

# Particle-based solid for nonsmooth multidomain dynamics

John Nordberg<sup>1</sup> · Martin Servin<sup>1</sup> 

Received: 5 January 2017 / Accepted: 14 March 2017  
© The Author(s) 2017. This article is an open access publication

**Abstract** A method for simulation of elastoplastic solids in multibody systems with nonsmooth and multidomain dynamics is developed. The solid is discretised into pseudo-particles using the meshfree moving least squares method for computing the strain tensor. The particle's strain and stress tensor variables are mapped to a compliant deformation constraint. The discretised solid model thus fit a unified framework for nonsmooth multidomain dynamics simulations including rigid multibodies with complex kinematic constraints such as articulation joints, unilateral contacts with dry friction, drivelines, and hydraulics. The nonsmooth formulation allows for impact impulses to propagate instantly between the rigid multibody and the solid. Plasticity is introduced through an associative perfectly plastic modified Drucker–Prager model. The elastic and plastic dynamics are verified for simple test systems, and the capability of simulating tracked terrain vehicles driving on a deformable terrain is demonstrated.

**Keywords** Multidomain · Nonsmooth dynamics · Meshfree · Elastoplastic · Deformable terrain · Multibody dynamics

## 1 Introduction

We address the modelling and simulation of multidomain systems with nonsmooth dynamics and deformable solids of particulate nature. This includes mechatronic multibody systems with nonsmooth dynamics, such as vehicles, robots, and processing machinery that handle or operate on soil or

bulk materials. Fast multidomain simulation is useful for concept design exploration, development of control strategies, and for interactive real-time simulators, e.g. for operator training, human–machine interaction studies, and hardware-in-the-loop testing.

### 1.1 Nonsmooth multidomain dynamics

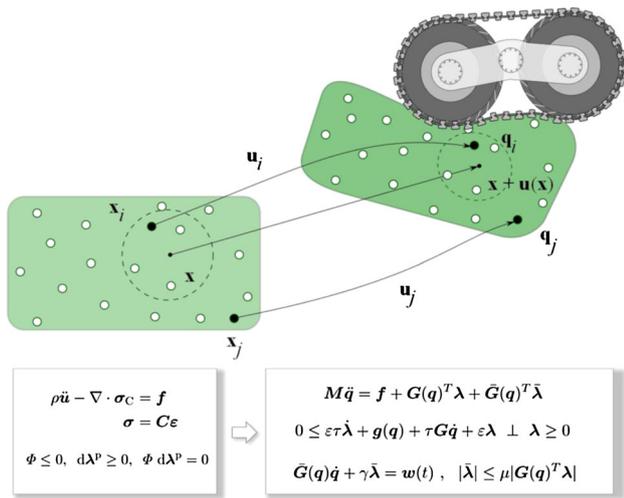
Multidomain system simulation requires integration of multiple heterogeneous subsystems into a single full-system model. Mechatronic systems are typically composed by rigid and flexible multibodies coupled by kinematic constraints for modelling of joints and sets of differential algebraic equation (DAE) models for electronics, hydraulics, and powertrain dynamics [1]. If the subsystems are not strongly coupled, the full-system simulation can be realised by means of co-simulation [2]. At coarse timescales, however, the dynamics need to be treated as nonsmooth [3–5]. This means that velocities are allowed to be arbitrarily discontinuous and impulses—from impacts, joint limits, and frictional stick–slip transitions—can propagate instantly throughout the system. This enables implicit time integration using large timesteps. To ensure numerical stability and computational efficiency, the full system must be approached with a consistent mathematical framework and adequate numerical solvers. Current techniques for co-simulation do not support this for strongly coupled systems with nonsmooth dynamics.

### 1.2 Particulate-based solids

The versatile and widely used discrete element method (DEM) [6] for simulating particulate solids have been extended to the framework of nonsmooth dynamics [4,5,7,8]. When the lengthscale of deformations is much larger than the particle size, it is more computationally favourable to

✉ Martin Servin  
martin.servin@umu.se

<sup>1</sup> Umeå University, Umeå, Sweden



**Fig. 1** Illustration of the idea of using particle-based discretisation and compliant deformation constraints to obtain a unified multibody system model for both deformable solids and mechatronic systems

model the solid as a continuum with elastoplastic constitutive law [9]. There are several arguments for using a meshfree discretisation [10, 11] over traditional mesh-based methods such as the finite element method (FEM). Meshfree methods allow large deformations and topological changes without need for remeshing [12]. This makes them well suited for fracturing materials and transitions from solid into viscous and free-surface flow, or from continuum to particulate level of detail when using multi-scale methods [13, 14] or model reduction techniques [15].

### 1.3 Our contribution

The main contribution in this paper, illustrated in Fig. 1, is a particle-based method for modelling and simulation of elastoplastic solids as a multibody system on descriptor form [2] and with nonsmooth dynamics. This enable numerical integration with large timestep size of multidomain systems containing deformable solids, mechatronics, and frictional contacts. The objective is to achieve real-time performance or better.

The solid is modelled as a particle system with strain tensor constraints introduced as the stiff limit of energy and dissipation potentials for elastic deformations, with constraint regularisation and stabilisation based on a discrete variational approach [16, 17]. An associative perfectly plastic Drucker–Prager model is employed using an elastic predictor–plastic corrector strategy to detect yielding and compute the plastic flow. In its basic form, this model does not yield in hydrostatic compression in contrast to many real materials. Many soils yield under hydrostatic compression by failure in the microscopic structures whereby air and fluid are released. Therefore, the Drucker–Prager model

is extended to a capped version [18] that model also plastic compaction. A meshfree method [10, 11] is chosen in order to handle large deformations and, for future development, support fracturing and transitions to viscous flow or to granular media represented by contacting discrete elements. The displacement gradient and strain tensor is approximated by the method of moving least squares (MLS) [19]. A stabilisation constraint is introduced to suppress spurious short wavelength modes associated with the MLS approximation [20].

The dynamics of other subsystems and their potentially strong coupling is treated within the same multibody dynamics framework using a variational time integrator [16, 17]. Each timestep involves solving a block-sparse mixed linear complementarity problem (MLCP) that also support modelling of frictional contacts and secondary constraints for joints and motors. The present paper is motivated by the exploration of ground vehicles on deformable terrain [21]. Current solutions enable simulation of complex vehicle models with real-time performance but not with dynamic terrain models firmly based on solid mechanics in three dimensions. Usually, empirical terrain models of Bekker–Wong type are used [22, 23]. On the other hand, there exist many solutions for more accurate offline simulations with fine spatial and temporal resolution for elastoplastic solids coupled with tire models [24, 25]. But few support inclusion of multibody systems and not in real time or faster.

### 1.4 Notations

Matrices and vectors are represented in bold face in capital and lower case,  $\mathbf{A}$  and  $\mathbf{x}$ , respectively. Latin superscripts indicate the index of a specific body  $i, j, k = 1, \dots, N$ , where  $N$  is the total number of bodies in the system. Greek subscripts indicate a specific scalar component of a vector or matrix. The Einstein summation convention is used where repeated indices imply summation over them, e.g. representing matrix–vector multiplication as  $\mathbf{Ax} = A_{\alpha\beta}x_{\beta}$ . Subscripts  $x, y$  or  $z$  indicate a specific component of a vector or matrix assuming a Cartesian coordinate system. As an example of these notations, the position vector of a body  $i$  is denoted  $\mathbf{q}^i$ . The relative position of particle  $j$  to  $i$  is  $\mathbf{q}^{ij} = \mathbf{q}^j - \mathbf{q}^i$ . The  $y$  component of this vector is  $q_y^{ij} = q_y^j - q_y^i$ . For rigid bodies, the position vector includes also the rotation variables of the body. The system position vector is  $\mathbf{q} = [\mathbf{q}^1, \mathbf{q}^2, \dots, \mathbf{q}^i, \dots, \mathbf{q}^N]$ . The gradient is denoted  $\nabla_{\mathbf{x}} = \frac{\partial}{\partial \mathbf{x}}$  and  $\nabla_{\alpha} = \frac{\partial}{\partial x_{\alpha}}$ . The dot notation is used for the time derivative  $\dot{\cdot} \equiv \frac{d}{dt}$ . The Cauchy stress tensor is represented by  $\sigma_C$  and the second Piola Kirchoff stress tensor by  $\sigma$ .

## 2 Discrete multidomain dynamics

This section provides a summary of discrete dynamics and of the numerical integration method that is used.

### 2.1 Lagrangian formulation

The Lagrangian of a constrained mechanical system with position  $\mathbf{q}$  and velocity  $\dot{\mathbf{q}}$  is

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\lambda}, \bar{\boldsymbol{\lambda}}) = T(\mathbf{q}, \dot{\mathbf{q}}) - U(\mathbf{q}) - \mathcal{R}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\lambda}^T \mathbf{g}(\mathbf{q}, t) + \bar{\boldsymbol{\lambda}}^T \bar{\mathbf{g}}(\mathbf{q}, \dot{\mathbf{q}}, t), \tag{1}$$

where  $T(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}}$  is the kinetic energy,  $U(\mathbf{q})$  is the potential energy, and  $\mathcal{R}(\mathbf{q}, \dot{\mathbf{q}})$  is the Rayleigh dissipation function. The system state vector  $(\mathbf{q}, \dot{\mathbf{q}})$  is constrained by holonomic constraints  $0 = \mathbf{g}(\mathbf{q})$ , with Jacobian  $\mathbf{G} = \partial \mathbf{g} / \partial \mathbf{q}$ , and nonholonomic (or nonintegrable kinematic) constraints  $0 = \bar{\mathbf{g}}(\mathbf{q}, \dot{\mathbf{q}}, t)$ . The corresponding Lagrange multipliers are  $\boldsymbol{\lambda}$  and  $\bar{\boldsymbol{\lambda}}$ , respectively. We will restrict to Pfaffian nonholonomic constraints,  $0 = \bar{\mathbf{g}}(\mathbf{q}, \dot{\mathbf{q}}, t) \equiv \bar{\mathbf{G}} \dot{\mathbf{q}} - \mathbf{w}(t)$ . Nonholonomic constraints can, by definition, not be deduced by differentiation of a holonomic constraint [26].

To ensure numerical stability, it is common to regularise the constraints. This may be done by treating them as the limit of strong potentials and dissipation functions  $U_\varepsilon(\mathbf{q}) = \frac{1}{2\varepsilon} \mathbf{g}^T \mathbf{g}$  and  $\mathcal{R}_\gamma(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2\gamma} \bar{\mathbf{g}}^T \bar{\mathbf{g}}$  with  $\varepsilon, \gamma \rightarrow 0$  [27]. In fact, any stiff force can be transformed into a regularised constraint through a Legendre transform [17]

$$U_\varepsilon(\mathbf{q}) = \boldsymbol{\lambda}^T \mathbf{g} - \frac{\varepsilon}{2} \boldsymbol{\lambda}^T \boldsymbol{\lambda}, \tag{2}$$

$$\mathcal{R}_\gamma(\mathbf{q}, \dot{\mathbf{q}}) = \bar{\boldsymbol{\lambda}}^T \bar{\mathbf{g}} - \frac{\gamma}{2} \bar{\boldsymbol{\lambda}}^T \bar{\boldsymbol{\lambda}}. \tag{3}$$

The corresponding Euler–Lagrange equations of motion are

$$\mathbf{M} \ddot{\mathbf{q}} + \dot{\mathbf{M}} \dot{\mathbf{q}} - \mathbf{G}(\mathbf{q})^T \boldsymbol{\lambda} - \bar{\mathbf{G}}(\mathbf{q})^T \bar{\boldsymbol{\lambda}} = \mathbf{f}, \tag{4}$$

$$\varepsilon \boldsymbol{\lambda} + \mathbf{g}(\mathbf{q}) = 0, \tag{5}$$

$$\gamma \bar{\boldsymbol{\lambda}} + \bar{\mathbf{G}}(\mathbf{q}) \dot{\mathbf{q}} = \mathbf{w}(t), \tag{6}$$

where  $\mathbf{f} \equiv -\nabla_{\mathbf{q}} U(\mathbf{q}) - \nabla_{\dot{\mathbf{q}}} \mathcal{R}(\mathbf{q}, \dot{\mathbf{q}})$  are the explicit forces from nonstiff potentials. This constitutes a system of DAE of index 1, which are easier to integrate numerically than the corresponding higher index DAE in the absence of regularisation and stabilisation.  $\varepsilon = \gamma = 0$ . A term for dissipation of motion orthogonal to the holonomic constraint surface can be added to improve the convergence. Also dissipation can be physically based by introducing it as a Legendre transform of a Rayleigh dissipation function  $\mathcal{R}_\tau(\mathbf{q}, \dot{\mathbf{q}}) = \frac{\tau}{2\varepsilon} \dot{\mathbf{g}}^T \dot{\mathbf{g}} \rightarrow \tau \dot{\boldsymbol{\lambda}}^T \dot{\mathbf{g}} - \frac{\tau\varepsilon}{2} \dot{\boldsymbol{\lambda}}^T \dot{\boldsymbol{\lambda}}$ , with damping parameter  $\tau$ . This modifies Eq. (5) to

$$\varepsilon \boldsymbol{\lambda} + \varepsilon \tau \dot{\boldsymbol{\lambda}} + \mathbf{g}(\mathbf{q}) + \tau \mathbf{G} \dot{\mathbf{q}} = 0. \tag{7}$$

The compliance and damping factors  $\varepsilon, \tau$  and  $\gamma$  are not restricted to being scalar or diagonal. In what follows these are assumed to be matrices.

### 2.2 Time discretisation

Variational integration [28] provides a systematic approach to derive time integration schemes with good properties, e.g. conservation of discrete symplectic forms and discrete momentum maps. Rather than discretising the Euler–Lagrange equations of motion directly, the Lagrangian and principle of least action are defined in time-discrete form. Employing semi-implicit Euler discretisation and linearising the constraint as  $\mathbf{g}(\mathbf{q} + \Delta \mathbf{q}) = \mathbf{g}(\mathbf{q}) + \mathbf{G} \Delta \mathbf{q}$  leads to the following stepping scheme [16, 17]

$$\mathbf{q}_{n+1} = \mathbf{q}_n + h \dot{\mathbf{q}}_{n+1}, \tag{8}$$

$$\begin{bmatrix} \mathbf{M} & -\mathbf{G}^T & -\bar{\mathbf{G}}^T \\ \mathbf{G} & \boldsymbol{\Sigma} & 0 \\ \bar{\mathbf{G}} & 0 & \bar{\boldsymbol{\Sigma}} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_{n+1} \\ \boldsymbol{\lambda} \\ \bar{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_n \\ \mathbf{v}_n \\ \boldsymbol{\omega}_n \end{bmatrix} \tag{9}$$

where  $\mathbf{p}_n = \mathbf{M} \dot{\mathbf{q}}_n + h \mathbf{f}_n$ ,  $\mathbf{v}_n = -\frac{4}{h} \boldsymbol{\Upsilon} \mathbf{g} + \boldsymbol{\Upsilon} \mathbf{G} \dot{\mathbf{q}}_n$  and regularisation and stabilisation matrices  $\boldsymbol{\Sigma} = \text{diag}(4\varepsilon/h^2[1 + 4\tau/h])$ ,  $\bar{\boldsymbol{\Sigma}} = \text{diag}(\gamma/h)$  and  $\boldsymbol{\Upsilon} = \text{diag}([1 + 4\tau/h]^{-1})$ . Equation (9) is a linear system of  $N = \text{dim}(\dot{\mathbf{q}}) + \text{dim}(\mathbf{g}) + \text{dim}(\bar{\mathbf{g}})$  equations. The matrix on the left-hand side is block-sparse, positive definite, and nonsymmetric. The regularisation appearing as the diagonal perturbation matrices  $\boldsymbol{\Sigma}$  and  $\bar{\boldsymbol{\Sigma}}$  is needed for handling otherwise ill-posed or ill-conditioned problems, e.g. systems with constraint degeneracy and large mass ratios. The stabilisation terms  $-\frac{4}{h} \boldsymbol{\Upsilon} \mathbf{g} + \boldsymbol{\Upsilon} \mathbf{G} \dot{\mathbf{q}}_n$  on the right-hand side counteract constraint violations, e.g. sudden and large contact penetrations at impacts or small numerical constraint drift. The presented stepping scheme, referred to as SPOOK, has been proved to be linearly stable [16], and numerical simulations suggest a large domain of nonlinear stability.

The combined effect of the regularisation and stabilisation terms is to bring elastic and viscous properties to motion orthogonal to the constraint surface  $\mathbf{g}(\mathbf{q}) = 0$ , e.g. for modelling elasticity in mechanical joints. The parameters  $\varepsilon$  and  $\gamma$  need not be chosen arbitrarily, as for the conventional approaches to constraint regularisation and stabilisation, but can be based on physics models using parameters that can be derived from first principles, found in the literature or be identified by experiments. This becomes straightforward when the regularisation is introduced by potential energy as quadratic functions in  $\mathbf{g}$ , i.e.  $U_\varepsilon(\mathbf{q}) = \frac{1}{2} \mathbf{g}^T \boldsymbol{\varepsilon}^{-1} \mathbf{g}$ . This has been exploited in previous work to constraint-based mod-

elling of lumped element beams [29], wires [30], meshfree fluids [31], and granular material [8].

### 2.3 Nonsmooth dynamics

In discrete time, some dynamics is best treated as *nonsmooth* [3], meaning that the velocity may change discontinuously in accordance with some impact law, expressed in terms of inequality constraints or complementarity conditions, in addition to the equations of motion. This is an efficient way of modelling dynamical contacts, dry friction, joint limits, electric and hydraulic circuit switching and actuator dynamics in discrete time with large step size. The alternative is to use fine enough temporal resolution where the dynamics appear smooth and may be modelled by DAEs or ordinary differential equations alone. In general, this is an intractable approach for full-system simulations and interactive applications. The nonsmooth dynamics can be formally treated as differential variational inequalities [32]. The discrete equations of motion (9) can be linearised into a mixed linear complementarity problem (MLCP), when nonsmooth dynamics is included

$$\begin{aligned} \mathbf{H}\mathbf{z} + \mathbf{r} &= \mathbf{w}_+ - \mathbf{w}_- \\ 0 &\leq \mathbf{w}_+ \perp \mathbf{z} - \mathbf{l} \geq 0 \\ 0 &\leq \mathbf{w}_- \perp \mathbf{u} - \mathbf{z} \geq 0, \end{aligned} \tag{10}$$

where  $\mathbf{H}\mathbf{z}$  corresponds to the left-hand side of Eq. (9),  $\mathbf{r}$  is the negated right-hand side, and  $\mathbf{w}_\pm$  are slack variables. The terms  $\mathbf{u}$  and  $\mathbf{l}$  correspond to the upper and lower limits on the solution vector  $\mathbf{z}$ , respectively. The original linear system of equations is recovered by assigning  $\pm\infty$  to the upper and lower limits, i.e. no limit.

Rigid body contacts are modelled by the Signorini–Coulomb law for unilateral dry frictional contacts [5], denoting the penetration depth by  $\delta$ , contact normal and tangent by  $\mathbf{n}$  and  $\mathbf{t}$ . The Signorini–Coulomb law states that if the nonpenetration constraint,  $g_n \equiv \delta \leq 0$ , is violated, then the normal contact velocity and constraint force are complementary to ensure separation,  $0 \leq \dot{g}_n \perp \lambda_n \geq 0$ , and the tangential friction force that acts to maintain zero slip,  $g_t \equiv \mathbf{t}^T \dot{\mathbf{q}} = 0$ , is bound by the Coulomb friction cone,  $|\lambda_t| \leq \mu \lambda_n$ . The latter may be linearised by approximating the cone with a polyhedral or a box. Impacts are treated post facto in a separate *impact stage* succeeding the update of velocities and positions. At the impact stage, an impulse transfer is applied enabling discontinuous velocity changes, i.e. from  $\dot{\mathbf{q}}^-$  to  $\dot{\mathbf{q}}^+$ . Newtons’ impact law,  $\mathbf{n}^T \dot{\mathbf{q}}^+ = -e \mathbf{n}^T \dot{\mathbf{q}}^-$ , is used with restitution coefficient  $e$  in conjunction with preserving all other constraints,  $\mathbf{G}\dot{\mathbf{q}} = 0$ . This amounts to solving the same MLCP (10) but with  $\mathbf{p}_n = \mathbf{M}\dot{\mathbf{q}}_n$  and  $\mathbf{v}_n = 0$  on the right-hand side of Eq. (9).

A fixed timestep approach is preferred when aiming for fast full-system simulation with many nonsmooth events per unit time, which is the case with many dynamic contacts. The alternative, using variable timestep and exact event location, makes the numerical integration computationally intractable and may fail by occurrence of Zeno points.

### 2.4 Heterogeneous multidomain dynamics

In multidomain simulation with strongly coupled subsystems, e.g. an elastoplastic solid and mechatronic system, the dynamics propagate instantly throughout the full system. The nonsmooth multidomain dynamics approach provides a general framework for implicit integration of such systems that enable stable simulation at large timesteps. This avoid the loss in performance and numerical stability associated with explicit co-simulation algorithms. A system composed of two strongly coupled subsystems,  $A$  and  $B$ , takes the following form

$$\mathbf{H}\mathbf{z} = \left[ \begin{array}{cc|c} \mathbf{H}_A & 0 & -\mathbf{G}_{AB_A}^T \\ 0 & \mathbf{H}_B & -\mathbf{G}_{AB_B}^T \\ \hline \mathbf{G}_{AB_A} & \mathbf{G}_{AB_B} & \boldsymbol{\Sigma}_{AB} \end{array} \right] \begin{bmatrix} \mathbf{z}_A \\ \mathbf{z}_B \\ \boldsymbol{\lambda}_{AB} \end{bmatrix}, \tag{11}$$

where  $\boldsymbol{\lambda}_{AB}$ ,  $[\mathbf{G}_{AB_A}, \mathbf{G}_{AB_B}]$ ,  $\boldsymbol{\Sigma}_{AB}$  are the multiplier, Jacobian, and regularisation of the subsystem coupling constraints. The individual subsystem matrices,  $\mathbf{H}_A$  and  $\mathbf{H}_B$ , take the generic form of a saddle point matrix as in Eq. (9).

## 3 Constraint-based meshfree elastoplastic solid

This section describes the elastoplastic model that is assumed, the spatial discretisation method and the compliant deformation constraint.

### 3.1 Elastoplastic solid

The solid is assumed to sustain geometrically large deformations. The St. Venant–Kirchhoff elasticity model is used in combination with the Drucker–Prager plasticity model [9]. The material strain is expressed by the Green–Lagrange strain tensor

$$\epsilon_{\alpha\beta}(\mathbf{x}) = \frac{1}{2} (\nabla_\alpha u_\beta + \nabla_\beta u_\alpha + \nabla_\alpha u_\gamma \nabla_\beta u_\gamma), \tag{12}$$

where  $\mathbf{u}(\mathbf{x})$  is the displacement field mapping a reference coordinate  $\mathbf{x}$  to displaced position  $\tilde{\mathbf{x}}(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x})$ . Observe that the Green–Lagrange strain tensor transforms under large rotations and no co-rotation procedure is needed. It is convenient to use the Voigt notation for representing the strain tensor on vector format,  $\boldsymbol{\epsilon} =$

$[\epsilon_{xx}, \epsilon_{yy}, \epsilon_{zz}, 2\epsilon_{yz}, 2\epsilon_{xz}, 2\epsilon_{xy}]^T$ , and the second Piola Kirchoff stress tensor  $\sigma = [\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{yz}, \sigma_{xz}, \sigma_{xy}]^T$ , related to the Cauchy stress by  $\sigma_C = J^{-1}F^T\sigma F$  with displacement gradient  $F = \nabla u$  and  $J = \det F$ . The linear constitutive law reads  $\sigma = C\epsilon$ , with stiffness matrix

$$C = \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix}, \quad (13)$$

where  $\lambda$  and  $\mu$  are the first and second Lamé parameters which are related to the Young’s modulus and Poisson’s ratio,  $E$  and  $\nu$  by  $\lambda = E\nu / (1 + \nu)(1 - 2\nu)$  and  $\mu = E/2(1 + \nu)$ . The corresponding strain energy density is

$$U(x) = \frac{1}{2}\epsilon^T C \epsilon = \lambda(\text{tr } \epsilon)^2 + 2\mu \text{tr}(\epsilon^2). \quad (14)$$

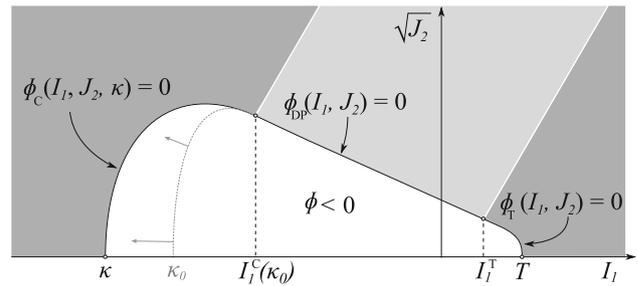
Plastic deformation occurs when the stress of a material reaches the critical yield strength,  $\Phi(\sigma) = 0$ . The choice of yield function,  $\Phi(\sigma)$ , is based on the type of material being modelled. For simplicity, we restrict ourself to small plastic deformation and use the second Piola Kirchoff stress to evaluate the yield function and plastic increment. The strain tensor is additively decomposed into an elastic and a plastic component,  $\epsilon = \epsilon^e + \epsilon^p$ , where the latter store the permanent plastic deformation. In ideal plasticity, the plastic deformation occurs instantly according to a flow rule  $d\epsilon^p = d\lambda^p \frac{\partial \Psi}{\partial \sigma}$ , where  $\Psi(\sigma)$  is the plastic potential and  $\lambda^p$  is the plastic multiplier. If  $\Psi(\sigma) = \Phi(\sigma)$ , the model is said to be associative and otherwise nonassociative. The plastic flow lasts as long as the plastic multiplier is positive  $d\lambda^p > 0$ , incrementally reducing the stress  $d\sigma^p$ , until it reaches the elastic regime,  $\Phi(\sigma) < 0$ . This constitutes a nonlinear complementarity problem known as Karush–Kuhn–Tucker conditions

$$\Phi \leq 0, \quad d\lambda^p \geq 0, \quad \Phi d\lambda^p = 0. \quad (15)$$

The plastic multiplier is computed from the constitutive law, which in the plastic flow phase is  $d\sigma = C^p d\epsilon$ , where the elastoplastic tangent stiffness matrix is

$$C^p = C - \frac{C \frac{\partial \Psi}{\partial \sigma} (\frac{\partial \Phi}{\partial \sigma})^T C}{(\frac{\partial \Phi}{\partial \sigma})^T C \frac{\partial \Psi}{\partial \sigma}}. \quad (16)$$

Predictor–corrector algorithms are conventionally used to integrate the plastic flow. In the absence of plastic hardening or softening, the plastic deformation, plastic multiplier, and total stress can be computed easily using the radial return algorithm [9] summarised in Algorithm 2.



**Fig. 2** Capped Drucker–Prager yield surface. The nonshaded region indicates the elastic domain, which grows under plastic compression

The assumed plasticity model is a capped Drucker–Prager model, following Dolarevic [18], with a compaction variable  $\kappa$ . The yield function  $\Phi(\sigma, \kappa)$  is  $C^1$  continuous consisting of the Drucker–Prager yield surface with a tension and a compression cap function according to

$$\Phi(\sigma, \kappa) = \begin{cases} \Phi_T(I_1, J_2) & I_1 \geq I_1^T \\ \Phi_{DP}(I_1, J_2) & I_1^T > I_1 > I_1^c(\kappa) \\ \Phi_c(I_1, J_2, \kappa) & I_1^c(\kappa) \geq I_1 \end{cases}, \quad (17)$$

where  $I_1 = \text{tr}(\sigma)$  is the first invariant of the stress tensor and  $J_2 = 0.5 \text{tr}(\bar{\sigma}^2)$  is the second invariant of the deviatoric stress tensor  $\bar{\sigma} = \sigma - \frac{1}{3}I_1 \mathbf{1}$ . The expressions for the tension cap  $\Phi_T(I_1, J_2)$  and compression cap  $\Phi_c(I_1, J_2, \kappa)$  are given in “Appendix”, and the Drucker–Prager surface is defined as

$$\Phi_{DP}(I_1, J_2) = \sqrt{J_2} + \frac{\eta}{3}I_1 - \xi c, \quad (18)$$

where  $\phi$  is the internal friction angle and  $c$  the cohesion parameter such that  $\eta = 6 \sin \phi / \sqrt{3} (3 - \sin \phi)$  and  $\xi = 6 \cos \phi / \sqrt{3} (3 - \sin \phi)$ .

The capped yield surface is illustrated in Fig. 2. The tension cap is fixed and is used to regularise the plastic flow behaviour in the corner region. The compression surface cap, on the other hand, is dynamic, and the maximum hydrostatic pressure,  $\kappa$ , is used as the main variable.

The conventional Drucker–Prager model is a common model for the plastic deformation dynamics of soils, e.g. wet or dry sand. These materials are weak under tensile stress ( $I_1 > 0$ ) and become stronger under compression ( $I_1 < 0$ ) where it may support large shear stresses ( $\sqrt{J_2} \neq 0$ ). The capped Drucker–Prager model is a generalisation that includes plastic compaction that occurs in many soils. The compaction mechanism is the failure of individual grains whereby air or water is displaced from the soil. The compaction saturates at a maximum level, where all voids vanish. The compaction dynamics is modelled by a variable cap on compressive side of the Drucker–Prager yield surface, intersecting the hydrostatic axis at  $-\kappa$ . The chosen compaction hardening law is [18]

**Table 1** Elastoplastic model parameters

$E, \nu$	Young's modulus and Poisson's ratio
$\phi, c$	Friction angle and cohesion
$W, D$	Max compaction and hardening rate
$T, \kappa$	Cap $I_1$ -intersections

$$\kappa = \kappa_0 + \frac{1}{D} \ln \left( 1 + \frac{\text{tr}(\epsilon^p)}{W} \right), \tag{19}$$

where  $W$  is the maximum volume compaction,  $D$  is the hardening rate, and  $\kappa_0$  is the initial cap position where compressive failure first occurs. Observe that when the plastic volume compaction  $-\text{tr}(\epsilon^p)$  approaches  $W$ , the cap variable  $\kappa$  goes to infinity. In this regime, the material no longer compact plastically and behaves like the standard Drucker–Prager model. The elastoplastic material parameters are summarised in Table 1. Expressions for the detailed shape of the compression cap and the derivatives of the yield functions are found in ‘‘Appendix’’.

### 3.2 Meshfree method

The continuous solid of mass  $m$  and reference volume  $V$  is discretised into  $N_p$  particles. The particles have mass  $m^i = m/N_p$ , volume  $V^i = V/N_p$ , position  $\mathbf{q}^i$ , and velocity  $\dot{\mathbf{q}}^i$ . The particle displacement is  $\mathbf{u}^i = \mathbf{q}^i - \mathbf{x}^i$  with reference position  $\mathbf{x}^i$  as illustrated in Fig. 1. A continuous and differentiable displacement field is approximated using the MLS method [19,33]. The displacement gradient field defines the strain tensor field in any point by Eq. (12). The particles can thus be understood as pseudo-particles having both particle and field properties.

The MLS approximation of the displacement field is

$$u_\alpha(\mathbf{x}) = \sum_j^{N_p} \Psi_j(\mathbf{x}) u_\alpha^j, \tag{20}$$

where the shape function and moment matrix use a quadratic basis

$$\begin{aligned} \Psi_j(\mathbf{x}) &= p_\gamma(\mathbf{x}) A_{\gamma\tau}^{-1}(\mathbf{x}) p_\tau(\mathbf{x}^j) W(\mathbf{x} - \mathbf{x}^j), \\ A_{\gamma\tau}(\mathbf{x}) &= \sum_j^{N_p} W(\mathbf{x} - \mathbf{x}^j) p_\gamma^j p_\tau^j, \\ \mathbf{p}(\mathbf{x}) &= \left[ 1, x, y, z, yz, xz, xy, x^2, y^2, z^2 \right]^T, \end{aligned} \tag{21}$$

and weight function  $W(\mathbf{x}) = \frac{315}{64\pi l^9} (l^2 - \mathbf{x}^T \mathbf{x})^3$  if  $|\mathbf{x}|$  is smaller than the influence radius  $l$ , and otherwise zero. The base function notation  $\mathbf{p}^j = \mathbf{p}(\mathbf{x}^j)$  is used to simplify the expressions. The gradient of the interpolated displacement field is

$$F_{\alpha\beta}(\mathbf{x}) \equiv \nabla_\beta u_\alpha(\mathbf{x}) = \sum_j^{N_p} \nabla_\beta \Psi_j(\mathbf{x}) u_\alpha^j, \tag{22}$$

$$\begin{aligned} \nabla_\beta \Psi_j &= \nabla_\beta p_\gamma A_{\gamma\tau}^{-1} p_\tau^j W^j + p_\gamma \nabla_\beta A_{\gamma\tau}^{-1} p_\tau^j W^j \\ &\quad + p_\gamma A_{\gamma\tau}^{-1} p_\tau^j \nabla_\beta W^j, \end{aligned} \tag{23}$$

with  $W^j = W(\mathbf{x} - \mathbf{x}^j)$ ,  $\nabla_\beta W^j = \nabla_{x_\beta} W(\mathbf{x} - \mathbf{x}^j)$ . The strain tensor field,  $\epsilon(\mathbf{x})$ , can thus be approximated by applying Eqs. (12)–(22). Observe that  $u_\alpha^j$  depends on current particle positions  $\mathbf{q}$  while  $\nabla_\beta \Psi_j(\mathbf{x})$  depends only on the reference positions.

### 3.3 Compliant deformation constraint

The strain energy density,  $U(\mathbf{x}) = \frac{1}{2} \epsilon^T \mathbf{C} \epsilon$ , is transformed into a compliant constraint via a Legendre transform as presented in Sect.2. For each particle  $i$ , a deformation constraint is imposed

$$0 = \mathbf{g}^i \equiv \begin{bmatrix} \epsilon_{xx}^i \\ \epsilon_{yy}^i \\ \epsilon_{zz}^i \\ \epsilon_{yz}^i + \epsilon_{zy}^i \\ \epsilon_{zx}^i + \epsilon_{xz}^i \\ \epsilon_{xy}^i + \epsilon_{yx}^i \end{bmatrix}, \tag{24}$$

with particle strain components  $\epsilon_{\alpha\beta}^i = \frac{1}{2} (F_{\alpha\beta}^i + F_{\beta\alpha}^i + F_{\gamma\alpha}^i F_{\gamma\beta}^i)$  based on the displacement gradient  $F_{\alpha\beta}^i(\mathbf{q}) = \nabla_\beta u_\alpha(\mathbf{x}^i, \mathbf{q})$ , computed by evaluating the field at  $\mathbf{x} = \mathbf{x}^i$ . Note that the particle strain depends on the displacement of all particles within the domain of influence with radius  $l$ . The compliance parameter becomes  $\mathbf{e}^i = (V^i \mathbf{C}^i)^{-1}$ , which depends on the material parameters, through Eq. (13), and on the spatial resolution through the particle volume factor  $V^i$  that appears when integrating from energy density to particle energy. As expected, softer materials will have larger compliance parameter and thus deform more easily. The Jacobian of the deformation constraint,  $\mathbf{G} = \partial \mathbf{g} / \partial \mathbf{q}$ , can be expanded through the chain rule into

$$G_{\alpha\beta}^i(\mathbf{q}) = \frac{\partial g_\alpha^i}{\partial \epsilon_{\gamma\eta}^i} \frac{\partial \epsilon_{\gamma\eta}^i}{\partial F_{\tau\kappa}^i} \frac{\partial F_{\tau\kappa}^i}{\partial q_\beta^j}. \tag{25}$$

The Jacobian for constraint  $i$  has block structure  $\mathbf{G}^i = [\mathbf{G}^{i1}, \dots, \mathbf{G}^{ij}, \dots, \mathbf{G}^{iN_p}]$ , where each block element  $\mathbf{G}^{ij}$  has dimension  $6 \times 3$ . For notational convenience, the constraint vector is split in two parts, one for the diagonal terms of the strain tensor,  $\mathbf{g}_d^i = [\epsilon_{xx}^i, \epsilon_{yy}^i, \epsilon_{zz}^i]^T$  and one for the off-diagonal terms  $\mathbf{g}_{od}^i = [\epsilon_{yz}^i + \epsilon_{zy}^i, \epsilon_{zx}^i + \epsilon_{xz}^i, \epsilon_{xy}^i + \epsilon_{yx}^i]^T$ . The Jacobian blocks  $\mathbf{G}^{ij}$  are split correspondingly into two  $3 \times 3$  Jacobian blocks,  $\mathbf{G}^{ij} = [\mathbf{G}_d^{ijT}, \mathbf{G}_{od}^{ijT}]^T$ . After some lengthy algebra, the following expressions for the Jacobians are found

$$\mathbf{G}_d^{ij} = \begin{bmatrix} (1 + F_{xx}^i) \Lambda_x^{ij} & F_{yx}^i \Lambda_x^{ij} & F_{zx}^i \Lambda_x^{ij} \\ F_{xy}^i \Lambda_y^{ij} & (1 + F_{yy}^i) \Lambda_y^{ij} & F_{zy}^i \Lambda_y^{ij} \\ F_{xz}^i \Lambda_z^{ij} & F_{yz}^i \Lambda_z^{ij} & (1 + F_{zz}^i) \Lambda_z^{ij} \end{bmatrix}, \tag{26}$$

$$\mathbf{G}_{od}^{ij} = \begin{bmatrix} F_{xy}^i \Lambda_z^{ij} + F_{xz}^i \Lambda_y^{ij} & (1 + F_{yy}^i) \Lambda_z^{ij} + F_{yz}^i \Lambda_y^{ij} & F_{zy}^i \Lambda_z^{ij} + (1 + F_{zz}^i) \Lambda_y^{ij} \\ F_{xz}^i \Lambda_x^{ij} + (1 + F_{xx}^i) \Lambda_z^{ij} & F_{yz}^i \Lambda_x^{ij} + F_{yx}^i \Lambda_z^{ij} & (1 + F_{zz}^i) \Lambda_x^j + F_{zx}^i \Lambda_z^j \\ (1 + F_{xx}^i) \Lambda_y^j + F_{xy}^i \Lambda_x^{ij} & F_{yx}^i \Lambda_y^j + (1 + F_{yy}^i) \Lambda_x^{ij} & F_{zx}^i \Lambda_y^j + F_{zy}^i \Lambda_x^{ij} \end{bmatrix}, \tag{27}$$

where  $\Lambda_\beta^{ij} = \nabla_\beta \Psi_j(\mathbf{x}^i)$ . Observe that only  $F_{\alpha\beta}^i(\mathbf{q})$  depends on the current particle positions while  $\Lambda_\beta^{ij}$  depends only on the reference positions and may thus be pre-computed. In the derivation of the Jacobians, the following partial derivatives are used

$$\frac{\partial \epsilon_{\gamma\eta}^i}{\partial F_{\tau\kappa}^i} = \frac{1}{2} (\delta_{\gamma\tau}^{\eta\kappa} + \delta_{\eta\tau}^{\gamma\kappa} + \delta_{\kappa\gamma} F_{\tau\eta}^i + \delta_{\eta\kappa} F_{\tau\gamma}^i),$$

$$\frac{\partial F_{\tau\kappa}^i}{\partial q_\beta^j} = \delta_{\tau\beta} \nabla_\kappa \Psi_j(\mathbf{x}^i),$$

with Kronecker notation  $\delta_{\gamma\tau}^{\eta\kappa} \equiv \delta_{\gamma\tau} \delta_{\eta\kappa}$ .

### 3.3.1 Stabilisation constraint

The MLS approximation has the deficiency of underestimating the strain for deformation modes on short length scales. This makes the method unstable, manifested by spurious short wavelength deformation modes. Following [20,34], we eliminate this instability and associated discretisation error by adding the following deformation energy

$$\tilde{U}^i = \frac{\alpha l^2 V^i}{E} [(\nabla \cdot \boldsymbol{\sigma})_{\mathbf{x}=\mathbf{x}^i}]^2, \tag{28}$$

where  $\alpha$  is a dimensionless stabilisation parameter and  $\mathbf{f}^i$  are the forces on the pseudo-particle other than elastic ones. The corresponding stabilisation constraint becomes  $\tilde{\mathbf{g}}^i = (l/E) [(\nabla \cdot \boldsymbol{\sigma})_{\mathbf{x}=\mathbf{x}^i}]$  with regularisation  $\tilde{\epsilon}^i = (\alpha V^i E)^{-1}$  and after a suitable normalisation. The stress divergence is computed using a MLS approximation of the stress tensor field  $\boldsymbol{\sigma}(\mathbf{x}) = \sum_j^{N_p} \Psi^j(\mathbf{x}) \mathbf{C}(\boldsymbol{\epsilon}^j - \boldsymbol{\epsilon}_p^j)$ . The stabilisation constraint becomes

$$\tilde{g}_\alpha^i = \frac{l}{E} \left[ \sum_j^{N_p} (\hat{D}_{\alpha\beta} \Psi^j)_{\mathbf{x}=\mathbf{x}^i} C_{\beta\tau} (\epsilon_\tau^j - \epsilon_{p\tau}^j) \right], \tag{29}$$

where the stress and strain are represented using Voigt notation such that the divergence operator is

$$\hat{\mathbf{D}} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ 0 & 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{bmatrix}. \tag{30}$$

The Jacobian of the stabilisation constraint becomes

$$\tilde{G}_{\alpha\sigma}^{ik} \equiv \frac{\partial \tilde{g}_\alpha^i}{\partial q_\sigma^k} = \frac{l}{E} \sum_j^{N_p} (\hat{D}_{\alpha\beta} \Psi^j)_{\mathbf{x}=\mathbf{x}^i} C_{\beta\tau} G_{\tau\sigma}^{jk}, \tag{31}$$

where  $G_{\tau\sigma}^{jk} = \partial \epsilon_\tau^j / \partial q_\sigma^k$  is already derived for the compliant constraint.

### 3.4 Contacts

Contacts between elastoplastic solids and rigid or static bodies are modelled by assigning a rigid spherical shape of radius  $l$  to each pseudo-particle and imposing the Signorini–Coulomb law for unilateral dry frictional contacts [5], as described in Sect. 2.3 and in greater detail in [8]. A contact situation between a solid and three rigid bodies is illustrated in Fig. 3. The contact overlap and normal are indicated as well as the contact point, which is defined at half distance between the overlapping geometries. Contacts between neighbouring pseudo-particles are ignored. Using spherical shapes to approximate the contact boundary of the solid has obvious drawbacks, e.g. perfectly smooth surfaces cannot be represented. This should be considered as an intermediate solution only, practical to implement in prototype code. More sophisticated representations of boundaries and contact detection algorithms for meshfree methods exist [35,36] and should replace the use of spherical shapes.

## 4 Simulation

The implementation details and results from nonsmooth multidomain dynamic simulation including elastoplastic solids is presented in this section. The main simulation algorithm is given in Sect. 4.1. The elastoplastic solid model is implemented in C++ making use of the simulation software library

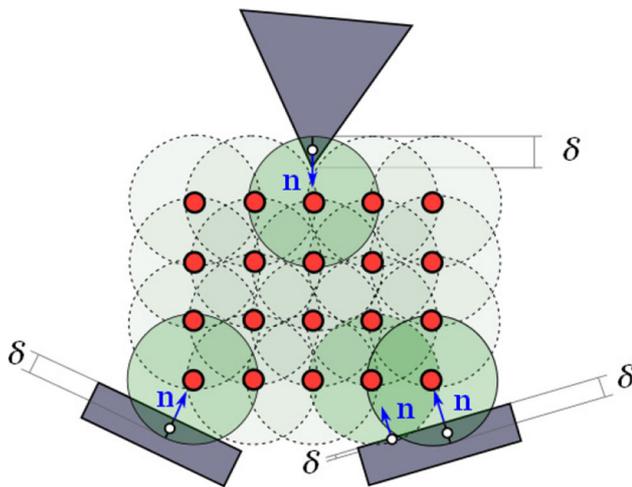


Fig. 3 Illustration of contact handling

AGX Dynamics [37] for data types, collision detection algorithms, and MLCP solver framework, covered in Sect. 4.2. The elastic and plastic model and implementation is verified by performing load–displacement simulation tests presented in Sect. 4.3. Finally, in Sect. 4.4, the applicability of the approach is demonstrated by multidomain examples including a simple terrain vehicle with tracked bogies driving over a deformable terrain and a simulated cone penetrometer test in soil with embedded rock.

### 4.1 Main algorithm

Algorithm 1 lists the main steps in running a multidomain dynamics simulation including both elastoplastic solids and contacting rigid multibodies.

---

#### Algorithm 1 Main simulation algorithm

---

- 1: initialisation of bodies and constraints
  - 2: compute MLCP blocks  $\mathbf{M}$ ,  $\Sigma$ ,  $\mathbf{Y}$
  - 3: compute MLS blocks  $\mathbf{A}_i^{-1}$ ,  $\mathbf{A}_i$
  - 4: **for all** timestep  $n = 1, 2, \dots$  **do**
  - 5:   get input signals and explicit forces
  - 6:   do contact detection
  - 7:   **for all** particles  $i = 1, 2, \dots, N_p$  **do**
  - 8:     compute  $\mathbf{u}_n^i$  and  $\boldsymbol{\epsilon}_n^i = \mathbf{J}_n^i \mathbf{T} \mathbf{J}_n^i - \mathbf{1}$
  - 9:      $[\boldsymbol{\epsilon}_n^{e,i}, \boldsymbol{\epsilon}_n^{p,i}] = \text{radial\_return}(\boldsymbol{\epsilon}_n^i, \boldsymbol{\epsilon}_{n-1}^{p,i})$
  - 10:   **end for**
  - 11:   compute constraint data  $\mathbf{G}$ ,  $\mathbf{g}$
  - 12:   build MLCP data  $\mathbf{H}$ ,  $\mathbf{r}$ ,  $\mathbf{l}$ ,  $\mathbf{u}$
  - 13:   solve  $\mathbf{z} = \text{MLCP}(\mathbf{H}, \mathbf{r}, \mathbf{l}, \mathbf{u})$
  - 14:   get  $\dot{\mathbf{q}}_{n+1}$  and constraint force  $\mathbf{G}^T \boldsymbol{\lambda}$
  - 15:   update position  $\mathbf{q}_{n+1} = \mathbf{q}_n + h \dot{\mathbf{q}}_{n+1}$
  - 16:   store and visualise the new state
  - 17: **end for**
  - 18: post-process and visualise stored data
- 

The elastoplastic solid is initialised by defining a solid volume and assigning material parameters. The solid is discretised into  $N_p$  pseudo-particles according to a given spatial resolution. For simplicity, the particles are positioned in a regular cubic pattern. This defines the reference state with vanishing strain and stress. The elastoplastic constraints are initialised and connectivity data listing the particles involved in each constraint is created. Similarly, rigid bodies and kinematic constraints are defined, for instance, to form an articulated vehicle and powertrain. Each body is assigned a geometric 3D shape. Contact material properties are assigned to each body, including coefficient of restitution, elasticity, and friction. These are used to generate contact constraint data included in the MLCP when triggered by contact detection during simulation. Certain quantities are computed once before time integration and then reused for the sake of optimisation, e.g. the inverse moment matrix  $\mathbf{A}^{-1}$  in the MLS approximation,  $\nabla_\beta \Phi$  and  $\boldsymbol{\Lambda}$ .

---

#### Algorithm 2 Plastic radial return algorithm

---

- 1: input total strain  $\boldsymbol{\epsilon}_n$  and stored plastic  $\boldsymbol{\epsilon}_{n-1}^p$
  - 2: trial plastic strain  $\boldsymbol{\epsilon}_n^p = \boldsymbol{\epsilon}_{n-1}^p$
  - 3: compute trial stress  $\boldsymbol{\sigma}_n = \mathbf{C}(\boldsymbol{\epsilon}_n - \boldsymbol{\epsilon}_n^p)$
  - 4: **if** yielding  $\Phi(\boldsymbol{\sigma}_n) > 0$  **then**
  - 5:   return to surface
  - 6:   **while**  $|\Phi(\boldsymbol{\sigma}_n)| > \epsilon_{\text{tol}}$  **do**
  - 7:      $\Delta \lambda = \frac{\Phi(\boldsymbol{\sigma}_n)}{\frac{\partial \Phi}{\partial \boldsymbol{\sigma}} \mathbf{C} \frac{\partial \Psi}{\partial \boldsymbol{\sigma}}}$
  - 8:      $\Delta \boldsymbol{\epsilon}^p = \Delta \lambda \frac{\partial \Psi}{\partial \boldsymbol{\sigma}}$
  - 9:      $\Delta \boldsymbol{\sigma} = \Delta \lambda \mathbf{C} \frac{\partial \Psi}{\partial \boldsymbol{\sigma}}$
  - 10:      $\boldsymbol{\epsilon}_n^p = \boldsymbol{\epsilon}_n^p + \Delta \boldsymbol{\epsilon}^p$
  - 11:      $\boldsymbol{\sigma}_n = \boldsymbol{\sigma}_n - \Delta \boldsymbol{\sigma}$
  - 12:   **end while**
  - 13:   update cap variables  $\kappa_n(\boldsymbol{\epsilon}^p)$  and  $I_{1n}^c(\boldsymbol{\sigma}_n)$
  - 14: **end if**
  - 15: compute elastic strain  $\boldsymbol{\epsilon}_n^e = \mathbf{C}^{-1} \boldsymbol{\sigma}_n$
- 

The simulation is run with a fixed timestep, following the stepper in Sect. 2.2. Each timestep begins with reading input signals, e.g. from operator and control system, and computation of explicit forces. Next, contact detection algorithms produce a set of contacts for intersecting geometries. Each contact position and velocity, penetration depth, normal and tangent is stored. Contacts are classified as either impacting, continuous or separating, depending on the sign of the relative contact velocity. The strain and stress fields are computed as described in Sect. 3.1 using the MLS approximation of the displacement field for the spatial discretisation described in Sect. 3.2. The trial stress is tested against one of the three yield surfaces decided by Algorithm 3. If it is outside the elastic domain, the radial return Algorithm 2 computes the plastic deformation that returns the stress to the yield surface with an error threshold  $\epsilon_{\text{tol}}$ . The compaction variable  $\kappa$  is also updated by this. Constraint violation and Jaco-

**Algorithm 3** Deciding yield function

- 1: Drucker–Prager tangent  $\mathbf{t}_{DP} = [I_1^C, \sqrt{J_2^C}] - [I_1^T, \sqrt{J_2^T}]$
- 2: input stress  $\boldsymbol{\sigma}$
- 3: compute stress invariant vector  $\mathbf{v} = [I_1, \sqrt{J_2}]$
- 4: compression cap intersection  $\mathbf{v}_C = [I_1 - I_1^C, \sqrt{J_2}]$
- 5: tension cap intersection  $\mathbf{v}_T = [I_1 - I_1^T, \sqrt{J_2}]$
- 6: **if**  $\mathbf{v}_C \cdot \mathbf{t}_{DP} > 0$  **then**
- 7:   compression cap region
- 8: **else if**  $\mathbf{v}_T \cdot \mathbf{t}_{DP} < 0$  **then**
- 9:   tension cap region
- 10: **else**
- 11:   Drucker–Prager region
- 12: **end if**

bians are computed from elastic strain, contact data, and state of the multibodies. The matrix blocks and limits for the MLCP are computed and fed to the MLCP solver outlined in Sect. 4.2. The main solve is preceded with an *impact solve stage*, in which impacting contacts are solved using the Newton impulse law  $\mathbf{G}_n \dot{\mathbf{q}}^+ = -e \mathbf{G}_n \dot{\mathbf{q}}^-$ , with coefficient of restitution  $e$ , while satisfying all other constraint velocity conditions,  $\mathbf{G} \dot{\mathbf{q}}^+ = 0$ , and complementarity conditions. The main MLCP solve determines the new velocities and constraint forces. After this the positions and orientations are integrated. At the end of each timestep, data of the new state are stored for post-processing and visualisation.

**4.2 Overview of the MLCP solver**

The main computational task for each timestep is solving the MLCP in Eq. (10) with saddle point matrix structure given in Eq. (9). Each sub-matrix is block-sparse and the routines for building, storing, and solving are tailored for this and exploit BLAS3 operations as much as possible to gain computational speed. In order to achieve high performance, e.g. real-time simulation of complex multidomain systems, splitting is applied [38]. Each subproblem can then be treated with different MLCP solvers that best fit the requirements of accuracy, stability, and scalability. For the elastoplastic solid and for jointed rigid multibodies, with relatively few complementarity conditions, a direct MLCP solver is used that is described in Sect. 4.2.2.

*4.2.1 Split solve*

Assume three sets of constraints, labelled  $A$ ,  $B$ , and  $AB$  for two subsystem and their coupling. The linear system (11) is split into two subproblems by duplicating the variables  $\dot{\mathbf{q}}$  and  $\boldsymbol{\lambda}_{AB}$  and reordering the augmented system. A stationary iterative update procedure can then be formed based on the four matrix blocks

$$\mathbf{H}_{AB} \mathbf{z}_{AB}^{k+1} = -\mathbf{r}_{AB} + \mathbf{G}_B^T \boldsymbol{\lambda}_B^k, \tag{32}$$

$$\mathbf{H}_{BA} \mathbf{z}_{BA}^{k+1} = -\mathbf{r}_{BA} + \mathbf{G}_{BA}^T \boldsymbol{\lambda}_A^{k+1}, \tag{33}$$

with

$$\mathbf{H}_{AB} = \begin{bmatrix} \mathbf{M} & -\mathbf{G}_A^T & -\mathbf{G}_{AB}^T \\ \mathbf{G}_A^T & \boldsymbol{\Sigma}_A & 0 \\ \mathbf{G}_{AB}^T & 0 & \boldsymbol{\Sigma}_{AB} \end{bmatrix}, \tag{34}$$

$$\mathbf{z}_{AB} = \begin{bmatrix} \dot{\mathbf{q}} \\ \boldsymbol{\lambda}_A \\ \boldsymbol{\lambda}_{AB} \end{bmatrix} \quad \mathbf{r}_{AB} = \begin{bmatrix} \dot{\mathbf{p}} \\ \mathbf{v}_A \\ \mathbf{v}_{AB} \end{bmatrix} \tag{35}$$

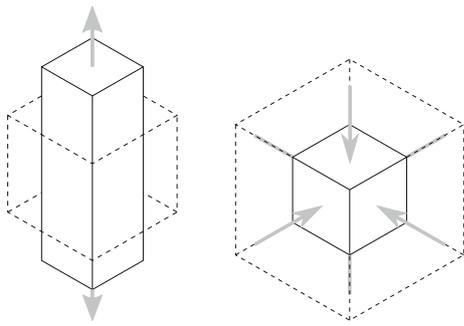
and the same for  $\mathbf{H}_{BA}$ ,  $\mathbf{z}_{BA}$ , and  $\mathbf{r}_{BA}$  with  $A$  and  $B$  permuted. The iterative procedure converges if the spectral radius of the augmented system  $\mathbf{H}$  fulfils  $\rho([\mathbf{D} + \mathbf{L}]^{-1} \mathbf{U}) < 1$ . The key point is that each subproblem can be approached using different solvers. In the case of a machine interacting with an elastoplastic terrain, the machine and solid terrain constraints,  $A$ , are split from the tangential contact forces related to dry friction,  $B$ , both systems sharing the normal contact constraints (nonpenetration),  $AB$ . The subsystem (32), with the solid, machine, and contact normals is solved first using a direct solver, while the subsystem (33), containing friction constraints, is solved using an iterative projected Gauss–Seidel solver. The split solve may be terminated after the first iteration, accepting potential errors from the friction forces, or continued with a final stage 1 solve. Experiments show that further iterations do not necessarily reduce the errors. For large contact systems ( $>1000$  contacts), with many complementarity conditions, both normal and friction constraints may be moved to an iterative projected Gauss–Seidel solver [8].

*4.2.2 Direct solver*

The direct solver is a block pivot method [38] for MLCPs based on Newton–Rhapson iterations applied to nonsmooth formulation. The detailed algorithm is an adaptation of Murty’s principal pivoting method, and is found in the reference [39] as Algorithm 21.2. Before the solve step, the matrix  $\mathbf{H}$  is factorised by a permuted LDL-factorisation,  $\mathbf{H} = \mathbf{P} \mathbf{L} \mathbf{D} \mathbf{L}^T \mathbf{P}^T$ , where the permutation  $\mathbf{P}$  is used to minimise fill-ins and may include leaf-swapping to avoid pivoting on  $\boldsymbol{\Sigma}$  since it is often close to zero. The factorisation requires symmetric indefinite matrices, but this is not the case with  $\mathbf{H}$  initially as shown in Eq. (11). This is fixed by absorbing a negative sign in the vector  $\mathbf{z}$ .

**4.3 Verification tests**

The rotational invariance of the elastic solid model is verified in simulations confirming that the strain invariants are unaffected by rigid rotations of the body.



**Fig. 4** Illustration of the verification tests by uniaxial stretch (left) and hydrostatic compression (right) of an elastic cube

**Table 2** Elastic verification parameters

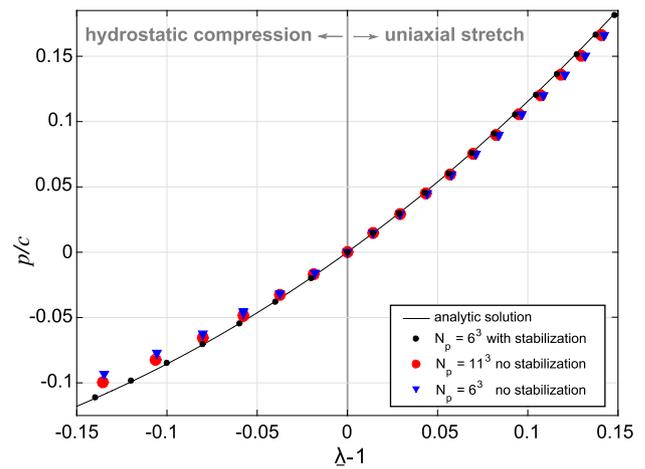
Parameter	Value
Side length	1 m
$N_p$	$(6^3, 9^3, 11^3)$
Resolution	$1/N_p^3$ m
Influence radius	$2 \times$ resolution
Mass density	$2000 \text{ kg/m}^3$
Young's modulus $E$	$(10^6, 10^8) \text{ Pa}$
Poisson ratio $\nu$	$(0.1, 0.25, 0.49)$
Timestep $h$	$10^{-3} \text{ s}$
Stabilisation $\alpha$	$(0, 100)$

### 4.3.1 Elasticity

The validity of the model is confirmed by a number of tests of macroscopic relations between an applied load and the resulting displacement. Uniaxial stretching ( $u$ ) and hydrostatic compression ( $c$ ) are investigated, as described in Fig. 4.

In the uniaxial test, the boundary condition in the tensile direction is a plane constraint with free slip in plane. The other boundaries are free. In the hydrostatic compression test, the boundary conditions are dynamically generated unilateral contact constraints. The boundary force is increased linearly with time, by regulating the position of the boundary geometry, while measuring the stress and strain components of Eqs. (36) and (37). All tests are simulated on a uniformly discretised homogeneous isotropic three-dimensional solid with parameters given in Table 2.

The simulation result is compared with exact solutions for from elasticity theory for uniform deformations of St. Venant–Kirchhoff materials. A test cube with initial rest length  $l_0$  and cross-sectional area  $A_0$  is considered. The deformed side length is denoted  $l$ , and the stretch ratio in that direction is  $\underline{\lambda} = l/l_0$ . The following exact relations can be derived



**Fig. 5** Verification test of elastic response to hydrostatic compression ( $\underline{\lambda} < 1$ ) and uniaxial stretch ( $\underline{\lambda} > 1$ ) with  $E = 10^6 \text{ Pa}$  and  $\nu = 0.25$ . The analytical solution is represented with a solid curve. Simulations are made with ( $\alpha = 100$ ) and without ( $\alpha = 0$ ) stabilisation constraint and for different discretisations

$$p_u = \frac{1}{2} c_u \underline{\lambda}_u (\underline{\lambda}_u^2 - 1), \tag{36}$$

$$p_c = \frac{1}{2} c_c \underline{\lambda}_c (\underline{\lambda}_c^2 - 1), \tag{37}$$

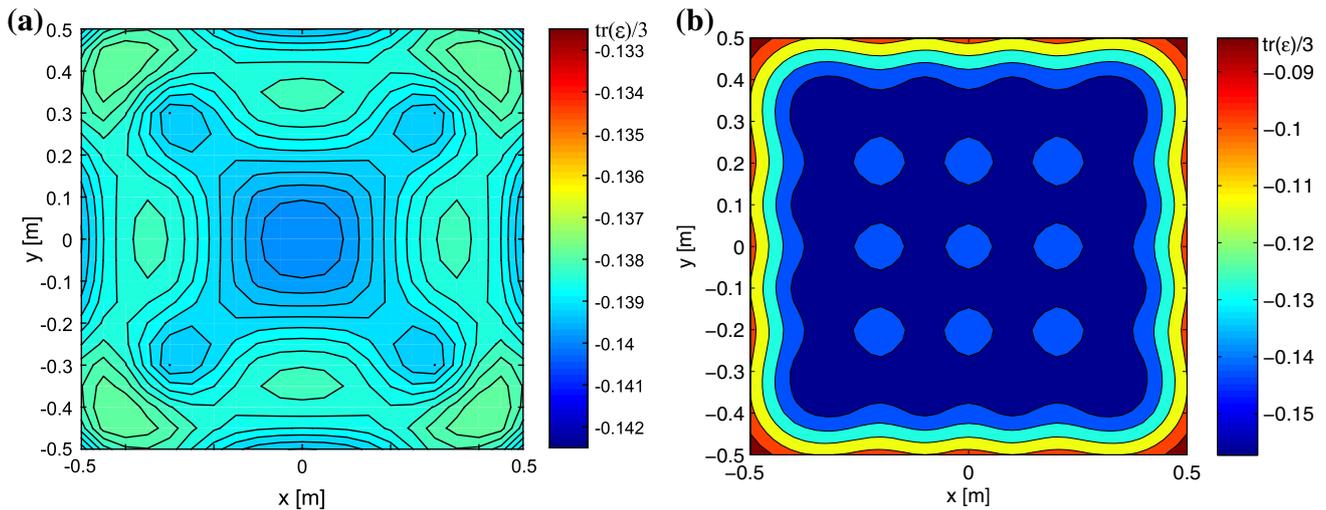
where  $p = F/A_0$  is the applied boundary pressure, and  $c_u = E$  and  $c_c = \frac{E}{(1-2\nu)}$  are the Young's and bulk modulus. The simulation results for  $E = 10^6 \text{ Pa}$  and  $\nu = 0.25$  are presented in Fig. 5. The influence radius was chosen experimentally to be as low as possible, reducing the number of neighbours and the simulation time, while maintaining physically accurate results. The chosen influence radius results in 27 neighbours for any evaluation point that is at least half an influence radius away from the boundary.

The simulation result with stabilisation constraint ( $\alpha = 100$ ) agrees very well with the exact solution for large deformations. Without the stabilisation constraint, the simulated stress–strain relation deviates by 5–10% from the theory at large deformations  $|\underline{\lambda} - 1| \lesssim 0.15$ . The error decrease with finer resolution. Observe that for infinitesimal deformations  $\frac{1}{2} \underline{\lambda} (\underline{\lambda}^2 - 1) \approx \Delta l/l_0$ , Eqs. (36) and (37) approximate to the well-known expressions from Hooke's law. Increasing the stiffness or resolution further requires a smaller timestep for numerical stability.

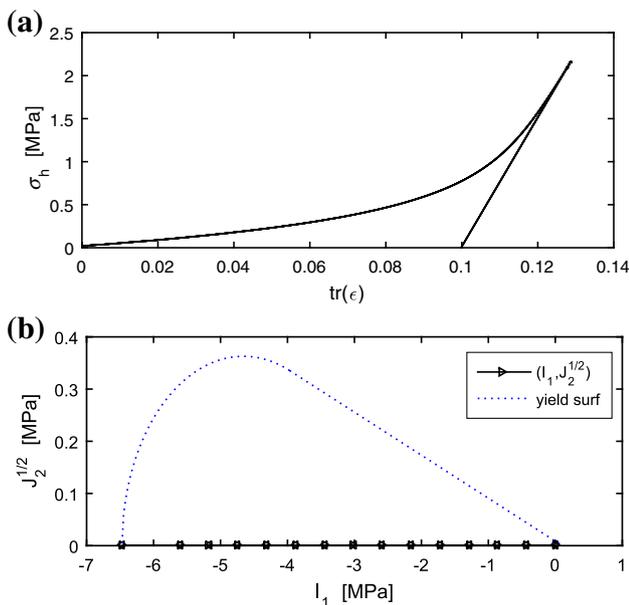
The strain field under hydrostatic compression is displayed in Fig. 6. With stabilisation constraint, the strain field is nearly uniform through the material. Without the stabilisation constraint, the strain deviates by as much as 40% with the largest deviations at the boundary.

### 4.3.2 Plasticity

The Drucker–Prager cap model is tested under hydrostatic and triaxial loading and unloading, following the tests made



**Fig. 6** Cross-sectional  $\text{tr}(\epsilon)$  through the elastic test cube with  $N_p = 6^3$  under hydrostatic compression to  $\lambda = 0.850$  for which  $\text{tr}(\epsilon) = 0.139$ . **a** Cross-sectional  $\text{tr}(\epsilon)$  with stabilisation constraint. **b** Cross-sectional  $\text{tr}(\epsilon)$  with no stabilisation constraint.



**Fig. 7** Stress evolution during hydrostatic compression and unloading of an elastoplastic solid cube. **a** Stress and strain. **b** Mean stress invariants and yield surface

in [18] with  $E = 91 \text{ MPa}$ ,  $\nu = 0.3$ ,  $c = 14 \text{ kPa}$ ,  $\phi = 22^\circ$ ,  $W = 0.1$ ,  $D = 1 \text{ mm}^2/\text{N}$ . Stabilisation constraint is used with  $\alpha = 100$ . The hydrostatic test examines the relation between the pressure,  $\sigma_h = I_1/3$ , and the volumetric strain,  $\text{tr}(\epsilon)$ , while undergoing plastic deformation on the compression cap. The simulation result is presented in Fig. 7. The result is a permanent deformation of the cube by  $\text{tr} \epsilon = 0.1$  compared to  $\text{tr} \epsilon = 0.12$  in [18]. The evolution of the stress invariants,  $I_1$  and  $J_2$ , in the yield space shows that the deviatoric stress is negligible and stress evolves purely along the hydrostatic axis during loading and unloading.

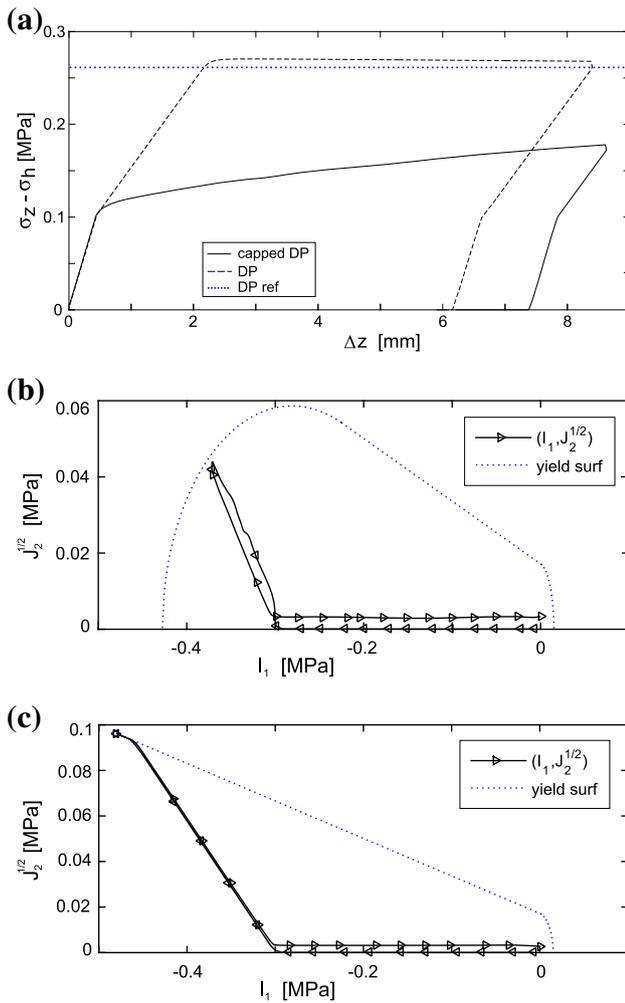
In the triaxial test, a hydrostatic pressure of  $\sigma_h = 100 \text{ kPa}$  is first established. The load along the  $z$ -axis is then increased up to a level where the material yields while the side wall pressure is kept constant. The material is finally unloaded. The simulated evolution of the deviatoric stress as function of strain is displayed in Fig. 8 for a Drucker–Prager material with and without cap model. The simulated Drucker–Prager material is found to yield at a critical stress of  $270 \text{ kPa}$ . This is in good agreement with the analytical prediction from Eq. (18). As expected, the yield plateau is lower and less pronounced in the case of the capped Drucker–Prager material, because the stress reaches the compression cap surface before the Drucker–Prager surface. The triaxial test can be used to determine the value of the plastic hardening parameter  $D$ , whereas the maximum compaction parameter,  $W$ , can be determined from the hydrostatic compression test.

#### 4.3.3 Dynamic contacts

Dynamic contact handling is demonstrated by dropping a beam to rest on two thin cylinders and then deforming it by pressing a larger cylinder down on the beam. The result for an elastic and elastoplastic beam is displayed in Fig. 9.

#### 4.4 Multidomain demonstration

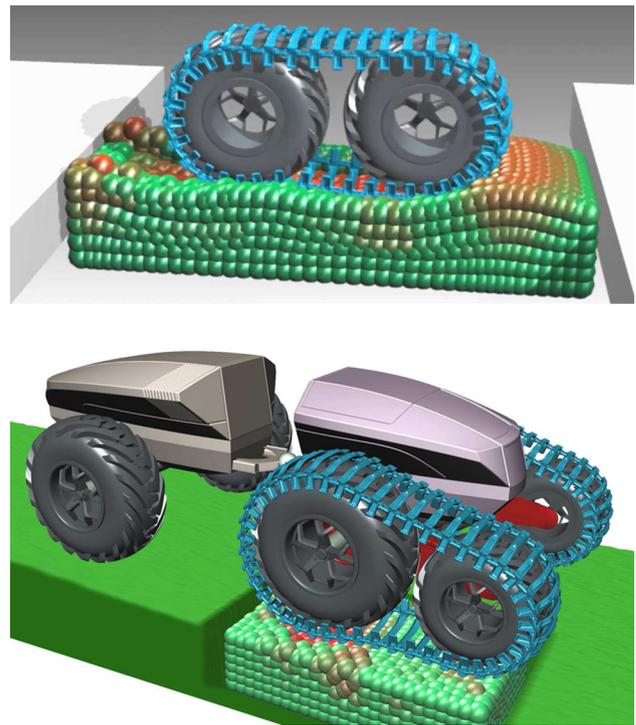
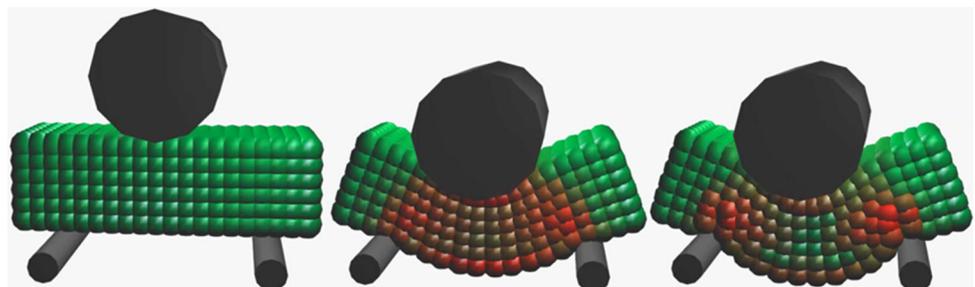
Two example systems are simulated to demonstrate multidomain capability. The first system is an articulated terrain vehicle with tracked bogies driving over a deformable terrain. Images from simulation of a bogie and of a full vehicle are shown in Fig. 10. Videos from simulations are available as supplementary material at <http://umit.cs.umu.se/elastoplastic/>. The vehicle weighs 4 tons and consist of roughly 200 rigid bodies including a front and rear chassis,



**Fig. 8** Triaxial test of an elastoplastic solid cube with capped and conventional Drucker–Prager model. **a** Applied stress over deformation. **b** Stress evolution for the capped Drucker–Prager model. **c** Stress evolution for the conventional Drucker–Prager model

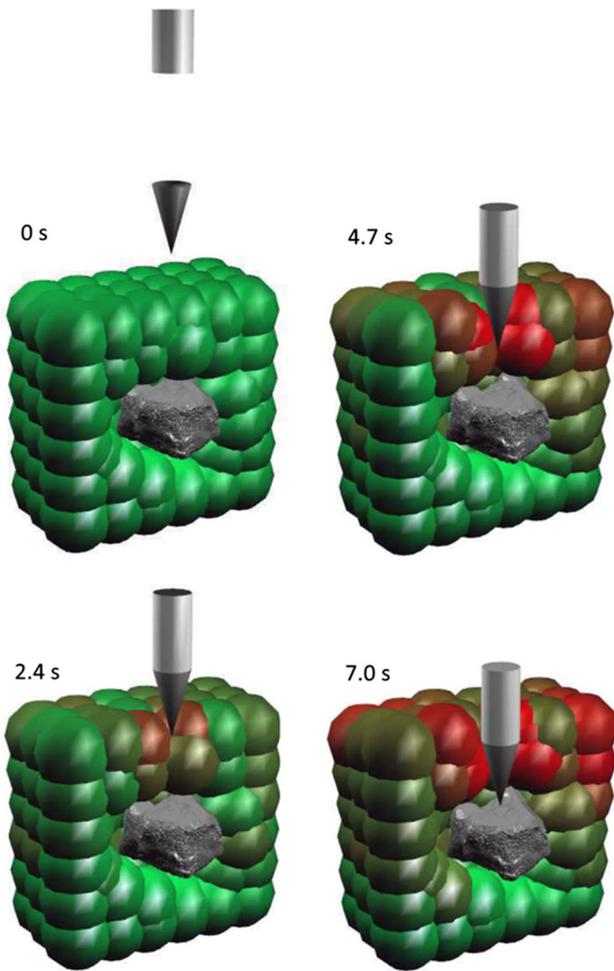
bogie frames, wheels and track elements. These are interconnected with kinematic constraints to model chassis articulation, bogie and wheel axes, and tracks covering the wheeled bogies. The vehicle drivetrain consists of an engine, torque converter, drive shafts, and differentials that distribute the engine power to the front and rear parts and further between left and right bogies to the wheels. The terrain is modelled

**Fig. 9** Deformation of an elastic and elastoplastic beam (middle and right from initial state to the left). The colour codes the Frobenius norm of the strain tensor, ranging from 0 (green) to 0.20 (red)



**Fig. 10** Images from simulation of a tracked bogie (top) and a full terrain vehicle (bottom) passing over a zone with elastoplastic material

as a static trimesh and a rectangular ditch with elastoplastic material. The elastoplastic material parameters are set to  $Y = 1 \text{ MPa}$ ,  $\nu = 0.25$ ,  $\phi = 22^\circ$ ,  $c = 1$ ,  $R_C = 4$ ,  $T = 5 \text{ kPa}$ ,  $\kappa_0 = -0.5 \text{ MPa}$ ,  $D = 10^{-6}$ ,  $W = 0.2$ ,  $\rho = 2000 \text{ kg/m}^3$ , which represents a weak and soft forest terrain. The tracked bogie and vehicle create a rutting with permanent deformation and the rut depth can be measured. In the full vehicle simulation, the solid is discretised into  $N_p = 2100$  pseudo-particles. The coupled system of vehicle and terrain thus have about 20.000 variables. With 10 ms timestep and using a single CPU on a conventional desktop computer, the simulation is of the order 1/100 from real time. It should be emphasised that this is measured on prototype code with no attempt for optimisation and parallelisation. Also, the implementation uses high-fidelity direct solver that delivers higher accuracy than what is typically required in terrain-vehicle simulations.



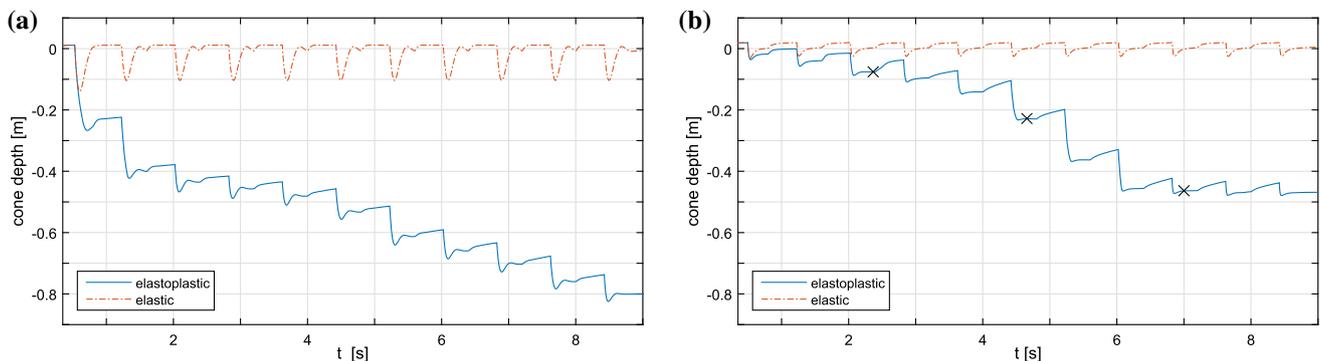
**Fig. 11** Image sequence from a simulated cone penetration test on an elastoplastic terrain with an embedded rock

The second demonstration example is a dynamic cone penetration test, where a cylindrical weight is dropped repeatedly on a cone measuring the penetration depth. This is a common way of measuring the mechanical properties of ter-

rain. The penetration depth of the cone is simulated for two different cases. The first case is a homogeneous material, and the second is material with an embedded rock, represented by a rigid body, as illustrated by Fig. 11. The colours indicate the magnitude of displacement. The simulated cone penetration is presented in Fig. 12. The resistance is higher in the ground with an embedded rock, before the cone actually comes in contact with the rock.

### 5 Discussion

A meshfree elastoplastic solid model is made compatible with nonsmooth multidomain dynamics. The solid appears as a particle system in a constrained multibody system, on the same footing as articulated rigid multibodies and power transmission systems. The pseudo-particles carry field variables, e.g. stress and strain tensor fields, approximated using the moving least squares method. The dynamic interaction between the deformable solid, rigid multibodies, and other geometric boundaries is modelled by unilateral contact constraints with dry friction. The full system can be processed using the same numerical integrator and solver framework without introducing additional coupling equations with unknown parameters. Impulsive behaviour can automatically be transmitted instantly through the full and strongly coupled system. This enables fast and stable simulation of complex mechatronic systems interacting with elastoplastic materials. The MLS approximation underestimates the strain for short wavelength deformation modes. A stabilisation constraint is developed to compensate for this and it improves the accuracy and stability significantly. The explicit form of the deformation constraint and its Jacobian is given in Eqs. (24)–(26), and the stabilisation constraint in Eqs. (29)–(31). The Jacobians can be factored to a constant matrix that can be pre-computed, multiplied with current particle displacements.



**Fig. 12** Simulation measurement of a cone penetrometer on elastic and elastoplastic terrain with and without embedded rock. **a** Homogeneous terrain, **b** terrain with embedded rock. The  $\times$  indicate the time instances in Fig. 11b–d

The verification tests show good agreement with analytical reference solutions for large deformations of elastic and elastoplastic test cubes. The stabilisation constraint is essential for the accuracy and stability. Demonstration is made with a tracked terrain vehicle driving over deformable terrain using the capped Drucker–Prager model and a cone penetrometer test on terrain with and without an embedded rock.

The computational bottleneck of the simulation lies in solving the block-sparse mixed linear complementarity problem (10). With dedicated hardware, parallel factorisation algorithms, and iterative solver for a more approximate solution of the terrain dynamics, the performance can be expected to increase by several orders in magnitude. Exploring this is left for future work. Another area of improvement is to integrate the plastic yield and flow computations with the mixed complementarity problem for the constraint forces and velocity update. This will enable integration with larger timestep for strongly coupled problems than the current predictor–corrector method allow.

The presented method rest on the St. Venant–Kirchhoff elasticity model in combination with the Drucker–Prager plasticity model and the additive decomposition of the elastic and plastic strain. These assumptions are in general only valid under small deformations and alternatives should be considered in future work.

**Acknowledgements** The Project was funded in part by the Kempe Foundation Grant JCK-1109, Umeå University and supported by Algoryx Simulation. The authors thank Prof. Urban Bergsten from the Swedish University of Agriculture, Komatsu Forest and Olofsfors AB, for fruitful discussions and ideas.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

### Appendix: Capped Drucker–Prager yield surface

This supplements Sect. 3 with further details of the Capped Drucker–Prager yield surfaces in Eqs. (17) and (18) illustrated in Fig. 2. We follow the smooth cap model of Dolarevic and Ibrahimbegovic [18] with some corrections to the compression cap. The three yield functions are

$$\Phi_{DP}(I_1, J_2) = \sqrt{J_2} + \frac{\eta}{3} I_1 - \xi c, \tag{38}$$

$$\Phi_T(I_1, J_2) = (I_1 - T + R_T)^2 + J_2 - R_T^2, \tag{39}$$

$$\Phi_C(I_1, J_2, \kappa) = \frac{[I_1 - a(\kappa)]^2}{R_C^2} + J_2 - b(\kappa)^2. \tag{40}$$

The centre, radius, and intersection point of the tension cap are

$$C = \frac{T - \frac{3\xi c}{\eta} \sin(\varphi)}{1 - \sin(\varphi)}, \tag{41}$$

$$R_T = T - C, \tag{42}$$

$$I_t = \frac{3\xi c}{\eta} - \beta \cos(\varphi), \tag{43}$$

where the Drucker–Prager cone angle  $\phi = \arctan(\eta)$  in the pressure–shear plane,  $T$  is the tension cap cut-off, and  $\beta = [(3\xi c/\eta - C)^2 - R_T^2]^{1/2}$ . The compression cap intersection point is

$$I_1^C = \frac{3\xi c}{\eta} - \frac{3b(\kappa)}{\eta \sqrt{1.0 + (\frac{\eta}{3})^2 R_C^2}}, \tag{44}$$

and  $a(\kappa)$  and  $b(\kappa)$  are the centre and main radius of the compression cap ellipse [18]. The stress gradient of the yield functions is

$$\frac{\partial \Phi_{DP}}{\partial \sigma} = \frac{\bar{\sigma}}{2\sqrt{J_2}(\sigma)} + \frac{\eta(\phi)}{3} \mathbf{1}, \tag{45}$$

$$\frac{\partial \Phi_T}{\partial \sigma} = \bar{\sigma} + 2(I_1 - T + R_T) \mathbf{1}, \tag{46}$$

$$\frac{\partial \Phi_C}{\partial \sigma} = \bar{\sigma} + \frac{2(I_1 - a(\kappa))}{R^2} \mathbf{1}. \tag{47}$$

### References

1. Burgermeister B, Arnold M, Eichberger A (2011) Smooth velocity approximation for constrained systems in real-time simulation. *Multibody Syst Dyn* 26(1):1–14
2. Kübler K, Schiehlen W (2000) Two methods of simulator coupling. *Math Comput Model Dyn Syst* 6(2):93–113
3. Acary V, Brogliato B (2008) Numerical methods for nonsmooth dynamical systems: applications in mechanics and electronics. Springer, Berlin
4. Moreau JJ (1999) Numerical aspects of the sweeping process. *Comput Methods Appl Mech Eng* 177:329–349
5. Jean M (1999) The non-smooth contact dynamics method. *Comput Methods Appl Mech Eng* 177(3–4):235–257
6. Cundall PA, Strack ODL (1979) A discrete numerical model for granular assemblies. *Geotechnique* 29:47–65
7. Radjai F, Richefeu V (2009) Contact dynamics as a nonsmooth discrete element method. *Mech Mater* 41(6):715–728
8. Servin M, Wang D, Lacoursière C, Bodin K (2014) Examining the smooth and nonsmooth discrete element approach to granular matter. *Int J Numer Methods Eng* 97:878–902
9. de Souza Neto E A, Peric D, Owen DRJ (2008) Computational methods for plasticity: theory and applications. Wiley, Chichester
10. Liu GR, Gu YT (2010) An introduction to meshfree methods and their programming. Springer Publishing Company, Dordrecht
11. Nguyen VP, Rabczuk T, Bordasc S, Duot M (2008) Meshless methods: a review and computer implementation aspects. *Math Comput Simul* 79:763–813

12. Ullah Z, Augarde CE (2013) Finite deformation elasto-plastic modelling using an adaptive meshless method. *Comput Struct* 118:39–52
13. Wellmann C, Wriggers P (2012) A two-scale model of granular materials. *Comput Methods Appl Mech Eng* 205–208:46–58
14. Rycroft CH, Wong Y, Bazant MZ (2010) Fast spot-based multiscale simulations of granular flow. *Powder Technol* 200:111
15. Servin M, Wang D (2016) Adaptive model reduction for nonsmooth discrete element simulation. *Comput Part Mech* 3(1):107121
16. Lacoursière C (2007) Ghosts and machines: regularized variational methods for interactive simulations of multibodies with dry frictional contacts. PhD thesis, Umeå university
17. Lacoursière C (2007) Regularized, stabilized, variational methods for multibodies. In: Fritzson D, Bunus P, Führer C (eds) *The 48th Scandinavian conference on simulation and modeling (SIMS 2007)*, Göteborg, 30–31 Oct 2007
18. Dolarevic S, Ibrahimbegovic A (2007) A modified three-surface elasto-plastic cap model and its numerical implementation. *Comput Struct* 85(7–8):419–430
19. Belytschko T, Lu YY, Gu L (1994) Element-free Galerkin methods. *Int J Numer Methods Eng* 37(2):229–256
20. Beissel S, Belytschko T (1996) Nodal integration of the element-free Galerkin method. *Comput Methods Appl Mech Eng* 139:49–74
21. Taheri S, Sandu C, Taheri S, Pinto E, Gorsich D (2015) A technical survey on Terramechanics models for tire–terrain interaction used in modeling and simulation of wheeled vehicles. *J Terramech* 57:1–22
22. Azimi A, Holz D, Kövecses J, Angeles J, Teichmann M (2015) A multibody dynamics framework for simulation of rovers on soft terrain. *J Comput Nonlinear Dyn* 10(3):031004
23. Madsen J, Seidl A, Negrut D (2013) Compaction-based deformable terrain model as an interface for real-time vehicle dynamics simulations. SAE World Congress, Detroit
24. Shoop SA (2001) Finite element modeling of tire–terrain interaction. CRREL technical report, ERDC/CRREL TR-01-16, US Army Cold Regions Research and Engineering Laboratory, Hanover
25. Xia K (2013) Finite element modeling of tire/terrain interaction: application to predicting soil compaction and tire mobility. *J Terramech* 48:113–123
26. Shabana A (2005) *Dynamics of multibody systems*. Cambridge University Press, New York
27. Bornemann F, Schütte C (1997) Homogenization of Hamiltonian systems with a strong constraining potential. *Phys D* 102(1–2):57–77
28. Marsden JE, West M (2001) Discrete mechanics and variational integrators. *Acta Numer* 10:357–514
29. Servin M, Lacoursière C (2008) Rigid body cable for virtual environments. *IEEE Trans Vis Comput Graph* 14(4):783–796
30. Servin M, Lacoursière C, Nordfelth F, Bodin K (2011) Hybrid, multiresolution wires with massless frictional contacts. *IEEE Trans Vis Comput Graph* 17(7):970–982
31. Bodin K, Lacoursiere C, Servin M (2012) Constraint fluids. *IEEE Trans Vis Comput Graph* 18(3):516–526
32. Pang JS, Stewart D (2008) Differential variational inequalities. *Math Program* 113(2):345–424
33. Wasfy M, Noor A (2003) Computational strategies for flexible multibody systems. *Appl Mech Rev* 56(6):553–613
34. Miguel J, Onate P (2001) A finite point method for elasticity problems, computers and structures. *Comput Struct* 79:2151–2163
35. Li S, Qian D, Liu W, Belytschko T (2001) A meshfree contact-detection algorithm. *Comput Methods Appl Mech Eng* 190:3271–3292
36. Wang H-P, Wu C-T, Chen J-S (2014) A reproducing kernel smooth contact formulation for metal forming simulations. *Comput Mech* 54:151–169
37. Algoryx Simulations (2016) AGX dynamics. September
38. Cottle R, Pang J-S, Stone R (1992) The linear complementarity problem. *Computer Science and Scientific Computing*
39. Lacoursière C, Linde M (2011) Spook: a variational time-stepping scheme for rigid multibody systems subject to dry frictional contacts. UMINF report 11.09, Umeå University