INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2016; **105**:416–439
Published online 6 August 2015 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nme.4977

Efficient numerical methods for the large-scale, parallel solution of elastoplastic contact problems

Jörg Frohne[1,*,†], Timo Heister[2] and Wolfgang Bangerth[3]

[1]*Institute of Applied Mathematics, Technische Universität Dortmund, 44221 Dortmund, Germany*
[2]*Mathematical Sciences, Clemson University, O-110 Martin Hall, Clemson, SC 29634-0975, USA*
[3]*Department of Mathematics, Texas A&M University, College Station, TX 77843-3368, USA*

SUMMARY

Quasi-static elastoplastic contact problems are ubiquitous in many industrial processes and other contexts, and their numerical simulation is consequently of great interest in accurately describing and optimizing production processes. The key component in these simulations is the solution of a single load step of a time iteration. From a mathematical perspective, the problems to be solved in each time step are characterized by the difficulties of variational inequalities for both the plastic behavior and the contact problem. Computationally, they also often lead to very large problems. In this paper, we present and evaluate a complete set of methods that are (1) designed to work well together and (2) allow for the efficient solution of such problems. In particular, we use adaptive finite element meshes with linear and quadratic elements, a Newton linearization of the plasticity, active set methods for the contact problem, and multigrid-preconditioned linear solvers. Through a sequence of numerical experiments, we show the performance of these methods. This includes highly accurate solutions of a three-dimensional benchmark problem and scaling our methods in parallel to 1024 cores and more than a billion unknowns. Copyright © 2015 John Wiley & Sons, Ltd.

Received 28 August 2014; Revised 16 March 2015; Accepted 14 June 2015

KEY WORDS: elastoplasticity; contact problems; adaptive finite element methods; active set method; parallel computing

1. INTRODUCTION

Elastoplasticity is a phenomenon that refers to the fact that materials initially deform elastically in reaction to forces, but deformation becomes plastic if the internal stresses exceed a certain threshold. Plastic deformation does not revert to the original frame of reference once the external forces are removed, unlike elastic deformation. Consequently, plastic deformation is purposefully used in many industrial forming processes [1–4], and their control is important to achieve material shapes close to the desired ones. Elastoplasticity is also important when it happens inadvertently, for example, in the investigation of the long-term deformation of machine components under external loads [5–7], in the case of geophysical deformation processes such as plate deformation on long-time scales [8–10] or the response of soil to nearby buildings, dams, or earthquakes [11].

Given this importance in both the control of industrial processes and in understanding natural phenomena that are difficult to investigate experimentally, it is not surprising that a large body of work has been devoted to the development of methods for the computational simulation of elastoplastic processes. Like many other practically important cases, this has proven to be a difficult area characterized by at least the following obstacles.

*Correspondence to: Jörg Frohne, Institute of Applied Mathematics, Technische Universität Dortmund, 44221 Dortmund, Germany.
†E-mail: joerg.frohne@math.tu-dortmund.de

- The realistic simulation of almost all purposeful industrial forming processes as well as the investigation of geophysical problems requires full three-dimensional simulations. This immediately yields problems that have millions of unknowns, and often many more than that. Consequently, efficient computational methods are indispensable for any realistic use of computational tools.
- Plasticity is mathematically described as a variational inequality in which the properties of the material depend in a non-smooth manner on the stress and other possible internal variables like the plastic strain at any given point. This results in a solution that is typically not smooth, and highly accurate simulations can only be achieved by adequately resolving the boundaries of the plastic region, for example, using fine meshes or adaptive mesh refinement. The non-smoothness of the material behavior also requires sophisticated nonlinear solvers.
- Most practical forming methods—rolling, drilling, machining, and cutting—are in fact *contact problems* in which the desired deformation is affected by a tool that exerts external forces to the surface of the body under deformation. Where these two bodies that are in contact are a priori unknown and are only a result of the deformation of one body or both bodies. The mathematical description of contact problems is again a variational inequality with the resulting need for adequate mesh resolution and nonlinear solvers.

In this article, we propose an integrated set of computational methods based on the finite element method that addresses these difficulties and apply them to examples of elastoplastic contact problems. The motivation of our study is the development of methods that are capable of solving problems of realistic complexity. In particular, this means that we need to be able to resolve the solution adequately and that we are able to solve problems that may require many millions or up to billions of unknowns. The building blocks of our approach will be the following:

- *Mesh adaptation.* Resolving the boundaries of the contact area as well as of the plastic zone is indispensable for accurate computations. However, given that these are lower dimensional objects, it is inefficient to do this using global mesh refinement. Furthermore, the locations to be resolved with a finer mesh are not known a priori but must be determined as part of the solution process. Optimal algorithms therefore must incorporate local mesh adaptation, and the remainder of the numerical methods—for example, the partitioning strategy for parallel computations—must be able to cope with locally adaptive and dynamically changing meshes without penalty on speed of convergence.
- *Efficient linear and nonlinear solvers.* Solving large problems requires linear and nonlinear solvers that do not degrade as the problem size grows. We will base our implementation on a damped Newton method for the plastic material behavior and show experimentally that the number of iterations remains constant with increasing problem sizes if the plastic regions are sufficiently resolved (Section 4). Our treatment of the contact conditions uses a primal–dual active set method that we can reformulate as a semi-smooth Newton method; this gives rise to a problem that can be transformed into a linear system that is amenable to multigrid solvers and, thus, to a more or less constant number of inner linear iterations.
- *Massive parallelization.* In order to solve the largest problems, we need to rely on clusters with hundreds or more processor cores. This places tight limits on the algorithms we can use. For example, preconditioners that only work on the part of the matrix that is available locally on a processor will not be able to scale adequately.

The point of this paper is to evaluate an integrative approach in which we show that the components we choose for each of the aforementioned steps interact well and form a combination that is capable of solving complex, realistic problems with almost optimal scalability up to the very largest problem sizes. These building blocks will be discussed in detail in the following sections. We will demonstrate their performance experimentally in Section 4 where we also compare them with other established and currently used methods.

DOI: 10.1002/nme

The implementation of these algorithms is available under an open source license as the step-42 tutorial program [‡] of the DEAL.II finite element library [12, 13]. It uses the TRILINOS library [14, 15] for its parallel linear algebra and P4EST [16] for parallel mesh handling.

Literature overview. In this paper, we consider a unilateral contact problem with a rigid obstacle impinging on a deformable, three-dimensional body. The material of the body is modeled by an elastoplastic constitutive law with linear isotropic hardening. The mathematical description of such problems results in two nonlinearities: the obstacle condition and material behavior.

There is a great deal of literature on such problems. The best and most efficient ways to deal with the plastic behavior are typically variants of Newton's method; see, for example, [17–23]. Recently, a total finite element tearing and interconnect (TFETI) domain decomposition solver was introduced in [24]. To make these methods more robust, one needs to globalize them, for example, using a line search for a damping parameter [4, 23, 25] and we will describe such a method in Section 3.5. Another strategy to choose the damping parameter can be found in [25] and the references therein.

For the contact problem, many methods have been proposed. An overview can be found in the book [26] and in [27–33], which also discuss the primal–dual active set algorithms we will use in the succeeding text. For scalable finite element tearing and interconnect (FETI) domain decomposition methods on this topic, see [34, 35] for example. In contrast, references [4, 36] discuss conjugate gradient-based projected Gauss–Seidel methods (CGPSSOR), which are very efficient for problem sizes up to a few 100,000 degrees of freedom. However, we will here consider problems that are much bigger and we will demonstrate the lack of efficiency of CGPSSOR for such problems in Section 4.4.

The combination of the two preferred algorithms—Newton's method for the nonlinear material law and the primal–dual active set strategy based on a semi-smooth Newton interpretation for the contact problem—has previously been shown to be effective [22, 31, 37], and we will therefore follow this approach here as well. However, previous studies have only considered problems several orders of magnitude smaller than our target and it is not a priori clear that they will scale well to large problem sizes or processor counts. We will show that, with careful consideration of how one reformulates the basic methods, an implementation can indeed scale to very large problems. This aspect and its implications on how one needs to choose the combination of basic algorithms and their various reformulations has, to our knowledge, not been investigated in the available literature. To the best of our knowledge, the literature also does not contain demonstrations of algorithms for elastoplastic contact problems applied to problems as large as the ones we will use in this paper.

We base our implementation on the DEAL.II, TRILINOS, and P4EST libraries for which scalability to large problem sizes has previously been demonstrated for entirely different applications (e.g., [38–40]). We will demonstrate that excellent scalability is also possible in the current context.

Overview. The structure of this paper is as follows. In Section 2, we present the mathematical model we will consider in this paper. The interplay of the numerical methods to compute efficient and accurate numerical solutions is described in Section 3. The efficiency of our approach is demonstrated using numerical examples in Section 4, where we also define a benchmark problem. Finally, we will conclude in Section 5.

2. MATHEMATICAL DESCRIPTION OF THE PROBLEM

In this section, let us give an overview of the mathematical formulation of the problem. We imagine a body that is being elastoplastically deformed through contact with an obstacle that we consider to be rigid; see Figure 1. In practice, the motion of the obstacle is often slow compared with the speed of sound in the body and we can then describe the process as quasi-static; that is, we can

[‡]See http://www.dealii.org/developer/doxygen/deal.II/step_42.html. We will make the programs (all of them are simple variations of step-42) used in the computations shown in Section 4 available as Supplementary Information of this article.

DOI: 10.1002/nme

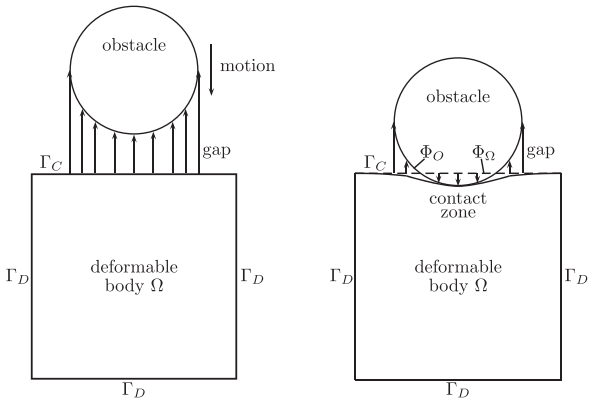

Figure 1. Left: starting configuration. Right: deformed body. In each load step, we only consider static deformations and therefore solve for the static deformation corresponding to the right picture.

consider a sequence of 'load steps' where the solution of each load step only depends on the previous one through the accumulated strain and its hardening effect on the plastic behavior. Consequently, the key component to a numerical solution is a procedure that can solve a single load step efficiently. In the following, we will therefore outline the mathematical formulation for the problem that corresponds to one such load step and point out the places where the solution of previous time steps enters.

To describe the variables that appear in the formulation, consider the situation in Figure 1. Let there be an obstacle that we imagine is pressed into a deformable, bounded polygonal body Ω, where contact may happen on a subset Γ_C of its boundary. We can describe the relative positions of obstacle and body by defining, for every point $\mathbf{x} \in \Gamma_C$ of the *undeformed* object, the closest point of the obstacle as $\mathbf{\Phi}_O(\mathbf{x})$ and by $\mathbf{\Phi}_\Omega(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x})$ the position of the *deformed* object where $\mathbf{u}$ is the deformation field. For small displacements, the contact (non-penetration) condition then requires that $\mathbf{\Phi}_O(\mathbf{x}) \cdot \mathbf{n} \geqslant \mathbf{\Phi}_\Omega(\mathbf{x}) \cdot \mathbf{n}$ where $\mathbf{n} = \mathbf{n}(\mathbf{x})$ is the outward normal to Γ_C at $\mathbf{x}$. We rewrite this condition as $g(\mathbf{x}) - \mathbf{u}(\mathbf{x}) \cdot \mathbf{n} \geqslant 0$ where $g = (\mathbf{\Phi}_O(\mathbf{x}) - \mathbf{\Phi}_\Omega(\mathbf{x})) \cdot \mathbf{n}$ is the commonly used *gap function* indicating the distance between obstacle and undeformed object. Note that on the other hand, $g - \mathbf{u} \cdot \mathbf{n}$ is the distance between obstacle and deformed object. The latter is consequently the quantity that is constrained. For more information, see [27].

In the following description, we will first discuss the strong form of the equations and inequalities we intend to solve. As is common in formulating plastic deformations, this formulation will involve both displacements and stresses (i.e., it is a mixed formulation). However, many numerical methods are most efficient for symmetric and positive definite operators, and consequently, we reformulate the problem so that it only involves the displacement variable by eliminating the stress and using a nonlinear radial-return mapping to replace the plasticity inequalities. Similarly, we will show that the contact inequalities can be replaced by a semi-smooth nonlinearity, resulting in a formulation that only contains (nonlinear) variational equations and that is amenable to a Newton iteration.

2.1. Classical formulation

Elastoplastic contact problems are formulated as partial differential equations with inequalities for the plastic behavior of the medium and the contact condition. In particular, plastic materials are only capable of carrying internal stresses $\sigma(\mathbf{x})$ limited by some maximal stress, i.e., that satisfy an inequality $\mathcal{F}(\sigma) \leqslant 0$ at every point $\mathbf{x}$. The simplest example of such a function $\mathcal{F}$ is the von Mises flow function $\mathcal{F}(\sigma) = |\sigma^D| - \sigma_y$ where σ_y is the yield stress, $\sigma^D = \sigma - \frac{1}{3}\text{trace}(\sigma) I$ is the deviatoric part of the stress tensor, and $|\cdot|$ denotes the Frobenius norm of a tensor. If the stress increases to σ_y, the material ceases to be elastic, and plastic yielding occurs.

Given this setup, let us first consider the classical formulation of the problem for a single load step starting from the stress-free, undeformed rest configuration (we will comment on the quasi-static

DOI: 10.1002/nme

time-dependent case at the end of this section). Its formulation then reads as follows (e.g., [4]). Find a displacement $\mathbf{u} = \mathbf{u}(\mathbf{x})$, a stress $\sigma = \sigma(\mathbf{x})$, and a tensor-valued Lagrange multiplier $\varepsilon^p = \varepsilon^p(\mathbf{x})$ in a domain $\Omega \subset \mathbb{R}^3$ so that

$$\varepsilon(\mathbf{u}) = A\sigma + \varepsilon^p \qquad \text{in } \Omega, \tag{1}$$

$$-\operatorname{div}\sigma = \mathbf{f} \qquad \text{in } \Omega, \tag{2}$$

$$\varepsilon^p : (\tau - \sigma) \geqslant 0 \quad \forall \tau \text{ with } \mathcal{F}(\tau) \leqslant 0, \quad \varepsilon^p \sigma = 0 \qquad \text{in } \Omega, \tag{3}$$

$$\mathbf{u} = 0 \qquad \text{on } \Gamma_D, \tag{4}$$

$$\sigma\mathbf{n} - [\mathbf{n} \cdot (\sigma\mathbf{n})]\mathbf{n} = 0, \quad \mathbf{n} \cdot (\sigma\mathbf{n}) \leqslant 0 \qquad \text{on } \Gamma_C, \tag{5}$$

$$(\mathbf{n} \cdot (\sigma\mathbf{n}))(\mathbf{n} \cdot \mathbf{u} - g) = 0, \quad \mathbf{n} \cdot \mathbf{u} - g \leqslant 0 \qquad \text{on } \Gamma_C. \tag{6}$$

Here, (1) defines the relationship between strain $\varepsilon(\mathbf{u}) = \frac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right)$ and stress σ via the fourth-order compliance tensor A; ε^p provides the plastic component of the strain to ensure that the stress does not exceed the yield stress. In the remainder of this paper, we will only consider isotropic materials for which A can be expressed in terms of the Lamé moduli λ and μ or alternatively in terms of the bulk modulus κ and μ. Equation (2) is the force balance; we will here not consider any body forces and henceforth assume that $\mathbf{f} = 0$. The complementarity condition (3) implies that $\varepsilon^p = 0$ if $\mathcal{F}(\sigma) < 0$ but that ε^p may be a nonzero tensor if and only if $\mathcal{F}(\sigma) = 0$, and in particular that in this case ε^p must point in the direction $\partial\mathcal{F}(\sigma)/\partial\sigma$. In physical terms, this inequality corresponds to the maximum plastic work principle for time-dependent deformation problems. Equation (4) describes a fixed, zero displacement on Γ_D, and (5) prescribes that on the surface $\Gamma_C = \partial\Omega\backslash\Gamma_D$ where contact may appear, the normal force $\sigma_n = \mathbf{n} \cdot (\sigma(\mathbf{u})\mathbf{n})$ exerted by the obstacle is inward (no 'pull' by the obstacle on our body) and with zero tangential component $\boldsymbol{\sigma}_t = \sigma\mathbf{n} - \boldsymbol{\sigma}_n\mathbf{n} = \sigma\mathbf{n} - [\mathbf{n} \cdot (\sigma\mathbf{n})]\mathbf{n}$. Zero tangential stresses are often used to describe well-lubricated contact interfaces. The last condition, (6), is again a complementarity condition that implies that on Γ_C, the normal force can only be nonzero if the body is in contact with the obstacle; the second part describes the impenetrability of the obstacle and the body. The last two equations are commonly referred to as the Signorini contact conditions.

Materials change their atomic structure as they deform plastically, and consequently also their material properties. For metals, this typically results in a stiffening. This 'hardening' effect is described by letting the yield stress depend on the norm of the plastic strain, $|\varepsilon^p|$. [§] We will describe this in the simple von Mises flow function by using a linear isotropic hardening law where $\sigma_y = \sigma_0 + \gamma^{\text{iso}}|\varepsilon^p|$ and consequently $\mathcal{F}^{\text{iso}}(\sigma, |\varepsilon^p|) = |\sigma^D| - (\sigma_0 + \gamma^{\text{iso}}|\varepsilon^p|)$ with the isotropic hardening parameter $\gamma^{\text{iso}} > 0$.

2.2. *Formulation as a variational inequality*

Our algorithms for the solution of the problem outlined in the previous section will be based on a variational formulation. There are numerous variational formulations (e.g., primal mixed, dual mixed, and primal). In the following, we will only state the primal-mixed formulation that corresponds to the problem outlined earlier and then, in the next section, reformulate it as a nonlinear primal one.

[§]For some materials, such as brittle or ductile rocks as well as soil, material models typically also increase the yield stress with the pressure. However, for the case of metals we are interested in here, experimental data suggest that hydrostatic pressure causes no plastic flow [41] and should thus not be included in the definition of the yield stress.

DOI: 10.1002/nme

To this end, let us introduce the following spaces (our examples will all be in dimension $d = 3$):

$$\begin{aligned} V &= \left\{ \mathbf{u} \in \left[H^1(\Omega) \right]^d : \mathbf{u} = 0 \text{ on } \Gamma_D \right\}, \\ V^+ &= \{ \mathbf{u} \in V : \mathbf{n} \cdot \mathbf{u} \leqslant g \text{ on } \Gamma_C \}, \\ W &= L^2 \left(\Omega, \mathbb{R}^{d \times d}_{\text{sym}} \right), \\ \Pi(W \times L^2(\Omega, \mathbb{R})) &= \left\{ (\tau, \eta) \in W \times L^2(\Omega, \mathbb{R}) : \mathcal{F}^{\text{iso}}(\tau, \eta) \leqslant 0 \right\}. \end{aligned}$$

With these spaces, our problem is defined by the following primal-mixed variational formulation [42]. *Find* $\{(\sigma, \xi), \mathbf{u}\} \in \Pi(W \times L^2(\Omega, \mathbb{R})) \times V^+$ *so that*

$$(A\sigma - \varepsilon(\mathbf{u}), \tau - \sigma) + \gamma^{\text{iso}}(\xi, \eta - \xi) \geqslant 0, \qquad \forall (\tau, \eta) \in \Pi(W \times L^2(\Omega, \mathbb{R})), \tag{7}$$

$$(\sigma, \varepsilon(\varphi) - \varepsilon(\mathbf{u})) \geqslant 0, \qquad \forall \varphi \in V^+. \tag{8}$$

While not immediately obvious, this formulation introduces a function $\xi \in L^2(\Omega, \mathbb{R})$ that is zero wherever $\mathcal{F}^{\text{iso}}(\sigma, \xi) < 0$, i.e., in the elastic part of the domain. In the plastic regions of the domain where $\mathcal{F}^{\text{iso}}(\sigma, \xi) = 0$, following the normality principle as presented in physics and applying our yield function $\mathcal{F}^{\text{iso}}(\sigma, \xi)$, we obtain that $\xi = |\varepsilon^p| = |A\sigma - \varepsilon(\mathbf{u})|$. See [42] for more details.

2.3. Reformulating plasticity as a nonlinear equality

The mixed formulation (7)–(8) mentioned earlier, when used as the basis of the finite element method, yields a saddle point problem with all the usual consequences for solving the associated linear systems. For computational purposes, we prefer to transform the formulation into one that only includes the displacement variable $\mathbf{u}$ and from which the stress σ has been eliminated. Such a primal problem can be obtained by making use of the projector onto the set of admissible stresses [4, 43, 44]. For the case of isotropic materials with isotropic linear hardening, it is defined as

$$P_\Pi(\tau) := \begin{cases} \tau, & \text{if } |\tau^D| \leqslant \sigma_0, \\ \left[\dfrac{\gamma^{\text{iso}}}{2\mu + \gamma^{\text{iso}}} + \left(1 - \dfrac{\gamma^{\text{iso}}}{2\mu + \gamma^{\text{iso}}} \right) \dfrac{\sigma_0}{|\tau^D|} \right] \tau^D + \dfrac{1}{3} \text{trace}(\tau) I, & \text{if } |\tau^D| > \sigma_0, \end{cases} \tag{9}$$

where μ is the shear modulus of the material. We have also absorbed the hardening behavior into the nonlinearity, using the fact that we have assumed a linear growth of the yield stress with the plastic strain.

Introducing the stress–strain tensor $C = A^{-1}$, we can use this projector to define the desired primal formulation. *Find the displacement* $\mathbf{u} \in V^+$ *so that*

$$(P_\Pi(C\,\varepsilon(\mathbf{u})), \varepsilon(\varphi) - \varepsilon(\mathbf{u})) \geqslant 0, \quad \forall \varphi \in V^+. \tag{10}$$

Note that this formulation has moved the plasticity inequality into a (non-smooth) nonlinearity and now only contains the contact problem as an inequality. This formulation can be shown to have a unique solution; see [45].

We could in principle reformulate the problem again to also include the contact inequality as a (semi-smooth) nonlinearity [46]. We will in fact use this approach in the succeeding text when solving linearized problems in each Newton step using a primal–dual active set strategy. However, it is not necessary to introduce this formalism here yet, so we will defer the discussion until we have introduced the discrete problem.

2.4. Quasi-static, time-dependent deformation

In reality, deformation is always a time-dependent process. While history plays no role in quasi-static elastic deformation, it is important in plastic deformation because the amount of accumulated plastic strain affects whether the material at a given point deforms elastically or plastically.

DOI: 10.1002/nme

Using a simple backward Euler time discretization of this process, one typically formulates the problem in terms of displacement increments $\Delta\mathbf{u}^n := \mathbf{u}^n - \mathbf{u}^{n-1}$ that, in analogy to (10), have to satisfy the following set of equations in time step n:

$$\left(P_\Pi(\sigma^{n-1} + C\varepsilon(\Delta\mathbf{u}^n)), \varepsilon(\varphi) - \varepsilon(\Delta\mathbf{u}^n)\right) \geqslant 0, \qquad \forall\varphi \in V^{+,n}, \tag{11}$$

where $V^{+,n}$ is the set of feasible displacement increments in load step n, σ^n is updated at the end of each step using the formula

$$\sigma^n = P_\Pi(\sigma^{n-1} + C\varepsilon(\Delta\mathbf{u}^n)), \tag{12}$$

and we introduce a variable $\eta = \eta(\mathbf{x})$ that depends on the load history and that describes the accumulated plastic strain [4, pp. 70–72]

$$\eta^n = \frac{1}{2\mu + \gamma^{iso}}\left(|(\sigma^n)^D| - \sigma_0 + 2\mu\eta^{n-1}\right). \tag{13}$$

Compared with (9), the projector used here is then defined as

$$P_\Pi(\tau) := \begin{cases} \tau, & \text{if } |\tau^D| \leqslant \sigma_0 + \gamma^{iso}\eta^{n-1}, \\ \left[\dfrac{\gamma^{\text{iso}}}{2\mu + \gamma^{\text{iso}}} + \left(1 - \dfrac{\gamma^{\text{iso}}}{2\mu + \gamma^{\text{iso}}}\left(1 - \dfrac{2\mu}{\sigma_0}\eta^{n-1}\right)\right)\dfrac{\sigma_0}{|\tau^D|}\right]\tau^D & \\ \quad + \dfrac{1}{3}\text{trace}(\tau)I & , \text{ if } |\tau^D| > \sigma_0 + \gamma^{iso}\eta^{n-1}. \end{cases} \tag{14}$$

The similarity of (11), (14) to (10), (9) implies that solving a quasi-static load step is essentially equivalent to solving a single load step from scratch with a body force and a different yield stress. For simplicity of notation, we will therefore in the following discuss only the single load step case (9)–(10). We will, however, show time-dependent computations in Section 4.5.

2.5. Nonlinear isotropic hardening

The aforementioned model used is easily generalized to nonlinear isotropic hardening where the yield stress no longer has the form $\sigma_y = \sigma_0 + \gamma^{\text{iso}}|\eta|$ but satisfies a more general relationship $\sigma_y = \sigma_y(\eta)$. In such cases, the projector is defined as

$$P_\Pi(\tau) := \begin{cases} \tau, & \text{if } |\tau^D| \leqslant \sigma_y(\eta^n), \\ \sigma_y(\eta^n)\dfrac{\tau^D}{|\tau^D|} + \dfrac{1}{3}\text{trace}(\tau)I, & \text{if } |\tau^D| > \sigma_y(\eta^n), \end{cases} \tag{15}$$

and it is no longer possible to have an explicit update expression for the plastic strain as in (13). Instead, it is necessary to solve a scalar nonlinear equation at each point where the projector is evaluated (see also [17, 47]). However, while cumbersome to implement, this does not present any additional conceptual problems. We show a numerical example using a nonlinear hardening law in Section 4.5.

3. NUMERICAL METHODS

Our basic approach to solving (10) (or (11)) consists of two nested loops that in pseudo-code reads as follows and solves the problem on a sequence of finite element meshes

Here, the inner loop is the semi-smooth Newton iteration to compute the solution on a fixed mesh, while the outer loop is a typical mesh adaptation loop that determines a mesh with a higher resolution in each iteration based on the solution of the previous iteration.

DOI: 10.1002/nme

```
for (o=0; o<n_outer_iterations; ++o)
    if (o>0)
        adapt mesh and transfer solution to new mesh;
    while (not converged):
        determine active set;
        assemble linearized system for Newton step;
        remove constrained degrees of freedom from linear system;
        solve linear system for Newton update;
        determine step length;
        update solution;
```

Conceptually, we think of the Newton iteration as happening in function spaces, with every Newton step discretized individually so that it can be computed. In practice, of course, we intend these steps to happen on a sequence of finer and finer, adaptively refined meshes (with a small fraction of cells being coarsened) that are obtained through hierarchic refinement to make the transfer of solutions possible in an efficient way. The key point of the overall algorithm is to choose components that interact well with each other. We will comment on the individual pieces in the following subsections. However, the following overarching comments are in order already at this point.

- Newton's method is well known to interact well with (adaptive) mesh refinement: If the current Newton iterate is transferred from the previous to the next mesh upon mesh refinement, then one often finds that only a small number of iterations is necessary on each mesh to achieve convergence of the discrete nonlinear system there. This implies that in a scheme such as the one mentioned earlier, most of the Newton iterations happen on coarser meshes where they are, comparatively, very cheap. In other words, using both Newton iteration and refining meshes between iterations is synergetic. We will demonstrate this, with a certain wrinkle, in Section 4.1.
- Choosing an active set method for the contact problem also integrates well with the overall framework. For example, active set iterations can be run concurrently with the Newton iteration, and unlike some interior point or penalty methods, they do not increase the condition number of the linear systems to be solved in each iteration. This is important because for penalty methods, one has to increase the penalty parameter with mesh refinement to ensure a sufficient approximative fulfillment of the non-penetration condition, and this results in a deterioration of the condition number. One can avoid this by using an augmented Lagrangian formulation in which the Lagrange multiplier is estimated iteratively or by using an enlarged system matrix for the standard Lagrange multiplier method. In contrast to these expensive methods, the mesh dependence of the active set method we use here is only weak, as we will show in the numerical results in the succeeding text.
- Finally, we will show in the succeeding text how we can formulate the active set iteration in a way that allows us to retain linear systems that are structurally equivalent to discrete elasticity problems and, thus, amenable to solvers for symmetric and positive definite systems. Such solvers, based on Krylov subspace methods preconditioned by algebraic multigrid, have previously been shown to work well even for very large problems (e.g., [39, 40]). This is in contrast to most other preconditioners that have been proposed for this kind of problem, and we will compare our method with the CGPSSOR method in Section 4.4.

In the following, we will discuss the various numerical methods that form the basis of the algorithm outlined earlier.

3.1. Newton linearization of the plastic behavior

As discussed earlier, the first step in our algorithm is to apply a Newton method to compute a better approximation to the solution of the primal formulation (10). Formally, (10) is not differentiable. However, it satisfies the conditions of slant differentiability [46] and, consequently, we can hope that a formal linearization works. This ultimately leads to a method equivalent to the frequently used radial-return mapping algorithms [17].

DOI: 10.1002/nme

In Newton's method, we seek an updated solution $\mathbf{u}^i = \mathbf{u}^{i-1} + \alpha^i \delta\mathbf{u}^i \in V^+$. Because the step length α^i can only be determined once $\delta\mathbf{u}^i$ is known, we derive the equations for $\delta\mathbf{u}^i$ under the assumption that we will choose $\alpha^i = 1$. Then, $\delta\mathbf{u}^i = \tilde{\mathbf{u}}^i - \mathbf{u}^{i-1}$ is computed by solving for the full Newton step $\tilde{\mathbf{u}}^i$ using

$$\begin{aligned}\left(I_\Pi \varepsilon(\tilde{\mathbf{u}}^i), \varepsilon(\varphi) - \varepsilon(\tilde{\mathbf{u}}^i)\right) \geqslant \left(I_\Pi \varepsilon(\mathbf{u}^{i-1}), \varepsilon(\varphi) - \varepsilon(\tilde{\mathbf{u}}^i)\right) \\ - \left(P_\Pi(C\,\varepsilon(\mathbf{u}^{i-1})), \varepsilon(\varphi) - \varepsilon(\tilde{\mathbf{u}}^i)\right), \quad \forall \varphi \in V^+,\end{aligned} \tag{16}$$

where the rank-4 tensor $I_\Pi = I_\Pi(\varepsilon^D(\mathbf{u}^{i-1}))$ given by

$$I_\Pi = \begin{cases} C_\mu + C_\kappa, & \text{if } |C\,\varepsilon^D(\mathbf{u}^{i-1})| \leqslant \sigma_0, \\ \frac{\gamma^{\text{iso}}}{2\mu+\gamma^{\text{iso}}} C_\mu + \frac{\left(1-\frac{\gamma^{\text{iso}}}{2\mu+\gamma^{\text{iso}}}\right)\sigma_0}{|C\,\varepsilon^D(\mathbf{u}^{i-1})|} \left(C_\mu - 2\mu \dfrac{C\,\varepsilon^D(\mathbf{u}^{i-1}) \otimes C\,\varepsilon^D(\mathbf{u}^{i-1})}{|C\,\varepsilon^D(\mathbf{u}^{i-1})|^2} \right) + C_\kappa, & \text{else.} \end{cases} \tag{17}$$

Note that I_Π is the (formal) linearization of $P_\Pi(C\cdot)$ around $\varepsilon^D(\mathbf{u}^{i-1})$, with the projector P_Π defined in (9). For a linear isotropic material, the bulk and shear components of the projector are given by

$$C_\kappa = \kappa I \otimes I, \qquad\qquad C_\mu = 2\mu \left(\mathbb{I} - \frac{1}{3} I \otimes I \right),$$

where I and $\mathbb{I}$ are the identity tensors of ranks 2 and 4, respectively.

From this, we see that problem (16) that needs to be solved in each nonlinear step is, in essence, a linear elastic contact problem with spatially variable elasticity coefficients. In fact, the very first Newton iteration solves an elastic, constant-coefficient contact problem if we start from $\mathbf{u}^0 = 0$. We will first discretize this (inequality) problem and then, using an active set method, convert it into an elastic (equality) problem with non-constant coefficients. Those parts of the boundary in which the deformable object is in contact with the obstacle and forces that are inward are treated as Dirichlet boundaries, i.e., we enforce that the displacement $\tilde{\mathbf{u}}^i$ satisfies the contact condition with equality.

3.2. Discretization

Conceptually, our algorithm approximates the solution of the linear problem (16) that defines the Newton update by finite element discretization on a mesh $\mathbb{T}^i$. We will keep the mesh the same between iterations, $\mathbb{T}^i = \mathbb{T}^{i+1}$, unless we have found that the solution in step i was already sufficiently converged *within the approximation of this mesh* by measuring the size of the (discrete) residual as well as ensuring that the active set defined in Section 3.3 has not changed in the previous iteration. This typically happens within 6–30 Newton iterations (Section 4.1), after which the mesh is refined using a simple smoothness indicator (the widely used Kelly error estimator [48]). More sophisticated strategies for mesh refinement are certainly possible here and would likely yield further savings in computational complexity [20, 30, 49, 50].

The discrete problem corresponding to (16) asks for a function $\tilde{\mathbf{u}}_h^i \in V_h^{+,i}$ so that

$$\begin{aligned}\left(I_\Pi \varepsilon\left(\tilde{\mathbf{u}}_h^i\right), \varepsilon(\varphi_h) - \varepsilon\left(\tilde{\mathbf{u}}_h^i\right)\right) \geqslant \left(I_\Pi \varepsilon\left(\mathbf{u}_h^{i-1}\right), \varepsilon(\varphi) - \varepsilon\left(\tilde{\mathbf{u}}_h^i\right)\right) \\ - \left(P_\Pi\left(C\,\varepsilon\left(\mathbf{u}_h^{i-1}\right)\right), \varepsilon(\varphi) - \varepsilon\left(\tilde{\mathbf{u}}_h^i\right)\right), \quad \forall \varphi_h \in V_h^{+,i},\end{aligned} \tag{18}$$

where we define the discrete spaces of piecewise polynomial degree p as

$$\begin{aligned} V_h^i &= \left\{ \mathbf{u}_h \in \left[H^1(\Omega)\right]^d : \mathbf{u}_h|_K \circ F_K^{-1} \in Q_p \text{ for all } K \in \mathbb{T}^i, \mathbf{u}_h = 0 \text{ on } \Gamma_D \right\}, \\ V_h^{+,i} &= \left\{ \mathbf{u}_h \in V_h^i : \mathbf{n} \cdot \mathbf{u}_h \leqslant g \text{ on } \Gamma_C \cap \mathcal{N} \right\}. \end{aligned}$$

DOI: 10.1002/nme

Here, Q_p are tensor product polynomials up to degree p on the reference element, [¶] F_K is the mapping from the reference element to element K, and $\mathcal{N}$ is the set of nodal points $\mathbf{x}_p$ of all shape functions φ_p^i of the finite element space (e.g., the vertices of $\mathbb{T}^i$ for $\mathbb{Q}_1$ finite elements). In other words, we only enforce the contact condition at nodal points.

In the following, we will represent the function $\mathbf{u}_h^i$ using its expansion in terms of finite element shape functions

$$\tilde{\mathbf{u}}_h^i(\mathbf{x}) = \sum_{p \in \mathcal{S}} \tilde{U}_p^i \varphi_p^i(\mathbf{x}),$$

where $\mathcal{S}^i$ is the set of indices of all degrees of freedom.

3.3. Reformulation as an equality constrained problem via the primal–dual active set method

Problem (18) is still an inequality constrained, albeit finite dimensional, problem that cannot be solved in one step. In order to solve it, we apply a single step of a primal–dual active set method that replaces it by an equality constrained problem where boundary displacements are prescribed on parts of the contact surface Γ_C. Strictly speaking, the solution of this problem is not that of (18) because we can only guess where the discrete solution u_h^i touches the obstacle. We could iterate out the nonlinear contact problem with the linearized material model, but in practice, it is sufficient to just go with the first approximation of the contact problem and then re-linearize the plastic nonlinearity.

The general idea of active set methods is as follows: If we knew that the two objects are in actual contact at $\Gamma_C^{\mathcal{A}} \subset \Gamma_C$ (the *active* contact surface), then we could find the solution of (18) by instead searching for a $\tilde{\mathbf{u}}_h^i$ satisfying

$$\begin{aligned} \min_{\tilde{\mathbf{u}}_h^i \in V_h^i} \quad & \frac{1}{2}\left(I_\Pi \varepsilon\left(\tilde{\mathbf{u}}_h^i\right), \varepsilon\left(\tilde{\mathbf{u}}_h^i\right)\right) - \left\{\left(I_\Pi \varepsilon\left(\mathbf{u}_h^{i-1}\right), \varepsilon\left(\tilde{\mathbf{u}}_h^i\right)\right) - \left(P_\Pi\left(C\,\varepsilon\left(\mathbf{u}_h^{i-1}\right)\right), \varepsilon\left(\tilde{\mathbf{u}}_h^i\right)\right)\right\} \\ \text{subject to} \quad & \mathbf{n} \cdot \tilde{\mathbf{u}}_h^i = g \qquad \text{on } \Gamma_C^{\mathcal{A}} \cap \mathcal{N}. \end{aligned}$$

For numerical stability, it is best to understand the equality constraint in an integral sense, i.e., $\left\langle \mathbf{n} \cdot \tilde{\mathbf{u}}_h^i - g, \mathbf{n}\mu \right\rangle_{\Gamma_C^{\mathcal{A}},h} = 0$ for all $\mu \in V_h$. As mentioned earlier, we opt for enforcing the constraint in the nodal points of the finite element only. This is equivalent to approximating the integral $\langle f, g \rangle_{\Gamma_C^{\mathcal{A}}} = \int_{\Gamma_C^{\mathcal{A}}} f\, g \, \mathrm{d}x$ with $\langle f, g \rangle_{\Gamma_C^{\mathcal{A}},h}$ using quadrature that includes only quadrature points that are also nodal points of the finite element space. In practice, this means computing $\langle f, g \rangle_{\Gamma_C^{\mathcal{A}},h}$ via Gauss–Lobatto quadrature, using the fact that the support points of $\mathbb{Q}_p$ elements are defined at Gauss–Lobatto points (or at the vertices for $p = 1$).

Remark 1

An alternative description that also takes into account the function space stability of solutions of the saddle point problem that results from this minimization problem is given in [32]. The description given here leads to the exact same algebraic system and is, consequently, equivalent. The formulation in [32] explicitly constructs the Lagrange multipliers using a basis defined at the vertices (i.e., the Gauss–Lobatto points for the case of linear elements). This basis fulfills a so-called biorthogonality condition in combination with linear ansatz function for the displacements that eventually leads to the same diagonal matrix B.

With this, the solution of the aforementioned problem is given by the following optimality conditions:

$$\begin{aligned} \left(I_\Pi \varepsilon\left(\tilde{\mathbf{u}}_h^i\right), \varepsilon(\varphi)\right) + \left\langle \mathbf{n} \cdot \lambda_h, \mathbf{n}\varphi_h \right\rangle_{\Gamma_C^{\mathcal{A}},h} &= \left(I_\Pi \varepsilon\left(\mathbf{u}_h^{i-1}\right), \varepsilon(\varphi)\right) \\ &\quad - \left(P_\Pi\left(C\,\varepsilon\left(\mathbf{u}_h^{i-1}\right)\right), \varepsilon(\varphi)\right) \qquad \forall \varphi \in V_h^i, \end{aligned} \tag{19}$$

[¶] We consider hexahedral meshes, but the algorithm also works on tetrahedral meshes when using the space of polynomials $\mathbb{P}_p$ on the reference tetrahedron.

DOI: 10.1002/nme

$$\left\langle \mathbf{n}\cdot\tilde{\mathbf{u}}_h^i - g, \mathbf{n}\cdot\mu_h \right\rangle_{\Gamma_C^A,h} = 0 \qquad \forall \mu_h \in V_h^i. \tag{20}$$

This problem only defines the Lagrange multiplier λ_h in normal direction and only on Γ_C^A. We could extend it by zero but it is a quantity whose non-unique parts are never used, as we will see next—in fact, we will eliminate it altogether from the discrete problem and only compute its unique components from $\tilde{\mathbf{u}}_h^i$ in a post-processing step. λ_h can be interpreted as the (discrete approximation of the) force the obstacle exerts on the body. Its analysis can be found in [51].

The aforementioned problem can be written in matrix–vector form as

$$\begin{pmatrix} A(U^{i-1}) & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \tilde{U}^i \\ \Lambda \end{pmatrix} = \begin{pmatrix} F(U^{i-1}) \\ G \end{pmatrix}. \tag{21}$$

Using the inner product defined by quadrature as described earlier guarantees that B is a diagonal matrix. We will discuss solving this linear system in the next subsection where we will exploit the fact that B is diagonal. The remaining question for this section is how to determine Γ_C^A. Obviously, we do not know the exact contact area a priori. Let $\mathcal{S}^i$ be the set of all degrees of freedom (with $|\mathcal{S}^i| = \dim\left(V_h^i\right)$). For simplicity, let us assume that we have rotated degrees of freedom in such a way that at every boundary node, one is responsible for the normal displacement and the other two for tangential displacements. Then let $\mathcal{S}_C^i \subset \mathcal{S}^i$ be those degrees of freedom located at the contact boundary Γ_C and representing normal displacements. The simplest active set methods [52] then define the *active set* $\mathcal{A}^i \subset \mathcal{S}_C^i$ by looking at the sign (inward or outward) of the residual boundary force density

$$\mathcal{A}^i := \left\{ p \in \mathcal{S}_C^i : \left(-A(U^{i-1})U^{i-1} - F(U^{i-1})\right)_p > 0 \right\}.$$

Primal–dual active set methods, see [32, 46], improve on this by instead using the criterion

$$\mathcal{A}^i := \left\{ p \in \mathcal{S}_C^i : \left[-\left(A(U^{i-1})U^{i-1} - F(U^{i-1})\right) + c(\bar{B}^T U^{i-1} - G) \right]_p > 0 \right\}. \tag{22}$$

Here, $\bar{B}_{pq} = \langle \mathbf{n}\cdot\varphi_p, \mathbf{n}\cdot\varphi_q \rangle_{\Gamma_C,h}$ is computed using the entire contact boundary Γ_C, given that we do not know which parts of it will actually be in contact. As before, $\bar{B}$ is diagonal. Furthermore, $G_p = \langle g, \mathbf{n}\cdot\varphi_p \rangle_{\Gamma_C,h}$. For adequate scaling, the penalty parameter c in front of the term $\bar{B}^T U^{i-1}$ clearly needs to be proportional to the elastic coefficients that go into the computation of $A(U^{i-1})U^{i-1}$; in our examples, we will consequently choose it as $c = 100E = 300\kappa(1-2\nu)$.

Because we do not intend to iterate out the active set for the linearized (elastic) contact problem but instead re-linearize the plastic behavior after each active set step, one may argue that a better criterion to determine the active set would replace the *linearized residual* $-A(U^{i-1})U^{i-1}$ by the fully nonlinear one. To this end, in our work, we compute the active set using

$$\mathcal{A}^i := \left\{ p \in \mathcal{S}_C^i : \left[R(\mathbf{u}^{i-1})_p + c(\bar{B}^T U^{i-1} - G) \right]_p > 0 \right\} \tag{23}$$

instead of (22), where the nonlinear residual is defined as

$$R(\mathbf{u})_p = \left(P_\Pi(C\,\varepsilon(\mathbf{u})), \varepsilon\left(\varphi_p^i\right) \right)$$

in accordance with (10).

Remark 2

In parallel computations, the steps of our algorithm discussed in this section can be computed almost completely locally on every machine. This includes the nonlinear residual that only requires one exchange of vector ghost elements. This satisfies one of the goals of our algorithm, namely, that all methods should be designed to scale in a way that allows us to run problems on very large machines.

DOI: 10.1002/nme

At the end of this section, let us add that when using hierarchically refined meshes, one typically ends up with hanging nodes. The degrees of freedom on these are then constrained against neighboring degrees of freedom. On top of that, in three dimensions, some of these hanging nodes are on the boundary and may also be constrained by the contact. In this case, we must choose only one of the two constraints for these degrees of freedom, and we choose the one that results from the hanging node because we want our finite element approximation to be continuous, even if this (slightly) violates the contact condition. As a consequence, we ensure that the set $\mathcal{S}_C^i$ of normal displacement degrees of freedom at the contact boundary does not include degrees of freedom on hanging nodes.

3.4. Solution of the linear system

Using the methods of the previous section, we have arrived at the linear system (21) that now needs to be solved. One could do so as is, but it is difficult to find adequate iterative solvers and preconditioners for saddle point problems of this kind. However, this is also not necessary: Because we have determined an estimate $\mathcal{A}^i$ of the set of active constraints and computed B from it, the second line of (21) simply reads

$$\tilde{U}_p^i = g(\mathbf{x}_p) \qquad \forall p \in \mathcal{A}^i.$$

This corresponds to a Dirichlet boundary condition for the normal displacements at a subset of nodes. Consequently, the displacement part of the solution of (21) is given by solving $\hat{A}(U^{i-1})\tilde{U}^i = \hat{H}$ where

$$\hat{A}_{pq}(U^{i-1}) = \begin{cases} A_{pq}(U^{i-1}) & \text{if } p \notin \mathcal{A}^i, \\ 0 & \text{if, } p \in \mathcal{A}^i \wedge p \neq q, \\ 1 & \text{if, } p \in \mathcal{A}^i \wedge p = q, \end{cases} \qquad \hat{H}_p = \begin{cases} F(U^{i-1})_p & \text{if } p \notin \mathcal{A}^i, \\ G_p & \text{if } p \in \mathcal{A}^i. \end{cases}$$

In other words, we simply remove constrained rows from the original linear system. Because B is nonzero only in these rows, the term $B\Lambda$ completely disappears, rendering Λ a quantity that no longer appears in our computation. As already hinted earlier, the fact that Λ is not uniquely determined then poses no problem in practice. To restore symmetry of the matrix, we can further transform this linear system by Gauss elimination steps on each column $q \in \mathcal{A}^i$ to

$$\hat{\hat{A}}(U^{i-1})\tilde{U}^i = \hat{\hat{H}}(U^{i-1}), \tag{24}$$

where

$$\hat{\hat{A}}_{pq}(U^{i-1}) = \begin{cases} A_{pq}(U^{i-1}) & \text{if } p \notin \mathcal{A}^i \wedge q \notin \mathcal{A}^i, \\ 0 & \text{if } (p \in \mathcal{A}^i \vee q \in \mathcal{A}^i) \wedge p \neq q, \\ 1 & \text{if } p \in \mathcal{A}^i \wedge p = q, \end{cases}$$

$$\hat{\hat{H}}_p(U^{i-1}) = \begin{cases} F(U^{i-1}) - \sum\limits_{q \in \mathcal{A}^i} A_{pq}(U^{i-1})G_q & \text{if } p \notin \mathcal{A}^i, \\ G_p & \text{if } p \in \mathcal{A}^i. \end{cases}$$

In fact, it is this form of the linear system that we assemble directly.

It is not difficult to show that $\hat{\hat{A}}(U^{i-1})$ is a positive definite matrix if $A(U^{i-1})$ is positive definite; by construction, it also inherits the symmetry from $A(U^{i-1})$. Consequently, it is now amenable to solution by conjugate gradients (CG) or other Krylov space solvers with an algebraic multigrid as preconditioner.

Remark 3

Our reformulation has resulted in a linear system that can be assembled with local operations requiring a minimal amount of communication to add to elements of the matrix or vector that are not stored on the local processor. Furthermore, it can be solved by a preconditioned BiCGStab iteration.

DOI: 10.1002/nme

As mentioned earlier, this combination satisfies our requirement that we only use methods that are known to scale well even to very large systems. We will numerically confirm that this is indeed the case in Sections 4.2 and 4.3.

3.5. Line search

Once we have computed $\tilde{U}^i$, we can use a line search to determine the next Newton iterate U^i. We use the usual backtracking line search [52] to find the first step length $\alpha^i \in \{1, 2^{-1}, 2^{-2}, \ldots\}$ so that

$$\left\| \hat{R}(\mathbf{u}_\alpha) \right\|_{\ell_2} < \left\| \hat{R}\left(\mathbf{u}_h^{i-1}\right) \right\|_{\ell_2}, \tag{25}$$

where $\mathbf{u}_\alpha = (1-\alpha^i)\mathbf{u}_h^{i-1} + \alpha^i \tilde{\mathbf{u}}_h^i$ and where $\hat{R}(\mathbf{u}_\alpha) = R(\mathbf{u}_\alpha)$ with the exception of (1) elements $p \in \mathcal{A}^i$ where we set $\hat{R}(\mathbf{u}_\alpha)_p = 0$ and (2) elements that correspond to hanging nodes, which we eliminate in the usual manner.

As before, all of these steps can be carried out mostly locally, with a small amount of communication to exchange elements of the residual vectors $R(\mathbf{u}_\alpha)$ between processors.

In our computations, we typically observe that the line search is necessary mostly during the first Newton iterations.[‖] Furthermore, for elastoplasticity, one considers a time-dependent problem where it is always possible to reduce the time or load step if the line search does not lead to convergence of the Newton iteration. That said, the line search makes the method strictly more robust.

3.6. Summary of the algorithm

To summarize, the steps of our algorithm to be performed in Newton iteration i are as follows:

(1) Assemble the residual vector $R\left(\mathbf{u}_h^{i-1}\right)$ and the matrix $\bar{B}$ (or take the latter from the previous iteration, if the mesh has not changed).
(2) Compute the active set $\mathcal{A}^i$.
(3) Assemble the matrix $\hat{\hat{A}}$ and right-hand side $\hat{\hat{H}}$ in the same way as one would usually assemble A but eliminating rows and columns $p, q \in \mathcal{A}^i$ when copying local contributions from every cell to the global objects.
(4) Solve linear system (24).
(5) Find a step length using the line search procedure (25).

4. NUMERICAL RESULTS

In this section, we show results for a number of test cases that illustrate the performance of our methods. In particular, in Section 4.1, we consider the overall performance of our algorithms in terms of the number of Newton iterations and inner linear iterations, and we will demonstrate that interpolating from the previous to the next mesh significantly reduces the computational complexity. This example also shows some of the difficulties one encounters when working on problems with geometries and computational sizes relevant to real applications. In Section 4.2, we then illustrate the accuracy of our solutions by defining a benchmark problem. Sections 4.3 and 4.4 evaluate the parallel scalability of our methods up to 1024 processor cores and more than a billion unknowns and compare our algorithm with one previously described in the literature. Section 4.5 shows a quasi-static computation with a load that increases over time. We conclude our numerical results by showing a case of an obstacle with a realistic geometry used in metal drilling in Section 4.6.

[‖]For example, in the weak scaling test with 1024 cores and 1.2 billion unknowns shown in Section 4.3, we perform 17 Newton iterations. Of these, 9 use a full step size $\alpha^i = 1$, 7 use $\alpha^i = 0.5$, 1 uses $\alpha = 0.25$, and none use even smaller step sizes.

DOI: 10.1002/nme

Our implementation of the algorithms discussed earlier is available as the step-42 tutorial program of the open source finite element library DEAL.II [12, 13] and all computations in the succeeding text are based on variants of it. We use TRILINOS version 11.0.3 for parallel linear algebra [14, 15] and P4EST 0.3.4 to distribute meshes among processors [16].

4.1. Evaluating nonlinear and linear solvers

In order to evaluate the performance of nonlinear and linear solvers, we consider two test cases. The first simulates pressing a rigid sphere into an elastoplastic cube (see the left panel of Figure 2; this example is also the subject of Section 4.2). The second illustrates our ability to solve problems on unstructured meshes with complex obstacles: We consider a situation where we press a binary mask (i.e., an obstacle with a flat bottom for certain values of x_1, x_2 and an infinite distance for all other values of x_1, x_2) into a half sphere. We choose as our obstacle a shape corresponding to the Chinese symbol for 'force'; see the right panel of Figure 2. As will become clear from the following discussion, this is not simply a more complicated obstacle but one that provides for a fundamental lesson on dealing with complex shapes.

We solve these problems on a sequence of adaptively refined meshes, and Tables I and II summarize the number of cells and unknowns on each of these meshes, along with the number of nonlinear and the average number of linear iterations per Newton step. Each table compares two cases: where the solution on one mesh is interpolated onto the next one to use as a starting guess and where we start from scratch on each mesh. In the latter case, our starting solution is a zero vector with

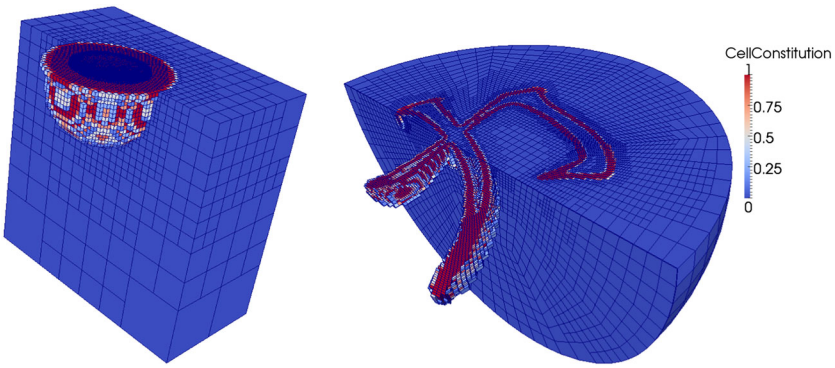

Figure 2. Adaptively refined mesh (cut away in the left half of the domain for all cells that are exclusively elastic) and fraction of quadrature points in each cell that are plastified (blue: none, red: all). Left: pressing a sphere into a cube, corresponding to the benchmark discussed in Section 4.2. Right: pressing the Chinese symbol for 'force' into a half sphere. The part of the top surface where the obstacle is in contact with the body is shown in Figure 6.

Table I. Performance characteristics of our algorithms when pressing a sphere into a cube.

Mesh	No. of cells	No. of DoFs	No. of Newton it.s	Average no. of linear it.s
0	512	2187	6 / 6	3 / 3
1	1548	6294	5 / 8	6 / 6
2	4768	18,207	12 / 11	10 / 9
3	14,652	52,497	8 / 10	12 / 10
4	45,368	154,647	9 / 21	15 / 15
5	140,344	461,106	7 / 31	22 / 22
6	435,044	1,400,961	6 / 31	20 / 23
7	1,347,690	4,297,257	6 / 28	23 / 23

The number of linear iterations is averaged over the Newton steps taken on that mesh. In the last two columns, numbers correspond to the algorithm where we do/do not interpolate the solution from the previous to the next mesh upon mesh refinement.
DoFs = degrees of freedom; it.s = iterations.

DOI: 10.1002/nme

Table II. Performance characteristics of our algorithms when pressing a Chinese symbol into a half sphere.

Mesh	No. of cells	No. of DoFs	No. of Newton it.s	Average no. of linear it.s
0	384	1443	5 / 5	1 / 1
1	1189	4770	13 / 6	5 / 5
2	3695	14,787	13 / 6	7 / 6
3	11,661	46,122	19 / 7	9 / 9
4	36,630	141,990	22 / 11	9 / 9
5	114,995	432,180	20 / 11	14 / 13
6	360,100	1,319,442	23 / 18	16 / 17
7	1,125,977	4,007,004	37 / 26	18 / 21
8	3,513,838	12,215,337	27 / 30	24 / 23
9	10,931,570	37,106,544	23 / 29	28 / 29
10	33,928,726	112,814,994	22 / 48	30 / 35

All columns as in Table I.

those elements that would violate the contact condition displaced in normal direction so that they are below the obstacle. This may lead to cells at the surface whose displaced image is inverted and in any case to unphysical strains, so the first Newton step is carried out using a completely elastic material model (equivalent to one with an infinite yield stress) to avoid the dependence of the next solution on the unphysical stress state of the initial solution. In both cases, we initialize the active set based on the starting solution.

The results of these tables show that for the case of the spherical obstacle (Table I), interpolating the solution guarantees an almost constant number of Newton iterations on each mesh, while this number increases if the solution on each mesh is computed without reference to that on the previous mesh. This shows that for this case, the majority of the numerical effort has indeed been pushed onto coarser and consequently much cheaper meshes. In addition, the number of inner linear iterations increases only slowly with the number of degrees of freedom (by a factor of less than 2 for an increase from a first reasonable mesh with 50,000 degrees of freedom to one with more than 4 million), yielding an almost linear overall complexity.

The situation for the Chinese symbol-shaped obstacle (Table II) is more complex. Here, interpolating the solution provides a *worse* starting solution than just a zero vector in the first few mesh refinement iterations. Our numerical investigations indicate that this is because for this rather irregular obstacle, the first meshes used are simply too coarse (the mesh shown in Figure 2 corresponds to mesh 5 of the table) and that the solution computed on each does not accurately reflect the solution on the next. On the other hand, once the mesh does resolve the obstacle, the number of Newton steps required when using the interpolated solution from the previous mesh reverts to a reasonable number and, more importantly, stays constant as the mesh is further refined. At the same time, the number of iterations when starting the Newton method anew on each mesh continues to grow, illustrating the significant computational advantage obtained by interpolating the solution. In this context, note in particular that we are most concerned with the number of iterations on the last few, most expensive meshes. As before, the number of inner linear iterations per Newton step does increase as the mesh becomes finer, albeit slowly: by a factor 3–4 as the number of degrees of freedom grows by three orders of magnitude from 10^5 to 10^8.

The discussion of these results points at a fundamental issue when moving from 'simple' benchmarks to realistic cases with complex geometries. There, the full performance of an algorithm may only become apparent when applying it to cases where the goal is to solve something to achieve at least engineering accuracy. As the next section will show, reaching this level of accuracy is not trivial even for the case of the sphere.

4.2. Evaluating accuracy: pressing a sphere into a cube

Because we know of no widely used benchmarks for elastoplastic contact problems, we have decided to use the first of the two cases discussed in the previous section as a benchmark problem. A

DOI: 10.1002/nme

Table III. Description of the benchmark discussed in Section 4.2.

Domain	Cube $(0,1\text{ mm})^3$
Obstacle	Sphere $\mathbf{x}_{\text{center}} = (0.5, 0.5, \text{and } 1.59\text{ mm})$, $r = 0.6$ mm
Contact surface	$\Gamma_C = \{\mathbf{x} \in \Omega : x_3 = 1\text{ mm}\}$
Boundary conditions	$\mathbf{u} = 0$ on the bottom of the cube
	$u_1 = u_2 = 0$ on the sides of the cube, u_3 is free
Material properties	$E = 200$ GPa, $\nu = 0.3$
	(or equivalently $\kappa = 166.67$ GPa, $\mu = 76.92$ GPa)
	$\sigma_0 = 400$ MPa, $\gamma^{\text{iso}} = 2\mu\dfrac{\gamma}{1-\gamma} = 1.55$ GPa, $\gamma = 0.01$
Evaluation point	$P = (0.5001, 0.5001, \text{and } 0.9501\text{ mm})^T$
	(in the reference configuration)

full description of all the parameters corresponding to the picture shown in the left panel of Figure 2 is given in Table III. [**]

To evaluate accuracy we consider three measures as follows:

(1) The vertical displacement $u_z(P)$ at point P.
(2) The diagonal elements $\sigma_{xx}(P), \sigma_{yy}(P), \sigma_{zz}(P)$ of the stress tensor at the same point.
(3) The integral $\int_{\Gamma_C} \lambda_z$ of the vertical component of the reaction force density $\boldsymbol{\lambda}$ exerted by the deformable body onto the obstacle. The integral over Γ_C is then the total vertical force.

For P (Table III), we have chosen a point slightly offset from the central axis to avoid the complication that computed stresses are discontinuous and, consequently, non-unique along cell interfaces. The chosen point is guaranteed to never be on such an interface upon regular mesh refinement of the cubic domain.

Given these considerations, Table IV shows our numerical results for both uniform and adaptive mesh refinement with linear ($p = 1$) and quadratic finite elements ($p = 2$). Because of the symmetry of the problem and the chosen location for the evaluation point P, we have $\sigma_{xx}(P) = \sigma_{yy}(P)$; consequently, only one of the two is shown. The displacements make sense given the indentation of -0.01 mm at the surface; we have verified that our computation of the total force is correct for an *elastic* body for which an analytic solution exists (the so-called Hertzian contact problem [53]). We therefore believe that the values shown in the table indeed converge to the correct ones. The last row of the table contains our best, extrapolated guess of the exact value. We show in Figure 3 convergence histories for $u_z(P)$ and the integrated contact force against these best guesses (convergence for $\sigma_{xx}(P)$ and $\sigma_{zz}(P)$ is less regular—as may be expected—and not shown here).

The table and figure make clear the degree of difficulty of this problem, despite its apparent simplicity: For example, computing the displacement to better than 0.1% already takes several hundred million unknowns with globally refined meshes for Q_1 elements. On the other hand, the results also make clear that adaptive meshes—in particular when combined with higher-order elements—can achieve the same level of accuracy with far fewer unknowns: in the case of Q_2 elements, just a few hundred thousands. Next, we cannot determine the stresses σ_{xx}, σ_{zz} to better than a few percent, even on meshes with a billion unknowns, unless we use both adaptivity and higher-order elements.

These results, considering a relatively simple model problem, suggest that problems of more realistic complexity such as the Chinese symbol considered in the previous section or the drill bit discussed in Section 4.6 can no longer be solved without adaptivity or higher-order elements, even just to rather moderate accuracies. Rather, realistic applications *require* the complex methods discussed here, as well as very large computations that can only be carried out in parallel. Consequently, we will consider the parallel scalability of our methods in the next section.

[**] We could also have evaluated the accuracy of our solution (relative to the solution on the finest mesh) using the Chinese symbol obstacle. However, such results would be difficult to reproduce by others in the community. Furthermore, as the results in the succeeding text will indicate, it is questionable whether highly accurate results could have been obtained for the complex obstacle even on the more than 100 million unknowns shown in the last line of Table II.

DOI: 10.1002/nme

Table IV. Computed values of displacements (in millimeter), stresses (in megapascal), and integrated forces (in Newton) for global and adaptive mesh refinement when pressing a sphere into an elastoplastic cube (Section 4.2).

Mesh	No. of cells	No. of DoFs	$u_z(P)$	$\sigma_{xx}(P)$	$\sigma_{zz}(P)$	$\int_{\Gamma_C} \lambda_z$
			Global mesh refinement with $p = 1$			
0	512	2187	−0.0075681	−5733.1	−6098.2	37.306
1	4096	14,739	−0.0070691	−3317.5	−3855.5	62.313
2	32,768	107,811	−0.0068296	−1946.6	−2565.8	59.099
3	262,144	823,875	−0.0066294	−1027.6	−1684.2	56.761
4	2,097,152	6,440,067	−0.0065339	−1155.4	−1832.8	55.751
5	16,777,216	50,923,779	−0.0064756	−1240.1	−1925.0	55.333
6	134,217,728	405,017,091	−0.0064602	−1170.2	−1856.2	55.221
			Global mesh refinement with $p = 2$			
0	512	14,739	−0.0061351	27.5	−605.7	66.640
1	4096	107,811	−0.0074271	−376.3	−1085.8	57.127
2	32,768	823,875	−0.0065627	−766.3	−1450.0	55.226
3	262,144	6,440,067	−0.0064618	−1107.4	−1794.1	55.247
4	2,097,152	50,923,779	−0.0064541	−1129.5	−1816.0	55.168
5	16,777,216	405,017,091	−0.0064547	−1156.1	−1842.7	55.183
			Adaptive mesh refinement with $p = 1$			
0	512	2187	−0.0075681	−5733.1	−6098.2	37.306
1	1548	6294	−0.0070687	−3317.7	−3855.7	62.323
2	4768	18,207	−0.0068284	−1947.1	−2566.4	59.118
3	14,652	52,497	−0.0066271	−1027.5	−1684.2	56.794
4	45,368	154,647	−0.0065296	−1156.8	−1834.2	55.835
5	140,344	461106	−0.0064715	−1244.6	−1929.3	55.492
6	435,044	1,400,961	−0.0064694	−1242.4	−1927.8	55.377
7	1,347,690	4,297,257	−0.0064587	−1171.9	−1857.8	55.291
8	4,175,172	13,075,026	−0.0064584	−1101.7	−1787.0	55.224
9	12,911,781	40,051,599	−0.0064562	−1150.4	−1836.5	55.197
10	39,915,821	122,655,213	−0.0064557	−1154.8	−1841.2	55.184
11	123,459,540	377,150,526	−0.0064553	−1149.8	−1836.2	55.184
			Adaptive mesh refinement with $p = 2$			
0	64	2187	−0.0064262	−5625.5	−6050.7	26.980
1	176	6087	−0.0061346	−27.7	−605.6	66.647
2	652	20,397	−0.0074270	−376.4	−1085.8	57.130
3	2192	64,731	−0.0065623	−766.3	−1450.1	55.229
4	6980	195,327	−0.0064618	−1107.4	−1794.1	55.251
5	21,652	585,603	−0.0064530	−1125.6	−1811.8	55.172
6	67,264	1,829,181	−0.0064549	−1136.0	−1822.6	55.182
7	208,804	5,842,461	−0.00645493	−1154.6	−1841.0	55.1782
8	647,144	18,341,805	−0.00645512	−1157.1	−1843.4	55.1771
9	2,006,964	56,467,965	−0.006455126	−1158.3	−1844.8	55.1783
10	6,224,212	175,552,233	−0.006455129	−1158.5	−1845.0	55.1789
			Extrapolated best guesses			
	∞	∞	−0.00645513(1)	−1158.(7)	−1845.(2)	55.179(4)

The last row contains extrapolated values; digits in parentheses are probably correct but with a low degree of certainty.

DOI: 10.1002/nme

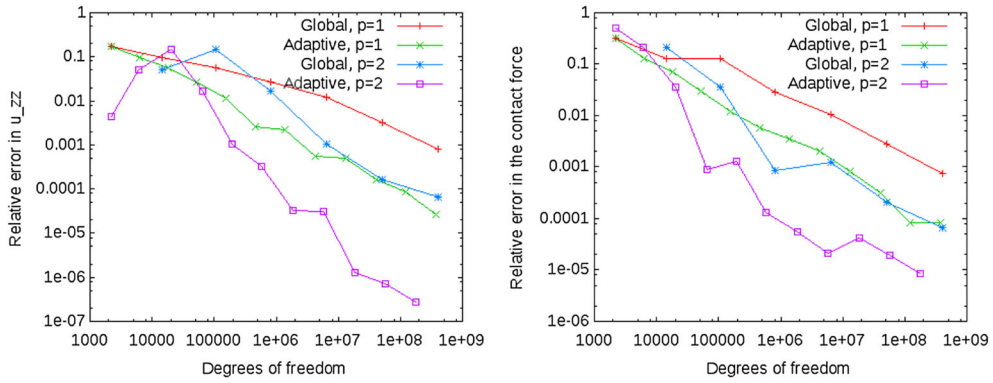

Figure 3. Convergence of $u_z(P)$ (left) and the contact force $\int_{\Gamma_C} \lambda_z$ for the benchmark problem discussed in Section 4.2. Convergence is measured against the extrapolated best guesses from Table IV.

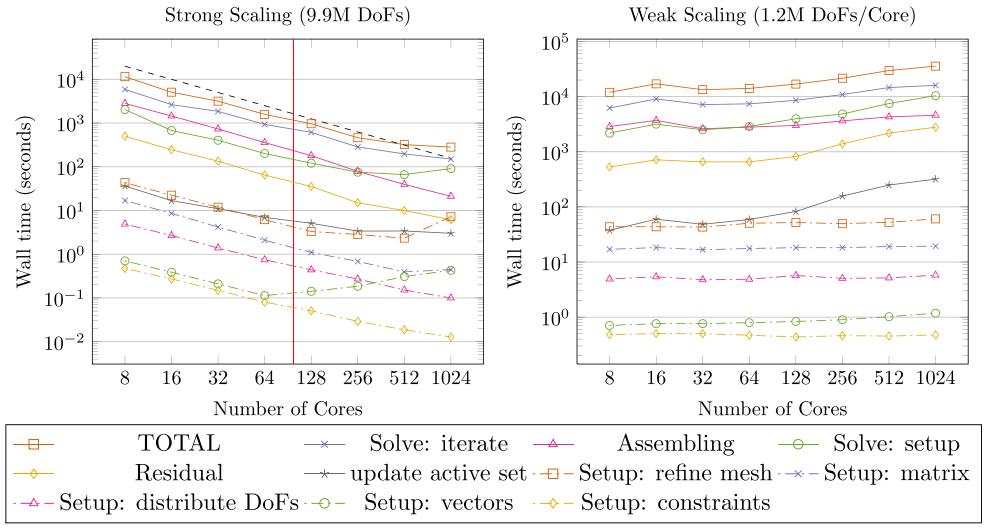

Figure 4. Results for strong (left) and weak (right) scaling as discussed in Section 4.3. The red line in the left panel corresponds to 100,000 degrees of freedom per core beyond which scalability deteriorates; the dashed line represents perfect speedup. In the weak scaling case, the slowdown is due to parallel inefficiencies as well as to a larger number of inner iterations per Newton step. See the main text for more information.

4.3. Parallel scalability

The results shown in Section 4.1 already indicate that the algorithms described in this contribution can, at least in principle, scale weakly to large problems because the number of Newton iterations does not grow with the problem size, and the number of linear iterations grows only slowly. Furthermore, the previous section shows that even for relatively simple problems, one needs very large numbers of unknowns to achieve an error of even just better than 1%, necessitating the use of parallel computers. To investigate whether our algorithms indeed scale, we have run a number of computations corresponding to the benchmark discussed in the previous section on the *Brazos* cluster at Texas A&M University. We have used up to 128 nodes, each equipped with one eight-core Opteron 2350 processor (2.5 GHz clock rate), 32 GB of memory per node, and an Infiniband interconnect.

To evaluate our algorithms, Figure 4 shows strong and weak scaling results for all significant components of our program.

For strong scaling, we have chosen a problem with 9.9 million unknowns (the largest size we can fit on a single node with eight cores). For each computation, we initially solve on a uniformly refined mesh, and the timings are then carried out for the following adaptively refined mesh to

DOI: 10.1002/nme

include the effects of adaptivity. As seen in the left panel of Figure 4, we obtain linear acceleration of all significant parts of the program until around 256 processors (approximately 40k unknowns per core). Beyond this point, the fraction of the problem each core has to solve simply becomes too small to fully amortize the cost of communication. Previous experience [38, 39] has shown that computations often fail to scale once the number of unknowns per core drops below approximately 100,000, indicated by the red line in the figure. Our results here scale somewhat further than that, possibly because the problem requires more work per degree of freedom.

To demonstrate weak scalability, we use meshes where the number of degrees of freedom is approximately equal to 1.2×10^6 times the number of cores. This is achieved by starting from $3 \times 3 \times 3$, $4 \times 4 \times 4$, or $5 \times 5 \times 5$ coarse mesh, refining globally a number of times and then refining adaptively once in such a way that we achieve the desired number of unknowns. Similar to before, timings are then carried out for the last, adaptive step. Coarse mesh sizes are chosen so that the last step can be achieved by refining approximately 10% of cells.

The results shown in the right panel of Figure 4 again show that our algorithms scale well, even to very large problems, losing only approximately a factor of 3 in the overall efficiency when increasing the number of processor cores from 8 to 1024. The figure shows that the primary obstacles to both strong and weak scalability are solver setup and solver iterations (including preconditioner). This preconditioner is provided by Trilinos' ML package. However, only part of the slowdown can be attributed to parallel inefficiencies: As shown in Tables I and II, as problems become larger, the number of inner iterations per Newton step increases slowly. Here, from 8 to 1024 processors, it increases from an average of 28 to 52, accounting for almost a factor of 2 of the slowdown in the solver (but not preconditioner setup). In other words, a significant fraction of the slowdown is due to a loss in quality of the preconditioner rather than inefficient parallel communication.

On the other hand, the third most expensive part of the program—assembling the linear systems, an almost completely local operation—scales almost linearly. The computation of the residual lacks some scalability because processors' work depends on the fraction of their cells located at the contact interface; this could be optimized, but we have not done so given that it takes up only 7% of run time even on the largest computations. All other operations are negligible.

We note that the last data point of these weak scaling results corresponds to a nonlinear, inequality constrained problem with more than 1.25 billion unknowns and is solved in about 10 h. Given that during this time we perform 17 Newton iterations, this corresponds to 35 min per Newton iteration for a system with 1.2 million unknowns per core. Taken together with the strong scaling results, this is consistent with experience from previous work that suggests a time of 1–2 min per linear solve (including assembly, setting up preconditioners and post-processing) when using approximately 100,000 unknowns per core for non-trivial problems of this size [38–40].

The results shown in this section demonstrate that the methods we have presented work together in a way that allows us to solve very large problems in a reasonable amount of time, utilizing large numbers of processors. Furthermore, the limitations we find—primarily the scalability of setting up and applying the algebraic multigrid preconditioner ML [54] in TRILINOS, i.e., the red and orange curves with boxes—are well known-bottlenecks to parallel scalability.

4.4. Comparison with an existing method

In order to put the results of the previous section in context, we have also implemented an entirely different solver for elastoplastic contact problems that has previously been described in the literature: The CG-based projection method CGPSSOR; see [36]. CGPSSOR uses conjugate gradients as the outer solver and a Gauss–Seidel-(or SSOR-)-based preconditioner in which all degrees of freedom that violate the obstacle condition are always projected. The CG iteration is modified in such a way that these projected degrees of freedom are left untouched and remain at their correct values. It is important to note that this modification does not just affect the inner linear solver but instead requires the assembly of different linear systems in each iteration and involves a different approach to solving the problem altogether. The primary advantage of the algorithm is that it treats the contact problem within the inner iteration rather than solving linear systems that correspond to a fixed

DOI: 10.1002/nme

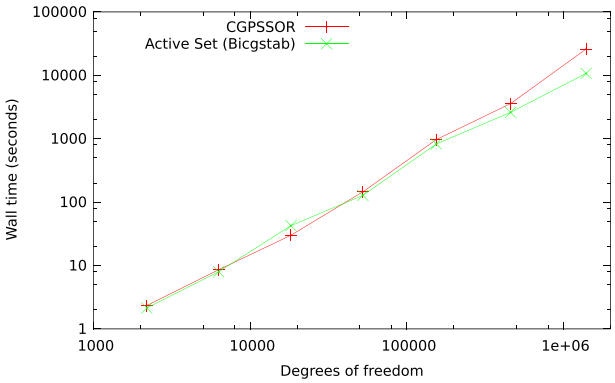

Figure 5. Comparison of the overall computation time for the active set strategy and the CGPSSOR method for a computation on a single processor.

set of contact nodes. A detailed description of this method can be found in [4, 36]. In contrast to the first of these references, we are here using a non-cascadic version.

We apply this method and compare it with our combination of algorithms to the benchmark example of Section 4.2. Figure 5 shows the run time for the two methods on a single processor. The curves compare the time necessary to solve the nonlinear system on each sequence of meshes and include the time to assemble and solve the linear and nonlinear systems but excluding the time for post-processing, mesh refinement, and generating graphical output. For small problems, the two methods are comparable in performance. However, beyond approximately 200,000 degrees of freedom, the run time of the CGPSSOR method increases faster than that of the active set/BiCGStab/algebraic multigrid method proposed here. Of course, 200,000 unknowns (corresponding to a 40^3 uniform mesh) is a vanishingly small number for many of the problems we have in mind for our approach. We did not compute numbers beyond 1.4 million unknowns because run times on a single processor become unreasonably large. The point, however, is not to establish that on a single processor, one method is better or worse than the other but simply to ensure that the solver we propose in this paper is asymptotically no worse than existing and documented methods.

We did not attempt to parallelize CGPSSOR because it cannot win there: While we have shown in the previous section that our methods provide scalability to large numbers of processors, SSOR is an inherently sequential algorithm. This also extends to the cascadic version of CGPSSOR discussed in [36]. Both can of course be parallelized by running the preconditioner only on that diagonal block of the matrix each processor stores locally, but it is well known that the quality of preconditioners then deteriorates, unlike our algebraic multigrid preconditioner.

The conclusion of this section is that for rather small problems, our proposed set of algorithms is comparable with one that has been described in the literature recently. Furthermore, our work here scales well in parallel, unlike the pre-existing CGSSOR algorithm.

4.5. A quasi-static example

As discussed in Section 2.4, the one-step situation of the previous examples is easily extended to quasi-static problems consisting of a sequence of loading steps [4, 21, 47]. To evaluate our methods in this context, we revisit the Chinese symbol example from Section 4.1 but apply the loading in steps; we also replace the hardening law by nonlinear isotropic hardening with a saturation term [17, 22]

$$\sigma_y = \sigma_0 + \gamma^{iso}|\varepsilon^p| + (\sigma_\infty - \sigma_0)\,(1 - \exp(k_e|\varepsilon^p|))\,.$$

This implies that we have to solve a scalar nonlinear equation in each integration point during assembly of the Newton matrix. For this purpose, we choose a Newton method again. Calculations are carried out on a relatively coarse, locally refined mesh with 137,157 degrees of freedom, which we obtained after five adaptive refinement cycles.

DOI: 10.1002/nme

Table V. Description of the quasi-static example of pressing the Chinese symbol into a half sphere.

Domain	Half sphere, $r = 0.8$ mm, $\mathbf{x}_{\text{center}} = (0.5, 0.5, \text{and}0.5 \text{ mm})$
Obstacle	Chinese symbol, initially in touching contact
Contact surface	$\Gamma_C = \{\mathbf{x} \in \Omega : x_3 = 0.5 \text{ mm}\}$
Time interval	[0s, 10s]
Obstacle velocity	2×10^{-4} mm/s, in negative z-direction
Boundary conditions	$\mathbf{u} = 0$ on the spherical surface; free surface or contact on the flat surface
Material properties	$E = 200$ GPa, $\nu = 0.3$
	(or equivalently $\kappa = 166.67$ GPa, $\mu = 76.92$ GPa)
	$\sigma_0 = 250$ MPa, $\sigma_\infty = 400$ MPa, $k_e = 17.0$,
	$\gamma^{\text{iso}} = 2\mu \dfrac{\gamma}{1-\gamma} = 1.55$ GPa, $\gamma = 0.01$

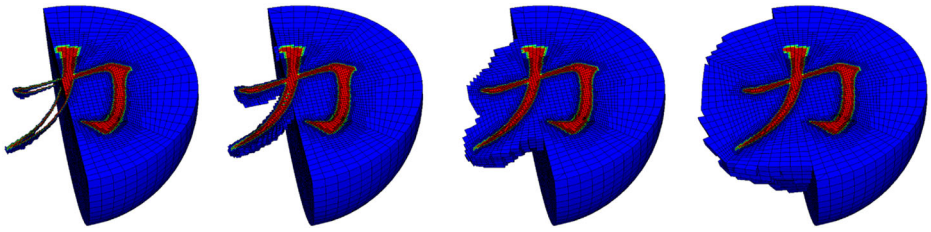

Figure 6. Quasi-static deformation: contact zone (red) and, in the left half of the domain, the plastified cells. Cells shown with colors other than red or blue have vertices both inside and outside the contact zone. Left to right: load steps 1, 2, 5, and 10 with one load step every second.

Table VI. Quasi-static deformation: influence of the time step size.

No. of time steps	No. of plastic Q-points	Plastic volume	Average no. of Newton iterations
1	212,810	0.1140	34.00
2	213,070	0.1146	11.50
5	211,352	0.1076	8.20
10	209,335	0.1019	7.90
20	207,378	0.0974	7.55
40	206,499	0.0947	7.65
100	205,684	0.0929	7.27
200	204,827	0.0914	6.95
500	205,133	0.09163	6.52

Table V describes the details of this case and Figure 6 shows the progress of plastic deformation as the obstacle is indented deeper into the body. We present numerical results in Table VI that show that, as expected, the average number of Newton iterations per time step (or load step, depending on viewpoint) decreases as time steps become smaller. Furthermore, the number of plastic quadrature points and the corresponding volume of the plasticized zone converge, as expected, if slowly. In deviation from the value given in Section 3.3, we have found that we obtain best results with a primal–dual penalty parameter of $c = 10^6 \cdot E$. The table uses this value.

4.6. A complex application: pressing a drill head into a cube

As a final example, we consider a more realistic obstacle: a drill head that is used as a test case in the project funded by the German Science Foundation that supports the first author. The geometry of this drill head, shown in Figure 7, contains two channels used to pump cooling fluid to the drill site. The presence of the cooling liquid also ensures that the contact is nearly frictionless and thus satisfies at least approximately one of our assumptions. On the other hand, it is clear that the actual drilling process cannot be modeled taking into account only continuous deformations. Furthermore,

DOI: 10.1002/nme

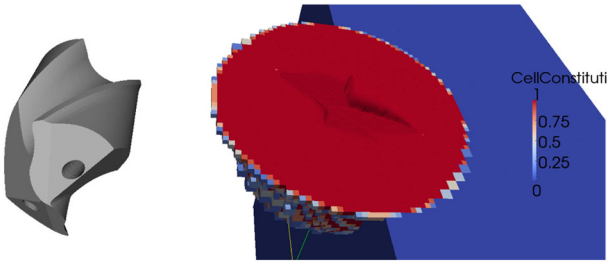

Figure 7. Left: drill head with two cooling canals. Right: distribution of plastified cells and elastoplastic deformation.

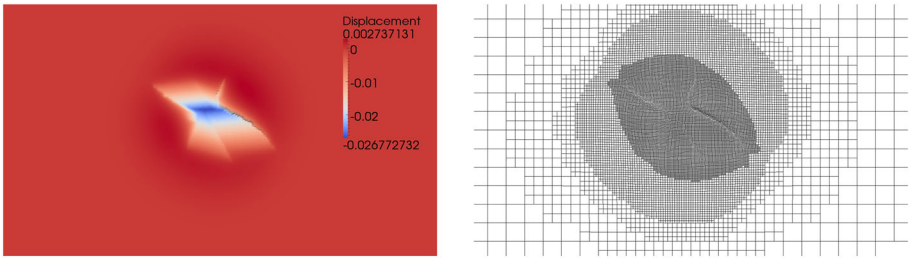

Figure 8. Left: displacements in z-direction. Right: adaptive refined mesh on the contact surface.

heat development and the temperature dependence of material parameters play an important role in drilling—both effects that are not considered here.

As a first step in the direction of such realistic models, we consider indenting the drill head into a block of metal. Figure 7 shows both the obstacle as well as the indentation and the distribution of plastified cells. Figure 8 shows a close-up of the complex structure of the vertical displacement (left) as well the adaptive mesh used to resolve it (right).

5. CONCLUSIONS AND OUTLOOK

In this paper, we have investigated a set of interconnected methods to solve complex elastoplastic contact problems. Our focus was on developing algorithms for mesh generation, iteration of the plastic and contact nonlinearities, and solution of the resulting linear systems that in each component provide almost optimal complexity and can scale to problems that require hundreds of millions of unknowns and more than a thousand cores. In the numerical results shown in Section 4, we have demonstrated the performance and accuracy of these methods as well as investigated their limits. We have also provided highly accurate results for a benchmark problem for comparison by others. The implementation of our methods is available under an open source license as the step-42 tutorial program of the DEAL.II finite element library.

Despite the accuracy and efficiency demonstrated in the results shown earlier, improvements continue to be possible. In particular, issues that could be addressed are the lack of scalability of the setup phase of the algebraic multigrid (e.g., by reusing the preconditioner from previous Newton steps), possible improvements in how solutions are transferred from one mesh to the next (e.g., by projections that use not only the displacement but also the stresses), or using higher-order polynomials or hp-adaptive refinement methods. Furthermore, more realistic applications require incorporating thermal effects and friction at the contact surface (e.g., [26, 27, 29, 30, 55–57]).

ACKNOWLEDGEMENTS

Parts of the work of the first author were supported by the Deutsche Forschungsgemeinschaft (DFG) within Priority Program 1480, ‘Modelling, Simulation and Compensation of Thermal Effects for Complex Machining Processes’, through grant number BL 256/11-3.

DOI: 10.1002/nme

The third author is supported by the National Science Foundation through award no. OCI-1148116. The second and third authors are supported in part by the Computational Infrastructure in Geodynamics initiative (CIG), through the National Science Foundation under award no. EAR-0949446 and The University of California—Davis. This publication is based in part on the work supported by award no. KUS-C1-016-04 made by King Abdullah University of Science and Technology (KAUST).

Some computations for this paper were performed on the Brazos and Hurr clusters at the Institute for Applied Mathematics and Computational Science (IAMCS) at Texas A&M University. Part of Brazos was supported by NSF award DMS-0922866. Hurr is supported by award no. KUS-C1-016-04 made by King Abdullah University of Science and Technology (KAUST).

REFERENCES

1. Stiemer M, Unger J, Svendsen B, Blum H. Algorithmic formulation and numerical implementation of coupled electromagnetic-inelastic continuum models for electormagnetic metal forming. *International Journal for Numerical Methods in Engineering* 2006; **68**(13):1301–1328.
2. Marusich TD, Usui S, Ma J, Stephenson DA, Shih A. Finite element modeling of drilling processes with solid and indexable tooling in metals and stack-ups. *Proceedings of 10th CIRP International Workshop on Modeling of Machining Operations*, Reggio Calabria, Italy, 2007; 7.
3. Valberg HS. *Applied Metal Forming: Including Fem Analysis*. Cambrige University Press: New York, 2010.
4. Frohne J. FEM-simulation der umformtechnik metallischer oberflächen im mikrokosmos. *Ph.D. Thesis*, Univeersity Siegen, Verlag Dr. Hut, München, 2011.
5. Lange K. *Handbook of Metal Forming*. McGraw-Hill Book Company, 1985.
6. Lemaitre J. *A Course on Damage Mechanics*. Springer-Verlag: Berlin Heidelberg, 1996.
7. Lemaitre J, Desmorat R. *Engineering Damage Mechanics*. Springer-Verlag: Berlin Heidelberg, 2005.
8. Poliakov ANB, Buck WR. Mechanics of stretching elastic-plastic-viscous layers: applications to slow-spreading mid-ocean ridges. In *Faulting and Magmatism at Mid-ocean Ridges*, Buck WR, Delaney PT, Karson JA, Lagabrielle Y (eds)., Geophys. Monogr. Ser. 106. American Geophysical Union: Washington, DC, 1998; 305–325.
9. Poliakov ANB, Herrmann HJ. Self-organized criticality in plastic shear bands. *Geophysical Research Letters* 1994; **21**:2143–2146.
10. Lavier LL, Buck WR, Poliakov ABN. Factors controlling normal fault offset in an ideal brittle layer. *Journal of Geophysical Research* 2000; **23**:23431–23442.
11. Lal R, Shukla MK. *Principles of Soil Physics*. CRC Press: New York, 2004.
12. Bangerth W, Hartmann R, Kanschat G. deal.II—a general purpose object oriented finite element library. *ACM Transactions on Mathematical Software* 2007; **33**(4):Article No. 24, 27 pages.
13. Bangerth W, Heister T, Heltai L, Kanschat G, Kronbichler M, Maier M, Turcksin B, Young TD. The `deal.ii` library, version 8.1. *arXiv preprint*, 2013. (Available from: http://arxiv.org/abs/1312.2266v4) [accessed on 25 July 2015].
14. Heroux MA, Bartlett RA, Howle VE, Hoekstra RJ, Hu JJ, Kolda TG, Lehoucq RB, Long KR, Pawlowski RP, Phipps ET, Salinger AG, Thornquist HK, Tuminaro RS, Willenbring JM, Williams A, Stanley KS. An overview of the Trilinos project. *ACM Transactions on Mathematical Software* 2005; **31**:397–423.
15. Heroux MA et al. Trilinos web page, 2013. (Available from: http://trilinos.sandia.gov) [accessed on 25 July 2015].
16. Burstedde C, Wilcox LC, Ghattas O. `p4est`: scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM Journal of Scientific Computing* 2011; **33**(3):1103–1133.
17. Simo JC, Hughes TJR. *Computational Inelasticity*. Springer-Verlag: New York, 1998.
18. Alberty J, Carstensen C, Zarrabi D. Adaptive numerical analysis in primal elastoplasticity with hardening. *Computer Methods in Applied Mechanics and Engineering* 1999; **3**(171):175–204.
19. Geiger C, Kanzow C. *Theorie Und Numerik Restringierter Optimierungsaufgaben*. Springer-Verlag: Berlin Heidelberg, 2002.
20. Stein E. *Error-controlled Adaptive Finite Elements in Solid Mechanics*. John Wiley & Sons Ltd.: Chichester, 2003.
21. Gruber PG, Valdman J. Solution of one-time-step problems in elastoplasticity by a slant Newton method. *SIAM Journal of Scientific Computing* 2009; **31**:1558–1580.
22. Hager C, Wohlmuth B. Nonlinear complementarity functions for plasticity problems with frictional contact. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**:3411–3427.
23. Sauter M, Wieners C. On the superlinear convergence in computational elasto-plasticity. *Computer Methods in Applied Mechanics and Engineering* 2011; **200**:3646–3658.
24. Cermak M, Kozubek T, Sysala S, Valdman J. A TFETI domain decomposition solver for elastoplastic problems. *Mathematics and Computers* 2014; **231**:634–653.
25. Sysala S. Application of a modified semismooth Newton method to some elasto-plastic problems. *Mathematics and Computers in Simulation* 2012; **82**:2004–2021.
26. Wriggers P. *Computational Contact Mechanics*. Springer-Verlag: Berlin Heidelberg, 2006.
27. Kikuchi N, Oden JT. *Contact Problems in Elasticity: A Study of Variational Inequalities and Finite Element Methods*. SIAM Studies in Applied Mathematics 8. SIAM: Philadelphia, 1988.
28. Haslinger J, Hlaváček I, Nečas J. Numerical methods for unilateral problems in solid mechanics. In *Handbook of Numerical Analysis*, Ciarlet P, Lions JL (eds). vol. 4. Elsevier: Amsterdam, 1996; 313–485.

DOI: 10.1002/nme

29. Schröder A. Fehlerkontrollierte adaptive h- und hp-finite-elemente-methoden für kontaktprobleme mit anwendungen in der fertigungstechnik. *Ph.D. Thesis*, University Dortmund, 2006.
30. Wohlmuth BI. Variationally consistent discretization schemes and numerical algorithms for contact problems. *Acta Numerica* 2011; **20**:569–734.
31. Brunssen S, Schmid F, Schäfer M, Wohlmuth BI. A fast and robust iterative solver for nonlinear contact problems using a primal–dual active set strategy and algebraic multigrid. *International Journal for Numerical Methods in Engineering* 2007; **69**:524–543.
32. Hüeber S, Wohlmuth BI. A primal–dual active set strategy for non-linear multibody contact problems. *Computer Methods in Applied Mechanics and Engineering* 2005; **194**:3147–3166.
33. Dickopf T, Krause R. Efficient simulation of multi-body contact problems on complex geometries: a flexible decomposition approach using constrained minimization. *International Journal for Numerical Methods in Engineering* 2009; **77**:1834–1862.
34. Dostál Z, Vondrák V, Horák D, Farhat C, Avery P. *Scalable FETI Algorithms for Frictionless Contact Problems*, Vol. 60. Springer: Berlin Heidelberg, 2008; 263–270.
35. Dostál Z, Kozubek T, Vondrák V, Brzobohaty T, Maropoulos A. Scalable TFETI algorithm for the solution of multibody contact problems of elasticity. *International Journal for Numerical Methods in Engineering* 2010; **82**: 1384–1405.
36. Blum H, Braess D, Suttmeier FT. A cascadic multigrid algorithm for variational inequalities. *Computing and Visualization in Science* 2004; **7**:153–157.
37. Popp A, WGee M, Wall WA. Finite deformation contact based on a 3D dual mortar and semi-smooth Newton approach 2011; **58**:57–77.
38. Bangerth W, Burstedde C, Heister T, Kronbichler M. Algorithms and data structures for massively parallel generic adaptive finite element codes. *ACM Transactions on Mathematical Software* 2011; **38**(2):Article No. 14, 28 pages.
39. Kronbichler M, Heister T, Bangerth W. High accuracy mantle convection simulation through modern numerical methods. *Geophysics Journal International* 2012; **191**:12–29.
40. White J, Borja RI. Block-preconditioned Newton–Krylov solvers for fully coupled flow and geomechanics. *Computers and Geosciences* 2011; **15**:647–659.
41. Brigdman PW. *Studies in Large Plastic Flow and Fracture*, Metallurgy and Metallurgical Engineering Series. McGraw-Hill Book Company: New York, 1952; 118–124.
42. Johnson C. On plasticity with hardening. *Journal of Mathematical Analysis and Applications* 1978; **62**(2):325–336.
43. Grossmann C, Roos HG. *Numerical Treatment of Partial Differential Equations*. Springer-Verlag: Berlin Heidelberg, 2007.
44. Suttmeier FT. On plasticity with hardening: an adaptive finite element discretisation. *International Mathematical Forum* 2010; **5**(52):2591–2601.
45. Rodrigues JF. *Obstacle Problems in Mathematical Physics*. North-Holland: Amsterdam, 1987.
46. Hintermüller M, Ito K, Kunisch K. The primal–dual active set strategy as a semismooth Newton method. *SIAM Journal on Optimization* 2003; **13**(3):865–888.
47. Rannacher R, Suttmeier FT. A posteriori error control and mesh adaptation for finite element models in elasticity and elasto-plasticity. *Computer Methods in Applied Mechanics and Engineering* 1999; **21**(2):333–361.
48. De SR, Gago JP, Kelly DW, Zienkiewicz OC, Babuška I. A posteriori error analysis and adaptive processes in the finite element method: part II—adaptive mesh refinement. *International Journal for Numerical Methods in Engineering* 1983; **19**:1621–1656.
49. Wohlmuth B. An a posteriori error estimator for two-body contact problems on non-matching meshes. *Journal of Scientific Computing* 2007; **33**:25–45.
50. Klein N. Consistent fe-analysis of elliptic variational inequalities. *Ph.D. Thesis*, University Siegen, 2013.
51. Glowinski R, Lions JL, Trémolières R. *Numerical Analysis of Variational Inequalities*. North-Holland, 1981.
52. Nocedal J, Wright SJ. *Numerical Optimization*, Springer Series in Operations Research. Springer: New York, 1999.
53. Hertz H. Über die Berührung fester elastischer Körper. *Journal Fur Die Reine Und Angewandte Mathematik* 1882; **92**:156–171.
54. Gee MW, Siefert CM, Hu JJ, Tuminaro RS, Sala MG. ML 5.0 Smoothed Aggregation User's Guide. *Technical Report 2006-2649*, Sandia National Laboratories, 2006.
55. Stadler G. Infinite-dimensional semi-smooth Newton and augmented Lagrangian methods for friction and contact problems in elasticity. *Ph.D. Thesis*, University of Graz, 2004.
56. Kunisch K, Stadler G. Generalized Newton methods for the 2D-Signorini contact problem with friction in function space. *M2AN Mathematical Modeling and Numerical Analysis* 2005; **39**:827–854.
57. Hüeber S, Stadler G, Wohlmuth BI. A primal–dual active set algorithm for three-dimensional contact problems with coulomb friction. *SIAM Journal of Scientific Computing* 2008; **30**:572–596.

SUPPORTING INFORMATION

Additional supporting information may be found in the online version of this article.

DOI: 10.1002/nme