{47}$$

where we have introduced the wave celerity c as,

$$c = \sqrt{\frac{E}{\rho}} \tag{48}$$

Alternative, rearranging both equations (45) and (46) it is possible to join them in a single system of equations as,

$$\frac{\partial}{\partial t} \begin{bmatrix} \sigma \\ v \end{bmatrix} + \begin{bmatrix} 0 & -E \\ -1/\rho & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial \sigma}{\partial x} \\ \frac{\partial v}{\partial x} \end{bmatrix} = \mathbf{0}. \tag{49}$$

This expression can be written in a more compact format as,

$$\frac{\partial \phi}{\partial t} + \mathbf{A} \frac{\partial \phi}{\partial x} = \mathbf{0} \tag{50}$$

where both variables are joined in a single vectorial variable ϕ and $\mathbf{A}$ in coupling matrix between both equations,

$$\phi = \begin{bmatrix} \sigma \\ v \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 0 & -E \\ -1/\rho & 0 \end{bmatrix}.$$

Note that the nature of is still hyperbolic despite the fact it does not have a second order temporal derivative as (47). A proof of this can be easily obtained if we get the zeros of the hypersurface defined by (47). And later the eigenvalues of $\mathbf{A}$ in (50). In both cases, eigenvalues are real and distinct ($\lambda = \pm\sqrt{\frac{E}{\rho}}$), therefore the system is called strictly hyperbolic.

For a more general description in the following, we will assume that $\mathbf{A}$ has n different eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_i, \ldots \lambda_n\}$ and n eigenvectors $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^i, \ldots \mathbf{x}^n\}$ satisfying that $\mathbf{Ax} = \lambda\mathbf{x}$. Now we introduce the matrix $\mathbf{P}$ whose columns are the n eigenvalues $\mathbf{x}$

$$\mathbf{P} = \{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \ldots \mathbf{x}^n\}. \tag{51}$$

Diagonalizing $\mathbf{A}$ using $\mathbf{P}$ we get

$$\Lambda = \mathbf{P}^{-1}\mathbf{A}\,\mathbf{P}, \tag{52}$$

where $\Lambda_{ii} = \lambda_i$. Next we will define a vector $\mathfrak{R}$ such that:

$$\phi = \mathbf{P}\,\mathfrak{R} \tag{53}$$

we will assume to be integrable. Expanding the above expression with the chain rule and passing the matrix $\mathbf{P}$ to left hand side of the equality we get,

$$d\mathfrak{R} = \frac{\partial\mathfrak{R}}{\partial t}dt + \frac{\partial\mathfrak{R}}{\partial x}dx = \mathsf{P}^{-1}\left(\frac{\partial\phi}{\partial t}dt + \frac{\partial\phi}{\partial x}dx\right) \tag{54}$$

and setting the terms we get,

$$\frac{\partial\mathfrak{R}}{\partial t} = \mathbf{P}^{-1}\frac{\partial\phi}{\partial t}, \quad \frac{\partial\mathfrak{R}}{\partial x} = \mathbf{P}^{-1}\frac{\partial\phi}{\partial x} \tag{55}$$

Next, if we multiply (50) by $\mathbf{P}^{-1}$ we get:

$$\mathbf{P}^{-1}\frac{\partial\phi}{\partial t} + \left(\mathbf{P}^{-1}\mathbf{AP}\right)\mathbf{P}^{-1}\frac{\partial\phi}{\partial x} = \mathbf{0} \tag{56}$$

finally introducing the expressions (55) we reach to

$$\frac{\partial\mathfrak{R}}{\partial t} + \Lambda\frac{\partial\mathfrak{R}}{\partial x} = \mathbf{0} \tag{57}$$

which consists of n uncoupled equations as Λ is diagonal matrix as we can see in (52). Each of this equations are 1D scalar convective transport equations, with solutions of the form:

$$\mathfrak{R}^{(i)} = F^{(i)}\left(x - \lambda^{(i)}t\right) \tag{58}$$

This uncoupled system, has, therefore, a set of n characteristics. These magnitudes $\mathfrak{R}_i$ which propagate along characteristics are known as "Riemann invariants" of the problem. Here we have a 1D configuration, so the domain is $\Omega : (0, L) \times (0, T)$. For the closure of the problem we require:

– "n" initial conditions of the form $\mathfrak{R}_i(x, t = 0) = h_i(x)$, where $i = 0, \ldots, n$, and $h_i(x)$ is a vectorial function given by the physical variables of the problem.
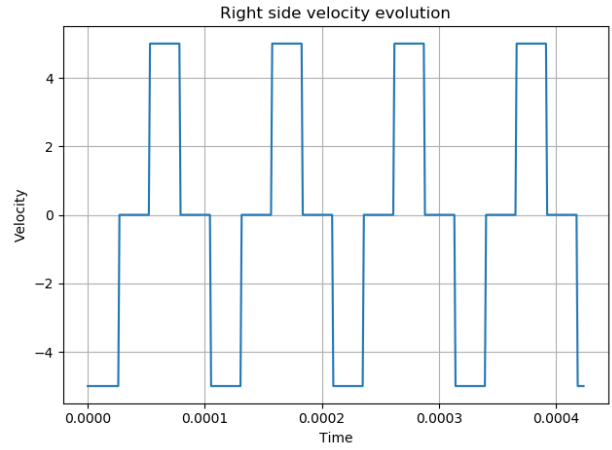– "n" boundary conditions.

Fig. 12: Analytical solution for the velocity in the right side of the Dyka bar.

Now particularizing the previous equations for the 1D elastic bar described in [?], we get that the matrix $\mathbf{P}$ is the following:

$$\mathbf{P} = \begin{bmatrix} -\sqrt{E\rho} & \sqrt{E\rho} \\ 1 & 1 \end{bmatrix}$$

and its inverse is:

$$\mathbf{P}^{-1} = \frac{1}{2\sqrt{E\rho}}\begin{bmatrix} -1 & \frac{1}{\sqrt{E\rho}} \\ 1 & \frac{1}{\sqrt{E\rho}} \end{bmatrix}$$

And introducing the value of the inverse matrix $\mathbf{P}^1$ in the Riemann definition (53) we get the following system of equations,

$$\mathfrak{R}^I = \frac{1}{2\sqrt{\rho E}}\left(-\sigma + v\sqrt{\rho E}\right) \tag{59}$$

$$\mathfrak{R}^{II} = \frac{1}{2\sqrt{\rho E}}\left(\sigma + v\sqrt{\rho E}\right) \tag{60}$$

From (59) and (60) we can obtain the values of the stress and the velocity as:

$$v = \mathfrak{R}^I + \mathfrak{R}^{II} \quad , \quad \sigma = \sqrt{E\rho}\left(\mathfrak{R}^{II} - \mathfrak{R}^I\right) \tag{61}$$

The boundary conditions are in both cases of radiation as there is not wave in-going from the exterior. So for the right side (fixed boundary) we get the following conditions:

$$\mathfrak{R}^{II} = 0 \quad and \quad v_{x=L} = 0$$

Therefore $\sigma_{x=L} = -2\sqrt{\rho E}\,\mathfrak{R}^I$. And in the left side (free boundary) we get the following conditions:

$$\mathfrak{R}^I = 0 \quad and \quad \sigma_{x=0} = 0$$

Therefore $v_{x=0} = 2\mathfrak{R}^{II}$. Finally, applying this conditions in the elastic bar sketched in 4, is possible to obtain the velocity history in the right side of the bar 12 and the stress in the last quarter side of the Dyka bar 13 as is demanded in [?].
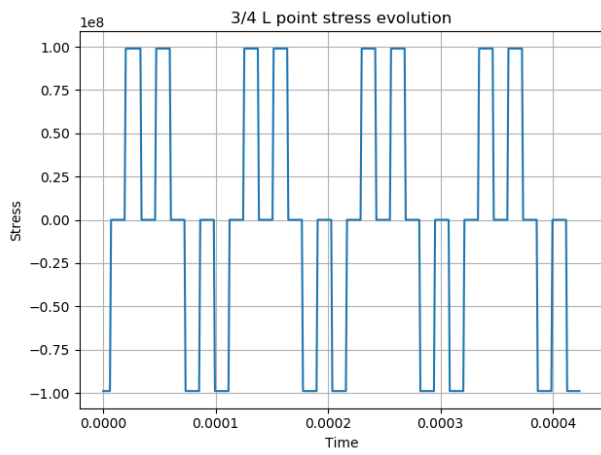
Fig. 13: Analytical solution for the stress in the last quarter
of the Dyka bar.