Commun. Comput. Phys.
doi: 10.4208/cicp.091010.140711s

Vol. **12**, No. 3, pp. 807-833
September 2012

Parallel Algorithms and Software for Nuclear, Energy, and Environmental Applications. Part I: Multiphysics Algorithms

Derek Gaston[1], Luanjing Guo[2], Glen Hansen[3,*], Hai Huang[2], Richard Johnson[1], Dana Knoll[4], Chris Newman[4], Hyeong Kae Park[4], Robert Podgorney[2], Michael Tonks[1] and Richard Williamson[1]

[1] *Nuclear Science and Technology, Idaho National Laboratory, Idaho Falls, ID 83415, USA.*
[2] *Energy and Environment Science and Technology, Idaho National Laboratory, Idaho Falls, ID 83415, USA.*
[3] *Multiphysics Simulation Technologies Dept. (1444), Sandia National Laboratories, Albuquerque, NM 87185, USA.*
[4] *Fluid Dynamics and Solid Mechanics Group (T-3), Los Alamos National Laboratory, Los Alamos, NM 87545, USA.*

Received 9 October 2010; Accepted (in revised version) 14 July 2011

Available online 1 March 2012

Abstract. There is a growing trend within energy and environmental simulation to consider tightly coupled solutions to multiphysics problems. This can be seen in nuclear reactor analysis where analysts are interested in coupled flow, heat transfer and neutronics, and in nuclear fuel performance simulation where analysts are interested in thermomechanics with contact coupled to species transport and chemistry. In energy and environmental applications, energy extraction involves geomechanics, flow through porous media and fractured formations, adding heat transport for enhanced oil recovery and geothermal applications, and adding reactive transport in the case of applications modeling the underground flow of contaminants. These more ambitious simulations usually motivate some level of parallel computing. Many of the physics coupling efforts to date utilize simple code coupling or first-order operator splitting, often referred to as loose coupling. While these approaches can produce answers, they usually leave questions of accuracy and stability unanswered. Additionally, the different physics often reside on distinct meshes and data are coupled via simple interpolation, again leaving open questions of stability and accuracy.

*Corresponding author. *Email addresses:* Derek.Gaston@inl.gov (D. Gaston), Luanjing.Guo@inl.gov (L. Guo), gahanse@sandia.gov (G. Hansen), Hai.Huang@inl.gov (H. Huang), Rich.Johnson@inl.gov (R. Johnson), nol@lanl.gov (D. Knoll), cnewman@lanl.gov (C. Newman), hkpark@lanl.gov (H. K. Park), Robert.Podgorney@inl.gov (R. Podgorney), Michael.Tonks@inl.gov (M. Tonks), Richard.Williamson@inl.gov (R. Williamson)

This paper is the first part of a two part sequence on multiphysics algorithms and software. Part I examines the importance of accurate time and space integration and that the degree of coupling used for the solution should match the requirements of the simulation. It then discusses the preconditioned Jacobian-free Newton Krylov solution algorithm that is used for both multiphysics and multiscale solutions. Part II [1] presents the software framework; the Multiphysics Object Oriented Simulation Environment (MOOSE) and discusses applications based on it.

AMS subject classifications: 65M12, 65M60, 65Y05, 65Z05, 65H10

Key words: Multiphysics simulation, Jacobian-free Newton Krylov, finite element applications, physics-based preconditioning.

1 Introduction

The use of multiphysics simulation is growing rapidly with the interest in more realistic and higher fidelity analysis of energy and environmental systems. This increase in activity is typically attributed to increasing computer power, but in truth, advanced numerical methods are playing an equal role. The phrase "multiphysics simulation" is used to describe analyses which include disparate physical phenomena. Examples of multiphysics problems in the nuclear energy field include coupling fluid flow, heat transfer and neutron kinetics for reactor dynamics, coupling fluid flow and structural dynamics to consider fluid-structure interactions for nuclear fuel rod fretting, and coupling nonlinear thermomechanics with contact, fission product behavior, and species transport to study fuel performance. In energy and environmental applications, one encounters problems involving most of the above physics; energy extraction involves geomechanics, flow through porous media and fractured formations, adding heat transport for enhanced oil recovery and geothermal applications, and adding reactive transport in the case of applications modeling the underground flow of contaminants. In addition to multiphysics coupling, most of these problems also have multiscale issues to resolve.

Important in higher fidelity multiphysics simulation are considerations of time integration and spatial discretization of the coupled system. It is popular to employ first-order operator splitting, or even explicit coupling of different codes, to perform the time integration. While this approach can produce results, it can also produce significant time integration error [2] and stability issues [3].

This paper (Part I of a two part sequence) discusses a modern multiphysics algorithm, the Jacobian-Free Newton-Krylov method (JFNK) combined with physics based preconditioning [4]. This approach can typically compete well with operator splitting, while significantly reducing time integration error and stability concerns. The first part of this presentation describes the JFNK methodology and its use in both a multiphysics and multiscale setting. Part II [1] describes an evolving software framework MOOSE [5] which utilizes this algorithmic basis and enables rapid development of multiphysics engineering analysis tools.

2 Multiphysics methods

The solution of multiphysics problems can be straightforward, or extremely involved, depending on the degree of which the physical properties are interdependent. If the processes being modeled are only weakly dependent on each other, operator splitting (solving for the state of each phenomena in turn, holding all others fixed) can be both stable and accurate, particularly if iteration is used to converge the composite system. However, if the processes are highly interdependent, it may not be straightforward to achieve convergence of the overall system due to the dependencies. Multiphysics simulation is often considered to obtain more accurate or higher fidelity solutions to systems traditionally analyzed by "single physics" approaches, where the most dominant effect is modeled while other effects are held to nominal values. Unfortunately, the utility of a multiphysics approach also depends on its efficacy over older approaches, and this efficacy depends on accurate time and spatial integration of the composite multiphysics system.

2.1 Time integration error

Time integration accuracy is usually one of the goals when one considers multiphysics simulation. In a coupled system, the time evolution of each phenomena affects the time evolution of the others, perhaps to a great degree. Failure to incorporate an effective time integration scheme that considers *both* the order of integration of each phenomena and the coupling between them, results in temporal integration error. An analysis [2,6,7] can be performed of the time discretization method under consideration for the multiphysics problem at hand to understand the basic nature of multiphysics time integration error. Frequently this error can be significant, especially if many time steps are considered. Additionally, the type of splitting and linearization employed can produce new time step constraints for nonlinear stability. Iteration on sources and temperature dependent coefficients within a time step can help minimize these errors and stability issues.

A nuclear engineering model problem will be the first multiphysics problem examined in this paper. The following example is derived from [7] and seeks to motivate the need to perform accurate time integration and to study issues that may impact accuracy of coupled multiphysics problems. Considering thermal conduction coupled to a one group diffusion model of neutron kinetics, ignoring delayed neutrons, leads to the equations

$$\frac{1}{v}\frac{\partial \phi}{\partial t}+\Sigma_a(T)\phi-\nu\Sigma_f(T)\phi=0, \tag{2.1}$$

$$\frac{\partial T}{\partial t}-e_f\Sigma_f(T)\phi+S=0. \tag{2.2}$$

In this greatly simplified homogeneous picture of a reactor, nuclear fission generates heat, and this heat is removed from the system through a thermal sink term S. Here, T is the

material temperature and e_f is the fission to heat conversion factor. This model problem is a nonlinear PDE system with multiple time and space scales. Without loss of generality, spatial discretization issues are ignored in this discussion.

The implicit first-order backward Euler (BE1) temporal discretization of (2.1) can be written as

$$\frac{1}{v}\frac{\phi^{n+1}-\phi^n}{\Delta t}+\Sigma_a(T^{n+1})\phi^{n+1}-\nu\Sigma_f(T^{n+1})\phi^{n+1}=0. \tag{2.3}$$

Similarly, an implicit second order backward difference (BDF2) discretization can be written as:

$$\frac{1}{v}\frac{\frac{3}{2}\phi^{n+1}-2\phi^n+\frac{1}{2}\phi^{n-1}}{\Delta t}+\Sigma_a(T^{n+1})\phi^{n+1}-\nu\Sigma_f(T^{n+1})\phi^{n+1}=0. \tag{2.4}$$

Use of (2.3) does not necessarily result in an accurate solution to (2.1), due to temporal discretization error. Indeed, the statement (2.3) describes a *modified problem*. The modified problem can be explored by performing modified equation analysis [2,8]. To analyse the behavior of the modified problem, one writes Taylor series expansions of ϕ^n and ϕ^{n-1} about t^{n+1}:

$$\phi^n=\phi^{n+1}-\Delta t\frac{\partial\phi}{\partial t}\bigg|_{n+1}+\frac{\Delta t^2}{2}\frac{\partial^2\phi}{\partial t^2}\bigg|_{n+1}-\frac{\Delta t^3}{6}\frac{\partial^3\phi}{\partial t^3}\bigg|_{n+1}+\mathcal{O}(\Delta t^4), \tag{2.5}$$

$$\phi^{n-1}=\phi^{n+1}-2\Delta t\frac{\partial\phi}{\partial t}\bigg|_{n+1}+\frac{(2\Delta t)^2}{2}\frac{\partial^2\phi}{\partial t^2}\bigg|_{n+1}-\frac{(2\Delta t)^3}{6}\frac{\partial^3\phi}{\partial t^3}\bigg|_{n+1}+\mathcal{O}(\Delta t^4). \tag{2.6}$$

Substituting (2.5) into (2.3) and collecting the terms of the source PDE that correspond to time t^{n+1} on the left hand side, one obtains

$$\left[\frac{1}{v}\frac{\partial\phi}{\partial t}+\Sigma_a\phi-\nu\Sigma_f\phi\right]^{n+1}=\frac{\Delta t}{2v}\frac{\partial^2\phi}{\partial t^2}+\mathcal{O}(\Delta t^2). \tag{2.7}$$

Similarly, substituting (2.5) and (2.6) into (2.4) yields

$$\left[\frac{1}{v}\frac{\partial\phi}{\partial t}+\Sigma_a\phi-\nu\Sigma_f\phi\right]^{n+1}=\frac{\Delta t^2}{3v}\frac{\partial^3\phi}{\partial t^3}+\mathcal{O}(\Delta t^3). \tag{2.8}$$

Note that modified equation analysis isolates terms that illustrate the temporal truncation error on the right hand side of the above expressions. Not surprisingly, we have shown that the first-order backward Euler approach has a first-order temporal error term, with a coefficient proportional to the second time derivative of the flux ϕ. The second order discretization is indeed second order in time, with a coefficient proportional to the third time derivative of ϕ. To stress an obvious but sobering point, for each time step taken using the above expressions an error results that is the magnitude of the right hand side of these equations; the error is reduced with smaller Δt. This error either accumulates or cancels as time steps are taken during the evolution of the problem. To further explore the temporal error behavior of a particular problem requires a numerical study.

It is our opinion that modern multiphysics simulations should be second-order accurate in time, or better[†]. An implicitly coupled solution strategy results when either (2.3) or (2.4) is combined with an implicit in time discretized version of (2.2). This approach will require nonlinear iteration within a time step. Use of operator splitting (often achieved by coupling two separate codes together) avoids the need for subiteration, as S, Σ_f, and Σ_a are usually lagged (evaluated using data from the previous time step). When coupling existing codes to execute a time step, sources are evaluated with an explicit dependence of variables from the other code, to simplify data exchange between codes. Such a time discretization strategy might look like

$$\frac{1}{v}\frac{\phi^{n+1}-\phi^n}{\Delta t}+\Sigma_a(T^n)\phi^{n+1}-\nu\Sigma_f(T^n)\phi^{n+1}=0, \tag{2.9}$$

$$\frac{T^{n+1}-T^n}{\Delta t}-e_f\Sigma_f(T^n)\phi^n+S^n=0. \tag{2.10}$$

To summarize, the spectrum of possible multiphysics coupling strategies for this example are:

1. Use of first order time integration, explicitly coupling two monophysics codes; the first solving (2.9) and the second solving (2.10). Data is exchanged between the codes at the end of each timestep. Note that solving (2.9) and (2.10) in a split fashion within the same code is equivalent.
2. Use of second order time integration employing separate codes that exchange information at the end of each time step (*e.g.*, BDF2 versions of (2.9) and (2.10)).
3. Use of first order time integration but employ an implicitly coupled (simultaneous) solution of this example using (2.3) and appropriately discretized versions of (2.2). Note that (2.3) requires the solution of (2.2), T^{n+1}. This is achieved by employing nonlinear iteration within each time step.
4. Use of second order time integration and an implicitly coupled solution of (2.4) and appropriately discretized versions of (2.2) (again employing nonlinear iteration within each time step).

The next example describes a simplified two-dimensional boiling water reactor (BWR) control rod ejection transient [9] to numerically study these four options. The problem consists of time-dependent two-group neutron diffusion and two-group delayed neutron precursor equations, (29) and (30) in [7], with the following adiabatic heat-up for temperature,

$$\frac{dT}{dt}=e_f\sum_g\nu\Sigma_{fg}\phi_g. \tag{2.11}$$

[†]This is no panacea; it is possible to accumulate appreciable temporal error even with high order time integration if care is not taken.

The control rod ejection event and Doppler feedback are modeled as:

$$\Sigma_{CR1}(t) = \Sigma_{CR1}\left[1 - 0.06068184t\right], \tag{2.12}$$

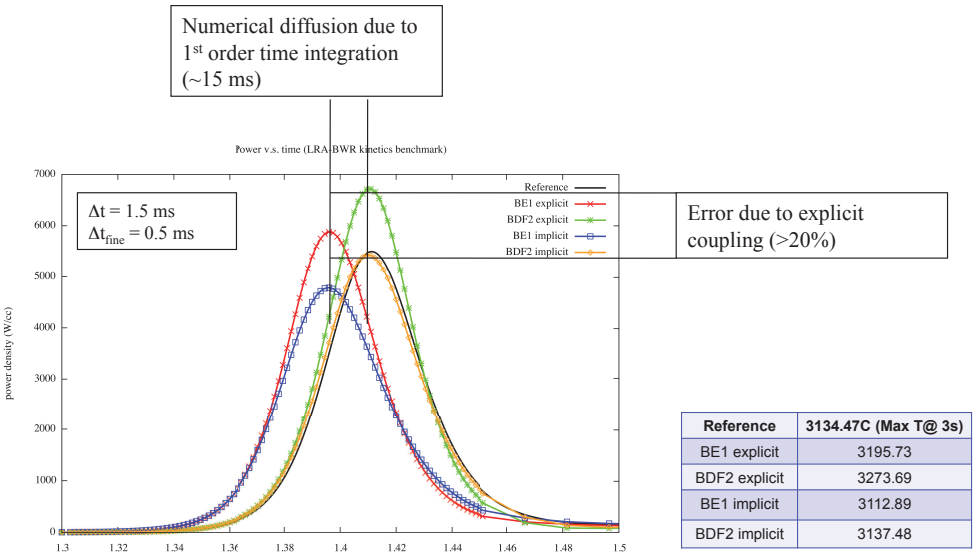
$$\Sigma_{a2}(\vec{r},t) = \Sigma_{a2}(\vec{r},0)\left[1 + \alpha_1\left(\sqrt{T} - \sqrt{T_0}\right)\right]. \tag{2.13}$$

Fig. 1 shows a plot of synthetic reactor power density vs. time. All cases were run with fixed time step size of $2\times10^{-3}s$. The reference solution was computed using BDF2 employing the implicitly coupled method with the time step size of $1\times10^{-4}s$. As expected, first-order time integration (the two curves labeled BE1) can introduce significant numerical dissipation, independent of if the problem is solved in an implicitly or explicitly coupled manner. In this example, the power peak occurred $1.2\times10^{-2}s$ earlier than the reference solution, and the peak power density was somewhat less. Note that the BDF2 implicit coupled result is very close to the reference solution (this result differs from the reference solution by employing a factor of 20 larger time step).

The first order explicitly coupled result produces an error in time, again the power peak occurred $1.2\times10^{-2}s$ earlier than the reference solution. However, in this case the observed peak amplitude was greater. Lastly, the second order explicit solution was timely, but roughly 20% high in peak amplitude. To summarize this example, explicit coupling results in a 20% misprediction of the peak amplitude (high), and first order time integration yields a $\approx$15ms misprediction of the peak amplitude in time (early). Please see [7] for a more thorough study of time integration error inherent in this coupled problem.

These results show that the methodology used to couple the physics is as important as higher order time integration. Indeed, with the explicitly coupled results, one would be hard pressed to argue that the second order result is "more accurate" than the first order result. Note however, when the coupling error is removed in the monolithic solution, the second order result is indeed clearly better than the first order result. Most importantly, the second order implicitly coupled result is quite accurate in predicting the reference solution which is the ultimate goal.

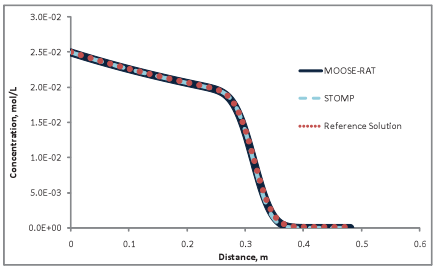
As a second example, in multi-component reactive transport systems interactions between multiple species result in nonlinear systems of governing equations for coupled fluid flow, solute transport, and chemical reaction processes. Popular subsurface reactive transport simulators, such as TOUGHREACT [10] and STOMP [11], often adopt an operator splitting approach that solves the transport equations and batch chemistry sequentially within a single time step. Iteration between transport and batch chemistry within calculation steps is also not always practiced. One obvious drawback of such an approach is that it only applies to loosely coupled reactive transport problems in which mineral-solution interactions are weak and kinetics of the reaction are typically slow, such as in natural biogeochemical degradation process in subsurface scenarios. For engineered environmental remediation problems where strong mineral-solution interactions and fast kinetics are typically encountered, chemical reactions and transport are tightly coupled together. Use of the operator splitting approach for such problems produces large decoupling errors even when using a very small time step.



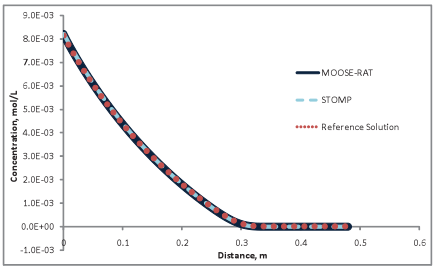
Reference	3134.47C (Max T@ 3s)
BE1 explicit	3195.73
BDF2 explicit	3273.69
BE1 implicit	3112.89
BDF2 implicit	3137.48

(a) Multiphysics Temporal Error. **(b)** Temperature Variation.

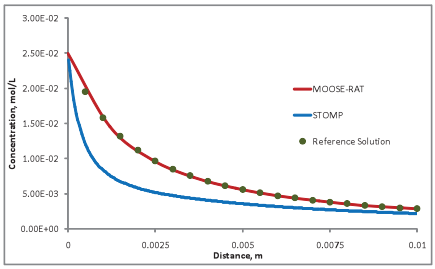
Figure 1: Comparison of four solutions of (2.1) and (2.2) showing solution differences between first and second order time integration and loose and tight coupling.



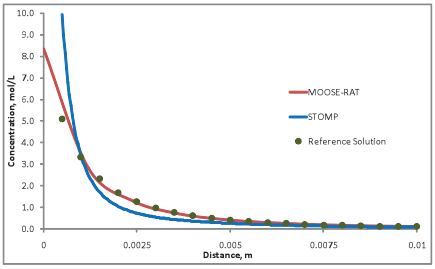
(a) Slow kinetics solution: Component A



(b) Slow kinetics solution: Component C



(c) Fast kinetics solution: Component A



(d) Fast kinetics solution: Component C

Figure 2: Comparison of operator split (STOMP) and tightly coupled solution strategies (MOOSE-RAT) for two reaction regimes. The top figures show a comparison of the codes for a slow kinetics problem, operator split methods perform well as the reaction equations are not strongly interdependent. The bottom figures show a fast kinetics problem where coupling issues negatively impact the accuracy of the operator split method.

Here, a simple synthetic case of kinetic mineral precipitation in a one-dimensional column is used to illustrate the validity of operator splitting and implicit coupled approaches. Reactants A and B are injected at constant concentrations and rates into a column from one end to react and form a solid species C in the column. The system of governing equations for this simple first-order mineral precipitation example is

$$\frac{\partial[\theta(C_A+C_{C(s)})]}{\partial t}+\nabla[\theta V\cdot C_A]-\nabla[\theta D\cdot\nabla C_A]=0, \tag{2.14a}$$

$$\frac{\partial[\theta(C_B+C_{C(s)})]}{\partial t}+\nabla[\theta V\cdot C_B]-\nabla[\theta D\cdot\nabla C_B]=0, \tag{2.14b}$$

$$\frac{d(C_{C(s)})}{dt}+k\cdot A\cdot\left(1-\frac{C_A\cdot C_B}{K_{eq}}\right)=0. \tag{2.14c}$$

The rate of precipitation was adjusted through the equilibrium constant K_{eq}. Two values, 10^{-5} and 10^{-8}, are used for K_{eq} in this example. This results in a 3 order of magnitude difference that spans both weakly coupled phenomena (slow kinetics) and strongly coupled scenarios (strong mineral-solution interaction and fast kinetics). STOMP [11] was chosen to represent the operator splitting approach, and ReActive Transport (RAT), developed using the Multiphysics Object Oriented Simulation Environment framework (MOOSE; see Part II of this issue [1]), generates the tightly-coupled solution. Both simulators were applied to simulate the spatial and temporal distributions of species within the column.

Fig. 2 shows a comparison of concentration profiles of the reactants and precipitated mineral using both RAT and STOMP for the two scenarios. A "reference" solution is also provided for comparison. Because this nonlinear system does not have an analytical solution, the reference solution was generated numerically by a MATLAB code specifically designed for this particular problem, using an extremely small time step size (10^{-9} s) and fine mesh. For slow chemical kinetics (*i.e.*, $K_{eq}=10^{-5}$) where the coupling between transport and reaction is weaker than the fast kinetics case (Fig. 2(a) and Fig. 2(b)), concentration profiles for both reactant (A) and precipitated mineral (C) obtained from the two simulators are in excellent agreement with the reference solution. Operator-splitting clearly works well for this problem. However, in the fast kinetics scenarios (*i.e.*, $K_{eq}=10^{-8}$) a implicit coupled approach produced solutions that match the reference solution much better than those generated by an operator splitting approach (Fig. 2(c) and Fig. 2(d)). This comparison concludes that for engineered systems where processes are generally tightly coupled, a implicit coupled solution approach is desirable. In closing, accurate time integration methods are currently an active area of research (*cf.* [3, 12, 13]).

2.2 Spatial integration error

The next major consideration for coupled multiphysics applications is the process by which solution data is shared between physical phenomena. Typically, the spatial discretization requirements for each phenomenon being considered differ, occasionally

markedly. For example, when one solves a fluid structure interaction problem, a mesh well tailored to the needs of the fluid mechanics problem is used, particularly in the boundary layer region where it is important to resolve the flow to obtain a correct solution to the force transferred to the structure. In the structure, one employs a mesh well suited for the structural mechanics problem to accurately calculate stresses and vibrational moments, etc. These requirements usually do not intersect where the meshes intersect; at the boundary of the structure and fluid region. It is common to compromise one or both meshes to match them at this point (to create a *conformal* mesh interface) or to couple the separate meshes using a mesh tying approach [14, 15].

A more difficult challenge ensues when the meshes span the same geometry, such as a fluid (coolant), thermal, and neutronics calculation inside of a nuclear reactor. Here, the physics are interdependent and each phenomenon is solved across the entire domain of the reactor. Further, the spatial discretization requirements may be quite different at certain locations within the domain. Again, one may choose a single compromise mesh on which to solve all of the problems simultaneously. Alternatively, one may choose a distinct mesh for each physics and transfer the required data between meshes and calculations as needed, perhaps using interpolation between meshes. This is often the approach used when codes are coupled to solve multiphysics problems, as each code may have unique and perhaps mutually exclusive mesh requirements. If the code employs internal coupling, interpolation may still be used to transfer the solution between meshes, but other options also exist. Recently, there has been substantial work in spatial integration approaches that accommodate spatial diversity within the solution method (such as the *multimesh* approach [16–20]). This section of the paper examines potential issues with the use of multiple meshes where data is exchanged between meshes.

There are many approaches [21–23] for transferring data from one mesh to another, depending on the nature of the two meshes. Jiao and Heath [22] study the use of analytical functions hosted on various one- and two-dimensional meshes, where the data are exchanged from a coarse to a fine mesh and back (defining one transfer cycle) using several different data transfer schemes (cubic spline, common refinement, common refinement 2, source, target, and linear interpolation), where common refinement uses linear finite elements while common refinement 2 used quadratic elements. The technique of applying many transfer cycles to a static function is an experimental approach useful to indicate possible cumulative effects of data transfer over many time steps. Both [22] and [24] employ the Runge function, $f(x)=1/(1+25x^2)$ on $-1\leq x\leq 1$ as a one-dimensional example function. To study the effect of function shape on these observations, [24] adds a second set of results generated using the sine-based function, $f(x)=0.6+0.1\sin(\pi x)$ on $0\leq x\leq 1$. For the Runge function, uniform grids having 45 and 32 nodes, denoted as (45, 32), are used. For the sine-based function, the uniform fine and coarse grids are (23, 16), which is about the same spacing as for the Runge function but on the smaller domain. Fig. 3 illustrates the error as a function of 64 data transfer (remapping) cycles for both functions, where the error is computed as $\|f_{\text{numerical}}-f_{\text{exact}}\|_2/\|f_{\text{exact}}\|_2$. While all of the data transfer schemes perform at a similar relative accuracy for the two examples

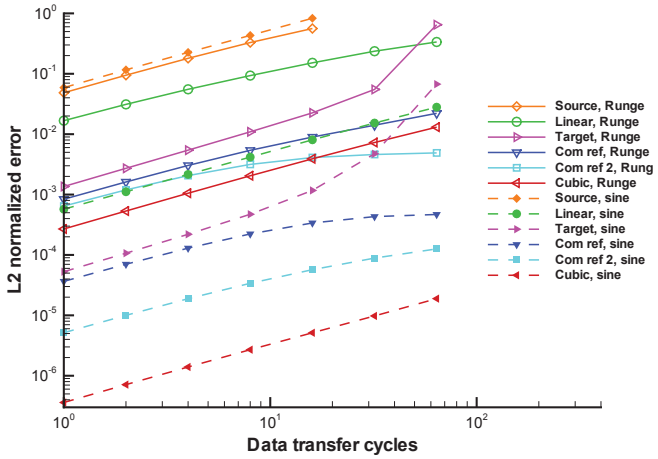


Figure 3: Normalized L^2 error vs. the number of data transfer cycles for the sine and Runge functions using cubic spline, common refinement, source, target, and linear interpolation.

examined, the magnitude of the L^2 error depends on the function (actually the smoothness of the function) being remapped. The figure shows the precision of each of the data transfer schemes considered on the Runge and sine-based functions, and provides a comparison of the relative performance between various schemes. Note that both the cubic spline and common refinement methods provide the lowest error results for multiple data transfer cycles. Secondly, note that linear interpolation does not compete well with these two, more involved approaches. Finally, note that none of the schemes is error free; it is important to realize that some precision loss occurs whenever data from one mesh are transferred to another. Whether in the end this is significant is problem dependent, and the impact of this must be verified by the analyst using the code. Two examples will be used to further study potential issues with multiphysics spatial integration.

The Brusselator is a system proposed in [25] to model a chemical reaction diffusion system, coupled through nonlinear source terms,

$$\frac{\partial T}{\partial t} = D_1 \frac{\partial^2 T}{\partial x^2} + \alpha - (\beta+1) + T^2 C, \tag{2.15a}$$

$$\frac{\partial C}{\partial t} = D_2 \frac{\partial^2 C}{\partial x^2} + \beta T - T^2 C, \tag{2.15b}$$

where Ω is the one dimensional domain $a \le x \le b$, $T(a,t) = T(b,t) = \alpha$ and $C(a,t) = C(b,t) = \beta/\alpha$. The system (2.15) has steady state, oscillatory and chaotic solutions. With the choice of $\alpha = 0.6$, $\beta = 2.0$ and $D_1 = D_2 = 0.025$, T and C are oscillatory [12]. The initial conditions, $T(x,0) = f(x)$, $C(x,0) = g(x)$, can be specified as any smooth functions, which allows the study of the effects of the initial conditions and evolving solutions on aggregate numerical error. It also allows the use of functions for $f(x)$ and $g(x)$ that have distinct natures that will allow the investigation of such disparity on the aggregate error given various data transfer strategies.

The study by Johnson *et al.* [24] presents two examples that apply the following initial conditions to investigate the effects of data transfer error:

$$f(x)=0.6+\frac{1}{1+25x^2}, \qquad g(x)=\frac{10}{3}+\frac{1}{1+25x^2}, \tag{2.16}$$

$$f(x)=0.6+\frac{\exp(ax)-1}{\exp(a)-1}, \qquad g(x)=\frac{10}{3}+\frac{1}{1+25x^2}, \tag{2.17}$$

where $a=15$ in (2.17).

To examine the effects of data transfer error on selected solutions of the Brusselator system (2.15), consider the Runge functions (2.16) as the initial condition on an interval of $-1\leq x\leq 1$ with $\Delta t=5.0\times 10^{-5}$. The uniform meshes consist of 45 $(\Delta x\approx 0.045)$ and 32 $(\Delta x\approx 0.065)$ nodes for C and T, respectively, which are both within the asymptotic convergence range for their respective functions. Fig. 4 shows T at $t=1$ using cubic spline, common refinement, source, target, and linear interpolation data transfer schemes. Also shown are the initial condition and a reference solution, computed using a common uniformly discretized 45 node mesh. There is a diversity of results that are a function of the data transfer scheme employed to transfer information between meshes. The common refinement and target schemes show similar results that match the reference solution closely. Both the source scheme and linear interpolation schemes show poor accuracy. Cubic spline results are slightly worse than those for common refinement and target. The percentage difference of the results for each scheme (compared to the reference solution at $x=0$) are 0.47% for cubic spline, 0.0003% for common refinement, -4.1% for source, -0.05% for target, and 2.9% for linear interpolation. Unlike [24], these results are based on a coarse reference solution.

Given this simple multiphysics problem, it is readily apparent that the choice of data transfer mechanism is an important consideration. While only the analyst that will employ the calculation can make the determination of what the acceptable accuracy is for the problem at hand, it is likely of significant general concern that the selection of the source transfer scheme results in a 4% error, and selecting linear interpolation results in a 3% error on this example, with all else being equal.

The next study examines results obtained by selecting different initial conditions (2.17). Here, the uniform meshes used were (45, 33) for T and C, respectively. These discretizations result in mesh node spacings of $(\Delta x\approx 0.045,0.063)$, which are spacings within the asymptotic range of convergence for (2.15). Fig. 5 shows T and C at $t=3$. In this result, there is significant diversity in the results for T near $x=0.9$, where a sharp peak has evolved in the solution. The L^2 norm of the error in T is 0.55% for common refinement, 0.14% for common refinement 2, 0.64% for cubic spline and 3.17% for linear interpolation as compared to the fine mesh reference solution. Similar errors are seen for C. These results again show that one may encounter significant error in the data transfer process between meshes in a multiphysics simulation. Further, these results indicate that while one may be within the asymptotic range of the physics approximations on all

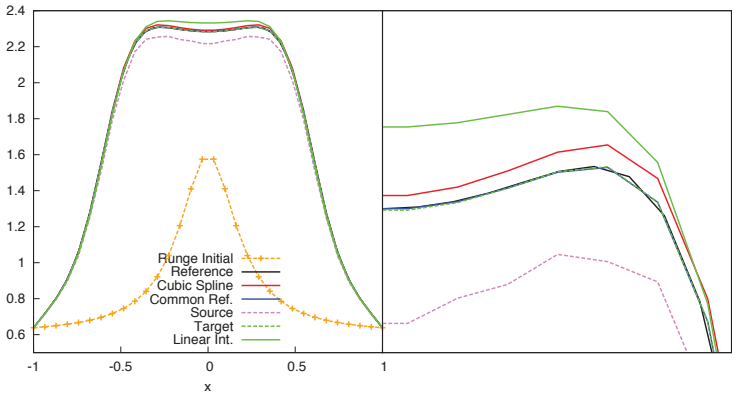


Figure 4: T at $t=1$ using cubic spline, common refinement, source, target, and linear interpolation schemes. The figure to the right is a magnification of the region $0\leq x\leq 0.5$. The initial condition and discretization is also shown.

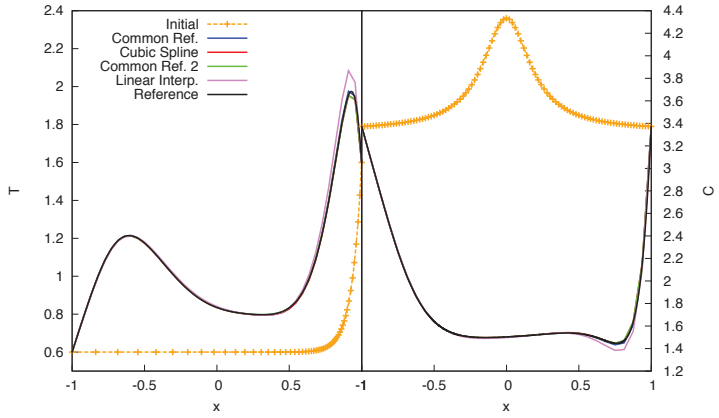


Figure 5: T and C at $t=3$ for cubic spline, common refinement, and linear interpolation for uniform meshes (45, 33) compared to a fine mesh reference solution and the initial condition.

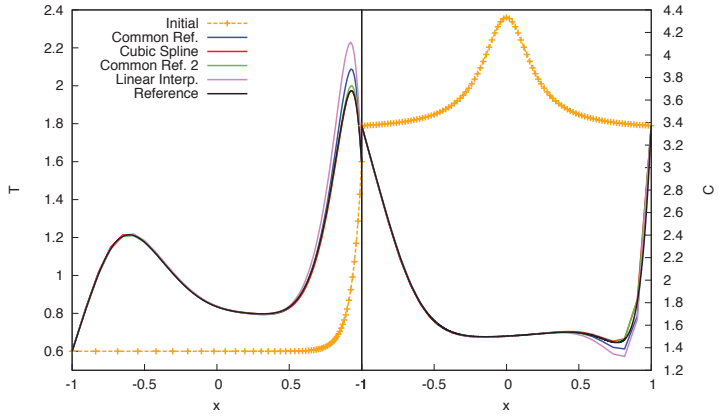


Figure 6: T and C at $t=3$ for cubic spline, common refinement and linear interpolation for nonuniform meshes of (67, 45) nodes compared to a fine mesh reference solution and the initial condition.

the meshes, one needs to verify that the data transfer scheme chosen for the problem supports the spatial resolution requirements of the aggregate problem.

The next example considers the use of nonuniform meshes optimized for the initial conditions on the aggregate error, perhaps as one might see in adaptive mesh refinement. Optimized meshes consisting of 67 and 45 nodes for the exponential and modified Runge functions were used. The mesh adaptation to the initial source data was accomplished using an approach based on curvature and then adjusted using a manufactured solution [24]. The maximum element sizes for these nonuniform meshes $(\Delta x \approx 0.096, 0.091)$ were verified to be within the asymptotic convergence range of (2.15).

Solutions for T and C using cubic spline, common refinement, common refinement 2 and linear interpolation compared to the fine mesh and initial solutions are shown in Fig. 6 for $t=3$. The solution for T at the location of the peak near $x=0.9$ significantly overshoots the fine mesh solution for linear interpolation and common refinement schemes; cubic spline results are slightly low at the peak. Common refinement 2 results are slightly high. The L^2 norm of the error in T is 2.51% for common refinement, 0.51% for common refinement 2, 0.57% for cubic spline and 6.59% for linear interpolation compared to the fine mesh reference solution. Again, similar errors are seen for C near $x=1$.

While the results for the cubic spline and common refinement 2 approaches are of the same magnitude as discretization error, both the linear interpolation and common refinement solutions depart significantly from the peak near $x=0.9$, likely by an unacceptable amount. Even though the maximum element sizes on both meshes fall within the asymptotic ranges of their respective physics solutions, the disparity of meshes near the peak combine with the tight coupling between the equations and with the data transfer algorithm to result in significant spatial error. Indeed, this leads to a hypothesis that even though the separate physics equations are spatially resolved, the coupled multiphysics problem (which includes the chosen data transfer algorithm), is not resolved at the location of the T peak. Said another way, even though the meshes are adaptively refined to their respective source functions, the C mesh is probably too coarse near the T peak ($x=0.9$) to support the data transfer operation there. In the case of adaptive mesh refinement in a multiple mesh multiphysics calculation, a global error indicator must be employed that includes a quantification of spatial error occurring within the data transfer operation between meshes. Note that even advanced schemes such as Fig. (4) in [26] do not consider the error within the data transfer process. The metric that must be used needs to consider the impact of the quantity being transferred on the aggregate solution. A metric based on an *a posteriori* reference solution [27] does provide this information (albeit expensively). Even better, it may be preferable to consider one of the advanced multimesh strategies based on *hp*-FEM that do not require data transfer between meshes [19, 20].

If one is developing a new coupled multiphysics application, it is important to carefully assess the advantages and disadvantages of a multiple mesh approach over using a single mesh to host the composite multiphysics problem. A multiple mesh approach has the advantage of using the minimum number of elements to host the solution; a single

mesh must be sufficiently fine at every location to support the most demanding physics there. But a multiple mesh approach is much more complex in that multiple meshes must typically be generated [28], data must be transferred between them, and adequate refinement must be maintained on each mesh, usually in parallel. Dynamic load balancing of a time adaptive multiple mesh application is a very sobering challenge, indeed.

Alternatively, while using a single mesh with sufficient refinement to satisfy the most demanding of the physics phenomena present may be computationally expensive, it is a straightforward and accurate approach. A single, adaptively refined mesh is employed in the results presented in Part II [1] of this paper. Here, data transfer error is not a concern as data is not transferred, all physics discretizations share the same mesh and coupling values between equations is straightforward.

2.3 Jacobian-free Newton-Krylov methods

The preconditioned Jacobian-free Newton-Krylov method is the cornerstone of the multiphysics approach presented in this study. This method is a popular coupled multiphysics solution algorithm [4,29,30] and it also has utility as a multiscale algorithm [31–33]. Following finite element discretization of (2.1) and (2.2), Newton's method can be used to solve the nonlinear problem

$$\mathbf{M}(\mathbf{u})\dot{\mathbf{u}}+\mathbf{K}(\mathbf{u})\mathbf{u}-\mathbf{f}(\mathbf{u})=\mathbf{F}(\mathbf{u}), \tag{2.18}$$

where $\mathbf{u}$ is the unknown solution vector, $\mathbf{M}$ and $\mathbf{K}$ are matrices, $\mathbf{f}$ are source terms, and $\mathbf{F}(\mathbf{u})$ is the system *residual*. Newton's method, given an *initial guess* that is within the sphere of convergence of the method, can provide quadratic convergence in driving the system residual toward zero. Note that one may write the entire composite multiphysics system in the monolithic form

$$\mathbf{F}(\mathbf{u})=0, \tag{2.19}$$

where the composite system is coupled through terms in the above matrices and the state $\mathbf{u}$.

Newton's method requires the solution of the linear system [6]

$$\mathbf{J}^k\delta\mathbf{u}^k=-\mathbf{F}(\mathbf{u}^k), \tag{2.20}$$

followed by the update

$$\mathbf{u}^{k+1}=\mathbf{u}^k+\delta\mathbf{u}^k, \tag{2.21}$$

where $\mathbf{J}$ is the Jacobian matrix the state vector and k is the nonlinear iteration index. In vector notation, the $(i,\ j)^{th}$ element of the Jacobian matrix is

$$\mathbf{J}_{i,j}=\frac{\partial F_i(\mathbf{u})}{\partial u_j}. \tag{2.22}$$

Forming each element of $\mathbf{J}$ requires taking analytic or numerical derivatives of $F_i(\mathbf{u})$ with respect to $\mathbf{u}$ at each grid point. This can be difficult, time consuming, and error-prone.

In the Jacobian free Newton Krylov method (JFNK), a Krylov method [34] is used to solve (2.20). An initial linear residual, $\mathbf{r}_o$, given an initial guess, $\delta\mathbf{u}_o$, is formed using

$$\mathbf{r}_o = -\mathbf{F}(\mathbf{u}) - \mathbf{J}\delta\mathbf{u}_o. \tag{2.23}$$

Note that the nonlinear iteration index k has been dropped as Krylov iteration is performed at each Newton iteration k until convergence is achieved. The approximate solution at the l^{th} Krylov iteration $\delta\mathbf{u}_l$ is constructed from a linear combination of the Krylov vectors (search directions)

$$\left\{\mathbf{r}_o,\ \mathbf{J}\mathbf{r}_o,\ (\mathbf{J})^2\mathbf{r}_o,\cdots,(\mathbf{J})^{l-1}\mathbf{r}_o\right\}, \tag{2.24}$$

which were constructed during the previous $l-1$ Krylov iterations. This linear combination of Krylov vectors can be written as

$$\delta\mathbf{u}_l = \delta\mathbf{u}_o + \sum_{j=0}^{l-1}\alpha_j(\mathbf{J})^j\mathbf{r}_o, \tag{2.25}$$

where evaluating the scalars α_j is part of the Krylov iteration. Upon examining (2.25) it is apparent that a Krylov method requires the action of the Jacobian on a vector, not the Jacobian alone. One may express the action of the Jacobian on a vector $\mathbf{v}$ with the difference expression [35]

$$\mathbf{J}\mathbf{v} \approx \frac{\mathbf{F}(\mathbf{u}+\epsilon\mathbf{v}) - \mathbf{F}(\mathbf{u})}{\epsilon}, \tag{2.26}$$

where $\mathbf{v}$ is a Krylov vector (*i.e.*, one of the set (2.24)) and ϵ is a small perturbation.

A fair amount of literature exists on the use of matrix free methods, motivated by ODEs and boundary value problems, while less exists for time-dependent PDEs. Standard "PDE motivated" references are [4, 35, 36]. The JFNK approach, besides its obvious memory advantage in not requiring storage for the Jacobian matrix, has other interesting features. The JFNK method provides Newton-like nonlinear convergence without forming the true Jacobian. The method also lends itself to a clean mathematical abstraction that results in a *façade design pattern* [37] that supports extensible implementation in software.

The JFNK method also has its advantages and disadvantages. The main advantage is that for a tight convergence tolerance there is no splitting or linearization error [2]. The algorithm provides a clean way to include other nonlinear phenomena, and the developer can readily add complex physics into the residual evaluation function $\mathbf{F}(\mathbf{u})$, and have the derivatives of this physics be computed and included into the action of the Jacobian. Indeed, one may even call an external nonlinear analysis code that operates at a different scale than the host, as is described in [32, 33]. Another advantage is that one can implement a variety of time discretizations such as higher-order backward difference formula (BDF) methods or implicit Runge-Kutta methods.

One major consideration in the use of the JFNK method is that one is solving a nonlinear problem, using iteration, to advance a time step. Furthermore, one is solving the full dimensional system implicitly. Thus a large matrix system must be approximately inverted on each Newton iteration. In order for this approach to be tractable one must produce an effective preconditioner, as the number of JFNK function evaluations needed to form (2.26) are directly proportional to the number of Krylov iterations required for convergence of the linear system. Clearly, strategic use of the inexact Newton method [38,39] will economize solution time. Further, note that the JFNK function evaluation may often be implemented to be "embarrassingly parallel." As a final note, on finite element problems this function evaluation short circuits the need for a global assembly operation as one is forming the action of the Jacobian on a per element basis not the global Jacobian itself, simplifying the code significantly in most cases.

2.4 Preconditioning of JFNK

The purpose of preconditioning a Krylov method is to efficiently cluster eigenvalues of the iteration matrix, which in turn will reduce the number of Krylov iterations required for converging (2.20). Traditionally, standard iterative methods such as Jacobi, SSOR, or ILU are applied to the system Jacobian matrix when constructing a preconditioner [34]. In the sequel, we refer to the set of approaches in this class as *algebraic preconditioning methods*. When applying physics-based preconditioning to the JFNK method the Jacobian matrix $\mathbf{J}$ is never formed.

When employing the right preconditioned form of (2.20), one solves the problem

$$(\mathbf{J}\mathbf{P}^{-1})(\mathbf{P}\delta\mathbf{u}) = -\mathbf{F}(\mathbf{u}), \tag{2.27}$$

where $\mathbf{P}^{-1}$ denotes a linear operator which symbolically represents the *preconditioning process*. By *symbolically*, we mean that the inverse of a preconditioning matrix $\mathbf{P}$ is functionally not required to implement the algorithm. For example, in physics based preconditioning, $\mathbf{P}^{-1}$ actually represents a linearized time step. The right preconditioned version of the JFNK matrix-vector multiply approximation (2.26) is

$$(\mathbf{J}\mathbf{P}^{-1})\mathbf{v} \approx \frac{\mathbf{F}(\mathbf{u}+\epsilon\mathbf{P}^{-1}\mathbf{v}) - \mathbf{F}(\mathbf{u})}{\epsilon}. \tag{2.28}$$

This preconditioning process is implemented using two steps:

1. Perform
$$\mathbf{y} = \mathbf{P}^{-1}\mathbf{v}. \tag{2.29}$$
2. Perform $\mathbf{J}\mathbf{y} = [\mathbf{F}(\mathbf{u}+\epsilon\mathbf{y}) - \mathbf{F}(\mathbf{u})]/\epsilon$.

2.4.1 Physics-based preconditioning

There exist numerous legacy algorithms to solve nonlinear systems, and these algorithms are typically based on linearization and time splitting. This linearization and time splitting has usually been developed with significant insight into the time scales or physical behavior of the problem. As a benefit of this insight, a reduced implicit system, or a sequence of segregated implicit systems may be solved in place of the implicit coupled system. In the end, these legacy algorithms may make excellent preconditioners for the Jacobian-free Newton-Krylov method.

This algorithmic concept is referred to as physics-based preconditioning. Here, the Jacobian matrix is not manipulated algebraically to find an approximate inverse; rather, approximations are made to the original differential system that result in a modified, more tractable, approximate Jacobian system for preconditioning.

Consider using the first-order in time operator split method (*e.g.*, explicitly coupled) for our model problem ((2.1) and (2.2)) as a preconditioner. The Newton step (2.20) can be expressed as:

$$\begin{bmatrix} J_{\phi,\phi} & J_{\phi,T} \\ J_{T,\phi} & J_{T,T} \end{bmatrix} \begin{bmatrix} \delta\phi \\ \delta T \end{bmatrix} = - \begin{bmatrix} F_\phi \\ F_T \end{bmatrix}, \tag{2.30}$$

with

$$\delta\mathbf{u} = [\delta\phi, \delta T]^T, \qquad \mathbf{F}(\mathbf{u}) = [F_\phi, F_T]^T.$$

Typically in problems such as this, simplification is made to the operators $J_{T,T}$ and $J_{\phi,\phi}$ within the preconditioner. As an example, one can linearize the temperature dependent coefficients, $\Sigma_a(T)$, $\Sigma_f(T)$, as $\Sigma_a(T^n)$, $\Sigma_f(T^n)$, thus simplifying the evaluation of $J_{T,T}$ and $J_{\phi,\phi}$. After enough simplification it may be more appropriate to call the preconditioning operators $P_{T,T}$ and $P_{\phi,\phi}$, as they may be very poor approximations of the system Jacobian. A multigrid method is often employed to smooth the two separate elliptic systems in the preconditioner, due to the potential of scalability of this approximate inversion method. This overall process is equivalent to using code coupling (*i.e.*, using the time discretization of (2.9) and (2.10)) as the system preconditioner.

Note that it is often key to carefully "engineer" the structure of $P_{T,T}$ and $P_{\phi,\phi}$; one desires that these matrices be easily and effectively addressed using multigrid but they must still include sufficient character of the coupled problem such that $\mathbf{P^{-1}}$ is an effective preconditioning process. This becomes quite important for larger, tightly coupled problems. To increase the effectiveness of the overall preconditioner without adding to the complexity of the diagonal blocks, one could add some source coupling while retaining a block lower triangular system such that:

$$\mathbf{P} = \begin{bmatrix} P_{\phi,\phi} & 0 \\ P_{T,\phi} & P_{T,T} \end{bmatrix}. \tag{2.31}$$

Then a preconditioning process $\delta\mathbf{u} \approx \mathbf{P}^{-1}(-\mathbf{F}(\mathbf{u}))$ is done in two steps:

1. Perform an approximate inversion of a parabolic system of order N (number of mesh nodes) to solve for $\delta\phi$ from
$$P_{\phi,\phi}\delta\phi = -F_\phi. \tag{2.32}$$
2. Perform an approximate inversion of a parabolic system of order N to solve for δT from
$$P_{T,T}\delta T = -F_T - P_{T,\phi}\delta\phi. \tag{2.33}$$

Fig. 7 shows the effectiveness of this blocked preconditioner strategy on a three dimensional coupled thermomechanical problem [6]. The diagram shows a result that does not employ preconditioning of the GMRES linear solver as a reference case. The other curves show a diagonal preconditioner $\mathbf{P}$ consisting of only the diagonal blocks $P_{\phi,\phi}$ and $P_{T,T}$, then a system that adds the lower triangular block $P_{T,\phi}$. Finally, a curve is shown that employs a second block sweep to the $L+D$ approach. Note that simple diagonal preconditioning of this example results in a significant decrease in the number of GMRES iterations to provide a given linear residual value. As one adds progressively more complexity ($L+D$ blocks, then two sweeps of $L+D$) the performance steadily improves further. Clearly, preconditioning is an important component to the overall solution strategy, and is quite effective in reducing the number of GMRES iterations required for a given level of accuracy even on this simple problem. Again, minimizing the number of GMRES iterations is paramount due to the function evaluations that form the action of the Jacobian that is the heart of GMRES.

For finite element problems (2.18), note that the Jacobian is

$$\mathbf{J} = \frac{\partial}{\partial \mathbf{u}}[\mathbf{M}(\mathbf{u})\dot{\mathbf{u}} + \mathbf{K}(\mathbf{u})\mathbf{u} - \mathbf{f}(\mathbf{u})]. \tag{2.34}$$

One may, upon analysis of the relative importance of the terms, choose to neglect portions of the complete finite element problem to develop a physics based preconditioner. In is case, one uses experience and intuition to neglect components, such as the source term and time derivative in the above. In this case, the approximate Jacobian of interest might be

$$\tilde{\mathbf{J}} = [\mathbf{K}(\mathbf{u})\mathbf{u}]' = \mathbf{K}(\mathbf{u})\mathbf{u}' + [\mathbf{K}(\mathbf{u})]'\mathbf{u}. \tag{2.35}$$

In many cases, linearizations of the stiffness matrix of the finite element problem might be convenient (or inexpensive) to form, one might consider using a linearization to form a *Picard* preconditioner [29],

$$\mathbf{P} = \mathbf{K}(\mathbf{u}^k), \tag{2.36}$$

that ignores the derivative of the stiffness matrix for economy and perhaps ease of approximate invertibility by a multigrid process. Variations of this approach, as well as comparison to other simplified stiffness matrix approaches were studied. Here, solutions of Laplace's equation in harmonic coordinates [40]

$$\Delta x^i \equiv \frac{1}{\sqrt{g}}\frac{\partial}{\partial u^\alpha}\left(\sqrt{g}g^{\alpha\beta}\frac{\partial x^i}{\partial u^\beta}\right) = 0, \tag{2.37}$$

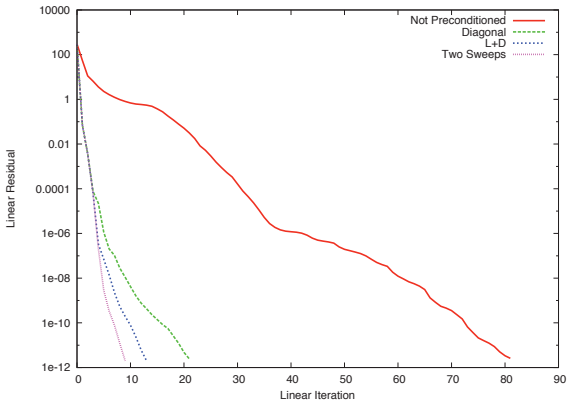


Figure 7: Performance of GMRES linear solver on an example problem described in [6], employing the blocking strategy discussed in (2.31). Plot show results for no preconditioning, diagonal (only the diagonal blocks $P_{\phi,\phi}$ and $P_{T,T}$ are nonzero), lower triangular (2.31), and (2.31) using two Gauss-Seidel relaxation sweeps.

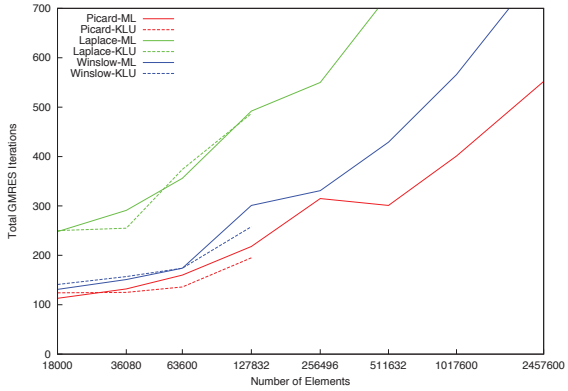


Figure 8: Performance of the Laplace-Beltrami preconditioner set applied to the mesh generation problem in [29]. Three preconditioner forms are considered here, an unstructured Laplace problem, a modified Winslow-Crowley problem without cross-terms, and a Picard linearization of the stiffness matrix. Two variants are shown in each case, the solid lines indicate that ML was used to obtain the inverse, and the dotted lines employed KLU (see text). KLU results stop at 127832 elements due to the space needed to store the inverse of the preconditioner. Vertical axis indicates the total number of GMRES iterations required to achieve complete convergence. In all cases, 5 Newton iterations were used.

are studied. Following discretization using the finite element method, the stiffness matrix K_{mn} is

$$K_{mn} = \int_\Omega \frac{\partial \psi_m}{\partial u^\alpha} \sqrt{g} g^{\alpha\beta} \frac{\partial \psi_n}{\partial u^\beta} d^3u. \tag{2.38}$$

The Picard preconditioner considered is the stiffness matrix linearized about the current Newton solution (or the initial guess). Two other simpler preconditioners were studied, a pure Laplacian stiffness form

$$K^{\text{Lap}}_{mn} = \int_\Omega \frac{\partial \psi_m}{\partial u^\alpha} \frac{\partial \psi_n}{\partial u^\beta} d^3u, \tag{2.39}$$

and a *Winslow-Crowley* form

$$K_{mn}^{\mathrm{WC}} = \int_{\Omega} \frac{\partial \psi_m}{\partial u^{\alpha}} \sqrt{g} g^{\delta_{\alpha\beta}\delta_{\alpha\beta}} \frac{\partial \psi_n}{\partial u^{\beta}} d^3u. \tag{2.40}$$

The linearized form of the stiffness matrix ((2.38), (2.39), or (2.40)) is symbolically **P** that is then passed to the ML algebraic multigrid package in Trilinos [41,42] for approximate inversion to form $\mathbf{P}^{-1}$. The ML package provides the mathematical operation (2.29).

Fig. 8 shows the performance of the Laplace–Beltrami solution (2.37) on an example from [43]. These results indicate the scalability of the GMRES Krylov solver used in the JFNK solution; the performance of the linear solver in this case is completely determined by the effectiveness of the preconditioner used. In this case, the Picard preconditioner is the best performing of the three, but it falls short of providing scalability (*i.e.*, the number of iterations taken by the linear solver increase with problem size). The Laplace-Beltrami problem considered is a very challenging numerical problem due to its nonlinearity and stiffness [44]. It is typically difficult to obtain scalability on highly nonlinear problems of this sort, on complex three dimensional domains such as this one. As such, this example serves as an excellent test problem on which to study physics based preconditioning ideas.

Several interesting points are apparent in Fig. 8. Note that the dotted lines show the convergence behavior obtained using a preconditioner that is based on direct inversion of each of the three preconditioners shown (KLU). This data was not continued beyond 127832 elements as the full inverse of the next largest problem could not be stored in the 2^{31} bit address space used to solve the problem. Secondly, the ML algebraic multigrid approach was nearly as effective as the direct approach, at least up to 127832 elements. Here, parameters appropriate for smoothed aggregation were chosen and ILU(0) was used as a smoother (using one sweep, zero overlap and fill in). KLU was used as a coarse grid solver, and a full-MGV strategy was employed limited to a maximum of six levels. Also note that the Picard preconditioner provides the best performance. Given this data, it is clear that ML is a very effective approximate inversion method for the three preconditioners studied. Further, even in the case of the Picard preconditioner, too much of the relevant physics has been removed from **P** to provide a scalable preconditioner for this problem. To conclude, 5 Newton iterations and 550 Picard-ML-preconditioned GMRES iterations were needed to solve this 7.4×10^6 degree of freedom (DOF) problem to steady state on the three dimensional horseshoe geometry in [43]. Other forms of physics-based preconditioning are discussed in [4].

2.5 JFNK as a multiscale solver

This section considers the use of the JFNK solver to simultaneously couple solutions at significantly difference scales. For example, assume that thermal conductivity impacts

heat conduction within a material according to a solution of a mathematical model,

$$\begin{aligned} &\rho c_p \frac{\partial T}{\partial t} - \nabla \cdot \mathbf{q} - Q = 0, && T \in \Omega, \\ &\mathbf{n} \cdot \mathbf{q} = q(T), && T \in \Gamma^C, \\ &\mathbf{n} \cdot \mathbf{q} = 0, && T \in \Gamma^T \cup \Gamma^B, \\ &T(t=0) = T_0, && T \in \Omega, \end{aligned} \tag{2.41}$$

where T is the internal temperature and that heat is generated within the bulk material at a uniformly distributed constant rate, Q. Further, assume that the object being analyzed is a cylindrical geometry, where Ω is the body of the cylinder, Γ^T denotes the top and Γ^B denotes the bottom boundary of the material, and Γ^C denotes the outer circumferential boundary such that $\Gamma = \Gamma^T \cup \Gamma^B \cup \Gamma^C$, and $q(T)$ is an expression of the heat rejected to surroundings around the outside radius of the cylinder. The heat flux $\mathbf{q}$ within Ω may be written as

$$q_i = -k \nabla T, \tag{2.42}$$

where k is the isotropic thermal conductivity of the material making up the cylinder. The heat conduction problem is further simplified in this example by assuming steady-state conditions, i.e. $\partial T / \partial t = 0$.

Further, assume that the value of k depends on the microstructure of the material, and that the microstructure changes over time. In this case, the evolving microstructure will influence the thermal conductivity of the material, *e.g.*, the curing of moist concrete or irradiation of nuclear fuel. In the case of fuel, the microstructure at a given point in time depends on both the irradiation rate and the temperature history of the material. A typical fuel thermal calculation might use an experimentally derived correlation such as those seen in [45] to account for the microstructure dependence of k. In this example, we assume that the value of k is determined by implicitly coupling a macroscale thermal calculation of the cylindrical problem (2.41) to a mesoscale phase field model that evolves k as a function of irradiation and temperature history.

Linking calculations between scales is relatively common [46–48]. However, linking mutually dependent (or coupled) multiscale simulations is more challenging, since operator splitting may result in both stability and accuracy issues as discussed above. In addition, the proposed multiscale calculation has significant diversity in spatial scales between the engineering and mesoscale solutions. To address both of these considerations, evaluation of the mesoscale physics within the JFNK *function evaluation process* is proposed; where the nonlinear engineering scale calculation is consistently coupled with the mesoscale solution and then solved using the JFNK algorithm presented in Section 2.3. Performing the mesoscale calculation within the function evaluation process eliminates the need to write down the derivatives of the state of the mesoscale properties, which may be difficult or impossible to obtain analytically. The concept of using the JFNK method to bridge scales in this manner was developed by Knoll [31], who employs JFNK as an accelerator for reactor neutronics calculations.

2.5.1 Macroscale (coarse) discretization

The macroscale model begins with a fully coupled finite element discretization of the heat equation (2.41) in a UO_2 fuel pellet. To develop this approximation, the coarse test space V is approximated by V^h using linear Lagrange finite elements such that $V^h \subset V$ and $V^h = \text{span}\{\phi_i\}_{i=1}^n$. The trial space hosting the engineering scale solution is similarly approximated, where $v \in V^h$ is

$$v(t,\mathbf{x}) = \sum_{j=1}^{n} v_j(t)\phi_j(\mathbf{x}), \tag{2.43}$$

which allows the governing equations for heat conduction to be written in weak form as in [49]

$$\mathbf{F}(\mathbf{T}) = \left(k\nabla T^{k+1}, \nabla\phi_i\right) - (Q,\phi_i). \tag{2.44}$$

The JFNK solution method as outlined in Section 2.3 is used to solve this residual equation. To precondition this equation, one could choose $P_{ij} = k\left(\nabla\phi_i, \nabla\phi_j\right)$ in (2.29).

Unlike [49], in this example the thermal conductivity k is a function of temperature and irradiation, and is based on the macroscale temperature field T coupled to the mesoscale model as demonstrated in Part II [1] of this paper. The coupling strategy is depicted in Fig. 9. Here, at each integration point of the macroscale calculation a call to the mesoscale code is initiated, and the current value of temperature at that location is passed to the mesoscale code as input. The mesoscale model then evolves the microstructure for a specified amount of time, at the provided temperature value. Finally, the mesoscale thermal conductivity is returned to the macroscale calculation, and is utilized within the residual calculation.

The matrix free solution strategy ensures that this solution method is self-consistent.

2.5.2 Algorithm for JFNK scale bridging

The spatial bridging method employed here is similar to the method proposed by Wagner and Liu [50], in which the multiscale problem is divided into coarse and fine spaces. Here, the heat flux in (2.42) is represented as

$$\mathbf{q} = \bar{\mathbf{q}} + \mathbf{q}', \tag{2.45}$$

where $\bar{\mathbf{q}}$ is the coarse space heat flux field and $\mathbf{q}'$ is the fine space flux field. The coarse space is the macroscale domain Ω, while the mesoscale solution exists only in the vicinity of the integration points of the macroscale mesh. Further, the spatial projection of the mesoscale domain is assumed to be of measure zero in the engineering scale.

Unlike [50], decoupling the spatial scales is accomplished here using Fourier's Law

$$q_i = -k(t,\mathbf{r},T)\nabla T, \tag{2.46}$$

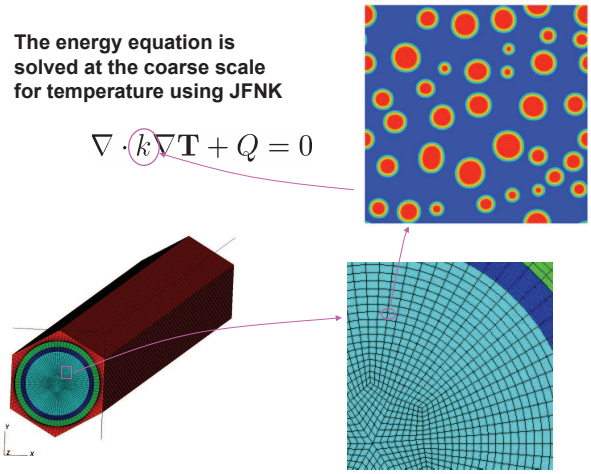


Figure 9: Pictorial of the scale bridging method used to calculate the effective thermal conductivity at the macroscale. At each integration point, the current macroscale temperature is passed to the mesoscale and the mesoscale model evolves the microstructure. Next, the effective thermal conductivity k is "homogenized" and passed back to the macroscale, all within a given function evaluation call. The heat conduction equation is then solved using the preconditioned JFNK algorithm.

where the engineering scale heat flux $\mathbf{q}$ results from the temperature gradient at that scale, while the coarse thermal conductivity k represent a spatial homogenization over the mesoscale. The mesoscale model calculates void evolution over the fine space, resulting in the spatially-dependent mesoscale thermal conductivity $k'(\mathbf{r})$. We determine the homogenized coarse thermal conductivity k by solving a steady-state conductivity equation [51],

$$\nabla \cdot (k'(\mathbf{r}) \nabla T) = 0, \tag{2.47}$$

in the fine space using the evolved microstructure. Eq. (2.47) is discretized and solved using the same grid as that used for the phase field simulations. A constant-temperature boundary condition of $T_l = 800$ K is applied on the left boundary and a constant heat flux of $\mathbf{q} = 50\ \hat{\mathbf{j}}$ MW/m^2 is applied across the right boundary as illustrated in Fig. 10;

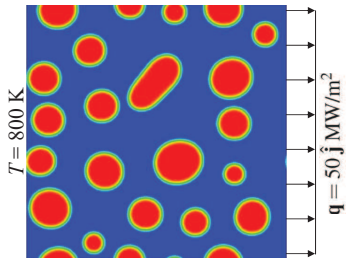


Figure 10: Effective thermal conductivity is obtained by solving (2.47) with a constant temperature ($T = 800$ K) on the left boundary and a constant heat flux of $\mathbf{q} = 50\ \hat{\mathbf{j}}$ MW/m^2 applied to the right boundary.

periodic boundary conditions are applied in the y-direction (note that the values of T_l and q' are arbitrarily selected and have no effect on the value of the calculated thermal conductivity). Assuming $\mathbf{q} = \mathbf{q}'$, k is determined with $q_i = -k\Delta T$, where ΔT is the difference between the temperature on the left boundary of the fine space ($T_l = 800$ K) and the average temperature on the right boundary. ∇T at the engineering scale equals ΔT at the mesoscale, since the mesoscale is of measure zero at the engineering scale.

Fig. 5 in Part II [1] of this paper shows the nonlinear convergence of the coupled multiscale calculation.

3 Conclusions

Part I of this paper motivated the need to carefully consider time and spatial integration for multiphysics problems. First, a simple example was used to demonstrate that operator splitting (or code coupling) in a coupled system will produce errors that vary with the size of the time step taken. Two numerical studies were presented; the first showed that both operator splitting and first order time integration resulted in significant errors in a nuclear engineering problem. In the second study, two kinetics problems were analyzed that show that operator splitting may be fully acceptable when the physics is weakly coupled, but not for strongly coupled problems.

Issues with spatial integration with multiple meshes were also studied. This paper concludes that, for accuracy, each solution on the mesh that hosts it must employ a resolution sufficiently fine to be in the asymptotic convergence range. Furthermore, all meshes must also be sufficiently fine over the domain of interest such that the data transfer mechanism used between solutions also yields the level of accuracy needed. This paper also described the modern algorithmic approach of JFNK, physics-based preconditioning, and the use of JFNK as a multiscale monolithic solution algorithm. In closing, Part II [1] of this work presents the software framework MOOSE that implements these ideas. Further, several applications based on these algorithms and software are presented, along with results.

Acknowledgments

The submitted manuscript has been authored by a contractor of the U.S. Government under Contract No. DE-AC07-05ID14517 (INL/JOU-10-20006). 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

[1] D. Gaston, L. Guo, G. Hansen, H. Huang, R. Johnson, D. Knoll, C. Newman, H. Park, R. Podgorney, M. Tonks, R. Williamson, Parallel algorithms and software for nuclear, en-

ergy, and environmental applications. Part II: Multiphysics software, Commun. Comput. Phys. 12 (2012) 834–865.
[2] D. A. Knoll, L. Chacon, L. G. Margolin, V. A. Mousseau, On balanced approximations for time integration of multiple time scales systems, J. Comput. Phys. 185 (2003) 583–611.
[3] D. L. Ropp, J. N. Shadid, Stability of operator splitting methods for systems with indefinite operators: Reaction-diffusion systems, J. Comput. Phys. 203 (2) (2005) 449–466.
[4] D. A. Knoll, D. E. Keyes, Jacobian-free Newton-Krylov methods: a survey of approaches and applications, J. Comput. Phys. 193 (2) (2004) 357–397.
[5] D. Gaston, C. Newman, G. Hansen, D. Lebrun-Grandié, MOOSE: A parallel computational framework for coupled systems of nonlinear equations, Nucl. Engrg. Design 239 (2009) 1768–1778.
[6] D. Gaston, G. Hansen, S. Kadioglu, D. Knoll, C. Newman, H. Park, C. Permann, W. Taitano, Parallel multiphysics algorithms and software for computational nuclear engineering, Journal of Physics: Conference Series 180 (1) (2009) 012012.
[7] H. Park, D. A. Knoll, D. R. Gaston, R. C. Martineau, Tightly coupled multiphysics algorithms for pebble bed reactors, Nuclear Science and Engineering 166 (2) (2010) 118–133.
[8] D. F. Griffiths, J. Sanz-Serna, On the scope of the method of modified equations, SIAM J. Sci. Statist. Comput. (1986) 994–1008.
[9] Argonne code center: Benchmark problem book, ANL-7416 supplement 2, Argonne National Laboratory (1977).
[10] T. Xu, E. L. Sonnenthal, N. Spycher, K. Pruess, TOUGHREACT user's guide: A simulation program for non-isothermal multiphase reactive geochemical transport in variable saturated geologic media, Tech. Rep. LBNL-55460, Berkeley National Laboratory, Berkeley, California (2004).
[11] M. D. White, B. P. McGrail, STOMP (Subsurface Transport over Multiple Phases) version 1.0 addendum: Eckechem equilibrium-conservation-kinetic equation chemistry and reactive transport, Tech. Rep. PNNL-15482, Pacific Northwest National Laboratory, Richland, Washington (2005).
[12] D. L. Ropp, J. N. Shadid, C. C. Ober, Studies of the accuracy of time integration methods for reaction-diffusion equations, J. Comput. Phys. 194 (2) (2004) 544–574.
[13] D. Estep, V. Ginting, D. Ropp, J. N. Shadid, S. Tavener, An A posteriori-A priori analysis of multiscale operator splitting, SIAM J. Numer. Anal. 46 (3) (2008) 116–1146.
[14] M. A. Puso, T. A. Laursen, Mesh tying on curved interfaces in 3D, Engineering Computations 20 (3) (2003) 305–319.
[15] M. W. Gee, G. A. Hansen, D. Andrs, Moertel mortar methods package, `http://trilinos.sandia.gov/packages/moertel`.
[16] P. Solin, J. Cerveny, L. Dubcova, I. Dolezel, Multi-mesh *hp*-FEM for thermally conductive incompressible flow, in: M. Papadrakakis, E. Onate, B. Schrefler (Eds.), ECCOMAS Conference on Coupled Problems, CIMNE, Barcelona, 2007, pp. 547–550.
[17] P. Solin, J. Cerveny, L. Dubcova, Adaptive multi-mesh *hp*-FEM for linear thermoelasticity, Tech. rep., Department of Mathematical Sciences, UTEP, research report No. 2007-08 (2008).
[18] P. Solin, L. Dubcova, J. Kruis, Adaptive *hp*-FEM with dynamical meshes for transient heat and moisture transfer problems, J. Comput. Appl. Math. 233 (2010) 3103–3112.
[19] P. Solin, J. Cerveny, L. Dubcova, D. Andrs, Monolithic discretization of linear thermoelasticity problems via adaptive multimesh *hp*-FEM, J. Comput. Appl. Math. 234 (2010) 2350–2357.
[20] L. Dubcova, P. Solin, G. Hansen, H. Park, Comparison of multimesh *hp*-FEM to interpolation and projection methods for spatial coupling of thermal and neutron diffusion calculations,

J. Comput. Phys. 230 (4) (2011) 1182–1197.
[21] R. K. Jaiman, X. Jiao, Geubelle, P. H., E. Loth, Assessment of conservative load transfer for fluid-solid interface with non-matching meshes, Internat. J. Numer. Methods Engrg. 65 (15) (2005) 2014–2038.
[22] X. Jiao, M. T. Heath, Common-refinement-based data transfer between non-matching meshes in multiphysics simulations, Internat. J. Numer. Methods Engrg. 61 (14) (2004) 2402–2427.
[23] J. Grandy, Conservative remapping and region overlays by intersecting arbitrary polyhedra, J. Comput. Phys. 148 (1999) 433–466.
[24] R. W. Johnson, G. Hansen, C. Newman, The role of data transfer on the selection of a single vs. multiple mesh architecture for tightly coupled multiphysics applications, Appl. Math. Comput. 217 (2011) 8943–8962.
[25] I. Prigogine, R. Lefever, Symmetry breaking instabilities in dissipative systems. II, J. Chem. Phys. 48 (4) (1967) 1695–1700.
[26] Y. Wang, J. Ragusa, Application of *hp* adaptivity to the multigroup diffusion equations, Nuclear Science and Engineering 161 (2009) 22–48.
[27] P. Solin, K. Segeth, I. Dolezel, Higher-Order Finite Element Methods, Chapman & Hall/CRC Press, Philadelphia, PA, 2003.
[28] G. Hansen, S. Owen, Mesh generation technology for nuclear reactor simulation; barriers and opportunities, Nucl. Engrg. Design 238 (10) (2008) 2590–2605.
[29] M. Berndt, J. D. Moulton, G. Hansen, Efficient nonlinear solvers for Laplace-Beltrami smoothing of three-dimensional unstructured grids, Comput. Math. Appl. 55 (12) (2008) 2791–2806.
[30] J. W. Bates, D. A. Knoll, On consistent time integration methods for radiation hydrodynamics in the equilibrium diffusion limit: Low energy density regime, J. Comput. Phys. 167 (2001) 99–130.
[31] D. Knoll, R. Park, K. Smith, Application of the Jacobian-free Newton-Krylov method in computational reactor physics, in: American Nuclear Society 2009 International Conference on Advances in Mathematics, Computational Methods, and Reactor Physics, Saratoga Springs, NY, 2009.
[32] M. R. Tonks, G. Hansen, D. Gaston, C. Permann, P. Millett, D. Wolf, Fully-coupled engineering and mesoscale simulations of thermal conductivity in UO_2 fuel using an implicit multiscale approach, Journal of Physics: Conference Series 180 (1) (2009) 012078.
[33] M. Tonks, D. Gaston, C. Permann, P. Millett, G. Hansen, D. Wolf, A coupling methodology for mesoscale-informed nuclear fuel performance codes, Nucl. Engrg. Design 240 (10) (2010) 2877–2883.
[34] Y. Saad, Iterative Methods for Sparse Linear Systems, The PWS Series in Computer Science, PWS Publishing Company, Boston, MA, 1995.
[35] P. N. Brown, Y. Saad, Hybrid Krylov methods for nonlinear systems of equations, SIAM J. Sci. Statist. Comput. 11 (3) (1990) 450–481.
[36] T. F. Chan, K. R. Jackson, Nonlinearly preconditioned Krylov subspace methods for discrete Newton algorithms, SIAM J. Sci. Statist. Comput. 5 (3) (1984) 533–542.
[37] D. Alur, D. Malks, J. Crupi, Core J2EE Patterns: Best Practices and Design Strategies, 2^{nd} Edition, Prentice Hall, New Jersey, USA, 2003.
[38] R. S. Dembo, S. C. Eisenstat, T. Steihaug, Inexact Newton methods, SIAM J. Numer. Anal. 19 (1982) 400–408.
[39] P. R. McHugh, D. A. Knoll, Inexact Newton's method solutions to the incompressible

Navier-Stokes and energy equations using standard and matrix-free implementations, AIAA J. 32 (1994) 2394.

[40] G. A. Hansen, R. W. Douglass, A. Zardecki, Mesh Enhancement: Selected Elliptic Methods, Foundations, and Applications, Research monograph, Imperial College Press, London, UK, 2005.

[41] M. Heroux, et al., Trilinos: an object-oriented software framework for the solution of large-scale, complex multi-physics engineering and scientific problems, `http://trilinos.sandia.gov` (2008).

[42] 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, K. S. Stanley, An overview of the trilinos project, ACM Trans. Math. Softw. 31 (3) (2005) 397–423.

[43] G. Hansen, A. Zardecki, D. Greening, R. Bos, A finite element method for three-dimensional unstructured grid smoothing, J. Comput. Phys. 202 (1) (2005) 281–297.

[44] L. Chacon, G. Lapenta, A fully implicit, nonlinear adaptive grid strategy, J. Comput. Phys. 212 (2) (2006) 703–717.

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

[46] P. Bochev, M. Christon, S. Collis, R. Lehoucq, J. Shadid, A. Slepoy, G. Wagner, A mathematical framework for multiscale science and engineering: the variational multiscale method and interscale transfer operators, Tech. Rep. SAND2004–2871, Sandia National Laboratories (Jun. 2004).

[47] Q. Yu, J. Fish, Multiscale asymptotic homogenization for multiphysics problems with multiple spatial and temporal scales: a coupled thermo-viscoelastic example problem, Int. J. of Solids and Structures 39 (2002) 6429–6452.

[48] J. Michopoulos, C. Farhat, J. Fish, Modeling and simulation of multiphysics systems, J. Comput. Inf. Sci. Eng. 5 (3) (2005) 198.

[49] C. Newman, G. Hansen, D. Gaston, Three dimensional coupled simulation of thermomechanics, heat, and oxygen diffusion in UO_2 nuclear fuel rods, Journal of Nuclear Materials 392 (2009) 6–15.

[50] G. Wagner, W. Liu, Coupling of atomistic and continuum simulations using a bridging scale decomposition, J. Comput. Phys. 190 (2003) 249–274.

[51] P. C. Millett, D. Wolf, T. D. Desai, S. Rokkam, A. El-Azab, Phase-field simulation of thermal conductivity in porous polycrystalline microstructres, J. Appl. Phys. 104 (2008) 033512.