

Solving Nonlinear Solid Mechanics Problems with the Jacobian-Free Newton Krylov Method

J. D. Hales¹, S. R. Novascone¹, R. L. Williamson¹
D. R. Gaston¹ and M. R. Tonks¹

Abstract: The equations governing solid mechanics are often solved via Newton's method. This approach can be problematic if the Jacobian determination, storage, or solution cost is high. These challenges are magnified for multiphysics applications. The Jacobian-free Newton-Krylov (JFNK) method avoids many of these difficulties through a finite difference approximation. A parallel, nonlinear solid mechanics and multiphysics application named BISON has been created that leverages JFNK. We overview JFNK, outline the capabilities of BISON, and demonstrate the effectiveness of JFNK for solid mechanics and multiphysics applications using a series of demonstration problems. We show that JFNK has distinct advantages in many cases.

Keywords: JFNK, multiphysics, solid mechanics, fully implicit, finite element, nonlinear solvers

1 Introduction

The standard numerical solution approach for implicit finite element nonlinear solid mechanics problems is Newton's method [Belytschko, Liu, and Moran (2000); Bathe (1982)]. This method updates a solution vector by solving a system of equations where the matrix is the Jacobian. For solid mechanics, this Jacobian matrix is commonly referred to as the stiffness matrix. At each iteration, the Jacobian matrix is reformed, requiring a new direct or iterative solve.

This approach has a well-known property that makes it attractive for solving systems of coupled nonlinear equations. If the estimate of the solution is close to the true solution, the method converges quadratically.

For a nonlinear system of coupled equations, Newton's method comes with some drawbacks as well. In order to achieve quadratic convergence, the exact Jacobian

¹ Idaho National Laboratory, Idaho Falls, ID, USA

must be available. For relatively simple systems, the Jacobian may be determined analytically, though it may be challenging. For more complicated systems, the difficulty increases until it is prohibitive to obtain the Jacobian. For example, if the calculation of the residual requires calling a third-party library, it may be impossible to determine the analytic Jacobian. Furthermore, forming the Jacobian may be expensive and require significant computer memory.

Other solution methods, such as Broyden's method [Broyden (1965)], may also be employed. Broyden's method may be started with the true Jacobian or with an approximation to it. This is attractive when the Jacobian is unavailable or expensive to form. Broyden's method, as well as other secant updating methods, quasi-Newton, and modified Newton methods have been developed to retain as much of the convergence characteristics of Newton's method as possible while reducing complexity and expense. For a discussion of linear and nonlinear solution methods, see [Kelley (1995)].

It is worth noting that approaches not associated with Newton's method can be robust and efficient enough to solve a variety of large, complex problems. The nonlinear conjugate gradient method [Fletcher and Reeves (1964)], though widely studied in the context of nonlinear optimization (see, e.g., [Hager and Zhang (2006)]), has not seen widespread use in nonlinear solid mechanics. Nevertheless, Sandia National Laboratories has written a series of successful three-dimensional, fully parallel, nonlinear solid mechanics packages that rely on nonlinear conjugate gradient solvers. See, for example, [SIERRA Solid Mechanics Team (2010)].

Another approach is to convert the nonlinear system of equations into a system of ordinary differential equations. The work of [Liu and Atluri (2008)] clearly outlines this approach and provides several examples of its use. A clear advantage of their work is that it does not rely on the derivative of the residual function. The method requires careful consideration of the time step size to prevent instability, and the convergence speed is parameter-dependent.

Nonlinear solvers remain an active area of research. Indeed, a wide variety of nonlinear solution techniques exist with advantages and disadvantages to each. One of the newer techniques is the Jacobian-free Newton Krylov (JFNK) method [Knoll and Keyes (2004)]. Although the Jacobian is not required, the true effect of the Jacobian is manifest in a quadratic convergence rate. For multiphysics analysis, JFNK has an attractive characteristic in that it is inherently a fully-coupled approach and does not rely upon fixed point iteration, operator splitting, or loose coupling. The method naturally accommodates multiple PDEs without the need to develop specialized elements that couple several unknowns.

The JFNK method has been applied to magnetohydrodynamics [Knoll, Mousseau,

Chacon, and Reisner (2005)], fluid dynamics [Geuzaine (2001); Banaś (2002); Knoll and Mousseau (2000)], groundwater modeling [Hammond, Valocchi, and Lichtner (2002)], ice sheet modeling [Lemieux, Price, Evans, Knoll, Sallinger, Holland, and Payne (2011)], multiscale modeling [Rahul (2011)], neutron transport [Roberts and Forget (2008)], and many other problems.

In this paper, we demonstrate the characteristics of the JFNK method as applied to solid mechanics problems and to multiphysics problems where solid mechanics is coupled to another PDE. Although it is generally difficult to compare nonlinear solution methods to one another [Matthies and Strang (1979)], we show that the JFNK method has distinct advantages over other solution techniques in many cases.

Note that no comparisons of CPU times between JFNK and other methods are made here. A given implementation of JFNK compared to an implementation of another method will likely perform better on some problems and worse on others. Much depends on preconditioning, problem size, the cost of computing the residual, the cost of computing the Jacobian, and other factors. While the computational expense of JFNK may be more or less than that of other methods for a given problem, the effort required to implement JFNK for that problem is a clear strength of the method.

We begin by reviewing JFNK, including the importance of preconditioning. We also discuss the expected convergence rate. In Section 3, solid mechanics equations, a set of constitutive laws, heat conduction, and gap heat conduction are reviewed. These equations are coupled together in our multiphysics examples and may in general be coupled with those from other PDEs (e.g., species diffusion). We overview four demonstration problems (general function minimization, a nonlinear spring, plasticity in a thick cylinder, and deflection of a clamped plate) in Section 4. Section 5 includes two multiphysics examples. The first is a simple problem coupling heat transfer across a gap with thermal expansion. We then present results for a complex nuclear fuel analysis problem (the performance of a light water reactor fuel rod), demonstrating the effectiveness of JFNK on large multiphysics problems. We give some details about BISON, the implicit multiphysics application used for these examples. We also discuss the parallel scalability of our approach. Finally, we conclude in Section 6 with a discussion of planned future work and possible extensions.

2 Jacobian-Free Newton Krylov

To solve a nonlinear system of equations, it is common to begin with a residual statement

$$\mathbf{f}(\mathbf{x}) = 0 \tag{1}$$

where \mathbf{f} is the residual with \mathbf{x} as the unknown solution. We write the Jacobian as

$$\mathbf{J}(\mathbf{x}) = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}. \quad (2)$$

Newton's method is then

$$\text{Compute } \mathbf{J}(\mathbf{x}_k), \mathbf{f}(\mathbf{x}_k) \quad (3)$$

$$\text{Solve } \mathbf{J}(\mathbf{x}_k)\mathbf{s} = -\mathbf{f}(\mathbf{x}_k) \text{ for } \mathbf{s} \quad (4)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s} \quad (5)$$

which is continued until the update is sufficiently small or some other criterion is met.

For any number of reasons, Newton's method may be problematic. For example, it may be difficult to obtain the analytic Jacobian. In a multiphysics setting, where there may be several variables and residual contributions, deriving all possible sensitivities may be prohibitive. Also, for large problems, the number of unknowns may invoke an unacceptable memory cost. Modified Newton or quasi-Newton algorithms attempt to remedy one or more of these concerns. See [Dennis, Jr. and Moré (1977)] for a review of these approaches.

Iterative solution methods may help alleviate the memory and solution time costs, particularly where parallel computing is needed. These methods, such as the conjugate gradient method (CG) [Hestenes and Stiefel (1952)] and the generalized minimum residual method (GMRES) [Saad and Schultz (1986)], do not require the factorization of a matrix or even its formation. Instead, they require only the action of a matrix on a vector. In the case of Newton's method, these methods require the action of the Jacobian on a vector, $\mathbf{J}\mathbf{v}$. This allows the effect of the matrix to be built up piece by piece, element by element, in a matrix-free form, if desired. See [Carey and Jiang (1986)] for a review of the matrix-free approach.

The JFNK method evaluates the action of the Jacobian through a finite difference approximation,

$$\mathbf{J}(\mathbf{x}_k)\mathbf{v} \approx \frac{\mathbf{f}(\mathbf{x}_k + \varepsilon\mathbf{v}) - \mathbf{f}(\mathbf{x}_k)}{\varepsilon}. \quad (6)$$

This is an attractive form since neither the full Jacobian nor its element-by-element contributions are required. Despite not requiring the analytic Jacobian, the effect of the full Jacobian is seen from the first iteration of the iterative solver, unlike modified Newton or quasi-Newton algorithms. Thus, with only GMRES or another iterative solver and a function that computes the residual, JFNK finds solutions to nonlinear coupled equations with the convergence rate of a traditional Newton algorithm.

The conjugate gradient method for solving Equation 4 with the finite difference approximation (Equation 6) appears below:

$$\mathbf{s}_0 = \mathbf{0}$$

$$\mathbf{r}_0 = -\mathbf{f}(\mathbf{x}_k)$$

$$\mathbf{v}_0 = \mathbf{r}_0$$

$$i = 0$$

Loop on i :

$$\mathbf{g}_i(\mathbf{x}_k) = \frac{\mathbf{f}(\mathbf{x}_k + \varepsilon \mathbf{v}_i) - \mathbf{f}(\mathbf{x}_k)}{\varepsilon} \quad (7)$$

$$\alpha_i = \frac{\mathbf{r}_i^T \mathbf{r}_i}{\mathbf{v}_i^T \mathbf{g}_i}$$

$$\mathbf{s}_{i+1} = \mathbf{s}_i + \alpha_i \mathbf{v}_i$$

$$\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathbf{g}_i$$

Exit if \mathbf{r}_{i+1} is below tolerance

$$\beta_i = \frac{\mathbf{r}_{i+1}^T \mathbf{r}_{i+1}}{\mathbf{r}_i^T \mathbf{r}_i}$$

$$\mathbf{v}_{i+1} = \mathbf{r}_{i+1} + \beta_i \mathbf{v}_i$$

$$i = i + 1$$

End loop

The only difference between a standard CG method and the JFNK approach is Equation 7. A standard CG algorithm for solving Equation 4 is recovered by replacing Equation 7 with $\mathbf{g}_i = \mathbf{J}\mathbf{v}_i$.

One might expect the finite difference approximation made in JFNK to degrade the convergence rate. However, for many problems, the error associated with the finite difference formula is so small that effectively quadratic convergence is in fact retained. Given that no analytical Jacobian is required, this is a very attractive combination.

The finite difference approach of JFNK is particularly useful in a multiphysics setting. With the finite difference approximation eliminating the need for the true Jacobian, only the residual contributions of new variables and/or new terms are needed. This makes numerical experimentation with different combinations of variables and terms in coupled PDEs straightforward.

Efficient solves using iterative methods require good preconditioners. The purpose of preconditioning is to decrease the condition number of the system being solved. Equivalently, preconditioning reduces the spread of eigenvalues in the system. In

JFNK, it is common to use right preconditioning,

$$\mathbf{J}(\mathbf{x}_k)\mathbf{M}^{-1}(\mathbf{M}\mathbf{s}) = -\mathbf{f}(\mathbf{x}_k) \quad (8)$$

where \mathbf{M} is the preconditioner or preconditioning process. In this form, the solution approach involves two steps. First, solve $\mathbf{J}(\mathbf{x}_k)\mathbf{M}^{-1}\mathbf{w} = -\mathbf{f}(\mathbf{x}_k)$ for \mathbf{w} . Then, compute $\mathbf{s} = \mathbf{M}^{-1}\mathbf{w}$. Note that if $\mathbf{M}^{-1} = \mathbf{J}^{-1}$ the iterative solve will converge in one iteration. However, computing \mathbf{J}^{-1} is equivalent to solving the original system and so is not advantageous. It is necessary, therefore, to choose a preconditioner that reflects the character of $\mathbf{J}(\mathbf{x}_k)$ in order to accelerate the iterative method but also one that is inexpensive to compute and apply. Simple approximations to sub-blocks of the Jacobian along the diagonal may suffice.

Using right preconditioning with the finite difference approximation in Equation 6 leads to

$$\mathbf{J}(\mathbf{x}_k)\mathbf{M}^{-1}\mathbf{v} \approx \frac{\mathbf{f}(\mathbf{x}_k + \varepsilon\mathbf{M}^{-1}\mathbf{v}) - \mathbf{f}(\mathbf{x}_k)}{\varepsilon}. \quad (9)$$

This is accomplished by computing $\mathbf{w} = \mathbf{M}^{-1}\mathbf{v}$ and then $\mathbf{J}(\mathbf{x}_k)\mathbf{w} \approx [\mathbf{f}(\mathbf{x}_k + \varepsilon\mathbf{w}) - \mathbf{f}(\mathbf{x}_k)]/\varepsilon$.

A host of preconditioning options are available. For example, see [Knoll and Keyes (2004)] for a discussion of JFNK with physics-based preconditioning for a stiff wave system. We explore the importance of preconditioning in more detail in Sections 4.3 and 5.5.

3 Fundamental Equations

In this section, we define the solid mechanics equations that will be used in most of the example problems and that have been implemented in our multiphysics code BISON. Further details concerning BISON are given in Section 5.2. We also review equations for heat transfer that will be used in the multiphysics examples.

3.1 Solid Mechanics Equations

The governing equations associated with quasistatic solid mechanics are

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = 0 \text{ in } \Omega, \quad (10)$$

$$\mathbf{u} = \bar{\mathbf{u}} \text{ on } \Gamma_{\bar{\mathbf{u}}}, \text{ and} \quad (11)$$

$$\mathbf{t} = \bar{\mathbf{t}} \text{ on } \Gamma_{\bar{\mathbf{t}}}, \quad (12)$$

where \mathbf{u} is displacement, $\boldsymbol{\sigma}$ is the Cauchy stress tensor, \mathbf{b} is the body force, $\bar{\mathbf{u}}$ is the prescribed displacements, and $\bar{\mathbf{t}}$ is the prescribed tractions. With an eye toward the

weak form, we follow the well-established pattern and multiply Equation 10 by a test function \mathbf{v}

$$-\mathbf{v} \cdot (\nabla \cdot \boldsymbol{\sigma} + \mathbf{b}) = 0. \quad (13)$$

We then integrate over the volume

$$-\int_{\Omega} \mathbf{v} \cdot (\nabla \cdot \boldsymbol{\sigma} + \mathbf{b}) \, d\Omega = 0. \quad (14)$$

Employing the divergence theorem, we obtain

$$\int_{\Omega} \nabla \mathbf{v} : \boldsymbol{\sigma} - \int_{\partial\Omega} \mathbf{v} \cdot \boldsymbol{\sigma} \mathbf{n} - \int_{\Omega} \mathbf{v} \cdot \mathbf{b} = 0. \quad (15)$$

Formally, the problem becomes

$$\begin{aligned} \text{Find } \mathbf{u} \in \mathcal{U} \exists \\ G(\mathbf{u}, \mathbf{v}) &= \int_{\Omega} \nabla \mathbf{v} : \boldsymbol{\sigma} - \int_{\partial\Omega} \mathbf{v} \cdot \boldsymbol{\sigma} \mathbf{n} \\ &\quad - \int_{\Omega} \mathbf{v} \cdot \mathbf{b} = 0 \quad \forall \mathbf{v} \in \mathcal{V} \end{aligned} \quad (16)$$

where

$$\begin{aligned} \mathcal{U} &= \{ \mathbf{u} \in H^1(\Omega) \mid \mathbf{u} = \bar{\mathbf{u}} \text{ on } \Gamma_{\bar{\mathbf{u}}} \} \\ \mathcal{V} &= \{ \mathbf{v} \in H_0^1(\Omega) \}. \end{aligned} \quad (17)$$

3.1.1 The finite element approximation

We proceed through the finite element approximation for completeness. For a Galerkin method, we approximate \mathbf{u} and \mathbf{v} for an element as

$$\mathbf{u} \approx \underline{\underline{N}} \underline{\underline{u}}, \quad (18)$$

$$\mathbf{v} \approx \underline{\underline{N}} \underline{\underline{v}}, \quad (19)$$

with $\underline{\underline{N}}$ as the shape function matrix and $\underline{\underline{u}}$ and $\underline{\underline{v}}$ as the discretized nodal displacement and virtual displacement vectors, respectively. Similarly, using a constitutive law we can write

$$\boldsymbol{\sigma} \approx \underline{\underline{C}} \underline{\underline{\boldsymbol{\varepsilon}}}, \quad \text{where } \underline{\underline{\boldsymbol{\varepsilon}}} = \underline{\underline{B}} \underline{\underline{u}} \quad (20)$$

with $\underline{\underline{B}}$ as the strain displacement matrix, which is comprised of derivatives of shape functions. The matrix $\underline{\underline{C}}$ is the material matrix (or, more generally, a material operator). The discretized version of \mathbf{b} is $\underline{\underline{b}}$, and the discretized version of $\bar{\mathbf{t}}$ is $\bar{\underline{\underline{t}}}$.

The resulting equation is

$$G(\underline{u}, \underline{v}) = \sum_{n=1}^{nelem} \left[\underline{v}^T \int_{\Omega_{elem}} \underline{B}^T \underline{C} \underline{B} \underline{u} \, d\Omega - \underline{v}^T \int_{\Omega_{elem}} \underline{N}^T \underline{b} \, d\Omega - \underline{v}^T \int_{\Gamma_{elem_t}} \underline{N}^T \underline{\bar{t}} \, d\Gamma_{elem_t} \right] = 0. \tag{21}$$

3.1.2 Element kinematics

For geometrically linear analysis, $\underline{\varepsilon}$ is defined as the discrete form of $1/2[\nabla \underline{u} + \nabla \underline{u}^T]$. Furthermore, with a linear elastic constitutive model, the stress is simply $\underline{C} \underline{\varepsilon}$. We now outline our approach for 3D nonlinear analysis. We follow the approach in [Rashid (1993)] and [Key (2011)].

We begin with a complete set of data for step n and seek the displacements and stresses at step $n + 1$. We first compute an incremental deformation gradient,

$$\hat{\mathbf{F}} = \frac{\partial x^{n+1}}{\partial x^n}. \tag{22}$$

With $\hat{\mathbf{F}}$, we next compute a strain increment that represents the rotation-free deformation from the configuration at n to the configuration at $n + 1$. Following [Rashid (1993)], we seek the stretching rate \mathbf{D} :

$$\mathbf{D} = \frac{1}{\Delta t} \log(\hat{\mathbf{U}}) \tag{23}$$

$$= \frac{1}{\Delta t} \log(\text{sqrt}(\hat{\mathbf{F}}^T \hat{\mathbf{F}})) \tag{24}$$

$$= \frac{1}{\Delta t} \log(\text{sqrt}(\hat{\mathbf{C}})). \tag{25}$$

Here, $\hat{\mathbf{U}}$ is the incremental stretch tensor, and $\hat{\mathbf{C}}$ is the incremental Green deformation tensor. Through a Taylor series expansion, this can be determined in a straightforward, efficient manner. \mathbf{D} is passed to the constitutive model as an input for computing σ at $n + 1$.

The next step is computing the incremental rotation, $\hat{\mathbf{R}}$ where $\hat{\mathbf{F}} = \hat{\mathbf{R}} \hat{\mathbf{U}}$. Like for \mathbf{D} , an efficient algorithm exists for computing $\hat{\mathbf{R}}$. It is also possible to compute these quantities using an eigenvalue/eigenvector routine.

With σ and $\hat{\mathbf{R}}$, we rotate the stress to the current configuration.

3.2 Constitutive laws

While other constitutive laws are available in BISON, we mention only the two used in the demonstration problems: elasticity and linear strain hardening plasticity.

3.2.1 Elastic

We use a hypoelastic formulation for elasticity. Specifically,

$$\sigma_{ij}^{n+1} = \sigma_{ij}^n + \Delta t \mathbf{C}_{ijkl} \mathbf{D}_{kl} \quad (26)$$

where \mathbf{C} is the elasticity tensor. For isotropic elasticity, this becomes

$$\sigma_{ij}^{n+1} = \sigma_{ij}^n + \Delta t (\delta_{ij} \lambda \mathbf{D}_{kk} + 2\mu \mathbf{D}_{ij}) \quad (27)$$

with λ as Lamé's first parameter and μ as the shear modulus. This stress update occurs in the configuration at n . Thus, as mentioned, as a final step the stress must be rotated to the configuration at $n + 1$.

3.2.2 Linear strain hardening plasticity

Strain hardening plasticity is calculated implicitly utilizing the radial return method. An excellent and detailed description of this procedure is given in [Dunne and Petrinic (2005)]. The radial return method consists of calculating an elastic trial stress by using a total strain increment

$$\sigma^{trial} = \sigma^n + \mathbf{C} \Delta \boldsymbol{\varepsilon} \quad (28)$$

where \mathbf{C} is the linear isotropic elasticity tensor, $\Delta \boldsymbol{\varepsilon}$ is the total strain increment tensor, and σ^n is the stress from the previous time step. A yield function is evaluated to determine if the trial stress exceeds a given yield stress. If yielding has occurred, a plastic strain increment is calculated via Newton's method (independent of the system-wide nonlinear solve, which may be based on JFNK, Newton, etc.). This plastic strain increment is subtracted from the total strain increment to give the elastic strain increment

$$\Delta \boldsymbol{\varepsilon}_e = \Delta \boldsymbol{\varepsilon} - \Delta \boldsymbol{\varepsilon}_p. \quad (29)$$

The elastic strain increment is used to calculate a new stress, returning the stress back to the yield surface

$$\Delta \boldsymbol{\sigma} = \mathbf{C} \Delta \boldsymbol{\varepsilon}_e, \quad (30)$$

$$\sigma^{n+1} = \sigma^n + \Delta\sigma. \quad (31)$$

The plastic strains are updated and the process is repeated as necessary

$$p^{n+1} = p^n + \Delta\varepsilon_p. \quad (32)$$

3.3 Heat Conduction

The heat conduction equation is

$$\rho C_p \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) - q = 0, \quad (33)$$

where T , ρ and C_p are the temperature, density and specific heat, respectively, k is the thermal conductivity of the material, and q is a volumetric heat generation rate.

When analyzing heat transfer between independent bodies, heat conduction across a gap must be addressed. In our work, gap conductance h in a gas-filled gap between two surfaces is described using the form proposed in [Ross and Stoute (1962)]:

$$h = \frac{k_g(T_g)}{L_g + 1.5(r_1 + r_2)} \quad (34)$$

where k_g is the thermal conductivity of the gap medium, T_g is the temperature in the gap, L_g is the gap length, and r_i is a distance associated with the roughness of the i side of the gap. Assuming the gap is helium-filled, the temperature dependent conductivity from [Ramirez, Stan, and Cristea (2006)] is

$$k_g(T_g) = 0.0468 + 3.81 \times 10^{-4} T_g - 6.79 \times 10^{-8} T_g^2. \quad (35)$$

4 Demonstration problems

In this section, we show results obtained using JFNK for four nonlinear problems. Each demonstrates that correct solutions are possible with JFNK. The first problem is a general minimization problem while the remainder are solid mechanics examples.

4.1 Rosenbrock function

The Rosenbrock function [Rosenbrock (1960)] is given as

$$f = (1 - x)^2 + 100(y - x^2)^2. \quad (36)$$

It is a commonly-used function for evaluating minimization algorithms. It is characterized by a curved valley with a global minimum at $(1, 1)$.

Starting at position $(-1, 1)$, we found the minimum of the Rosenbrock function with Newton's method and JFNK. A contour plot of the function and the iteration history of Newton's method is shown in Figure 1(a). Here we see that Newton's method (with a simple step limiting algorithm) follows the slight descent of the valley until it reaches the minimum. A plot of the function value at each iteration is shown for both methods in Figure 1(b). This figure shows the quadratic convergence of both methods. The JFNK method results in a slightly higher final solution for the same number of iterations. This is not unexpected but is so slight as to be negligible.

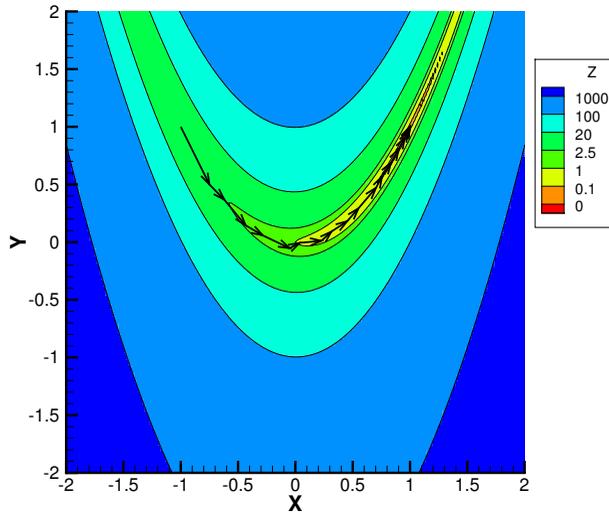
4.2 Nonlinear spring

As a first nonlinear solid mechanics example, we consider a pair of one-dimensional springs in series. The first, fixed at its left end, is a linear spring with a stiffness E . The second is a nonlinear spring with a stiffness $E(2.5 - 0.5 \tan^{-1}(a(u_2 - u_1 - b)))$. $E = 0.3$, $a = 50$, $b = 0.1$, and u_1 and u_2 are the displacements at the middle and right of the two-spring assembly. This choice for the stiffness function of the second spring was made since it results in a dramatic change in stiffness over a small range of displacement. Thus, the spring is strongly nonlinear.

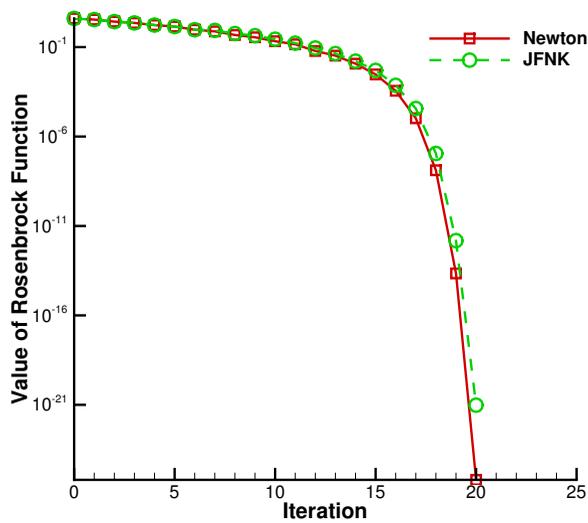
We vary the load at the free end from 0.05 to 0.1 in increments of 0.005. We solve this small nonlinear problem for each of those loads using a simple fixed point method, Newton's method, Broyden's method, nonlinear conjugate gradient [Fletcher and Reeves (1964)], and JFNK. The iterations required for each load are plotted in Figure 2. The dramatic increase in number of iterations around a load of 0.075 is due to the nonlinear behavior at that load.

The number of iterations required for each method at a given load should not be considered a meaningful measure of the time to solution for each method. Clearly, the work required for an iteration is different for each of the methods. In other words, this small problem is by no means useful in determining the relative efficiency of each method.

For this problem, all five methods are robust in finding a solution. At every load, all methods converged to a solution.



(a)



(b)

Figure 1: Results of minimization of the Rosenbrock function. Contour lines and iteration history for Newton's method are shown in (a). The function value at each iteration is shown for Newton's method and JFNK in (b). Note that both methods show the characteristic rapid convergence near the solution.

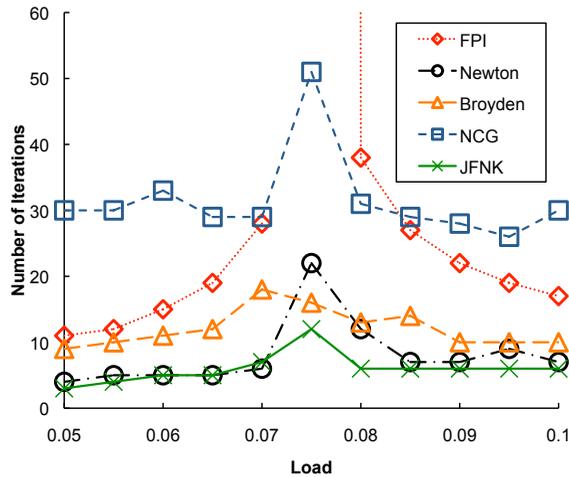


Figure 2: Number of iterations required by each nonlinear solver at each load level for the nonlinear spring problem. The problem is particularly nonlinear at a load of 0.075.

All five methods found the same solution at every load, with one exception. At a load of 0.075, two valid solutions were found. The fixed point iteration, Newton’s, nonlinear CG, and JFNK methods found a solution at $(0.25, 0.35)$. Broyden’s method resulted in a solution of $(0.25, 0.363279)$. Both solutions are valid. In other words, for every load case, every method found a valid solution, but they did not always find the same solution.

This problem illustrates one of the well-known difficulties associated with nonlinear analysis: multiple valid solutions may exist. If desired for problems with multiple solutions, additional techniques are required to direct the solver toward a preferred solution. This is true whether nonlinear CG, JFNK, or Newton’s method is employed.

4.3 Axisymmetric thick cylinder

This demonstration problem is taken from “4.6.2 NL2: Axisymmetric thick cylinder” in *Abaqus Benchmarks Manual* [aba (2009)]. The problem involves an elastic-perfectly plastic material and increasing internal pressure (80, 100, 120, 140, and 160 MPa). In the BISON JFNK solution, the nonlinear relative convergence tolerance for the L2-norm of the residual was 5×10^{-3} . The residual force criterion in Abaqus was also 5×10^{-3} .

Considering both the r - and θ -directions, the maximum error in the element average

stresses compared to Abaqus was 0.00087%.

The number of Newton iterations used by Abaqus to reach the solution at each load, as well as the number of JFNK iterations required by BISON, are in Table 1. Recall that JFNK uses an approximation to the Jacobian for preconditioning the linear solves but is Jacobian-free with respect to the overall nonlinear solution method. For this problem, an identical number of nonlinear iterations were required by JFNK in BISON compared to Newton's method in Abaqus. The advantage of the JFNK method in this case is the fact that a material tangent need not be derived, coded, and debugged.

Table 1: Number of iterations required to compute solution at each load.

	Internal Pressure (MPa)				
	80	100	120	140	160
Abaqus (Newton)	1	2	3	4	4
BISON (JFNK)	1	2	3	4	4

To demonstrate the effects of preconditioning, we converted this problem to a 3D form. The 3D solid model height is the same as in the axisymmetric case, but the solid model for the 3D case is a quarter annulus. The mesh had 24576 elements.

We ran the problem with the following preconditioners from the PETSc solver library [Balay, Buschelman, Eijkhout, Gropp, Kaushik, Knepley, McInnes, Smith, and Zhang (2004)]: AMG (algebraic multigrid (hypr BoomerAMG, [Falgout and Yang (2002)]), ASM (additive Schwarz method), ILU (incomplete LU factorization), LU (LU factorization), and NONE (no preconditioning). The preconditioning matrix supplied by BISON was a block-diagonal, elastic approximation to the Jacobian. The finite difference approximation of JFNK together with preconditioning followed the form in Equation 9.

The time to solution using each preconditioner is given in Table 2. Also shown is the number of nonlinear iterations required at each timestep. These runs were made with 8 processors except for the LU case, where a single processor was used.

The purpose here is not to present a definitive comparison of these preconditioners but instead to show that the choice of preconditioning matters. Indeed, no effort was taken to optimize the preconditioning settings. As expected, the preconditioner has a significant effect on the efficiency of the JFNK method. In fact, *whether* a solution is obtained, and not just how quickly it is obtained, may depend on the preconditioner chosen. This is seen by the fact that the run with the ASM preconditioner did not find a solution for the final load step.

Table 2: Time to solution for a set of preconditioners for the thick cylinder problem. The preconditioner has a significant impact on the time to solution as well as the number of nonlinear solves at each timestep.

Preconditioner	Time (s)	Nonl. Its. per Timestep
AMG	844	2, 4, 5, 6, 6
ASM	12688	6, 6, 11, 26, 50*
ILU	9253	2, 4, 6, 7, 37
LU	5123	2, 4, 5, 5, 6
NONE	10832**	14, 15, 24, 34, -

* Did not converge. ** Time for four solves.

4.4 Deflection of a Clamped Plate

Many authors have examined the problem of small displacements of thin plates (see, for example, [Timoshenko and Woinowski-Krieger (1959)]). The particular problem studied here is one of a clamped square plate with a uniform distributed load such that the displacements are relatively large.

The length of each side of the square plate is a , the plate thickness is t , and the distributed load is q . Young's modulus is E , and Poisson's ratio is ν . D is a plate stiffness:

$$D = \frac{Et^3}{12(1-\nu^2)}. \quad (37)$$

The deflection at the center of the plate is given by [Timoshenko and Woinowski-Krieger (1959)] as

$$w = \frac{0.00126qa^4}{D} \quad (38)$$

where w is the transverse displacement. This is valid for small displacements; i.e., when w is small compared to the plate thickness.

Substituting the values used in this study ($E = 1e7$, $\nu = 0.3$, $a = 1.0$, $t = 0.005$, $q = 2$), $w = 0.0220$. Since the displacement is considerably larger than the thickness, we would not expect the small displacement theory to accurately predict the displacement in this case.

We have chosen to use 8-node hexahedral elements for this study understanding that shell elements are much better suited for this problem. Our purpose is not to

demonstrate the qualities of particular finite elements but to show that the JFNK method inherently accommodates geometric nonlinearity.

As a first test, we ran a series of meshes to see that the results were converging to a solution with mesh refinement. Our meshes had 2 elements through the thickness and 20, 40, 60, 80, or 100 elements in each of the other two directions. Results from Abaqus and BISON are plotted in Figure 3(a).

As a second test, we ran a mesh with $100 \times 100 \times 2$ elements with varying magnitudes of the distributed load (1, 2, 3, 4, 5, 10, 15, 20, 25, and 30). Figure 3(b) shows that the geometric stiffness is engaged (the displacement is not a linear function of the load).

Thus, without requiring the geometric stiffness terms of the Jacobian, the JFNK method is able to compute solutions to geometrically nonlinear problems.

4.5 Summary of JFNK for solid mechanics

The four problems in this section demonstrate that JFNK is a reasonable solver for nonlinear solid mechanics problems. This is the case without the explicit construction of the stiffness matrix. In the next section, we demonstrate that JFNK is well-suited for highly coupled multiphysics analysis as well.

5 Coupled Multiphysics

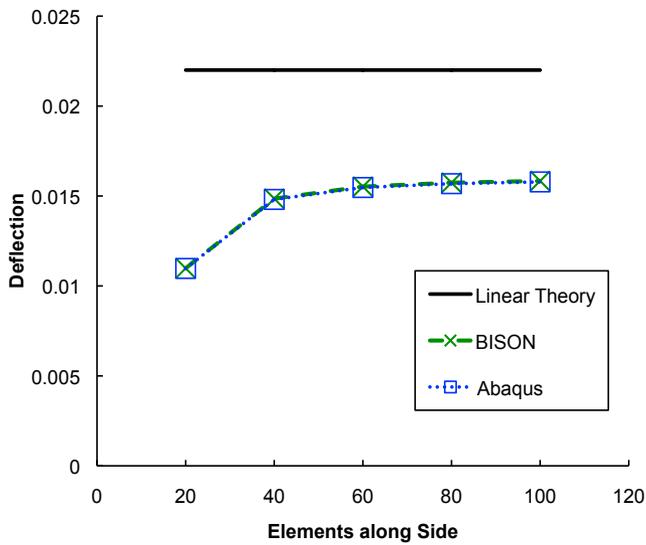
5.1 Coupled heat transfer across a gap

As a simple multiphysics example, we consider a one-dimensional, coupled solid mechanics and heat transfer problem where heat moves between a gap between two elements (Equations 33-35). The mesh consists of two unit-length elements designated AB and CD . A unit gap lies between them. The initial temperature of both elements is 100° . From time 0 to 1, the temperature at A (the left node of the left element) increases linearly to 200° and is held at 200° from then on. From time 1 to 2, the node at A moves to the right, closing the gap between the elements. The node at D (the right node of the right element) is fixed at 100° and zero displacement. The conductivity of the left element is 1, while the conductivity of the right element is 10.

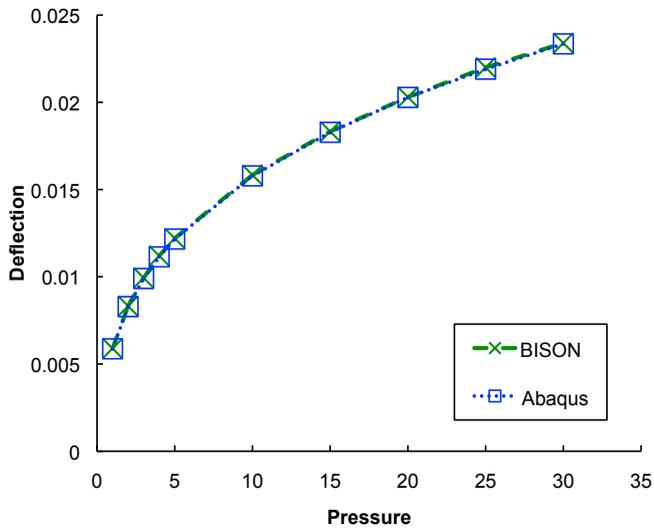
With the coefficient of thermal expansion set to zero for both elements, we displace the node at A to the right 1 unit. This results in a final gap length of 0.

We solved this simple multiphysics problem using BISON. The results, which have been verified, are given in Table 3.

As a second test, we set the coefficient of thermal expansion to 1×10^{-6} and 1×10^{-5} in the left and right elements, respectively. In this case, we displace node A



(a)



(b)

Figure 3: Deflection of square clamped plate with distributed load. Center plate displacement with increasing mesh density is shown in (a). Center plate displacement with increasing pressure is shown in (b) with 100x100x2 elements in the x, y, and z directions.

Table 3: Solution for gap heat transfer with a zero coefficient of thermal expansion.

Node	Time			
	1.0		2.0	
	Displacement	Temperature	Displacement	Temperature
A	0	200	1	200
B	0	190.9143	1	109.0937
C	0	100.9086	0	109.0906
D	0	100	0	100

0.9999 units. This results in a final gap length of approximately zero. In this test, the JFNK method must solve for the displacement of the interior nodes as well as their temperatures. The results are in Table 4. It is worthwhile to note that the final temperatures from the two analyses match.

The feature of this problem that makes it difficult is of course the expression for the conductance of the gap. The conductivity is a quadratic function of the temperature (though the coefficients are small), making the heat conduction nonlinear. More onerous is the inverse relationship to the gap size, coupling the mechanical response to the thermal. The presence of the roughness factors prevents the conductance from reaching infinity at a zero gap length. Still, the solution becomes very sensitive to small changes in displacement at the gap. This nonlinearity, along with mechanical contact constraints at the gap, makes the analysis of the interface behavior of systems such as nuclear fuel within cladding (see Section 5.3) a formidable challenge.

This multiphysics problem is small but highlights an important advantage of JFNK. The determination, programming, and debugging of the full Jacobian for the gap heat transfer is not required with JFNK. Specialized elements that couple variables from different physics are also not required. Despite this, the method retains the quadratic convergence rate of Newton’s method when the estimated solution is near the true solution.

5.2 BISON overview

In a typical light water reactor (LWR) nuclear fuel analysis, a stack of ceramic nuclear fuel pellets (typically UO₂) will heat up, thermally expand, densify, and swell according to specialized material models and power inputs. These pellets interact with metal cladding (typically a zirconium alloy), which serves to contain fission products and transfer heat from the pellets to the exterior coolant. The cladding is initially slightly larger than the fuel pellets in the radial direction, with a gas-filled gap between them. At the top portion of the cladding is a plenum, providing space

Table 4: Solution for gap heat transfer with a non-zero coefficient of thermal expansion. Note that the final temperatures from Table 3 match.

Node	Time			
	1.0		2.0	
	Displacement	Temperature	Displacement	Temperature
A	0	200	0.9999	200
B	9.5461×10^{-5}	190.9134	9.9995×10^{-1}	109.0937
C	-4.5433×10^{-6}	100.9087	-4.5453×10^{-5}	109.0906
D	0	100	0	100

for the gaseous fission products that will escape from the fuel pellets. See Figure 4.

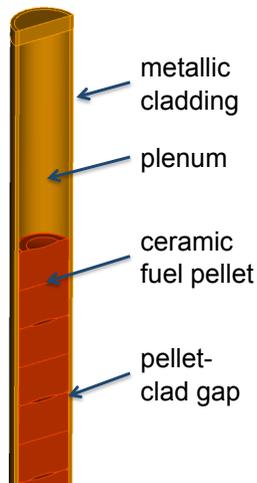


Figure 4: Schematic diagram of ceramic fuel pellets, cladding, gap, and plenum for the upper region of a typical UO_2 ceramic fuel rod.

Idaho National Laboratory is developing BISON, a three-dimensional, nonlinear, parallel multiphysics application based on JFNK and intended primarily for the analysis of nuclear fuel. For details on nuclear fuel performance analysis and typical equations used, see [Olander (1976)] and [Williamson (2011a)]. In an early LWR application, BISON was used to simulate thermomechanics and oxygen diffusion in a single fuel pellet [Newman, Hansen, and Gaston (2009)]. It was demonstrated that fully-coupled, three dimensional fuel performance solutions were quite

plausible and efficient using the JFNK method. BISON has also been applied to TRISO-coated fuel particles (the fuel form used in high temperature gas-cooled reactors) by investigating coupled heat transfer and fission product transport for a variety of fuel particle configurations [Williamson (2011b)]. For a detailed paper on BISON and fuel performance modeling, see [Williamson, Hales, Novascone, Tonks, Gaston, Permann, Andrs, and Martineau (2012)].

The solid mechanics capabilities of BISON include finite strain kinematics, nonlinear constitutive laws, and contact. BISON automatically couples to nonlinear heat conduction and can couple to other physics as well, such as meso-scale calculations that determine thermal conductivity [Tonks, Gaston, Permann, Millett, Hansen, and Wolf (2010)] and species diffusion models.

All of these capabilities are built upon MOOSE, *Multiphysics Object Oriented Simulation Environment* [Gaston, Newman, Hansen, and Lebrun-Grandié (2009)]. MOOSE leverages the advantages of the JFNK method for solving fully coupled nonlinear systems of equations and presents an interface for specifying the PDEs to be solved.

By incorporating a select set of third-party libraries, MOOSE supplies a fairly large set of common finite element software needs. MOOSE relies heavily on libMesh [Kirk, Peterson, Stogner, and Carey (2006)], a finite element framework providing parallel I/O services and interfaces to solver packages. In particular, MOOSE accesses solvers in PETSc [Balay, Buschelman, Eijkhout, Grop, Kaushik, Knepley, McInnes, Smith, and Zhang (2004)].

Other applications being built on MOOSE include: a multi-phase flow code with conjugate heat transfer; a pebble-bed reactor code (heat conduction, transport, and neutronics); a reactive transport code; and a phase field code for modeling microstructure evolution [Tonks, Gaston, Millett, Andrs, and Talbot (2012)].

5.3 Description of Nuclear Fuel Analysis Requirements

Generally the most important calculation in nuclear fuel performance simulation is the fuel centerline temperature. A correct fuel centerline temperature requires many simulation capabilities, the most fundamental being fully coupled thermomechanics. The source term in the heat conduction equation is volumetric heat generation in the fuel due to fission. This fission heat source drives the coupled problem. In addition to fully coupled thermomechanics within the fuel and clad, a capability to model heat transfer and mechanical contact in the gap between the fuel and the clad is critical. Finite strain deformation is important since both the fuel and the cladding swell and contract considerably (especially in abnormal environments) due to thermal expansion and thermal- and irradiation-driven creep. The fuel has additional

volumetric strains due to fission product swelling and densification. Many of the thermal and mechanical material models for the fuel and clad depend on temperature and burnup, where burnup is a measure of the energy extracted from the fuel. The fission process also produces gases, which must be tracked. Some of this gas is released into the plenum, resulting in increased pressure loading on the cladding and decreased thermal conductance of the gap between the fuel and clad. This gap thermal conductance is a function of gap size, temperature, and gas composition. The plenum pressure is a function of the evolving internal volume, the amount of gas (which increases due to fission gas release), and the gas temperature. On the exterior of the cladding, convective heat transfer carries away the heat, which in a nuclear reactor is used to generate electricity. It is also important to allow both steady-state and transient analysis.

Clearly, this is a complex multiphysics application where practical problems are very large. Thus, the software must run efficiently and make use of parallel computing. The JFNK method is well suited for such problems. To begin, it is based in iterative methods, which parallelize well. JFNK relies on the overall residual, which can be built up one piece at a time. Specialized elements (coupling temperature and displacement, for example) are not needed. Perhaps most importantly, the true Jacobian for this problem would be extremely difficult to derive and extremely difficult to program. In contrast, we supply block-diagonal approximations of the Jacobian for preconditioning only. This vastly simplifies the effort required to develop a fuel performance analysis tool.

5.4 Axisymmetric Discrete-pellet Fuel Rod

The various modeling capabilities of BISON are demonstrated using a 2D axisymmetric analysis of a simplified fuel rodlet. The assumed geometry is shown in Figure 5. The problem includes ten individual UO_2 pellets, Zr-4 cladding, an initial 80 μm pellet-clad gap, and an open region to simulate the upper plenum. The plenum volume was set assuming a plenum to fuel length ratio of 0.045, typical for PWR fuel [Bailly, Ménessier, and Prunier (1999), p. 282]. A uniform convective boundary condition at the clad outer wall simulates heat transfer to the flowing coolant. Operating conditions typical of a PWR reactor were used, as given in Table 5.

UO_2 properties were taken from [Olander (1976), Chap. 10 and 16], while temperature dependent thermal and elastic properties for Zr-4 were taken from MATPRO [Allison, Berna, Chambers, Coryell, Davis, Hagrman, Hagrman, Hampton, Hohorst, Mason, McComas, McNeil, Miller, Olsen, Reymann, and Siefken (1993)]. The contact surface between the fuel and clad is assumed frictionless. For this demonstration problem, mesh resolution studies were not performed. Note that results in which the 10 discrete pellets are simulated with a single smeared pellet are

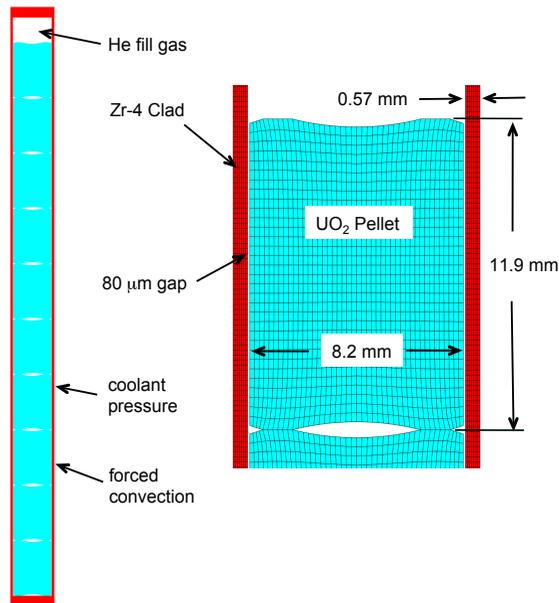


Figure 5: Axisymmetric problem geometry, materials, and typical mesh.

shown for comparison.

Results for temperature at the fuel centerline, fuel radial surface, and inner clad surface for the smeared and discrete-pellet simulations are shown in Figure 6(a). Following the initial power ramp, the fuel centerline temperature first increases slightly due to an increase in the fuel-clad gap as a result of fuel densification. After densification, fuel swelling and clad creep combine to reduce the gap size, resulting in a decrease in fuel temperature.

Fission gas release begins at a burnup of 20 MWd/kgU, and plots of fission gas release and the corresponding increase in plenum pressure are shown in Figure 6(b). As gas is released to the gap a corresponding increase in fuel temperature occurs, evident in Figure 6(a), due to a reduction in gap gas thermal conductivity as the fission gas mixes with the helium fill gas. This temperature increase is gradually reversed by continual gap closure, until the gap is fully closed at approximately 34 MWd/kgU. Fuel temperatures increase from this point on due to decreasing fuel thermal conductivity with burnup.

The predicted gap width versus burnup, at both a pellet end and axial centerline, is shown in Figure 6(c). Results from both the discrete and smeared-pellet calculations are shown for comparison. Note that the gap closes earlier at the ends

Table 5: Input parameters for the axisymmetric problem.

Linear average power (W/cm)	200
Fast neutron flux ($n/m^2\cdot s$)	9.5×10^{17}
Coolant pressure (MPa)	15.5
Coolant temperature (uniform) (K)	530
Coolant convection coefficient ($W/m^2\cdot K$)	7500
Rod fill gas	Helium
Fill gas initial pressure (MPa)	2.0
Initial fuel density	95% theoretical
Fuel densification	1% theoretical
Burnup at full densification (MWd/kgU)	5

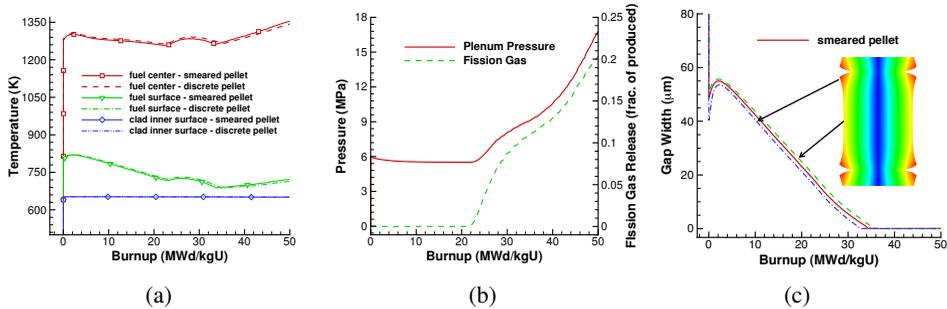


Figure 6: Results from the 2D axisymmetric ten pellet rodlet simulation for both smeared and discrete pellets, where (a) shows temperature plots at three locations in the fuel and (b) shows the pressure and fission gas release. The gap size is found in (c).

of the pellet, which is evidence of the so-called “bambooning” effect that has been observed in LWR fuel rods. This effect can be clearly seen in Figure 7, which compares the cladding radial displacement for the smeared and discrete pellet cases, at four burnup levels. The multidimensional bambooning effect of the discrete pellets becomes obvious following pellet-clad contact.

A coupled multiphysics application like BISON is needed to obtain such results. The fact that JFNK requires only approximate Jacobians (and then only for preconditioning) makes the development of such an analysis tool much more efficient. We next discuss the effectiveness of our approach for very large analyses.

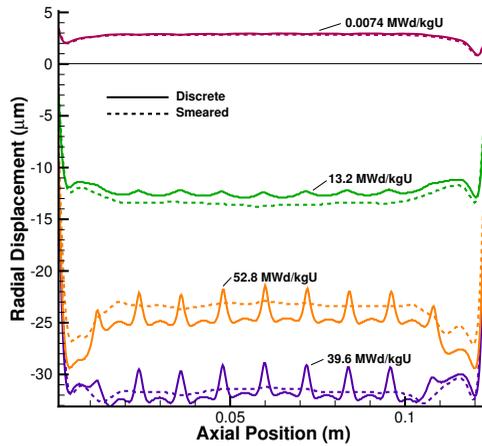


Figure 7: Radial displacement of the cladding vs. axial position. Note the bambooing effect due to discrete pellets.

5.5 Parallel Scaling

High-fidelity multiphysics modeling is a computationally intense task. As the number of coupled equations grow the computational burden quickly outpaces the resources available on a standard workstation. Further, highly resolved, three-dimensional geometry compounds the issue, necessitating the use of clusters or even supercomputers.

Efficient utilization of such machines requires a robust parallel algorithm with a minimum amount of communication and effective load balancing [Farhat and Lesoinne (1993)]. When utilizing a Newton-Krylov method for implicit solid mechanics much of the burden of ensuring efficient parallel execution falls to the linear solver used. However, the finite difference approximation in JFNK, requiring a full residual vector computation for each linear iteration of the solver, must also perform well in parallel. Any load imbalance will significantly impact overall solve performance as processors with a lighter load sit idle waiting on the others to finish their piece of the residual computation. Similarly, communication must be minimized in order to produce an efficient JFNK residual evaluation. If a poor subdomain decomposition is chosen, the communication of off-processor residual vector computations will stall the entire simulation. Finally, the need for forming effective preconditioners in parallel presents yet another obstacle to efficient solution of coupled multiphysics problems utilizing JFNK.

In spite of these issues, solid mechanics with JFNK can be extremely parallelizable. Given a competent partitioning library such as METIS [Karypis and Kumar (1996)] or Zoltan [Devine, Boman, Heaphy, Hendrickson, and Vaughan (2002)] the load imbalance and communication during normal finite element residual and Jacobian assembly can be minimized. For preconditioning, the strongly elliptic nature of the stress divergence equation lends itself well to the use of multigrid methods. As shown previously in Table 2, the Hypre algebraic multigrid (AMG) preconditioner can be extremely effective. In the context of parallel computation, AMG methods are attractive for their ability to maintain efficacy as the problem is spread among more parallel partitions. Many previous researchers have shown the utility in applying AMG to solid mechanics problems in parallel [Wriggers and Boersma (1998); Adams, Bayraktar, Keaveny, and Papadopoulos (2004); Lang, Wieners, and Wittum (2003)].

As mentioned previously, BISON is built using the MOOSE computational framework which, in turn, utilizes the libMesh finite element library. Through libMesh a number of partitioning libraries including those mentioned above can be easily applied to create communication minimizing, load balanced mesh partitions for distributed memory parallelism. MOOSE, following the libMesh convention, handles distributed memory parallel assembly of residuals and Jacobians through the use of “ghost elements” as described in [Kirk, Peterson, Stogner, and Carey (2006)] and parallel vectors/matrices supplied by an underlying linear or nonlinear solver library such as PETSc [Balay, Buschelman, Eijkhout, Groppe, Kaushik, Knepley, McInnes, Smith, and Zhang (2004)] or Trilinos [Heroux et al. (2008)]. MOOSE also provides a hybrid-parallelism model consisting of a distributed memory parallel model utilizing MPI and a shared memory model through use of threading. These two models can be used concurrently allowing for matching parallel communication hierarchies to cluster network topologies, enabling efficient utilization of massively parallel computers.

To demonstrate the efficacy of this approach for solving implicit, fully-coupled solid mechanics systems in parallel, a high-fidelity, three-dimensional nuclear fuel simulation has been run on a massively-parallel computer. Two different scaling studies were performed: a “strong scaling” study where a fixed simulation was run using successively more processors and a “weak scaling” study where the problem size is scaled with the number of processors. The strong scaling simulation involved 320 nuclear fuel pellets stacked inside the protective cladding tube as shown in Figure 8. This simulation utilized approximately 234 million total unknowns for temperature and displacements in three directions. The weak scaling study used this same configuration but scaled the number of pellets (and the cladding height) with the number of processors being used. All of these simulations were performed

on Idaho National Laboratory’s Fission supercomputer containing over 12,000 processing cores.



Figure 8: Full length fuel rod containing 320 nuclear fuel pellets inside cladding analyzed in 3D for strong scaling study. Colors indicate the temperature distribution after 14 months in the reactor.

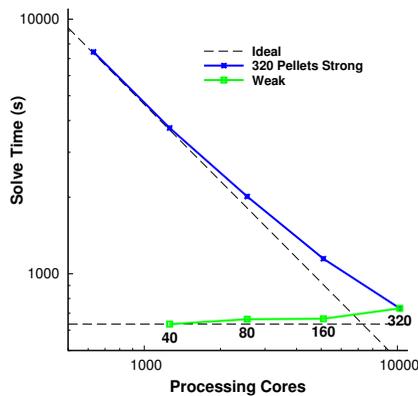


Figure 9: Strong and weak scaling results. The number of pellets in each of the weak scaling cases is listed in bold type.

Results from these scaling studies are given in Figure 9. It is clear that the use of preconditioned JFNK for solution of solid mechanics simulations allows for excellent scaling even out to over 10,000 processors. Both strong and weak scaling lines follow their respective ideal speedup lines, only significantly diverging over 5,000 processors. Utilization of this parallel scaling capability enables the study of extremely complex, multiphysics and even multiscale phenomena on high-fidelity, three-dimensional grids.

6 Summary

While the solution of the equations governing solid mechanics is often obtained via Newton’s method, this approach can be problematic if the determination, storage,

or solution cost associated with the Jacobian is high. JFNK methods employ a finite difference approximation to the Jacobian and offer compelling advantages for multiphysics modeling and parallel computing. Primary among these is the fact that the analytic Jacobian need not be determined, coded, and debugged. Despite this, the quadratic convergence rate of Newton's method is retained. JFNK is well-suited for multiphysics analysis since additional variables and/or terms in PDE's are easily incorporated. The method naturally accommodates multiple PDEs without the need to develop specialized elements that couple several unknowns. JFNK is inherently a fully-coupled approach and does not rely upon fixed point iteration, operator splitting, or loose coupling. JFNK also works well in a parallel computing environment.

The finite element code BISON is a parallel, object-oriented, nonlinear solid mechanics and multiphysics application that leverages JFNK methods. BISON demonstrates the robustness and flexibility of JFNK and is capable of solving traditional solid mechanics problems as well as large multiphysics problems.

Further work is underway to improve and extend BISON's set of features. In particular, we are interested in using BISON to study abnormal, 3D nuclear fuel scenarios which are beyond the reach of traditional nuclear fuel analysis codes.

Acknowledgements

The submitted manuscript has been authored by a contractor of the U.S. Government under Contract DE-AC07-05ID14517. Accordingly, the U.S. Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

References

. *Abaqus 6.9 Documentation*. Dassault Systèmes.

Adams, M. F.; Bayraktar, H. H.; Keaveny, T. M.; Papadopoulos, P. (2004): Ultrascaleable implicit finite element analyses in solid mechanics with over a half a billion degrees of freedom. *SC Conference*.

Allison, C. M.; Berna, G. A.; Chambers, R.; Coryell, E. W.; Davis, K. L.; Hagrman, D. L.; Hagrman, D. T.; Hampton, N. L.; Hohorst, J. K.; Mason, R. E.; McComas, M. L.; McNeil, K. A.; Miller, R. L.; Olsen, C. S.; Reymann, G. A.; Siefken, L. J. (1993): SCDAP/RELAP5/MOD3.1 code manual, volume IV: MATPRO—A library of materials properties for light-water-reactor accident analysis. Technical Report NUREG/CR-6150, EGG-2720, Idaho National Engineering Laboratory, 1993.

- Bailly, H.; Ménessier, D.; Prunier, C.**(Eds): *The Nuclear Fuel of Pressurized Water Reactors and Fast Neutron Reactors*. Lavoisier Publishing, Paris, France.
- Balay, S.; Buschelman, K.; Eijkhout, V.; Gropp, W. D.; Kaushik, D.; Knepley, M. G.; McInnes, L. C.; Smith, B. F.; Zhang, H.** (2004): PETSc users manual. Technical Report ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2004.
- Banaś, K.** (2002): A newton-krylov solver with multiplicative schwarz preconditioning for finite element compressible flow simulations. *Comm. Numer. Methods Engrg.*, vol. 18, pp. 269–275.
- Bathe, K.-J.** (1982): *Finite Element Procedures in Engineering Analysis*. Prentice-Hall.
- Belytschko, T.; Liu, W. K.; Moran, B.** (2000): *Nonlinear Finite Elements for Continua and Structures*. Wiley, New York.
- Broyden, C. G.** (1965): A class of methods for solving nonlinear simultaneous equations. *Math. Comput.*, vol. 19, no. 92, pp. 577–593.
- Carey, G. F.; Jiang, B.-N.** (1986): Element-by-element linear and nonlinear solution schemes. *Comm. Appl. Numer. Methods*, vol. 2, no. 2, pp. 145–153.
- Dennis, Jr., J. E.; Moré, J. J.** (1977): Quasi-newton methods, motivation, and theory. *SIAM Review*, vol. 19, no. 1, pp. 46–89.
- Devine, K.; Boman, E.; Heaphy, R.; Hendrickson, B.; Vaughan, C.** (2002): Zoltan data management services for parallel dynamic applications. *Comput. Sci. Eng.*, vol. 4, no. 2, pp. 90–97.
- Dunne, F.; Petrinic, N.** (2005): *Introduction to Computational Plasticity*. Oxford University Press, Oxford.
- Falgout, R. D.; Yang, U. M.** (2002): HYPRE: A library of high performance preconditioners. In *International Conference on Computational Science (3)*, pp. 632–641.
- Farhat, C.; Lesoinne, M.** (1993): Automatic partitioning of unstructured meshes for the parallel solution of problems in computational mechanics. *Internat. J. Numer. Methods Engrg.*, vol. 36, no. 5, pp. 745–764.
- Fletcher, R.; Reeves, C. M.** (1964): Function minimization by conjugate gradients. *Comput. J.*, vol. 7, no. 2, pp. 149–154.
- Gaston, D.; Newman, C.; Hansen, G.; Lebrun-Grandié, D.** (2009): MOOSE: A parallel computational framework for coupled systems of nonlinear equations. *Nucl. Eng. Design*, vol. 239, pp. 1768–1778.
- Geuzaine, P.** (2001): Newton-krylov strategy for compressible turbulent flows on unstructured meshes. *AIAA Journal*, vol. 39, no. 3, pp. 528–531.

- Hager, W. W.; Zhang, H.** (2006): A survey of nonlinear conjugate gradient methods. *Pacific Journal of Optimization*, vol. 2, no. 1, pp. 35–58.
- Hammond, G. E.; Valocchi, A. J.; Lichtner, P. C.** (2002): Modeling multicomponent reactive transport on parallel computers using jacobian-free newton krylov with operator-split preconditioning. *Developments in Water Science*, vol. 47, pp. 727–734.
- Heroux, M. et al.** (2008): Trilinos: an object-oriented software framework for the solution of large-scale, complex multi-physics engineering and scientific problems. <http://trilinos.sandia.gov>, 2008.
- Hestenes, M. R.; Stiefel, E.** (1952): Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Standards*, vol. 49, no. 6, pp. 409–436.
- Karypis, G.; Kumar, V.** (1996): Parallel multilevel k-way partitioning scheme for irregular graphs. *SC Conference*, vol. 0, pp. 35.
- Kelley, C. T.** (1995): *Iterative Methods for Linear and Nonlinear Equations*. SIAM, Philadelphia, PA.
- Key, S.** (2011): Fma-3d theoretical manual. Technical Report Ver 32, FMA Development, LLC, Great Falls, Montana, 2011.
- Kirk, B. S.; Peterson, J. W.; Stogner, R. H.; Carey, G. F.** (2006): libMesh: A C++ library for parallel adaptive mesh refinement/coarsening simulations. *Engineering with Computers*, vol. 22, no. 3-4, pp. 237–254.
- Knoll, D. A.; Keyes, D. E.** (2004): Jacobian-free Newton-Krylov methods: a survey of approaches and applications. *J. Comput. Phys.*, vol. 193, no. 2, pp. 357–397.
- Knoll, D. A.; Mousseau, V. A.** (2000): On Newton-Krylov-multigrid methods for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, vol. 163, pp. 262–267.
- Knoll, D. A.; Mousseau, V. A.; Chacon, L.; Reisner, J. M.** (2005): Jacobian-free Newton-Krylov methods for the accurate time integration of stiff wave systems. *SIAM J. Sci. Comput.*, vol. 25, pp. 213–230.
- Lang, S.; Wieners, C.; Wittum, G.** (2003): The application of adaptive parallel multigrid methods to problems in nonlinear solid mechanics. In *Error-controlled adaptive finite elements in solid mechanics*, pp. 347–384. Wiley.
- Lemieux, J.-F.; Price, S. F.; Evans, K. J.; Knoll, D.; Sallinger, A. G.; Holland, D. M.; Payne, A. J.** (2011): Implementation of the jacobian-free newton-krylov method for solving the first-order ice sheet momentum balance. *J. Comput. Phys.*, vol. 230, pp. 6531–6545.

- Liu, C.-S.; Atluri, S. N.** (2008): A novel time integration method for solving a large system of non-linear algebraic equations. *CMES: Computer Modeling in Engineering & Sciences*, vol. 31, no. 2, pp. 71–83.
- Matthies, H.; Strang, G.** (1979): The solution of nonlinear finite element equations. *Int. J. Numer. Meth. Engrg.*, vol. 14, pp. 1613–1626.
- Newman, C.; Hansen, G.; Gaston, D.** (2009): Three dimensional coupled simulation of thermomechanics, heat, and oxygen diffusion in UO₂ nuclear fuel rods. *Journal of Nuclear Materials*, vol. 392, pp. 6–15.
- Olander, D. R.** (1976): *Fundamental aspects of nuclear reactor fuel elements*. Technical Information Center, Energy Research and Development Administration.
- Rahul, D. S.** (2011): An efficient preconditioner for jacobian-free global–local multiscale methods. *Internat. J. Numer. Methods Engrg.*, vol. 87, pp. 639–663.
- Ramirez, J. C.; Stan, M.; Cristea, P.** (2006): Simulations of heat and oxygen diffusion in UO₂ nuclear fuel rods. *J. Nucl. Mater.*, vol. 359, no. 3, pp. 174–184.
- Rashid, M. M.** (1993): Incremental kinematics for finite element applications. *Internat. J. Numer. Methods Engrg.*, vol. 36, pp. 3937–3956.
- Roberts, J.; Forget, B.** (2008): Solving eigenvalue response matrix equations with jacobian-free newton-krylov methods. In *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, Rio de Janeiro, Brazil.
- Rosenbrock, H. H.** (1960): An automatic method for finding the greatest or least value of a function. *Comput. J.*, vol. 3, pp. 175–184.
- Ross, A. M.; Stoute, R. L.** (1962): Heat transfer coefficient between UO₂ and Zircaloy-2. Technical Report AECL-1552, Atomic Energy of Canada Limited, 1962.
- Saad, Y.; Schultz, M. H.** (1986): GMRES: a generalized minimal residual algorithm for solving linear systems. *SIAM J. Sci. Statist. Comput.*, vol. 7, pp. 856.
- SIERRA Solid Mechanics Team** (2010): Adagio 4.18 user's guide. Technical Report 2010-3111, Sandia National Laboratories, Albuquerque, NM 87185, 2010.
- Timoshenko, S. P.; Woinowski-Krieger, S.** (1959): *Theory of Plates and Shells*. McGraw-Hill, 2 edition.
- Tonks, M.; Gaston, D.; Millett, P.; Andrs, D.; Talbot, P.** (2012): An object-oriented finite element framework for multiphysics phase field simulations. *Comp. Mat. Sci.*, vol. 51, no. 1, pp. 20–29.

Tonks, M.; Gaston, D.; Permann, C.; Millett, P.; Hansen, G.; Wolf, D. (2010): A coupling methodology for mesoscale-informed nuclear fuel performance codes. *Nucl. Eng. Design*, vol. 240, pp. 2877–83.

Williamson, R.; Hales, J.; Novascone, S.; Tonks, M.; Gaston, D.; Permann, C.; Andrs, D.; Martineau, R. (2012): Multidimensional multiphysics simulation of nuclear fuel behavior. *Journal of Nuclear Materials*, vol. 423, pp. 149–63.

Williamson, R. L. (2011): Enhancing the ABAQUS thermomechanics code to simulate multipellet steady and transient LWR fuel rod behavior. *Journal of Nuclear Materials*, vol. 415, pp. 74.

Williamson, R. L. (2011): Simulation of NGNP fuel using the BISON fuel performance code: 2010 progress report. Technical Report INL/EXT-11-21403, Idaho National Laboratory, 2011b.

Wriggers, P.; Boersma, A. (1998): A parallel algebraic multigrid solver for problems in solid mechanics discretized by finite elements. *Comput. Struct.*, vol. 69, no. 1, pp. 129 – 137.

