

EFFICIENT NUMERICAL METHODS FOR THE LARGE-SCALE, PARALLEL SOLUTION OF ELASTOPLASTIC CONTACT PROBLEMS

JÖRG FROHNE*, TIMO HEISTER†, AND WOLFGANG BANGERTH‡

Abstract. Elastoplastic contact problems with hardening are ubiquitous in industrial metal forming processes as well as many other areas. From a mathematical perspective, they are characterized by the difficulties of variational inequalities for both the plastic behavior as well as the contact problem. Computationally, they also often lead to very large problems. In this paper, we present and evaluate a set of methods that allows us to efficiently solve 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 benchmark problem and scaling our methods to 1,024 cores and more than a billion unknowns.

Key words. Elastoplasticity, contact problems, adaptive finite element methods, active set method, parallel computing

AMS subject classifications. 65K15, 65N22, 65N30, 65N50, 65Y05, 65Y15

1. Introduction. Elastoplasticity is a phenomenon that refers to the fact that materials can 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 [9, 34, 44, 47], and their control is important to achieve material shapes close to the desired ones. On the other hand, elastoplasticity is also important when it happens inadvertently, for example in the investigation of the long term deformation of machine components under external loads [30, 32, 33], in the case of geophysical deformation processes such as plate deformation on long time scales [31, 36, 37], or the response of soil to nearby buildings, dams or earthquakes [29].

Given this importance in both the control of industrial processes as well as 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:

- The realistic simulation of almost all purposeful industrial forming processes as well as the investigation of geophysical problems requires full 3d simulations. This immediately yields problems that have millions of unknowns, or even many more than that. Consequently, efficient computational methods are indispensable for any realistic use of computational tools.

*Institute of Applied Mathematics, Technische Universität Dortmund, 44221 Dortmund, Germany (joerg.frohne@math.tu-dortmund.de) and Department Mathematik, Universität Siegen, 57258 Siegen, Germany (frohne@mathematik.uni-siegen.de). Part of this work was done while being a long-term visitor at the Department of Mathematics, Texas A&M University, College Station, TX 77843-3368, USA.

†Mathematical Sciences, O-110 Martin Hall, Clemson University, Clemson, SC 29634-095, USA (heister@clemson.edu).

‡Department of Mathematics, Texas A&M University, College Station, TX 77843-3368, USA (bangerth@math.tamu.edu).

- 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 non-linear solvers.
- Most practical forming methods – rolling, drilling, machining, cutting – are in fact *contact problems* in which the desired deformation is affected by a tool that exerts external forces to the surface to the body under deformation. Where these two bodies are in contact is a priori unknown and is only a result of the deformation of one 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 address these difficulties and apply them to metal forming 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:

- *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 can not efficiently be done using global mesh refinement. Optimal algorithms therefore must be able to use 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. The locations to be resolved with a finer mesh are not known a-priori but must be determined automatically.
- *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 (see 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 on every processor will not be able to scale adequately.

The point of this paper is not to introduce any of these methods in isolation, but to show that they all interact well and form a combination that is capable of solving complex, realistic problems. 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.

The implementation of these algorithms is available under an open source license as the step-42 tutorial program¹ of the DEAL.II finite element library [3, 4]. It uses the TRILINOS library [18, 19] for its parallel linear algebra and P4EST [8] for parallel mesh handling.

Literature overview. We are here considering a unilateral contact problem with a rigid obstacle impinging on a deformable, three dimensional body. The material of the body is modeled by an elasto-plastic 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 [1, 12, 15, 16, 39, 41, 43]. To make these methods more robust, one needs to globalize them, for example using a line search for a damping parameter [9, 39, 46] and we will describe such a method in Section 3.5. Another strategy to choose the damping parameter can be found in [46] and the references therein.

For the contact problem, many methods have been proposed. An overview can be found in the book [51] and in [7, 17, 23, 25, 40, 50] which also discuss the primal-dual active set algorithms we will use below. In contrast, references [5, 9] 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 very efficient [7, 16]. We will therefore follow this approach here as well.

As mentioned, our goal is to solve very large problems on adaptively refined meshes. 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 (see, for example, [2, 27, 48]). 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 model problem. In this section, let us give an overview of the mathematical formulation of the problem. In the following, 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) where the inequality

¹At the time of writing, the documentation of this program is not yet complete. However, we commit to finishing it before publication should this paper be accepted. The current state of the program can be found at http://www.dealii.org/developer/doxygen/deal.II/step_42.html.

We will make the programs (all variants of step-42) used in the computations shown in Section 4 available as supplementary material to this article.

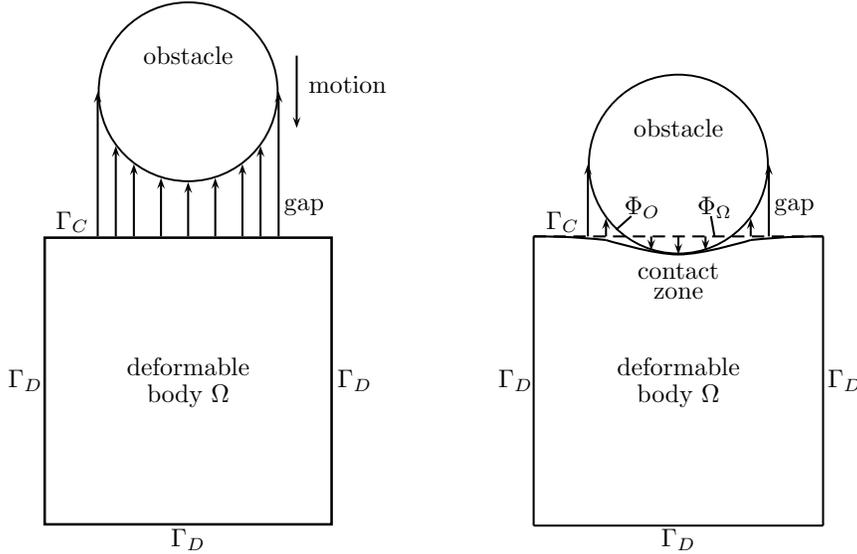


FIG. 2.1. *Left: Starting configuration. Right: Deformed body. We consider only static deformations and therefore solve for the static deformation corresponding to the right picture.*

describing the contact is on the displacements and the inequality describing plasticity is on the stress. We will then show a variational formulation corresponding to this problem. All algorithms discussed in the next sections are based on this variational (weak) formulation, and it turns out that they can be most efficiently formulated if they only involve the displacement variable as this leads to a positive definite operator. Consequently, we will eliminate the stress and use 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.

A sketch of the situation we have in mind is shown in Fig. 2.1. 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 $\Phi_O(\mathbf{x})$ and by $\Phi_\Omega(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x})$ the position of the *deformed* object where \mathbf{u} is the deformation field. For small displacements, the contact (non-penetration) condition then requires that $\Phi_O(\mathbf{x}) \cdot \mathbf{n} \geq \Phi_\Omega(\mathbf{x}) \cdot \mathbf{n}$ where $\mathbf{n} = \mathbf{n}(\mathbf{x})$ is the outward normal to Γ_C at \mathbf{x} . We rewrite this condition as $g(\mathbf{x}) - \mathbf{u}(\mathbf{x}) \cdot \mathbf{n} \geq 0$ where $g = (\Phi_O(\mathbf{x}) - \Phi_\Omega(\mathbf{x})) \cdot \mathbf{n}$ 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 [25].

2.1. Classical formulation. Elastoplastic contact problems are typically formulated as partial differential equations with multiple inequalities due to the plastic behavior of the medium and the contact condition. In particular, plasticity is commonly described in the following way: While the medium is linear elastic for small displacements, it is only capable of carrying internal stresses $\sigma(\mathbf{x})$ limited by some

maximal stress, i.e., that satisfies an inequality $\mathcal{F}(\sigma) \leq 0$ at every point \mathbf{x} . The simplest example of such a function \mathcal{F} is the von Mises flow function $\mathcal{F}(\sigma) = |\sigma^D| - \sigma_0$ where σ_0 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_0 , the material ceases to be elastic and plastic yielding occurs.

Given this setup, the classical formulation of the problem we will consider here reads as follows (see, for example, [9]): 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

$$\varepsilon(\mathbf{u}) = A\sigma + \varepsilon^p \quad \text{in } \Omega, \quad (2.1)$$

$$-\text{div } \sigma = \mathbf{f} \quad \text{in } \Omega, \quad (2.2)$$

$$\varepsilon^p : (\tau - \sigma) \geq 0 \quad \forall \tau \text{ with } \mathcal{F}(\tau) \leq 0 \quad \text{in } \Omega, \quad (2.3)$$

$$\mathbf{u} = 0 \quad \text{on } \Gamma_D, \quad (2.4)$$

$$\sigma \cdot \mathbf{n} - [\mathbf{n} \cdot (\sigma \cdot \mathbf{n})]\mathbf{n} = 0, \quad \mathbf{n} \cdot (\sigma \cdot \mathbf{n}) \leq 0 \quad \text{on } \Gamma_C, \quad (2.5)$$

$$(\mathbf{n} \cdot (\sigma \cdot \mathbf{n}))(\mathbf{n} \cdot \mathbf{u} - g) = 0, \quad \mathbf{n} \cdot \mathbf{u} - g \leq 0 \quad \text{on } \Gamma_C. \quad (2.6)$$

Here, (2.1) defines the relationship between strain $\varepsilon(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T)$ 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 (2.2) is the force balance; we will here not consider any body forces and henceforth assume that $\mathbf{f} = 0$. The complementarity condition (2.3) implies that $\varepsilon^p = 0$ if $\mathcal{F}(\sigma) < 0$ but that ε^p may be a nonzero tensor if and only if $\mathcal{F}(\sigma) = 0$. Equation (2.4) describes a fixed, zero displacement on Γ_D and (2.5) prescribes that on the surface $\Gamma_C = \partial\Omega \setminus \Gamma_D$ where contact may appear, the normal force $\sigma_n = \mathbf{n} \cdot (\sigma \cdot \mathbf{n})$ exerted by the obstacle is inward (no “pull” by the obstacle on our body) and with zero tangential component $\sigma_t = \sigma \cdot \mathbf{n} - \sigma_n \mathbf{n} = \sigma \cdot \mathbf{n} - [\mathbf{n} \cdot (\sigma \cdot \mathbf{n})]\mathbf{n}$. The last condition, (2.6), 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. As a consequence, their material properties also change as a result of plasticity. For metals, which we here consider as the case of interest, plastic deformation typically results in a stiffening. This “hardening” effect is described by letting the yield stress depend on the norm of the plastic strain, $|\varepsilon^p|$.² We will thus replace the simple von Mises flow function introduced above by the simplest generalization, namely linear isotropic hardening: $\mathcal{F}^{\text{iso}}(\sigma, |\varepsilon^p|) = |\sigma^D| - (\sigma_0 + \gamma^{\text{iso}}|\varepsilon^p|)$ 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

²For 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 suggests that hydrostatic pressure causes no plastic flow [6] and should thus not be included in the definition of the yield stress.

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 above and then, in the next section, reformulate it as a nonlinear primal one.

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

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

With these spaces, our problem is defined by the following primal-mixed variational formulation (see [24]): *Find* $\{(\sigma, \xi), \mathbf{u}\} \in \Pi(W \times L^2(\Omega, \mathbb{R})) \times V^+$ *so that*

$$(A\sigma - \varepsilon(\mathbf{u}), \tau - \sigma) + \gamma^{\text{iso}}(\xi, \eta - \xi) \geq 0, \quad \forall (\tau, \eta) \in \Pi(W \times L^2(\Omega, \mathbb{R})) \quad (2.7)$$

$$(\sigma, \varepsilon(\varphi) - \varepsilon(\mathbf{u})) \geq 0, \quad \forall \varphi \in V^+. \quad (2.8)$$

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 [24] for more details.

2.3. Reformulating plasticity as a nonlinear equality. The mixed formulation (2.7)–(2.8) above, 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 (see [9, 14, 45]). For the case of isotropic materials with isotropic linear hardening, it is defined as

$$P_{\Pi}(\tau) := \begin{cases} \tau, & \text{if } |\tau^D| \leq \sigma_0, \\ \left[\frac{\gamma^{\text{iso}}}{2\mu + \gamma^{\text{iso}}} + \left(1 - \frac{\gamma^{\text{iso}}}{2\mu + \gamma^{\text{iso}}} \right) \frac{\sigma_0}{|\tau^D|} \right] \tau^D + \frac{1}{3} \text{trace}(\tau) I, & \text{if } |\tau^D| > \sigma_0, \end{cases} \quad (2.9)$$

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*

$$(P_{\Pi}(C\varepsilon(\mathbf{u})), \varepsilon(\varphi) - \varepsilon(\mathbf{u})) \geq 0, \quad \forall \varphi \in V^+. \quad (2.10)$$

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 [38].

We could in principle reformulate the problem again to also include the contact inequality as a (semi-smooth) nonlinearity [21]. We will in fact use this approach below 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 already, and we will defer the discussion until we have introduced the discrete problem.

3. Numerical methods. Our basic approach to solving (2.10) consists of two nested loops that, in pseudo-code can be represented as follow and that solve the problem on a sequence of finite element meshes:

```

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;

```

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.

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 subsections below. 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 above, most of the Newton iterations happen on coarser meshes where they are, comparatively, very cheap. 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 an approach that varies a penalty parameter may have to choose this parameter dependent on the mesh size, potentially increasing the number of Newton iterations on the finest mesh or increasing the cost of each iteration as a result of worse conditioning.
- Finally, we will show below 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 Conjugate Gradients preconditioned by Algebraic Multigrid (AMG) methods have previously been shown to work well even for very large problems (see, for example, [27, 48]). 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 below.

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

3.1. Newton linearization of the plastic behavior. As discussed above, the first step in our algorithm is to apply a Newton method to compute a better approximation to the solution of the primal formulation (2.10). Formally, (2.10) is not differentiable. However, it satisfies the conditions of slant differentiability [21] 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 [41].

In Newton's method, we seek an updated solution $\mathbf{u}^i = \mathbf{u}^{i-1} + \alpha^i \delta \mathbf{u}^i \in V^+$. Since 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

$$(I_{\Pi} \varepsilon(\tilde{\mathbf{u}}^i), \varepsilon(\varphi) - \varepsilon(\tilde{\mathbf{u}}^i)) \geq 0, \quad \forall \varphi \in V^+. \quad (3.1)$$

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

$$I_{\Pi} = \begin{cases} C_{\mu} + C_{\kappa}, & \text{if } |C \varepsilon^D(\mathbf{u}^{i-1})| \leq \sigma_0, \\ \frac{\gamma^{\text{iso}}}{2\mu + \gamma^{\text{iso}}} C_{\mu} + \frac{(1 - \frac{\gamma^{\text{iso}}}{2\mu + \gamma^{\text{iso}}}) \sigma_0}{|C \varepsilon^D(\mathbf{u}^{i-1})|} \left(C_{\mu} - 2\mu \frac{C \varepsilon^D(\mathbf{u}^{i-1}) \otimes C \varepsilon^D(\mathbf{u}^{i-1})}{|C \varepsilon^D(\mathbf{u}^{i-1})|^2} \right) + C_{\kappa}, & \text{else.} \end{cases} \quad (3.2)$$

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 (2.9). For the linear isotropic material we consider here, the bulk and shear components of the projector are given by

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

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

From this, we see that problem (3.1) that needs to be solved in each nonlinear step is, in essence, a linear elastic contact problem with spatially variable elasticity coefficients. The very first Newton iteration in fact 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 at which the deformable object is in contact with the obstacle and forces 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 (3.1) 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 (see Section 4.1), after which the mesh is refined using a simple smoothness indicator (the widely used Kelly error estimator [10]). More sophisticated strategies for mesh refinement are certainly possible here and would likely yield further savings in computational complexity [26, 43, 49, 50].

The discrete problem corresponding to (3.1) asks for a function $\tilde{\mathbf{u}}_h^i \in V_h^{+,i}$ so that

$$(I_{\Pi}\varepsilon(\tilde{\mathbf{u}}_h^i), \varepsilon(\varphi_h) - \varepsilon(\tilde{\mathbf{u}}_h^i)) \geq 0, \quad \forall \varphi_h \in V_h^{+,i} \quad (3.3)$$

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

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

Here, Q_p are tensor product polynomials up to degree p on the reference element,³ F_K is the mapping from the reference element to the element K , \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.

REMARK 1. *Error estimation, hierarchic mesh refinement, and the construction of finite element spaces on such meshes are all techniques that have been well tested on problems up to very large machine sizes and have been shown to scale almost perfectly (see, for example, [2]). Consequently, they qualify for our goal of using only methods that work well even to large problem sizes.*

3.3. Reformulation as an equality constrained problem via the primal-dual active set method. Problem (3.3) is still an inequality constrained, albeit finite dimensional, problem that can not 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 (3.3) 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 $\Gamma_C^A \subset \Gamma_C$ (the *active* contact surface), then we could

³We consider hexahedral meshes, but the algorithm also works on tetrahedral meshes when using the space of polynomials \mathbb{P}_p on the reference tetrahedron.

find the solution of (3.3) by instead searching for a $\tilde{\mathbf{u}}_h^i$ satisfying

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

For numerical stability, it is best to understand the equality constraint in the sense $\langle \mathbf{n} \cdot \tilde{\mathbf{u}}_h^i - g, \mathbf{n}\mu \rangle_{\Gamma_C^A, h} = 0$ for all $\mu \in V_h$. Here, $\langle f, g \rangle_{\Gamma_C^A} = \int_{\Gamma_C^A} f g \, dx$, and $\langle f, g \rangle_{\Gamma_C^A, h}$ approximates $\langle f, g \rangle_{\Gamma_C^A}$ using quadrature that includes only quadrature points that are also nodal points of the finite element space.⁴ In practice, this means computing $\langle f, g \rangle_{\Gamma_C^A, h}$ via Gauss-Lobatto quadrature, using the fact that the shape functions for \mathbb{Q}_p elements are defined at the Gauss-Lobatto points.

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

$$(I_\Pi \varepsilon(\tilde{\mathbf{u}}_h^i), \varepsilon(\varphi)) + \langle \mathbf{n} \cdot \lambda_h, \mathbf{n}\varphi_h \rangle_{\Gamma_C^A, h} = 0 \quad \forall \varphi \in V_h^i, \quad (3.4)$$

$$\langle \mathbf{n} \cdot \tilde{\mathbf{u}}_h^i - g, \mathbf{n} \cdot \mu_h \rangle_{\Gamma_C^A, h} = 0 \quad \forall \mu_h \in V_h^i. \quad (3.5)$$

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 [13].

The problem above can be written in matrix-vector form as

$$\begin{pmatrix} A(U^{i-1}) & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \tilde{U}^i \\ \Lambda \end{pmatrix} = \begin{pmatrix} 0 \\ G \end{pmatrix}. \quad (3.6)$$

Using the inner product defined by quadrature as described above 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 $|\mathcal{S}^i| = \dim(V_h^i)$). 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 [35] 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:

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

Primal-dual active set methods, see [21, 23], improve on this by instead using the criterion

$$\mathcal{A}^i := \left\{ p \in \mathcal{S}_C^i : [-A(U^{i-1})U^{i-1} + c(\bar{B}^T U^{i-1} - G)]_p > 0 \right\}. \quad (3.7)$$

⁴An 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 [23]. The description given here leads to the exact same algebraic system and is, consequently, equivalent.

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, $G_p = \langle g, \mathbf{n} \cdot \varphi_p \rangle_{\Gamma_{C,h}}$. Numerical experiments suggest that $c = 100E = 300\kappa(1 - 2\nu)$ is a good choice.

Since 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

$$\mathcal{A}^i := \left\{ p \in \mathcal{S}_C^i : [R(\mathbf{u}^{i-1})_p + c(\bar{B}^T U^{i-1} - G)]_p > 0 \right\} \quad (3.8)$$

instead of (3.7), where the nonlinear residual is defined as

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

in accordance with (2.10).

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 which 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.*

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 3d, 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 since 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 (3.6) 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: Since we have determined an estimate \mathcal{A}^i of the set of active constraints and computed B from it, the second line of (3.6) simply reads

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

This corresponds to a Dirichlet boundary condition for the normal displacements at a subset of nodes. Consequently, the displacement part of the solution of (3.6) is given by solving $\hat{A}(U^{i-1})\tilde{U}^i = \hat{H}$ where

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

In other words, we simply remove constrained rows from the original linear system. Since B is non-zero only in these rows, the term $B\Lambda$ completely disappears, rendering

Λ a quantity that no longer appears in our computation. As already hinted at above, 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

$$\hat{A}(U^{i-1})\tilde{U}^i = \hat{H}(U^{i-1}) \quad (3.9)$$

where

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

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

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

It is not difficult to show that $\hat{A}(U^{i-1})$ 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 the Conjugate Gradient (CG) solver with an algebraic multigrid (AMG) 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 CG iteration. As above, 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 so also in the current context 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 [35] to find the first step length $\alpha^i \in \{1, 2^{-1}, 2^{-2}, \dots\}$ so that

$$\left\| \hat{R}(\mathbf{u}_\alpha) \right\|_{\ell_2} < \left\| \hat{R}(\mathbf{u}_h^{i-1}) \right\|_{\ell_2}, \quad (3.10)$$

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 (i) elements $p \in \mathcal{A}^i$ where we set $\hat{R}(\mathbf{u}_\alpha)_p = 0$, and (ii) elements that correspond to hanging nodes, which we eliminate in the usual manner.

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

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 $R(\mathbf{u}_h^{i-1})$ and the matrix \bar{B} (or take the latter from the previous iteration, if the mesh hasn't changed).
2. Compute the active set \mathcal{A}^i .
3. Assemble the matrix \hat{A} and right hand side \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.

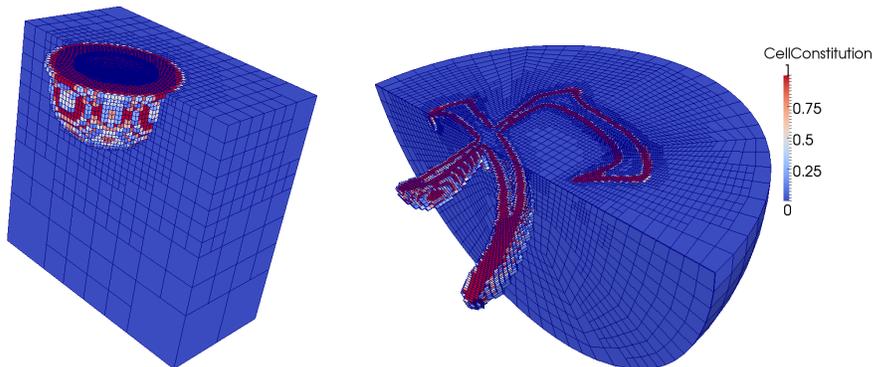


FIG. 4.1. *Adaptively refined mesh (cut away in the left half of the domain for all cells that are exclusively elastic) and fraction of quadrature points in each cell that are plastified (blue: none, red: all). Left: Pressing a sphere into a cube, corresponding to the benchmark discussed in Section 4.2. Right: Pressing the Chinese symbol for “force” into a half sphere.*

4. Solve linear system (3.9).
5. Find a step length using the line search procedure (3.10).

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 realistic complexity. In Section 4.2 we then illustrate the accuracy of our solutions by defining a benchmark problem. Subsections 4.3 and 4.4 evaluate the parallel scalability of our methods up to 1,024 processor cores and more than a billion unknowns, and compare our algorithm with one previously described in the literature. We conclude our numerical results by showing a case of an obstacle with a realistic geometry used in metal drilling in Section 4.5.

Our implementation of the algorithms discussed above is available as the step-42 tutorial program of the open source finite element library DEAL.II [3,4] and all computations below are based on variants of it. We use TRILINOS version 11.0.3 for parallel linear algebra [18,19] and P4EST 0.3.4 to distribute meshes among processors [8].

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 4.1; 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 4.1. As will become clear from the discussion below, 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 4.1 and 4.2 summarize the number of cells and unknowns on each of these meshes, along

TABLE 4.1

Performance characteristics of our algorithms when pressing a sphere into a cube. DoFs = degrees of freedom. It.s = iterations. 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.

Mesh	# cells	# DoFs	# Newton it.s	avg. # linear it.s
0	512	2,187	6 / 6	3 / 3
1	1,548	6,294	5 / 8	6 / 6
2	4,768	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

TABLE 4.2

Performance characteristics of our algorithms when pressing a Chinese symbol into a half-sphere. All columns as in Table 4.1.

Mesh	# cells	# DoFs	# Newton it.s	avg. # linear it.s
0	384	1,443	5 / 5	1 / 1
1	1,189	4,770	13 / 6	5 / 5
2	3,695	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

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 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 done 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.

The results of these tables show that for the case of the spherical obstacle (Table 4.1), 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

TABLE 4.3
Description of the benchmark discussed in Section 4.2.

Domain	Cube $[0, 1\text{m}]^3$
Obstacle	Sphere $\mathbf{x}_{\text{center}} = (0.5\text{m}, 0.5\text{m}, 1.59\text{m})$, $r = 0.6\text{m}$
Contact surface	$\Gamma_C = \{\mathbf{x} \in \Omega : x_3 = 1\}$
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)
Evaluation point	$\sigma_0 = 400$ MPa, $\gamma^{\text{iso}} = 0.01$ $P = (0.5001, 0.5001, 0.9501)^T$ (in the reference configuration)

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 4.2) 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 due to the fact that for this rather irregular obstacle, the first meshes used are simply too coarse (the mesh shown in Fig. 4.1 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. On the other hand, 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 application cases with complex geometries. There, the full performance of an algorithm may only become apparent when applying it to cases where the goal is not just to solve something but 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. Since 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 full description of all the parameters corresponding to the picture shown in the left panel of Fig. 4.1 is given in Table 4.3.⁵

⁵We 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 below 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 4.2.

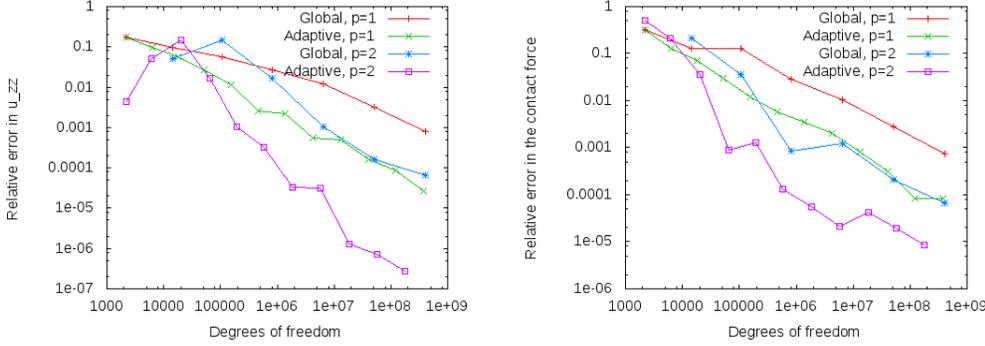


FIG. 4.2. Convergence of $u_z(P)$ (left) and the contact force $\int_{\Gamma_C} \lambda_z$ for the benchmark problem discussed in Section 4.2. Convergence is measured against the extrapolated best guesses from Table 4.4.

To evaluate accuracy we consider three measures:

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

For P (see Table 4.3), 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 4.4 shows our numerical results for both uniform and adaptive mesh refinement with linear ($p = 1$) and quadratic finite elements ($p = 2$). Due to 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 -0.01m 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 (so called Hertzian contact [20]). 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 and we show in Fig. 4.2 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 per cent 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 less unknowns: in the case of Q_2 elements, just a few hundred thousand. Secondly, we can not determine the stresses σ_{xx}, σ_{zz} to better than a few per cent, 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 pre-

TABLE 4.4

Computed values of displacements, stresses and integrated forces for global and adaptive mesh refinement when pressing a sphere into a cube (see Section 4.2). The last row contains extrapolated values; digits in parentheses are probably correct but with a low degree of uncertainty.

Mesh	# cells	# DoFs	$u_z(P)$	$\sigma_{xx}(P)$	$\sigma_{zz}(P)$	$\int_{\Gamma_C} \lambda_z$
Global mesh refinement with $p = 1$						
0	512	2,187	-0.0075681	-5733.1	-6098.2	37.306
1	4,096	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	4,096	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	2,187	-0.0075681	-5733.1	-6098.2	37.306
1	1,548	6,294	-0.0070687	-3317.7	-3855.7	62.323
2	4,768	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	461,106	-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	2,187	-0.0064262	-5625.5	-6050.7	26.980
1	176	6,087	-0.0061346	-27.7	-605.6	66.647
2	652	20,397	-0.0074270	-376.4	-1085.8	57.130
3	2,192	64,731	-0.0065623	-766.3	-1450.1	55.229
4	6,980	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)

vious section or the drill bit discussed in Section 4.5 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 done in parallel. Consequently, we will consider the parallel scalability of our methods in the next section.

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 since 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 one per cent, 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 8-core Opteron 2350 processor (2.5 GHz clock rate), 32GB of memory per node, and an Infiniband interconnect.

To evaluate our algorithms, Fig. 4.3 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 8 cores). We here solve once on a uniformly refined mesh first and timings are then shown for a once adaptively refined mesh to include the effects of adaptivity. We show results for multiples of 8 cores to avoid the effect of partially empty nodes. As seen in the left panel of Fig. 4.3, we get 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 [2, 27] 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 due to the fact that 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 million times the number of cores that participate in the computation. This is achieved by starting from either a $3 \times 3 \times 3$, $4 \times 4 \times 4$, or $5 \times 5 \times 5$ coarse mesh, refining it globally a number of times, and then refining it adaptively once in such a way that we achieve the desired number of unknowns. Similar to before, the timings are then done for the last, adaptive step. The coarse mesh sizes are chosen in such a way that the last step can be achieved by refining approximately 10% of cells.

The results shown in the right panel of Fig. 4.3 again show that our algorithms scale well, even to very large problems, losing only approximately a factor of 3 in overall efficiency when increasing the number of processor cores from 8 to 1024. The figure shows that the primary obstacles to 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 4.1 and 4.2, 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

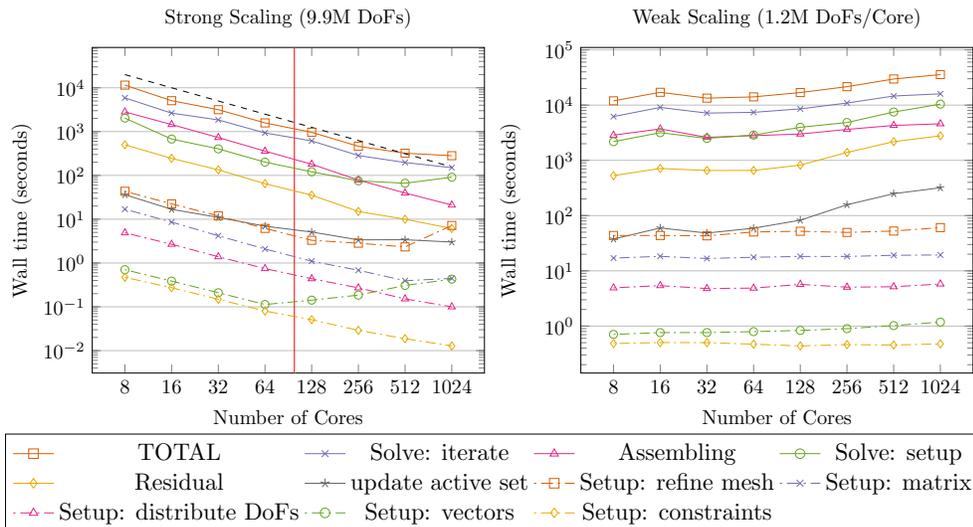


FIG. 4.3. Results for strong (left) and weak (right) scaling as discussed in Section 4.3. The red line in the left panel corresponds to 100,000 degrees of freedom per core beyond which scalability deteriorates. In the weak scaling case, the slowdown is due to parallel inefficiencies as well as to a larger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

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 component 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; it may be possible to optimize it, 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 hours. Given that during this time we perform 17 Newton iterations, this corresponds to 35 minutes 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 minutes per linear solve (including assembly, setting up preconditioners and postprocessing) when using approximately 100,000 unknowns per core for non-trivial problems of this size [2, 27, 48].

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 [11] 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:

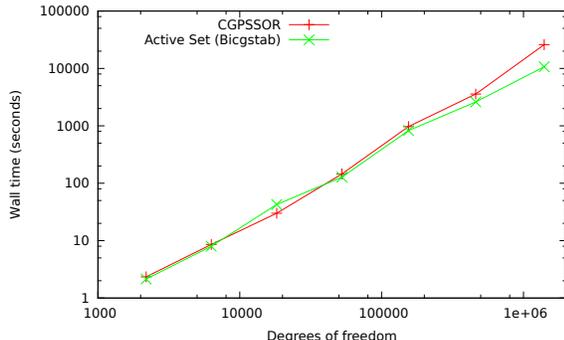


FIG. 4.4. Comparison of the overall computation time for the active set strategy and the CG-PSSOR method for a computation on a single processor.

The CG-based projection method CGPSSOR, see [5]. 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 set of contact nodes. A detailed description of this method can be found in [5, 9]. 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. Fig. 4.4 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 of a sequence of meshes and includes the time to assemble and solve the linear and nonlinear systems, but excluding the time for postprocessing, 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. The figure does not show numbers beyond 1.4 million unknowns because run times on a single processor become unreasonably large.

The figure does not show parallel results since CGPSSOR can not win there: While we have shown in the previous section that our methods provides scalability to large numbers of processors, SSOR is an inherently sequential algorithm. It can of course be parallelized by running the preconditioner only on those diagonal blocks of the matrix that each processor stores locally, but it is well known that this deteriorates the quality of a preconditioner and increases the number of linear iterations, unlike the algebraic multigrid preconditioner demonstrated above.

The conclusion of this section is that for rather small problems, our proposed set of algorithms is as fast as one that has been described in the literature recently, but

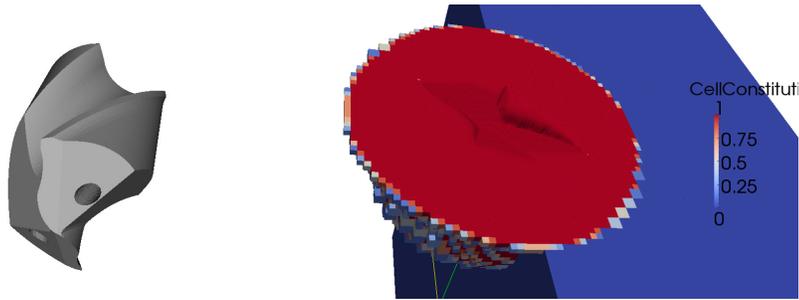


FIG. 4.5. *Left: Drill head with two cooling canals. Right: Distribution of plastified cells and elastoplastic deformation.*

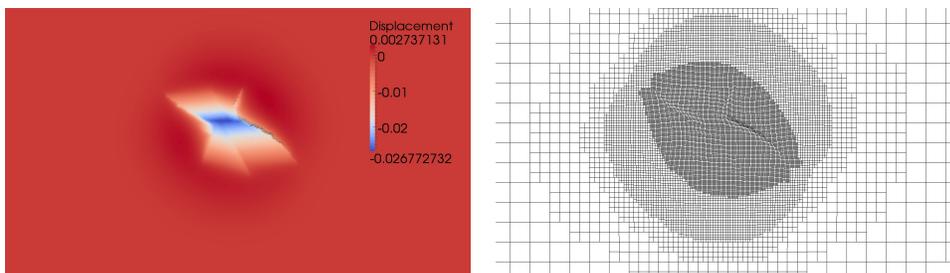


FIG. 4.6. *Left: Displacements in z -direction. Right: Adaptive refined mesh on the contact surface.*

that for large problems it is superior. Furthermore, unlike the previous algorithm, our work here scales well in parallel.

4.5. 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 (DFG) that supports the first author. The geometry of this drill head, shown in Fig. 4.5, 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 can not be modeled taking into account only continuous deformations. Furthermore, heat development and the temperature dependence of material parameters play an important role in drilling – both effects that are not taken into account here.

As a first step in the direction of such realistic models, we consider indenting the drill head into a block of metal. Fig. 4.5 shows both the obstacle as well as the indentation and the distribution of plastified cells. Fig. 4.6 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 to develop 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 results shown above, further improvements to the algorithm described here are certainly possible, and will be explored in the future. In particular, the following points warrant further work:

- A commonly acknowledged bottleneck for algebraic multigrid implementations is the scalability of the setup phase. This is obvious from Fig. 4.3, but has also been observed in [2, 27] and elsewhere. In the current context, it is possible that one can avoid this setup phase most of the time by observing that the Hessian matrix in one Newton step is likely similar to that of the previous and that an AMG preconditioner built in the previous step is then likely also a good preconditioner for the current one. However, how often one needs to rebuild it without unduly affecting the performance of the CG iteration involves a trade-off that isn't immediately obvious. Numerical experiments will be able to assess this question.
- As shown in Section 4.1, the number of Newton iterations required on one mesh depends sensitively on how good an approximation we get as a starting point by interpolating the solution from the previous mesh. In our work, we simply interpolate the displacement field, but better strategies are conceivable. In particular, we conjecture that a better starting point could be obtained by not only interpolating the displacement but also the stress or strain fields, and using these either to obtain a better displacement field or to determine the first linearized operator I_{Π} on the new mesh that describes the linearization of plasticity.
- One should be able to further improve the computational efficiency by using higher order polynomials or *hp*-adaptive refinement methods, coupled with better refinement indicators. This results from the observation that at least in the elastic region, we can expect the solution to be very smooth and therefore amenable to higher order finite element approximation.

Further extensions to the methods discussed here are motivated by more realistic applications. In particular, most metal forming processes are in fact time dependent and often also involve thermal effects. This requires not only solving a heat equation and coupling the temperature into the material model, but also keeping track of the plastic behavior through history variables. Secondly, while we have considered frictionless contact, realistic processes display friction, oftentimes with a limit tangential stress above which slip occurs. One would include such behavior into our model by another inequality, to be solved with a separate set of active constraints. Methods to treat friction can be found, for example, in [22, 25, 28, 40, 42, 50, 51]. In particular, the methods introduced in [22] provide a promising approach to extend our work.

Finally, for more sophisticated constitutive laws one has to iterate out nonlinear versions of the hardening process. To do so a small local nonlinear equation system has to be solved in each quadrature point, for example using a local Newton iteration. The difficulty is that this vastly increases the cost of plasticized quadrature points over elastic ones, posing a challenge to parallel scaling and requiring load-balancing of meshes even in the absence of mesh refinement. We will address these challenges

in future work.

Acknowledgments. 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 numbers BI 498/24-2, BL 256/11-2, and SU 245/6-1.

The third author is supported by the National Science Foundation through Award No. OCI-1148116. The second and third authors are supported in part through 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 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] J. ALBERTY, C. CARSTENSEN, AND D. ZARRABI, *Adaptive numerical analysis in primal elastoplasticity with hardening*, Comput. Meth. Appl. Mech. Engrg., (1999), pp. 175 – 204.
- [2] W. BANGERTH, C. BURSTEDDE, T. HEISTER, AND M. KRONBICHLER, *Algorithms and data structures for massively parallel generic adaptive finite element codes*, ACM Trans. Math. Softw., 38 (2011).
- [3] W. BANGERTH, R. HARTMANN, AND G. KANSCHAT, *deal.II – a general purpose object oriented finite element library*, ACM Trans. Math. Softw., 33 (2007).
- [4] W. BANGERTH, T. HEISTER, AND G. KANSCHAT, *deal.II Differential Equations Analysis Library, Technical Reference*, 2013. <http://www.dealii.org/>.
- [5] H. BLUM, D. BRAESS, AND F. T. SUTTMEIER, *A cascadic multigrid algorithm for variational inequalities*, Computing and Visualization in Science, (2004), pp. 153 – 157.
- [6] P. W. BRIGDMAN, *Studies in large plastic flow and fracture*, Metallurgy and Metallurgical Engineering Series, (1952).
- [7] S. BRUNSEN, F. SCHMIDT, M. SCHÄFER, AND B. I. WOHLMUTH, *A fast and robust iterative solver for nonlinear contact problems using a primal-dual active set strategy and algebraic multigrid*, Int. J. Numer. Meth. Engrg., 69 (2007), pp. 524 – 543.
- [8] C. BURSTEDDE, L. C. WILCOX, AND O. GHATTAS, *p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees*, SIAM J. Sci. Comput., 33 (2011), pp. 1103–1133.
- [9] J. FROHNE, *FEM-Simulation der Umformtechnik metallischer Oberflächen im Mikrokosmos*, Ph.D. thesis University Siegen, Verlag Dr. Hut, München, 2011.
- [10] J. P. DE S. R. GAGO, D. W. KELLY, O. C. ZIENKIEWICZ, AND I. BABUŠKA, *A posteriori error analysis and adaptive processes in the finite element method: Part II — Adaptive mesh refinement*, Int. J. Num. Meth. Engrg., 19 (1983), pp. 1621–1656.
- [11] M. W. GEE, C. M. SIEFERT, J. J. HU, R. S. TUMINARO, AND M. G. SALA, *ML 5.0 Smoothed Aggregation User’s Guide*, Tech. Report 2006-2649, Sandia National Laboratories, 2006.
- [12] C. GEIGER AND C. KANZOW, *Theorie und Numerik restringierter Optimierungsaufgaben*, Springer-Verlag Berlin Heidelberg, 2002.
- [13] R. GLOWINSKI, J.L. LIONS, AND R. TRÉMOLIÈRES, *Numerical Analysis of Variational Inequalities*, North-Holland, 1981.
- [14] C. GROSSMANN AND H.-G. ROOS, *Numerical Treatment of Partial Differential Equations*, Springer-Verlag Berlin Heidelberg, 2007.
- [15] P. G. GRUBER AND J. VALDMAN, *Solution of one-time-step problems in elastoplasticity by a slant newton method*, SIAM J. Sci. Comput., 31 (2009), pp. 1558 – 1580.
- [16] C. HAGER AND B. WOHLMUTH, *Nonlinear complementarity functions for plasticity problems with frictional contact*, Comput. Meth. Appl. Mech. Engrg., 198 (2009), pp. 3411 – 3427.
- [17] J. HASLINGER, I. HLAVÁČEK, AND J. NEČAS, *Numerical methods for unilateral problems in solid mechanics*, In: Handbook of Numerical Analysis (P. Ciarlet and J.-L. Lions, eds), 4 (1996), pp. 313–485.

- [18] M. A. HEROUX, R. A. BARTLETT, V. E. HOWLE, R. J. HOEKSTRA, J. J. HU, T. G. KOLDA, R. B. LEHOUCQ, K. R. LONG, R. P. PAWLOWSKI, E. T. PHIPPS, A. G. SALINGER, H. K. THORNQUIST, R. S. TUMINARO, J. M. WILLENBRING, A. WILLIAMS, AND K. S. STANLEY, *An overview of the Trilinos project*, ACM Trans. Math. Softw., 31 (2005), pp. 397–423.
- [19] M. A. HEROUX ET AL., *Trilinos web page*, 2013. <http://trilinos.sandia.gov>.
- [20] H. HERTZ, *Über die Berührung fester elastischer Körper*, J. Reine Angew. Math., (1882), pp. 156–171.
- [21] M. HINTERMÜLLER, K. ITO, AND K. KUNISCH, *The primal-dual active set strategy as a semi-smooth Newton method*, SIAM J. Optim., 13 (2003), pp. 865 – 888.
- [22] S. HÜEBER, G. STADLER, AND B. I. WOHLMUTH, *A primal-dual active set algorithm for three-dimensional contact problems with coulomb friction*, SIAM J. Sci. Comput., 30 (2008), pp. 572 – 596.
- [23] S. HÜEBER AND B. I. WOHLMUTH, *A primaldual active set strategy for non-linear multibody contact problems*, Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 3147 – 3166.
- [24] C. JOHNSON, *On plasticity with hardening*, J. Math. Anal. Appl., (1978).
- [25] N. KIKUCHI AND J. T. ODEN, *Contact Problems in Elasticity: A Study of Variational Inequalities and Finite Element Methods*, SIAM, 1988.
- [26] N. KLEIN, *Consistent FE-Analysis of elliptic Variational Inequalities*, Ph.D. thesis University Siegen, 2013.
- [27] M. KRONBICHLER, T. HEISTER, AND W. BANGERTH, *High accuracy mantle convection simulation through modern numerical methods*, Geophysics Journal International, 191 (2012), pp. 12–29.
- [28] K. KUNISCH AND G. STADLER, *Generalized newton methods for the 2d-signorini contact problem with friction in function space*, M2AN Math. Model. Numer. Anal., 39 (2005), pp. 827 – 854.
- [29] R. LAL AND M. K. SHUKLA, *Principles of Soil Physics*, CRC Press, 2004.
- [30] K. LANGE, *Handbook of Metal Forming*, McGraw-Hill Book Company, 1985.
- [31] L. L. LAVIER, W. R. BUCK, AND A. B. N. POLIAKOV, *Factors controlling normal fault offset in an ideal brittle layer*, J. Geoph. Res, 23 (2000), pp. 23431–23442.
- [32] J. LEMAITRE, *A Course on Damage Mechanics*, Springer, 1996.
- [33] J. LEMAITRE AND R. DESMORAT, *Engineering Damage Mechanics*, Springer, 2005.
- [34] T. D. MARUSICH, S. USUI, J. MA, D. A. STEPHENSON, AND A. SHIH, *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).
- [35] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, Springer Series in Operations Research, Springer, New York, 1999.
- [36] A. N. B. POLIAKOV AND W. R. BUCK, *Mechanics of stretching elastic-plastic-viscous layers: Applications to slow-spreading mid-ocean ridges*, in Faulting and Magmatism at Mid-Ocean Ridges, W. R. Buck, P. T. Delaney, J. A. Karson, and Y. Lagabriele, eds., Geophys. Monogr. Ser. 106, American Geophysical Union, Washington, DC, 1998, pp. 305–325.
- [37] A. N. B. POLIAKOV AND H. J. HERRMANN, *Self-organized criticality in plastic shear bands*, Geoph. Res. Lett., 21 (1994), pp. 2143–2146.
- [38] J.-F. RODRIGUES, *Obstacle Problems in Mathematical Physics*, North-Holland, Amsterdam, 1987.
- [39] M. SAUTER AND C. WIENERS, *On the superlinear convergence in computational elasto-plasticity*, Comput. Meth. Appl. Mech. Engrg., 200 (2011), pp. 3646 – 3658.
- [40] A. SCHRÖDER, *Fehlerkontrollierte adaptive h- und hp-Finite-Elemente-Methoden für Kontaktprobleme mit Anwendungen in der Fertigungstechnik*, Ph.D. thesis University Dortmund, 2006.
- [41] J. C. SIMO AND T. J. R. HUGHES, *Computational Inelasticity*, Springer, 1998.
- [42] G. STADLER, *Innite-dimensional semi-smooth newton and augmented lagrangian methods for friction and contact problems in elasticity*, Ph.D. thesis, University of Graz, 2004.
- [43] E. STEIN, *Error-Controlled Adaptive Finite Elements in Solid Mechanics*, John Wiley & Sons Ltd., 2003.
- [44] M. STIEMER, J. UNGER, B. SVENDSEN, AND H. BLUM, *Algorithmic formulation and numerical implementation of coupled electromagnetic-inelastic continuum models for electormagnetic metal forming*, Int. J. Numer. Meth. Eng., (2006), pp. 1301–1328.
- [45] F. T. SUTTMEIER, *On plasticity with hardening: An adaptive finite element discretisation*, International Mathematical Forum, 5 (2010), pp. 2591 – 2601.
- [46] S. SYSALA, *Application of a modified semismooth newton method to some elasto-plastic problems*, Mathematics and Computers in Simulation, (2012), pp. 2004 – 2021.

- [47] H. S. VALBERG, *Applied Metal Forming*, Cambridge University Press.
- [48] J. WHITE AND R. I. BORJA, *Block-preconditioned Newton-Krylov solvers for fully coupled flow and geomechanics*, *Comp. Geosc.*, 15 (2011), pp. 647–659.
- [49] B. WOHLMUTH, *An a posteriori error estimator for two-body contact problems on non-matching meshes*, *J. Sci. Comput.*, (2007), pp. 25–45.
- [50] B. I. WOHLMUTH, *Variationally consistent discretization schemes and numerical algorithms for contact problems*, *Acta Numerica*, (2011), pp. 569 – 734.
- [51] P. WRIGGERS, *Computational Contact Mechanics*, Springer, 2006.