Nuclear Engineering and Design 239 (2009) 566–579

Contents lists available at ScienceDirect

Nuclear Engineering and Design

journal homepage: www.elsevier.com/locate/nucengdes

Consistent and accurate schemes for coupled neutronics thermal-hydraulics reactor analysis

Jean C. Ragusa*, Vijay S. Mahadevan

Texas A&M University, Department of Nuclear Engineering, College Station, TX 77843-3133, USA

ARTICLE INFO

Article history:
Received 7 May 2008
Received in revised form
15 November 2008
Accepted 20 November 2008

ABSTRACT

Conventional coupling paradigms currently used to couple different physics components in reactor analysis problems can be inconsistent in their treatment of the nonlinear terms due to the operator-split (OS) strategies employed. This leads to the usage of small time steps to maintain accuracy requirements, thereby increasing the overall computational time. This paper proposes some remedies to OS techniques that can restore consistency in the coupling of the nonlinear terms and explores high-order mono-block nonlinearly consistent techniques with time step control. The performance of the methods was studied for several transient scenarios using a 0D point-kinetics/thermal-hydraulics lumped model and a 1D neutronics/heat conduction/enthalpy balance model. The results prove that consistent approximations can be made to enhance the overall accuracy in conventional codes with simple nonintrusive techniques. Additionally, an analysis of a mono-block coupling strategy (without having recourse to an OS strategy) is carried out to assess automated time stepping control using higher order Implicit Runge–Kutta (IRK) schemes. The conclusions from these results indicate that nonlinearly consistent adaptive time stepping methods can provide better accuracy and reliability in the solution fields than constant time stepping methods, even for transients with rapid and discontinuous variations.

1. Introduction

Reliable and accurate numerical simulation of physical phenomena often requires simultaneous description of several physics components. In most cases, these physics are coupled in a nonlinear fashion, making it complicated to find the solution efficiently and accurately. There are several examples of such nonlinear physical phenomena in the field of nuclear engineering, including radiative transfer (Lowrie, 2004), nuclear reactor analysis (Woodruff, 1984; Housiadas, 2002), and fluid-structure interaction (Carlos et al., 2001). For decades, the modeling of nuclear cores has been divided into several distinct domains of physics: neutronics, which solves for the neutron population distribution; hydraulics, which solves for the moderator density and temperature fields; heat transfer, which solves for the temperature fields within the nuclear fuel, etc. Yet, these isolated physical models in reality describe physical processes that are intertwined and rely heavily on the solution field of one another's. In the last decade or so, various existing mono-disciplinary codes have been coupled together in a simple black-box fashion, where the output of one code serves as the input of another code. The schematics of such models is shown on Fig. 1, where each physics component is solved by an independent mono-physics code and the data from one code is exchanged through message passing paradigms such as PVM or MPI. In order to solve a given physics component, the other physics lag behind; this is typically achieved using a simple explicit linearization of the nonlinear coupling terms, but produces solutions which are of low-order accuracy due to the inconsistent treatment of the nonlinearities. Such schemes, which are still the main coupling paradigm today for solving nonlinear nuclear reactor physics equations, are based on a linearization that is never resolved.

In reactor analysis, numerous neutronics/thermal-hydraulics (N/TH) codes have been coupled in such a fashion: see for example the OECD/NEA – US/NRC PWR Main Steam-Line Break Benchmark (Taylor et al., 2002) and the OECD/NEA – US/NRC Boiling Water Reactor Turbine Trip Benchmark (Akdeniz et al., 2006), where all participants used the conventional coupling techniques described above. This coupling technique is also mathematically described as operator-splitting (OS) in the sense that the action of the governing equations on the variables (neutron flux, temperatures, etc) is decomposed into each individual physics. Due to its inconsistent treatment of the nonlinear terms, we hereafter denote such schemes as Nonlinearly-Inconsistent Operator-Split or NI-OS. In this paper, and following a common practice in coupled

* Corresponding author. Tel.: +1 979 862 2033.
E-mail addresses: ragusa@ne.tamu.edu (J.C. Ragusa), vijaysm@tamu.edu (V.S. Mahadevan).

0029-5493/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.nucengdes.2008.11.006

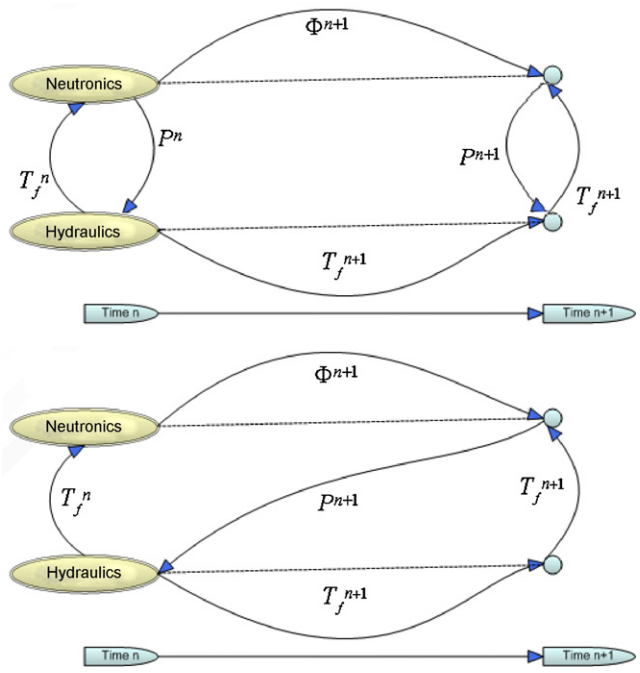

Fig. 1. Conventional operator-split coupling techniques. Top: simultaneous update; bottom: staggered update.

multi-physics simulations (see, for instance, Lowrie, 2004; Knoll and Keyes, 2004; Knoll et al., 2003), we refer to methods that lag behind one of the physics components and do not convergence nonlinearities within a computational time step as "nonlinearly inconsistent", i.e., not all terms in the nonlinear residuals are evaluated at the same time. "Nonlinearly consistent" methods evaluate the terms in the residual at the same time and converge the nonlinearities to a small tolerance. "Nonlinearly inconsistency" should not be confused with the mathematical concept of consistency, which requires that the discretized equations converged to the continuum partial differential equations as the mesh size (here, the time step size) goes to zero. More generally, an operator-split method follows a "divide and conquer" strategy by decomposing a system of coupled differential equations into simpler sub-systems and provides weak interaction between the different sub-systems to mimic the actual coupling. Should an attempt at iterating such strategies occur over a time step, the nonlinearities could be reduced or resolved but at a consequential CPU cost though; hence, an iteration is seldom performed, keeping the nonlinearities in the system due to the coupling terms unresolved and reducing the accuracy in the time stepping procedure to a first-order convergence in time, or $O(\Delta t)$. A few OS schemes that are second-order accurate in time have been devised (Curtis et al., 2004), but have not been employed in reactor analysis and we are not aware of higher order OS schemes that preserve the temporal accuracy. Despite these drawbacks, OS is still one of the major coupling paradigms used today for solving coupled physics systems. The direct implication of the loss of convergence order is typically addressed by using smaller time step sizes to achieve a reasonable accuracy. This, in turn, increases the computational time and resource requirements. Nonetheless, operator-split allows for the re-use of legacy mono-disciplinary codes, thereby inheriting many man-years of code development and validation and verification (V&V) efforts (though most of the V&V concerned the individual codes and little V&V has been carried out for coupled codes so far).

The coupled simulations of present and future nuclear reactors will rely increasingly on multi-dimensional, multi-physics computations, and these calculations should be performed in amenable wall-clock times and should yield highly accurate solutions. In that respect, the fundamental inefficiency of the current NI-OS methods needs to be tackled to create Nonlinearly Consistent (NC) schemes. Two paths are being pursued and reported here. One option consists in restoring the accuracy and convergence order of NI-OS through simple modifications or fixes. The second option involves recasting the whole nonlinear problem into a single block to be solved using Newton's method; thus, the explicit linearization of the nonlinear terms can be completely avoided and higher order Runge–Kutta (RK) methods for time integration can be employed in a very straightforward way. Nonetheless, Newton's method requires the Jacobian matrix of the nonlinear operator; this issue is addressed using a simple finite difference approximation of the action of the Jacobian matrix on a vector, a technique usually referred to as "Jacobian-free" which we will discuss later in the paper.

Finally, all time discretizations of practical interest are implicit ones. Indeed, the equations considered in reactor analysis are stiff, i.e., they contain modes of greatly disparate time scales, which forbids the use of explicit time integration whose stability requirements would lead to exceedingly small time steps.

This paper is organized as follows. In Section 2, the conventional coupling strategies and their deficiencies are reviewed. Next, in the same Section, the techniques available to either mend the traditional OS scheme or reformulate the whole problem as a Newton's solve. The time integration techniques employed here are discussed in Section 3. Section 4 describes the models used in demonstrating the various features of the coupling techniques described in this paper. Sections 5 and 6 present our numerical results, respectively for point kinetics test cases and one-dimensional test cases. Concluding remarks are offered in Section 7.

2. Description of coupling strategies

We review here the current conventional operator-split technique and describe (1) some simple improvements over the traditional OS technique and (2) time adaptive techniques based on a recasting of the problem using Newton's method.

2.1. Conventional operator-split coupling

A simple schematic, illustrating the time marching process of a neutronics/thermal-hydraulics system, coupled via the conventional OS strategies, is shown in Fig. 1. For instance, Fig 1(top) presents a block Jacobi type of coupling (simultaneous update), where each physics component advances simultaneously in time and then exchanges relevant data; each 'block' representing one physic component. Fig. 1(bottom) can be viewed as a block Gauss–Seidel type of coupling (staggered update), when one physic component is solved first and then passes on some data to the other component. It must be noted that, although the terminology of Jacobi and Gauss–Seidel schemes might be useful in description the techniques used, only one such iteration is performed per time step: the processes are not iterated, resulting in nonlinearly inconsistent schemes which are, at most, first order in time. These conventional coupling concepts spring from the fact that specialized mono-disciplinary codes are readily available and data communication at rendezvous points is easily achieved through message passing paradigms. To further illustrate the lost of accuracy of traditional coupling schemes, we write down the system of nonlinear differential equations, resulting from the coupled physics as follows:

$$y' = f(t, y) = L(t)y + N(t, y) \tag{1}$$

where y is the vector of unknowns, t represents time, f is a nonlinear vector-valued function, L is a linear operator, and N is a strictly nonlinear operator. For illustration purposes, applying the implicit Crank–Nicholson (CN) discretization scheme over the interval $[t^n, t^{n+1}]$ to Eq. (1) yields

$$\left[I - \frac{\Delta t}{2}L(t^{n+1})\right] y^{n+1} = \left[I + \frac{\Delta t}{2}L(t^n)\right] y^n + \frac{\Delta t}{2}[N(t^{n+1}, y^{n+1}) + N(t^n, y^n)] \quad (2)$$

Eq. (2) is nonlinear in the unknown y^{n+1} and has to be solved or approximated to some known precision to obtain a correct and accurate solution. This scheme is locally third order, $O(\Delta t^3)$, and hence is globally second order in time. Usually, in NI-OS schemes, a time explicit linearization of the nonlinear term is performed using a crude approximation such as

$$N(t^{n+1}, y^{n+1}) = N(t^{n+1}, y^n) + O(\Delta t) \quad (3)$$

Substituting Eq. (3) into Eq. (2) yields a locally $O(\Delta t^2)$ scheme, therefore, degrading the global convergence order to first order. Without converging the lagged nonlinearities in the coupling terms between physics, the global accuracy of the CN scheme has been reduced from second-order to first-order. This simple theoretical observation is quite general and illustrates the drawback of NI-OS methods and their inability to use higher order time integration schemes for better accuracy because of their handling of the coupling terms (the convergence accuracy degradation will also be demonstrated in the numerical results section). Additionally, having a scheme that is only first-order in time imposes the use of small time steps in order to obtain accurate solutions, thereby increasing the CPU time.

The current NI-OS coupling schemes, as seen from the above illustration are only first-order in time, regardless of the implicit time discretization technique chosen. Note that implicit methods are required for many problems due to the presence of stiff (i.e., rapidly varying) modes. In reactor analysis, these fast modes stem from the presence of the inverse neutron velocity terms. In order to achieve at least second-order convergence in time, second-order approximations that are nonlinearly consistent must be introduced. Some of the remedies covered in this paper include: (1) tackling the nonlinear inconsistency in OS schemes (presented in Section 2.2), (2) reformulating the whole system of equations as a nonlinear problem to be solved using Newton's method (Section 2.3).

2.2. Remedies to overcome drawbacks of traditional OS coupling schemes

2.2.1. Predict–evaluate–correct methods: higher order explicit linearization of the lagged nonlinear terms

For this class of methods, the idea is to employ a higher order linearization for the treatment of the lagged nonlinear terms. Specifically, a predicted value of y^{n+1} at time t^{n+1} (denoted by $y^{n+1,P}$) is to be derived in order to explicitly linearize the nonlinear term N with an accuracy of order q, i.e., $O(\Delta t^q)$, where q is the order of the time-integration numerical scheme. Then, the nonlinear term N can be evaluated with the predicted solution instead of the explicit solution at t^n (as was shown in Eq. (3)). This scheme is simply expressed as

$$N(t^{n+1}, y^{n+1}) = N(t^{n+1}, y^{n+1,P}) + O(\Delta t^q) \quad (4)$$

Replacing this approximation into Eq. (2), the resulting system of equations is then consistent (up to order q) and can be solved for the new value y^{n+1} (as in a corrector step). This prediction in the nonlinear term provides a better approximation to the traditional explicit linearization of the nonlinear system, and the entire process can be view as a predict–evaluate–correct (PEC) step.

The predictor step of Eq. (4) can be written as

$$y^{n+1,P} = y^n + \sum_i \alpha_i (\Delta t)^i \left.\frac{\mathrm{d}y}{\mathrm{d}t}\right|_{t^n} + O(\Delta t^q) \quad (5)$$

For instance, a second-order scheme is obtained by using:

$$y^{n+1,P} = 2y^n - y^{n-1} + O(\Delta t^2) \quad (6)$$

The method only involves approximating the derivatives of y, which can be expressed as a linear combination of the solution at previous temporal values. The PEC scheme can provide a more consistent approximation to the true solution and, therefore, the order of convergence that was lost in conventional schemes, can be restored. It should also be noted that the stability of this numerical method is still conditional, as with any NI-OS coupling strategy, including the conventional ones. For more details regarding the stability of implicit–explicit schemes, we refer the reader to (Gjesdal, 2007).

2.2.2. Picard iterations (fixed-point iterations)

Alternatively, one may view the time-discretized Eq. (2) as a nonlinear problem with variable y^{n+1} as the unknown. A simple technique for nonlinear root finding is the fixed-point iteration (FPI) process, which is also often referred to as Picard iteration in some other disciplines.

2.2.2.1. Unaccelerated Picard iterations. Picard iterations can fully restore the convergence order of a higher order time discretization scheme by solving the nonlinear problem iteratively. This eliminates the loss of accuracy due to crude explicit approximations such as Eq. (3). For illustrative purposes, during the time integration from t^n to t^{n+1}, the ℓ-th Picard step, when applied to Eq. (2), is given by

$$\left[I - \frac{\Delta t}{2}L(t^{n+1})\right] y^{n+1,\ell+1} = \left[I + \frac{\Delta t}{2}L(t^n)\right] y^n + \frac{\Delta t}{2}[N(t^{n+1}, y^{n+1,\ell}) + N(t^n, y^n)] \quad (7)$$

Basically, the nonlinearities are iterated upon until convergence within each time-step. Since such an iterative procedure at each time step may be computationally expensive, this strategy should be accelerated to obtain a faster nonlinear convergence.

2.2.2.2. Accelerated Picard iterations. To offset the potentially higher computational cost of Picard iterations, acceleration procedures can be employed to reduce the computational effort required for the solution process. The Aiken's Δ^2 acceleration (Stoer and Bulirsch, 1993) technique has been implemented for the Picard iterations. A brief introduction to this formulation is shown here.

For a given time step t^{n+1}, starting with the current Picard iterate y^ℓ, two unaccelerated fixed-point iterations are performed to obtain iterates $y^{\ell+1}$ and $y^{\ell+2}$. Then, the accelerated value, $\hat{y}^{\ell+2}$, is computed as follows:

$$\hat{y}^{\ell+2} = y^\ell - \frac{(\Delta y^\ell)^2}{\Delta^2 y^\ell} \quad (8)$$

where $\Delta y^\ell = y^{\ell+1} - y^\ell$ and $\Delta^2 y^\ell = y^{\ell+2} - 2y^{\ell+1} + y^\ell$. The whole process is repeated until convergence is reached. This method can be effective at reducing the total computational cost of iteration per time step. Eq. (8) is to be read component-wise.

Finally, an additional gain may be obtained by using a predicted value for the initial guess in the Picard iteration, by using Eq. (4) for instance, rather than initializing the process with the value from the previous time step.

2.3. Newton's technique applied to the whole coupled system (mono-block solution)

As described above, conventional operator-split (OS) strategies offer flexibility in the way the different physics are solved but involve inherent inconsistencies in some coupling terms. These can be mitigated if properly addressed. An alternative to OS techniques can be obtained by reformulating the whole coupled nonlinear system represented by Eq. (1) as a root finding problem for a vector-valued function:

$$F(y^{n+1}) = 0 \tag{9}$$

This equation was obtained after an implicit time discretization of Eq. (1) over the time interval $[t^n, t^{n+1}]$ was carried out. The vector-valued function F depends upon the physics (i.e., it contains vector-valued function f appearing on the right-hand-side of Eq. (1)) and upon the time discretization used. The unknowns in the nonlinear system (9) are the components of y^{n+1} at time t^{n+1}, which include (spatial) unknowns for all physics involved in the problem. In this mono-block solution technique, the nonlinear system (9) is solved using Newton's method. All nonlinearities inherent to the problem are thus resolved (to a set tolerance) in Newton's method. The main drawback of Newton's is the requirement of the Jacobian matrix of f, i.e., $\partial_y f$. For instance, using the CN time discretization, the Jacobian matrix arising in Newton's method is

$$J = I - \frac{\Delta t}{2}\partial_y f. \tag{10}$$

For any one-step implicit time integration technique, the Jacobian matrix will take the form $I - \gamma\Delta t\partial_y f$, where γ is a constant depending upon the numerical technique chosen. Therefore, reformulating the coupled problem as a monoblock solution opens up the possibility of employing higher order Implicit RK (IRK) schemes seamlessly. Such IRK schemes do not lend themselves to an easy or efficient implementation in OS strategies because the nonlinearities are not resolved (e.g., case of traditional OS strategies), making it absurd to utilize high-order schemes. Furthermore, it has long been known in numerical analysis that higher order schemes are known to converge faster when high accuracy results are desired, as opposed to using low-order schemes with a finer temporal discretization (see, for instance, Dormand, 1996).

Additionally, this reformulation enables us to use with relative ease adaptive time stepping techniques, which may lead to better usage of computer resources while providing answers to user-set tolerances. Indeed, it is well-established that IRK schemes can be devised with embedded formulae for error control and, therefore, allow for the simple implementation of adaptive time stepping (Hairer et al., 1987; Hairer and Wanner, 1996). By automatically controlling and selecting the time step size (and possibly rejecting some time steps during fast transient intervals), highly accurate solutions of the whole multi-physics simulation can obtained with the most sensible usage of available resources. These IRK embedded pairs provide efficient and inexpensive error control over the time step, therefore allowing for a fully automatic time step size control of the whole multi-physics simulation. Using time step control for the whole nonlinear system is a promising way to advance the various physics components of a simulation in a consistent fashion, while preserving a desired accuracy and sensibly managing the CPU resources. Hence, there are strong motivations for reformulating the whole nonlinear problem as a mono-block Newton iteration: (1) the assurance of converging the nonlinearities, and (2) the possibility of using efficient high-order time algorithms with time-step control.

The two key elements in successfully formulating the problem using Newton's methods are (1) avoiding the need for the Jacobian matrix J and (2) efficient preconditioning of the linear system arising in Newton's method.

Recent research by Knoll et al. (Knoll and Keyes, 2004; Knoll et al., 2003) provides concrete examples that Jacobian-Free Newton–Krylov (JFNK) techniques can be used to solve coupled multi-physics more efficiently than traditional methods. JFNK combines a Krylov subspace iterative method for its inner linear solver (typically GMRES is used), in which only matrix-vector $J \cdot v$ products are required. For instance, in JFNK, the matrix vector product is easily approximated using a finite difference formula:

$$J \cdot v \simeq v - \gamma\Delta t\frac{f(y^* + \epsilon v) - f(y^*)}{\epsilon} \tag{11}$$

where v is the current Krylov vector, y^* is the previous Newton iterate and ϵ is a small value. It is obvious from Eq. (11) that the matrix J never needs be formed, but only calls to the function f are required. Hence, it is usually straightforward to redirect the inner workings of existing mono-disciplinary codes to compute portions of the vector function f (one "portion" per physics). While using JFNK methods, it is also important to use good matrix preconditioners in order to limit the size of the Krylov subspace needs during each linear solve. A reasonable choice for the preconditioner would be the block diagonal of matrix J, where a 'block' is to be understood as a single physic component. Thus, the set of mono-disciplinary physics solvers used separately in the OS techniques can, in turn, be very valuable in preconditioning Newton's method. Such procedures are often referred to as "Physics-based preconditioned" JFNK techniques, and recent research by Mousseau (Mousseau, 2004; Mousseau et al. 2000) for coupled fluid flow and heat conduction problems and coupled radiation diffusion problems demonstrates the effectiveness of such preconditioners. In this paper, we analyze the performance of physics-based preconditioning for coupled reactor problems.

3. Time integration techniques used

We mostly have in mind employing implicit time integration schemes because of the presence of fast time scale modes in coupled neutronics/thermal-hydraulics calculations. Notably, numerical schemes and strategies for solving multi-physics coupled nonlinear stiff problems need to meet certain conditions, namely (a) Unconditional stability (A-stability) and, preferably, L-stability to avoid spurious oscillations, (b) Higher order accuracy in order to use larger time steps while still meeting user-specified accuracy criterion, and (c) a reduced average computational cost/step. Implicit Runge–Kutta (IRK) methods offer the possibility of higher accuracy orders, efficient treatment of the nonlinear terms, and ease in implementing time-adaptation strategies. Based on these criteria, several numerical methods and strategies for solving the OS and mono-block problems have been selected. In the current paper, the θ-time discretization method with constant step sizes is used to analyze the effects of nonlinear inconsistence and the remedies suggested in Section 2.2 to correct it; note that when $\theta = 1$, we retrieve the first-order Implicit Euler while $\theta = 1/2$ yields back the second-order Crank–Nicholson method. Obviously, these two methods can easily be recast as IRK schemes. IRK schemes with embedded formulae for error estimation will be primarily used in exploring the monoblock coupling strategy and the gains to be expected from time adaptation while retaining very precise and reliable solutions, even for transients involving sudden jumps and drops. We will not delve into the details of the IRK methods but simply state their most general form:

$$y^{n+1} = y^n + \Delta t\sum_{i=1}^{i=s} b_i Y_i \tag{12}$$

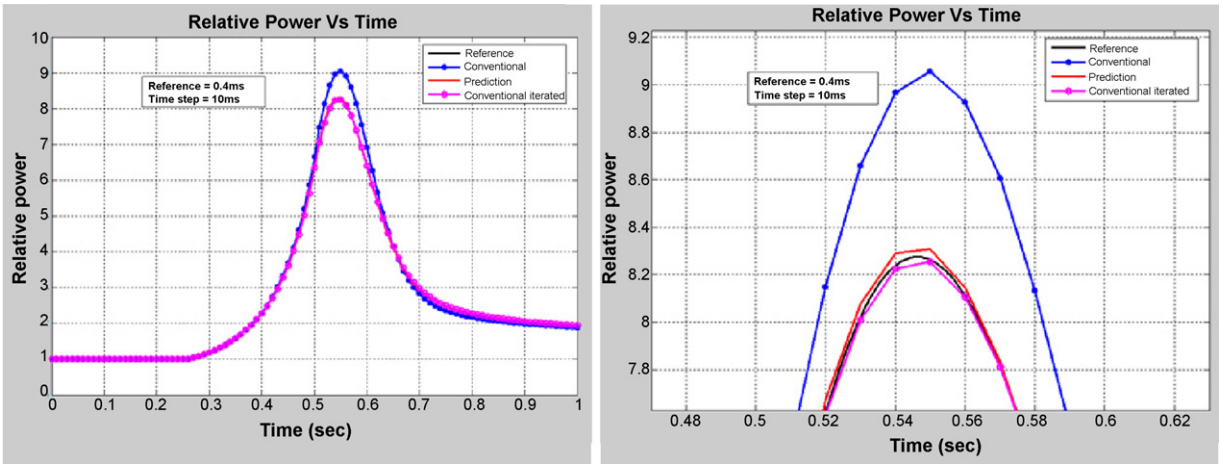

Fig. 2. Comparison of normalized power from various numerical schemes for the LWR rod-ejection test case. Left: power transient; right: enlarged at power peak.

where

$$Y_i = y^n + \Delta t \sum_{j=1}^{j=s} a_{ij} f(t^n + c_j \Delta t; Y_j) \quad \text{for } 1 \le i \le s \tag{13}$$

with the coefficient matrix a, and vectors b and c defining a particular Runge–Kutta (RK) method with s stages. If matrix a is strictly lower triangular, then the RK method is explicit in time; otherwise, it is implicit. If matrix a is lower triangular with a main diagonal consisting of identical nonzero terms, the method is said to be SDIRK (for Singly Diagonal IRK) and s nonlinear systems of size n are to be solved for the auxiliary values Y_i (with n the length of the solution vector y, which depends on the number of coupled physics and their spatial discretizations). Finally, if matrix a is full, then a single nonlinear system of size $s \times n$ needs to be solved to simultaneously provide all auxiliary values Y_i. The following RK methods were selected for our tests:

(1) DOPRI5 (Hairer et al., 1987), a 7-stage 5th-order explicit embedded scheme introduced by Dormand and Prince;
(2) GRK4A, GRK4T by Kaps and Rentrop (1979), two variants of 4-stage SDIRK methods with embedded lower-order formulae for error estimator (GRK4T is L-stable);
(3) SDIRK4 by Hairer and Wanner (1996), which is a 5-stage 4th-order embedded scheme that is both A-, L-stable.
(4) RADAU5, a 3-stage 5th-order fully implicit RK method based on the RADAU quadrature. This stiff integration scheme which is both A-, L-stable, has been tested by Hairer and Wanner (1996) for various stiff problems and has been proven to be successful in resolving the fast modes accurately.

More details concerning these techniques can be found in the cited references. To conclude this section related to time integration techniques, we specify the time step control procedures used here. Several such procedures have been devised for nonlinear dynamical systems by researchers Gustafsson (1991, 1994) and Watts (1984). In the current work, the ideas proposed by Gustafsson have been taken as the basis to implement the step control procedures. A standard step size controller can then be devised to choose a new size based on the local solution error alone without any dependence on the nature of the variation of the error over time.

$$\Delta t^{new} = S \Delta t^{old} \left(\frac{\tau}{err} \right)^{(1/(p+1))} \tag{14}$$

where p is the global order of the method, err is the estimated local truncation error (readily available with embedded methods), τ is the user-specified tolerance, and S is a safety factor used to account for the fact that the truncation error estimator only considers the leading order term of the error, which is only asymptotically correct as the time step size gets smaller. Such controllers have been proven to work well for explicit and one step Rosenbrock schemes but for complex algorithms such as in the RADAU5 solver, more intricate formulas to exploit the full potential of the stiff solvers are used. This has been covered in detail by Hairer and Wanner (1996).

4. Physical models used

Transient reactor analysis problems cover a broad spectrum of situations and the complex interaction between the different physics involved needs to be included when designing a model. In this paper, we will consider two simplified models of coupled neutronics/thermal-hydraulics: a Point Reactor Kinetics Equations (Duderstadt and Hamilton, 1976) and a 1D model. Both models will be used to demonstrate that conventional OS coupling techniques exhibit a reduced accuracy in the solution, while the remedies we propose for OS and the mono-block reformulation of the whole nonlinear problem restore the accuracy. First, the PRKE with feedback model employs the standard neutron point kinetic equations and couples them to simple 0D (core average) fuel heat conduction and fluid energy balance models via the reactivity function. In this model, the reactivity depends on fuel and coolant temperatures. Second, for our 1D model, 1D axial multi-group neutron diffusion and precursors equations are coupled to (i) a cylindrical fuel pellet heat conduction at each axial level (neglecting the axial heat conduction between pellets) and (ii) a 1D axial enthalpy balance. The equations for each of the models and physics are further detailed below.

4.1. PRKE with feedback

The PRKEs describe the transient behavior of the neutron distribution, or normalized power level (p), the delayed neutron precursor concentrations (ζ_j, $1 \le j \le J$ with $J = 6$), the decay heat components ω_k ($1 \le k \le K$ with $K = 3$), and the core average fuel and coolant temperatures, T_{eff}^{fuel} and T_{cool}. The resulting system of coupled nonlinear equations is given by

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} p \\ \zeta_1 \\ \vdots \\ \zeta_j \\ \vdots \\ \zeta_J \\ \omega_1 \\ \vdots \\ \omega_k \\ \vdots \\ \omega_K \\ T_{eff}^{fuel} \\ T_{\mathrm{cool}} \end{bmatrix} = \begin{bmatrix} \dfrac{\rho(t, T_{eff}^{fuel}, T_{\mathrm{cool}}) - \beta}{\Lambda} p + \dfrac{1}{\Lambda}\displaystyle\sum_{j=1}^{j=J} \lambda_j \zeta_j \\ \beta_1 p - \lambda_1 \zeta_1 \\ \vdots \\ \beta_j p - \lambda_j \zeta_j \\ \vdots \\ \beta_J p - \lambda_J \zeta_J \\ \kappa_1 p - \lambda_k^{FP} \omega_1 \\ \vdots \\ \kappa_k p - \lambda_k^{FP} \omega_k \\ \vdots \\ \kappa_K p - \lambda_K^{FP} \omega_K \\ \dfrac{\Omega_{pow}}{\rho_{fuel} C_p^{fuel}} \left\{ (1-\kappa) p + \displaystyle\sum_{k=1}^{k=K} \lambda_k^{FP} \omega_k \right\} - \dfrac{1}{\rho_{fuel} C_p^{fuel} \hat{R}_{th}} (T_{eff}^{fuel} - T_{\mathrm{cool}}) \\ -2\dfrac{u}{H}(T_{\mathrm{cool}} - T_{\mathrm{cool}}^{in}) + \dfrac{A_{fuel}}{A_{flow}} \dfrac{1}{\rho_{\mathrm{cool}} C_p^{cool} \hat{R}_{th}} (T_{eff}^{fuel} - T_{\mathrm{cool}}) \end{bmatrix} \tag{15}$$

with

p = normalized nuclear power (with $p(0) = 1.$)
ζ_j = precursor concentration
β_j = delayed neutron fraction
$\beta = \sum_{j=1}^{J} \beta_j$ = total delayed neutron fraction
Λ = neutron mean generation time (s)
$\rho(t, T_{eff}^{fuel}, T_{\mathrm{cool}}) = \rho_{ext}(t) + \alpha_{\mathrm{Doppler}}((T_{eff}^{fuel}(t))^d - (T_{eff,0}^{fuel})^d) + \alpha_{\mathrm{cool}}(T_{\mathrm{cool}}(t) - T_{\mathrm{cool}}, 0)$ = total reactivity
$\rho_{ext}(t)$ = external reactivity (e.g.,control rod)
$\alpha_{\mathrm{Doppler}}$ = Doppler coefficient
T_{eff}^{fuel} = fuel effective temperature
α_{cool} = coolant temperature coefficient
T_{cool} = coolant temperature
d = exponent for the Doppler law
$\Omega_{pow} = \dfrac{P_0}{V_{fuel}}$ = conversion factor to convert the normalized power p into power densities in $\mathrm{W/m^3}$
P_0 = total initial power in the entire core (MWth)
V_{fuel} = fuel volume in the entire core
$\kappa = \sum_{k=1}^{K} \kappa_k$ = total fraction of decay power
$A_{fuel} = \pi R_{fuel}^2$ = surface of a single fuel pin of radius R_{fuel} ($\mathrm{m^2}$)
A_{flow} = average flow area around a single fuel pin ($\mathrm{m^2}$)
H = reactor height (m)
u = inlet coolant speed (m/s)
$\hat{R}_{th} = A_{fuel}\left[\dfrac{1}{2\pi R_{gap} h_{gap}} + \dfrac{1}{2\pi k_{clad}} \ln\dfrac{R_{fuel}}{R_{gap}} + \dfrac{1}{2\pi R_{clad} h_{conv}(T_{\mathrm{cool}})} + \dfrac{w}{4\pi k_{fuel}(T_{eff}^{fuel})}\right]$ = fuel thermal resistance ($\mathrm{m^3\,C/W}$)
R_{gap} = gap radius (m)
h_{gap} = gap conductance ($\mathrm{W/m^2\,C}$)
$k_{fuel}(T_{eff}^{fuel})$ = temperature-dependent fuel conductivity (W/m C)
k_{clad} = clad conductivity (W/m C)
$h_{conv}(T_{\mathrm{cool}})$ = wall-coolant forced convection heat exchange coefficient ($\mathrm{W/m^2\,C}$)
$w = 4/9$ = weighting factor used in the effective fuel temperature formula

The effective temperature is computed using the fuel centerline and fuel pellet surface temperatures as follows:

$$\begin{aligned} T_{eff}^{fuel} &= wT_{centerline}^{fuel} + (1-w)T_{pellet\_surface}^{fuel} \\ &= wT^{fuel}(0) + (1-w)T^{fuel}(R_{fuel}) \end{aligned} \tag{16}$$

In this model, the neutronics equations (the first $1 + J + K$ equations in Eq. (15)) are coupled to the thermal-hydraulics equations (i.e., the last two equations) via the total reactivity ρ that contains the external reactivity term (e.g., control rod movement), the fuel temperature reactivity and the coolant temperature reactivity. In the thermal-hydraulics equations, the thermal resistance $\hat{R}_{th}$ accounts for the conduction through the fuel pellet, gap, and cladding and the convection at the clad-coolant interface. The lumped fluid equation is based on an average enthalpy conservation for a fluid being advected and heated in a channel of hydraulic cross sectional area A_{flow}. The convection heat exchange coefficient h_{conv} is given by the standard Dittus–Boelter single-phase correlation. The material properties (conductivities k_{fuel} and k_{clad}, densities ρ_{fuel} and ρ_{cool}, heat specific heat C_p^{fuel} and C_p^{cool}) are temperature-dependent.

The resulting system of equations forms a stiff nonlinear system of ODEs of the form

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t, y), \tag{17}$$

combining different physics that are nonlinearly coupled. For the coupled model described above in the zero-dimensional case, the number of unknowns involved is small. Due to such a simplification, an option for calculating the Jacobian matrix numerically by finite differences rather than using the Jacobian-free approach is feasible, when required. Additional data related to this model is given in Appendix A and in the references mentioned therein.

4.2. One-D models

In order to further verify the performance of the OS remedies and the monolithic solution procedure, we propose a 1D model for reactor transients, made of the following physics components:

- 1D axial neutron diffusion and precursors balance

$$\begin{aligned} \frac{1}{\nu}\partial_t \Phi^g &= \nabla \cdot D^g(z, T_{eff}, T_c)\nabla\Phi^g - \Sigma_r^g(z, T_{eff}, T_c)\Phi^g \\ &\quad + \chi^g \sum_{g'}^{G} \nu_p \Sigma_f^{g'}(z, T_{eff}, T_c)\Phi^{g'} \\ &\quad + \sum_{j=1}^{J} \chi_{d,j}^g \lambda_j C_j + \sum_{g'}^{G} \Sigma_s^{g' \to g}(z, T_{eff}, T_c)\Phi^{g'} \end{aligned} \tag{18}$$

$$\partial_t C_j = \sum_{g'}^{G} \nu_d \Sigma_f^{g'}(z, T_f, T_c)\Phi^{g'} - \sum_{j=1}^{J} \lambda_j C_j \tag{19}$$

  where the effective fuel temperature is again an average of the temperature profile obtained using the centerline and the fuel pellet values, as done previously;
- radial heat conduction at a given height z

$$\frac{\partial T^{fuel}}{\partial t} = \frac{1}{\rho_{fuel} C_p^{fuel}}\left[q''' - \frac{1}{r}\frac{\partial}{\partial r}\left(k(T^{fuel})\frac{\partial T^{fuel}}{\partial r}\right)\right] \tag{20}$$

  where q''' is the average fission power delivered in the nodal mesh located at height z (again, we neglect axial heat conduction between fuel pellets);
- 1D axial enthalpy conservation

$$\rho_{\mathrm{cool}} A_{flow}(\partial_t h + v\partial_z h) = P_{wet}\varphi_{th} \tag{21}$$

where $P_{wet} = 2\pi R_{fuel}$ is the perimeter of the fuel pin, and the heat flux at the wall φ_{th} is given by Newton's law of cooling

$$\varphi_{th} = h_{conv}(T_f - T_{\text{cool}}) \tag{22}$$

Additional data related to this model is given in Appendix A and in the references mentioned therein.

The neutronics is coupled to heat conduction and fluid enthalpy balance via the cross sections. Heat conduction, intrinsically nonlinear because of the temperature-dependent conductivities, is coupled to neutronics via the power density and to fluid via the heat flux and the convection exchange coefficient. After spatial discretization, a system of nonlinear coupled equation of the following form is, once again, obtained:

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t, y). \tag{23}$$

5. Results for the PRKE tests cases

In order to mimic, with the PRKE model, the standard OS methods, a neutronics and a thermal-hydraulics subset of equations should be solved individually, just as is the case when mono-disciplinary legacy codes are coupled via message passing paradigm. Here, this implies that the small system of up to 12 nonlinear equations is not solved at once, but that a neutronics sub-block (the first $1 + J + K$ equations) and a thermal-hydraulics sub-block (the last two equations) are solved separately. We stress again that there is no need to do this from a practical point of view, except to test and analyze the conventional coupling schemes using a reduced model.

5.1. Simple remedies to conventional OS schemes

To analyze the accuracy of the remedies suggested in Section 2.2, we compare a simple rod ejection accident for a LWR reactor (typical hydraulic flow area, dimensions and properties of LWR are used). On Fig. 2, the power level is displayed for a rod ejection performed at thinspace time $t = 250$ ms (the rod is fully withdrawn in 250 ms, i.e., the ramp ejection duration is 250 ms; the total rod worth is 1.2$). All calculations for this example use the Crank–Nicholson scheme (i.e., a second-order accurate scheme). The reference computation for the transient was performed using a time step size of 0.5 ms. The three other computations were performed using a time step size of 10 ms with the following numerical schemes:

(1) Conventional coupling paradigm – nonlinearities are not converged between the different physics
(2) Fixed-Point Iterations, FPI, (i.e., the conventional scheme with sub-iteration within each time step to converge the nonlinearities),
(3) Second-order explicit prediction of nonlinear terms (improved prediction)

On the left of Fig. 2, a zoom at the time of the power peak shows that the FPI scheme and the conventional scheme with prediction yield a solution that is significantly closer to the reference solution than the conventional OS scheme. The figure shows that the conventional coupling scheme over predicts the power level by more than 10% whereas the other schemes have a discrepancy of at most 1%. Obviously, the improved prediction scheme is cheaper in terms of computational effort than the fixed-point iterative method because iterations over all the entire physics within each time step are not performed, and hence the nonlinearities between the different physics are only resolved by using an explicit extrapolation that is accurate up to second-order. The improvement in the solution is impressive for the prediction case since the effort required to make the extrapolation is trivial and the effect of the modification results in a solution closer to the reference, even for a time-step that is much coarser (20 time larger time step than the one used in the reference solution). But it should also be noted that the solution resulting from the improved prediction is more accurate than the conventional scheme only and not the converged solution itself.

Table 1
Order of accuracy for the numerical schemes.

Case #	Fixed-point iterations within time step	Prediction in the nonlinear terms	Acceleration of iterations	Measured accuracy order
1	No	No	No	1.0448
2	Yes	No	No	1.9975
3	Yes	Yes	No	1.9975
4	Yes	Yes	Yes	1.9975
5	No	Yes	No	1.9675

We also performed a convergence analysis for the various schemes in order to confirm whether the remedies (FPI and/or prediction) can recover the second-order accuracy or not. Additionally, this convergence analysis will numerically validate the fact that conventional OS schemes are only first-order accurate, even when the time integration scheme used is second-order, due to the low-order treatment of nonlinearities. To analyze this, the same rod-ejection ramp transient was computed using the Crank–Nicholson scheme for various time step sizes. The error was computed as the difference in the power level at $t = 1$ sec between the reference value obtained with a small time step size ($\Delta t = 0.5$ ms) and the current numerical solution. Table 1 contains the various cases run and the calculated convergence orders (slope of the error versus time step size plot on a loglog scale).

As observed from Table 1, the conventional coupling scheme (case # 1) only yields first order accuracy even though a second-order scheme was used, whereas fixed-point iterations (cases # 2–4) and improved prediction (cases # 3–5) both yield the expected theoretical second order convergence rate. Case # 2 only employs FPI, case # 3 employs FPI and prediction in the nonlinear terms at the first iteration, and case # 4 uses FPI with prediction and Steffensen's method to reduce the number of iterations. Consequently, cases # 2–4 give exactly the same convergence rate since the same nonlinear equations are solved (only with different acceleration techniques). Case # 5 does not use FPI but only utilizes a second-order prediction for the nonlinear term. The observed rates are as expected theoretically, since the FPI and Predicted schemes are nonlinearly consistent whereas the conventional scheme is not.

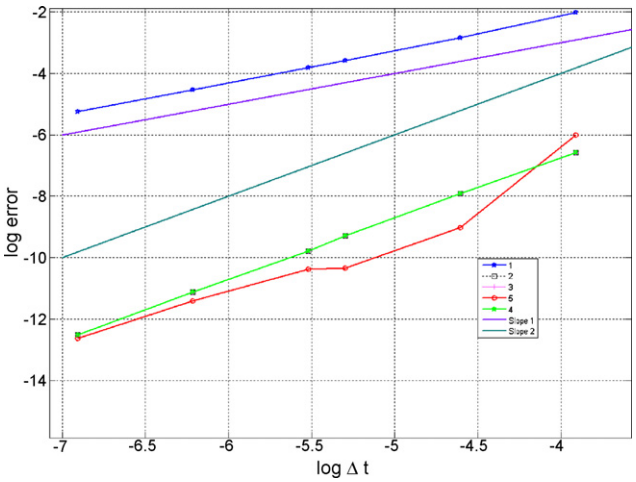

Fig. 3. Order of convergence for different numerical schemes for the LWR rod-ejection test case.

Table 2
Comparison of embedded higher order RK solvers with tolerance = 1E − 4.

Solver	DOPRI5	GRK4A	GRK4T	SDIRK4	RADAU5
Number of steps	139	29	26	22	25
Number of rejected steps	0	0	0	0	1
Number of function evaluations	835	365	326	428	365
Number of Jacobian evaluations	0	28	25	20	19
Average time step employed	0.0089928	0.178571	0.20012	0.238095	0.208333
Maximum time step employed	0.011238	0.887038	0.966367	1.18212	1.414738

Thus, whenever the nonlinearities are sufficiently resolved, the Crank–Nicholson scheme yields back a second order convergence.

Finally, Fig. 3 shows the order of convergence for different numerical schemes listed in Table 1. The convergence results for cases # 2–4 being identical, their graphs are identical and can hardly be distinguished. The second-order prediction method (case # 5) provides some good improvements in the convergence as compared to the conventional scheme, without the need for iterations in the FPI scheme. Both FPI and prediction techniques only involve a small amount of code modification and could easily be used to restore accuracy order in legacy codes.

5.2. Time adaptation

The results observed in the previous section used constant time steps. We now analyze the virtues of time-adaptive methods using implicit RK methods. In the RK framework, time step control is possible for IRK formulae containing embedded formulae of lower order; this usually requires a certain number of stages in the method, typically 3 stages or more. With a higher number of stages, the resulting methods are also of higher accuracy order, thus both concepts, time adaptivity and higher order, are jointly analyzed here. In this Section, our numerical examples are based on a rod-ejection for a sodium cooled fast reactor, with a UOX core ($\Lambda = 5 \times 10^{-5}$ s), MOX core ($\Lambda = 10^{-5}$ s), (standard hydraulic flow area, dimensions and properties of fast reactors are used, cf. Appendix A).

5.2.1. Step transient

An external step reactivity worth of 0.8$ is inserted at $t = 0$ s and the resulting transient up to $t = 5$ s is calculated using various IRK schemes with time adaptation. The different reactivity components as a function of time are plotted along with the various solution field transients in Fig. 4. The computational effort report for each of the adaptive solvers with a user-specified local accuracy of 10^{-4} is provided in Table 2.

The results shown in Table 2 present some noteworthy facts about the methods tested. First, we can see that the implicit, L-

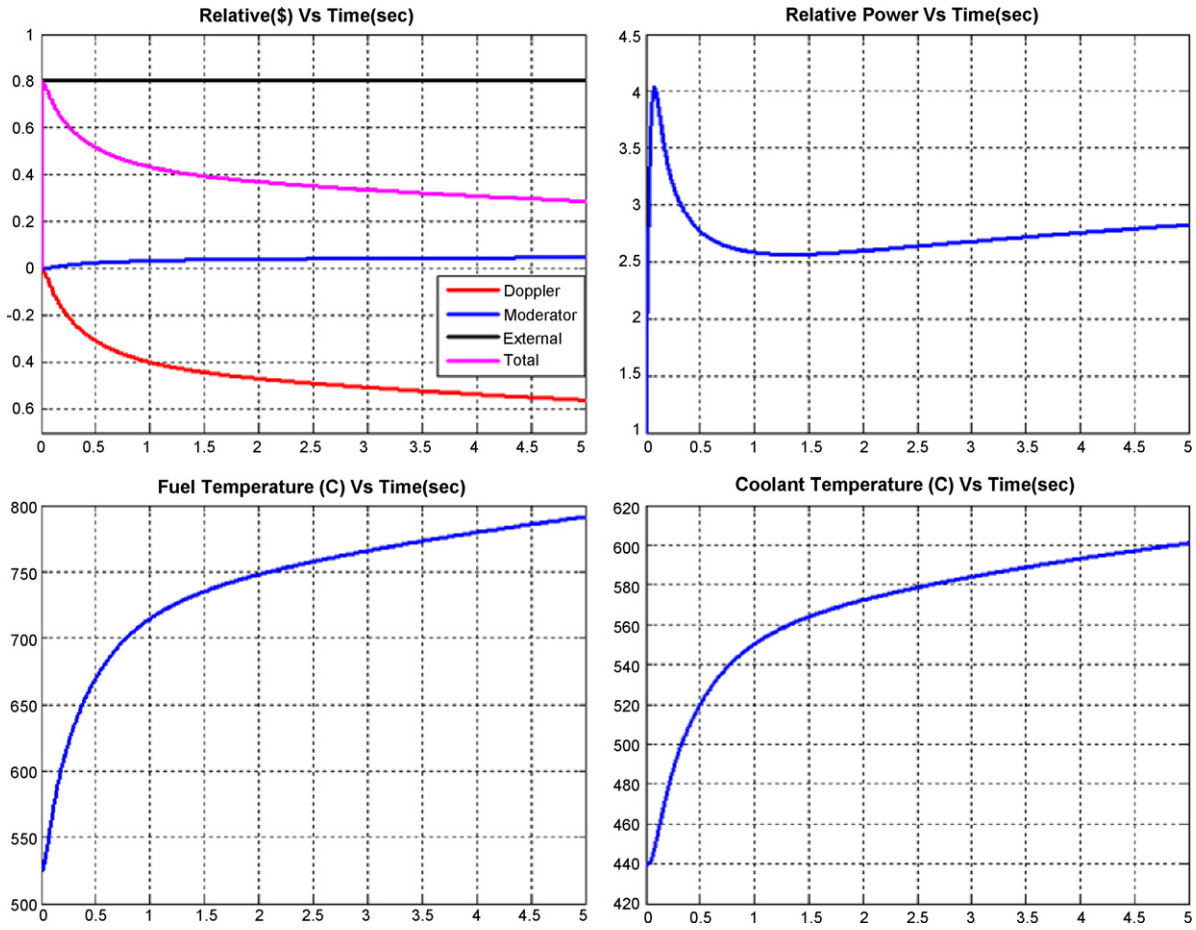

Fig. 4. Fast reactor step case. Top left: reactivity components in $; top right: power transient; bottom left: fuel temperature transient; bottom right: liquid sodium coolant temperature transient.

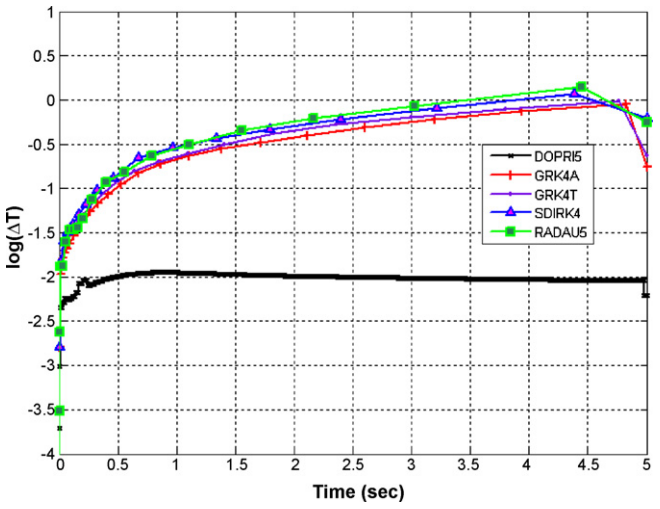

Fig. 5. Fast reactor step case: time steps chosen during transient by various solvers.

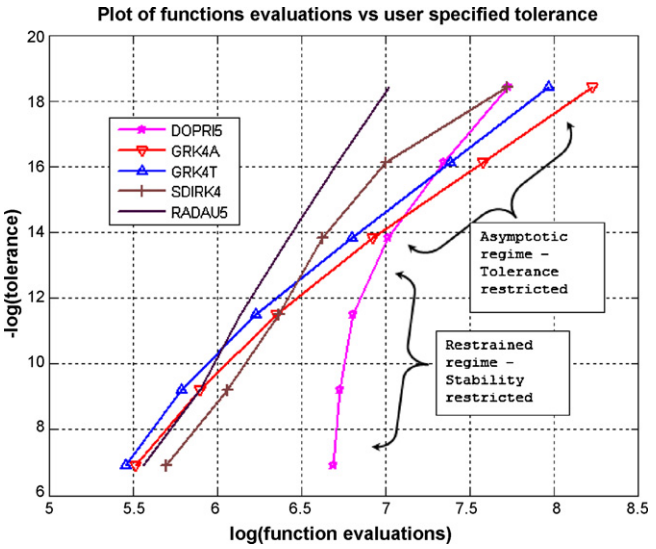

Fig. 6. Fast reactor step case: stability versus accuracy constraints (natural log plot).

stable, stiff solvers, namely SDIRK4 and RADAU5, accurately resolve the solution in very few time steps and use larger time steps once the fastest modes in the transient solution dissipate. The maximum step size used by these solvers is greater than 1 s after the initial activity in the transient. The A-stable GRK4A and the L-stable GRK4T schemes compare well with the other implicit solvers although the maximum step size reached is lower than SDIRK4 by a factor of 1.5. For comparison, a time adaptive explicit scheme, DOPRI5, was used. DOPRI5 is clearly restricted by stability conditions and hence is forced to maintain smaller step sizes than any other solver to resolve the solution accurately.

To better visualize the time adaptation of the various solvers, a plot of the time steps chosen as a function of time for the above transient is shown in Fig. 5. Fig. 5 clearly shows that, with the exception of the explicit DOPRI5 solver, the time adaptation process in all other solvers leads to a continuous increase in the time step sizes. This figure implies that using DOPRI5 for finding the solution for problems can be quite expensive since the time step chosen is severely restricted due to stability conditions of the explicit scheme and not by accuracy considerations.

To obtain a better insight into the behavior of the various solvers and to verify whether or not the solvers have the theoretical rate of convergence, the solver statistics were tabulated and listed in Table 3. In that table, the total computational cost is expressed purely in terms of the number of vector function evaluations (1 function evaluation gives a vector of length $12(= 1 + J + K + 2)$, one value for each component). Then, building the numerical Jacobian (with finite differences) can be expressed as 13 vector function evaluations (one base point and 12 perturbations); in the case of a Jacobian-free approach, at most 13 function evaluations are also needed (to solve the problem, GMRES requires at most 12 function evaluations for a system of size 12, plus the base point data to perform the finite difference). For such a small sized system, the relevance of the number of function evaluations is not necessarily very pertinent since all computations run extremely fast.

Table 3
Comparison of cost for RK solvers as a function of tolerance.

Tolerances used	DOPRI5	GRK4A	GRK4T	SDIRK4	RADAU5
1.00 E−03	805	248	235	298	261
1.00 E−04	835	365	326	428	365
1.00 E−05	907	573	508	581	464
1.00 E−06	1111	1015	898	755	619
1.00 E−07	1549	1964	1613	1103	826
1.00 E−08	2275	3745	2900	2257	1120

While refraining to conclude at this stage how the methods can be extrapolated to the multi-dimensional case, which we will do in Section 6, we can comment from Table 3 that RADAU5 is the most promising method, regardless of the user-specified tolerance, GRK4 methods are equally good at lower tolerances, and SDIRK4 is quite competitive for intermediate tolerances. We also note that the explicit DOPRI5 scheme tends to only perform well at very stringent tolerances; this is expected since, in the fine time step limit, the stability conditions for the explicit scheme are not restrictive in this asymptotic regime. Fig. 6 compares the behavior of the DOPRI5 solver with the ones of other solvers: for fine tolerances, DOPRI5 behaves like implicit schemes and the restriction imposed on the suitable time steps chosen depends only on the user specified tolerance; a more stringent tolerance requires smaller time step sizes, which in turn increases the number of function evaluations. For coarser tolerances, the number of function evaluations decreases gradually for all solvers, except for DOPRI5 because the time step sizes are restricted by stability issues, hence keeping the number of function evaluations significantly higher than for implicit methods. We note again that RADAU5 is the best performer overall, keeping the number of function evaluations to a minimum; GRK4 methods are equally good for coarser tolerance but tend to be more costly for finer tolerances.

5.2.2. SCRAM accident

The previous problem with a step transient is a good test case to gauge the efficiency of convergence of each solver but in order to understand how the solvers resolve local discontinuities in a problem, a new test case was simulated with external reactivity changes as shown in Fig. 7. The transient is simulated for a SFR with a MOX core. Since all solvers strive to contain the local error within the prescribed tolerance, the efficiency of a solver during a smooth continuous transient is not quite obvious. Hence, the 3 discontinuities in the SCRAM accident scenario will help us find out how the adaptive solvers perform and detect the sharp discontinuity in the reactivity function.

The solver performance can be expressed purely in terms of the vector function evaluations, since the cost/function evaluation is constant and the cost of a numerical Jacobian evaluation can be expressed in terms of function evaluations. Based on this idea, the performance of each of the RK solvers is computed and tabulated along with the average time steps and the maximum error in the solution above the user specified tolerance in Table 4. As seen from

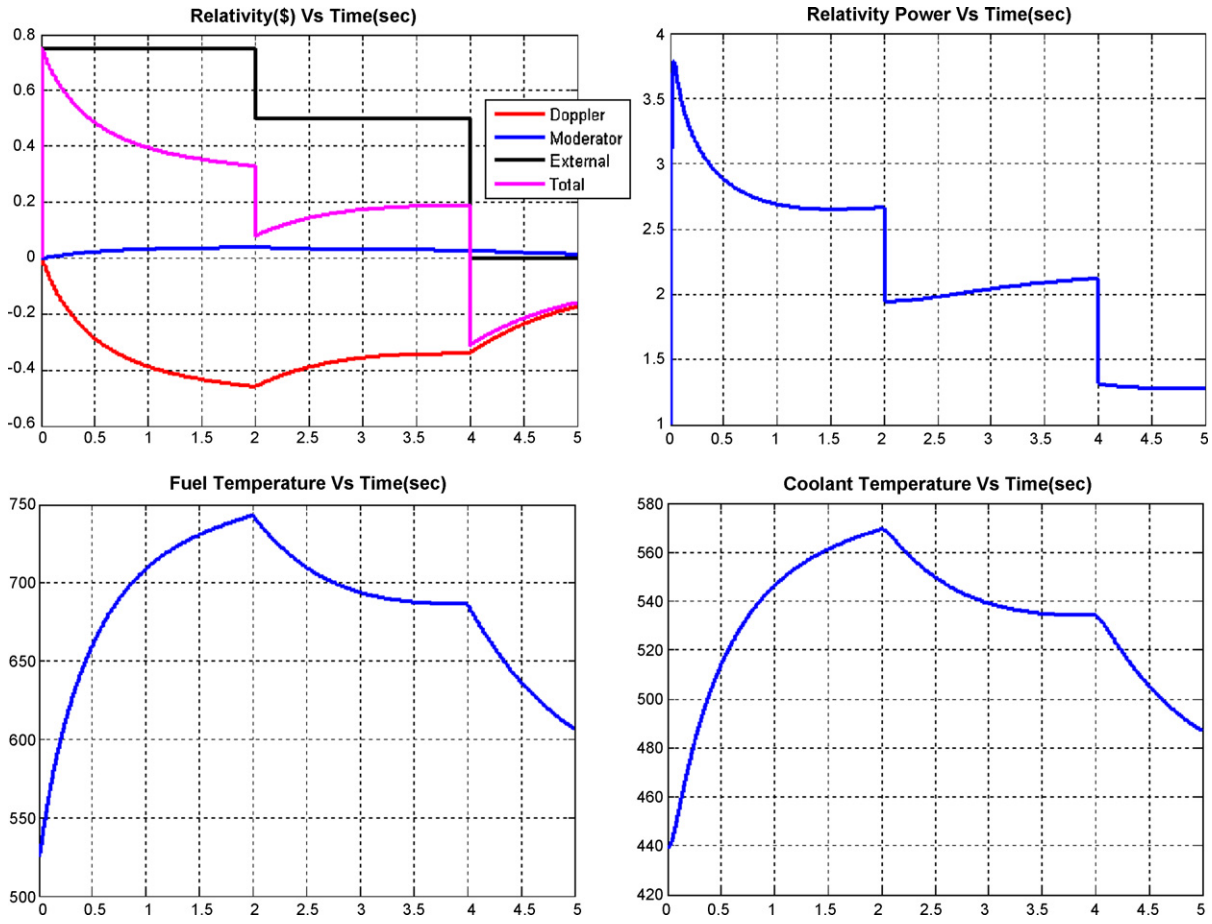

Fig. 7. Fast reactor multiple SCRAM case. Top left: reactivity components in $; top right: power transient; bottom left: fuel temperature transient; bottom right: liquid sodium coolant temperature transient.

Table 4
Solver statistics for a SCRAM accident problem with tolerance = 1E−5.

Solver	DOPRI5	GRK4A	GRK4T	SDIRK4	RADAU5
Number of steps	590	92	88	73	83
Number of rejected steps	35	0	0	0	0
Number of function evaluations	3751	1457	1393	1636	1043
Average time step employed	0.002119	0.05494	0.05748	0.06944	0.06024

Fig. 7, the discontinuities occur at $t = [0, 2, 4]$ s and the tested time-adaptive solvers are required to resolve these local discontinuities completely.

Fig. 8 plots the time step size used as a function of the time. It is again clear that the explicit solver performs comparably well when the fast modes dominate in the transient (¡0.1 s, i.e., in the accuracy-dominated regime) and beyond this, the performance degrades due to stability restrictions (we can clearly note the oscillations in step length on Fig. 8). All implicit time stepping schemes performed well, with the time step sizes increasing until a sudden rod insertion occurs, followed by another increase in time step length. RADAU5 employed the largest time step sizes.

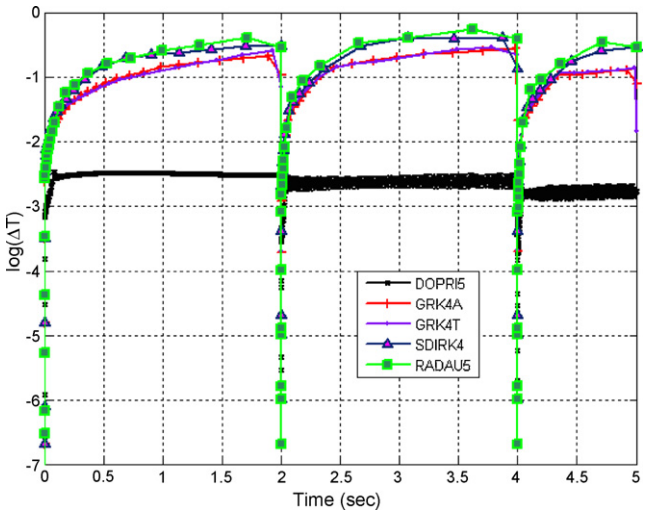

Fig. 8. Fast reactor SCRAM case: time steps chosen during transient by various solvers.

6. One-D results

In this section, the deficiency of standard OS schemes is analyzed using the 1D nonlinear model from Section 4.2. This model couples neutron diffusion, heat conduction, and fluid enthalpy balance. The data used here to model the reactor transient is borrowed from the MSLB benchmark (Ivanov et al., 1999). The 1D assembly is taken to be the 20th fuel assembly type, as described in Table 2.4.2

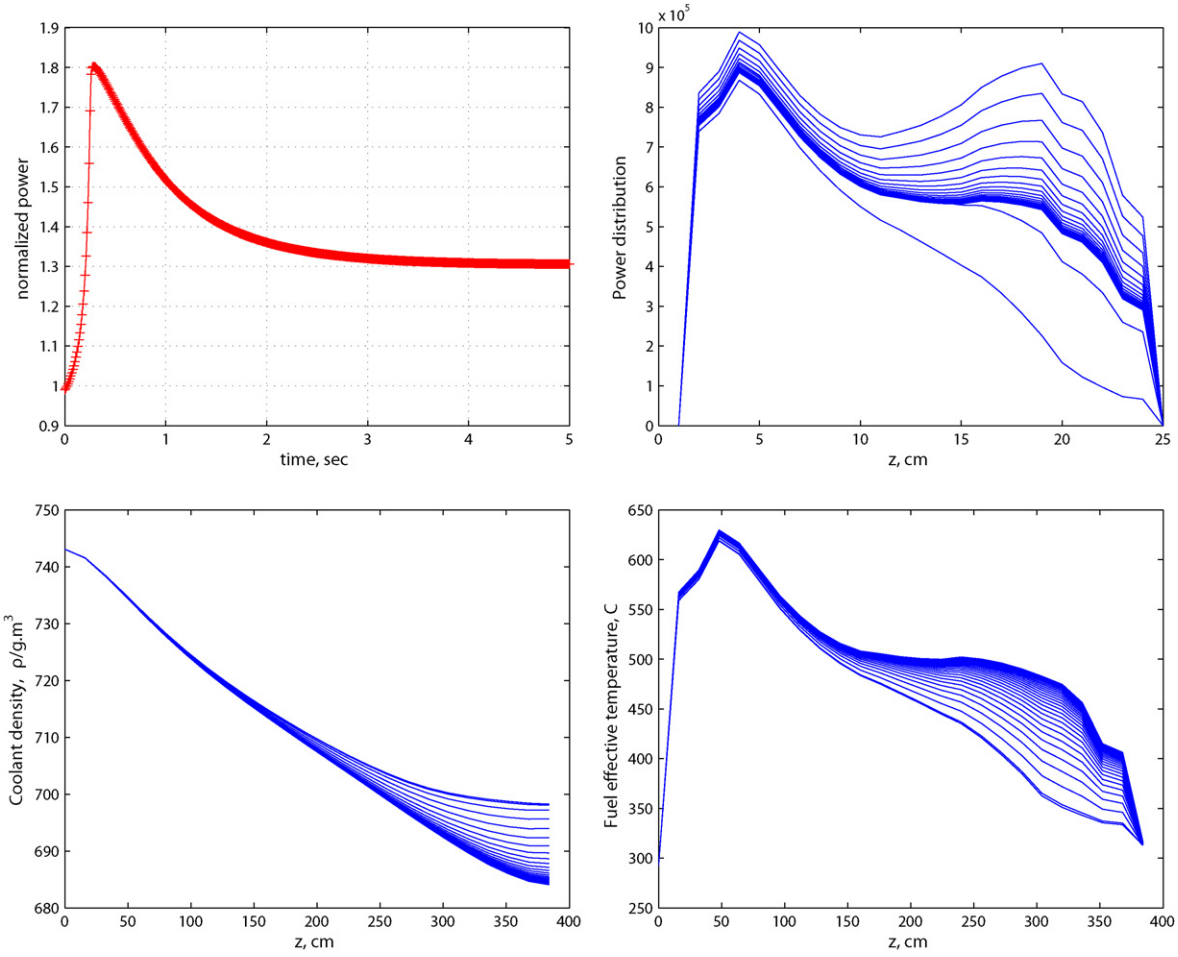

Fig. 9. 1D LWR test case: rod ejection. Top left: normalized power; top right: evolution with time of the power distribution; bottom left: evolution with time of the coolant density; bottom right: evolution with time of the fuel effective temperature.

of (Ivanov et al., 1999). A control rod, whose absorption properties have voluntarily been reduced by defining it to be 50% of unrodded material and 50% of rodded material, is inserted in the first five material zones of that fuel assembly. The steady state initial power was set to be 60% of the nominal power, to account for the fact that a partially rodded reactor does not operate at full power. At time $t = 0$, a ramp ejection of the control rod is performed for a duration of 250 ms. All additional data (neutronics, thermal-hydraulics, geometry) needed to reproduce our results can be found in Ivanov et al. (1999) and in Appendix A.

Fig. 9 shows the time evolution of the normalized power, axial fluid density, axial effective fuel temperature, and axial power profile for the first 5 s of the transient computed using a Crank–Nicholson scheme, with a fixed time step size of 0.01 s. On Fig. 10, we graph the convergence analysis performed for the conventional OS schemes (no prediction, no fixed-point iteration) using the Crank–Nicholson second-order method: regardless of whether the neutronics is started first or last in the OS staggered update, the accuracy order of the physics components are degraded back to first-order, as in the PRKE case. When replacing the OS scheme with the mono-block JFNK scheme, the second-order accuracy is recovered (on Fig. 10, the plain lines without markers represent a slope of 2, which is the theoretical convergence rate for second-order methods).

As mentioned in Section 5, it is more meaningful to compare the computational effort of JFNK and fixed-point OS methods for problems with dimensionality. Table 5 summarizes these findings. We compare the number of fixed-point iterations per time step for the OS scheme and the cost of the JFNK scheme. For coarse tolerances, the number of OS solves per time step ranges between 8

Table 5
Solver statistics for JFNK and conventional OS, 1D test case, coarse and tight tolerances.

Solver	OS	Unpreconditioned JFNK	Preconditioned JFNK
Tolerance 1 E−4			
Number of fixed-point iterations	8–15	n/a	n/a
Number of Newton's iterations	n/a	2–4	2–4
Size of the Krylov subspace	n/a	55–59	4–5
Tolerance 1 E−8			
Number of fixed-point iterations	30–120	n/a	n/a
Number of Newton's iterations	n/a	4–6	4–6
Size of the Krylov subspace	n/a	65–70	7–10

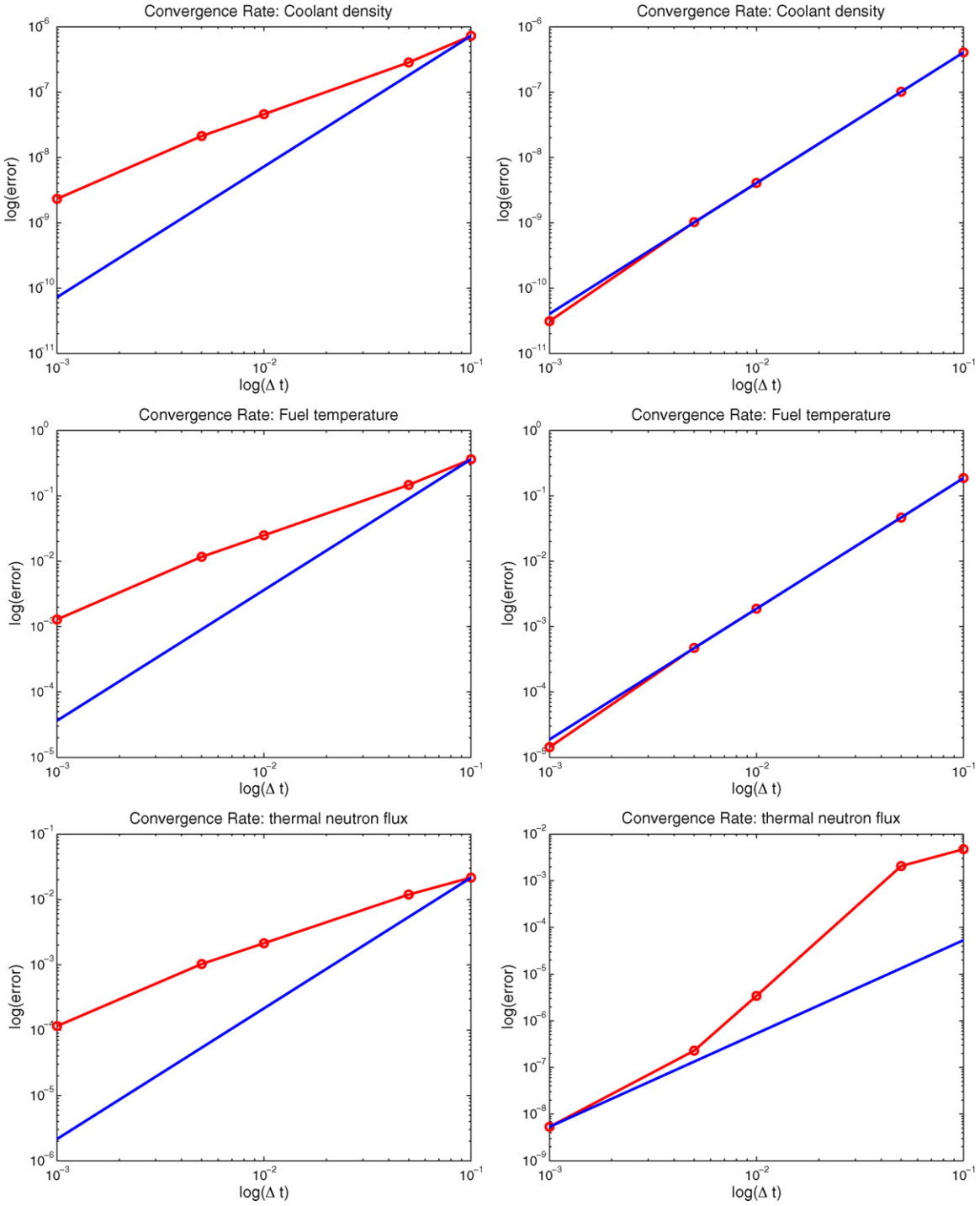

Fig. 10. 1D LWR test case: rod ejection. Left: OS results (fluid density, top, fuel temperature, middle, thermal neutron flux, bottom); right: preconditioned JFNK results: (fluid density, top, fuel temperature, middle, thermal neutron flux, bottom).

and 15 for this transient problem. Both the preconditioned and the unpreconditioned JFNK schemes require the same number of Newton's iterations, as expected since preconditioning only reduces the number of linear solves needed in each Newton's step. For the preconditioned JFNK case, the size of the Krylov subspace is about 4–5, hence requiring 8–20 preconditioner solves per time step, which is similar to the number of solves required in the iterated OS scheme. Thus, preconditioned JFNK is as competitive as the iterated OS scheme for coarse tolerance, whereas the unpreconditioned JFNK requires a significantly larger Krylov subspace.

For tighter tolerances, we note that the number of fixed-point iterations in the OS scheme has significantly increased and that, again, the preconditioned and unpreconditioned JFNK schemes require the same number of iterations in Newton's method. The preconditioned JFNK scheme only shows a small increase in the size of the Krylov subspace, resulting in about 30–60 linear solves. Thus, the preconditioned JFNK scheme requires about half as many OS preconditioner solves as the fixed point OS scheme.

Finally, preconditioned JFNK calculations was run with RADAU5 using time-step control (user-tolerance: tol = 1E-2, 1 E−3, 1 E−4) to

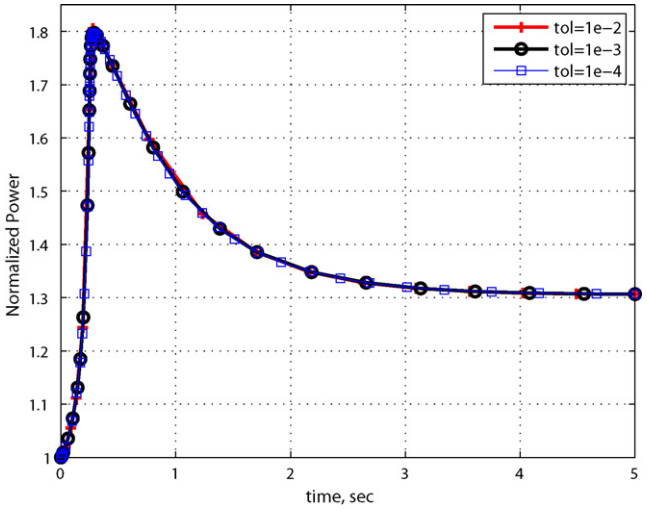

Fig. 11. 1D LWR test case: preconditioned JFNK results with time-adaptation using various tolerances.

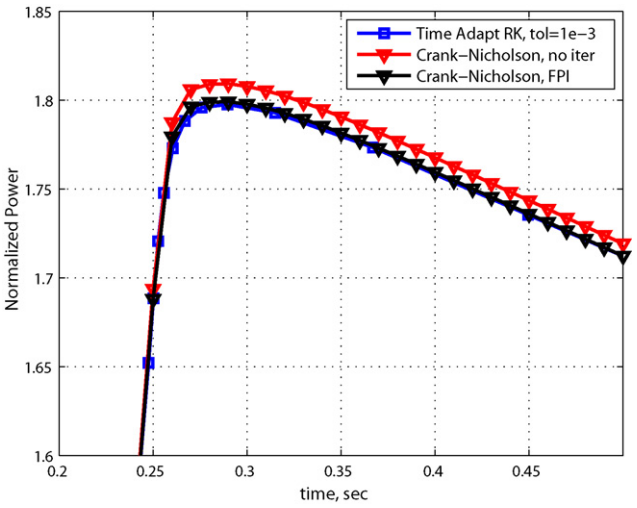

Fig. 13. 1D LWR test case: comparison of preconditioned JFNK results with time-adaptation and OS results with $\Delta t = 10$ ms.

compare the performance of preconditioned JFNK with fixed time step sizes and with automatic step size control. The initial time step size is chosen to be very small (10^{-8} s). Fig. 11 shows the normalized power as a function of time for the three tolerances used. The plots are indistinguishable, with less than 0.5% error in the peak value between a tolerance criterion of 1 E− 2 and 1 E−4. The total number of time steps used in the three calculations is: 24 (tol = 1E−2), 36 (tol = 1E−3), and 59 (tol = 1E−4). We can also clearly note that the time step sizes first become larger (due to our choice of initial time step) but, as the maximum power peak is approached, the time step sizes are automatically reduced to account for the peak before increasing again, showing that the time-step control captures the dynamical variations precisely.

We also compare the time adaptive results with fixed time step OS computations, performed with or without Picard iterations. Fig. 12 shows the normalized power as a function of time for time adaptive results (tolerance of 1E−3) and the OS results with $\Delta t = 100$ ms (50 steps total). The figure presents an enlarged view in the time interval [0.1 1.5] s of the 5-s transient. We note that: (1) the time step size is too large to capture the peak power accurately (with FPI, the magnitude of the peak is correct but the time of the peak is delayed, whereas without FPI, an error of 6.5% is committed) and (2) the dynamical time of the transient is such that the Crank–Nicholson scheme oscillates (a well-known feature for time step sizes larger than twice the fastest decaying mode). Fig. 13 shows the normalized power as a function of time for time adaptive results (tolerance of 1E−3) and the OS results with $\Delta t = 10$ ms (500 steps total). The figure presents an enlarged view over a short time interval around the peak. We note that: (1) with FPI, the magnitude of the peak and the time of the peak are correctly captured (2) whereas without FPI, an error of 1% still remains.

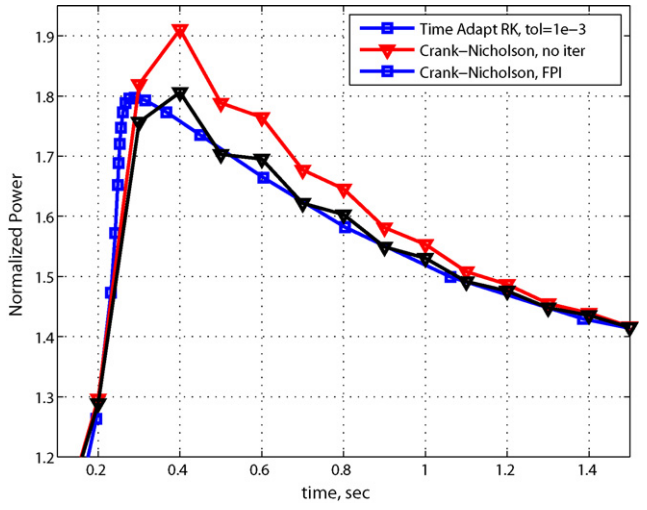

Fig. 12. 1D LWR test case: comparison of preconditioned JFNK results with time-adaptation and OS results with $\Delta t = 100$ ms.

7. Conclusions

Based on the coupled results for the reactor transients analyzed in this work, the following conclusions are proposed to (i) improve the performance of the existing coupled physics legacy codes by restoring consistency and (ii) point out the potential benefits of switching the coupled physics formalism to the JFNK framework with its ability to easily implement higher order methods and time step control.

- nonlinearly consistent OS schemes can be used in place of traditional OS coupling methods in order to restore the order of accuracy of the time discretization scheme from the conventional $O(\Delta t)$.
- the technique of explicit prediction in the formerly lagged nonlinear terms can serve as an alternative to enhance consistency with a minimal computational cost rather than using Picard iterations to converge the solution at every time step.
- a mono-block reformulation of the whole nonlinear problem is possible and opens up the option for high-order methods and time adaptation: both can clearly provide superior performance in terms of cost and accuracy compared to improved OS methods.
- the implicit stiff solvers tested (SDIRK, RADAU) resolve local discontinuities quite effectively and prove their potential applicability to larger and more complex coupled problems.

Appendix A. Correlations and additional data for test cases

The PRKE test case representative of a thermal reactor was modeled using the following data: fuel conductivity and specific heat from MatPro (Olsen and Miller, 1979)(also referenced in Ragusa (2001)), clad conductivity (Olsen and Miller, 1979; Ragusa, 2001), water properties (required to compute the Nu number and the liquid bulk temperature) from (Ragusa, 2001), convective heat

Table A.1
Additional data pertaining to the physical models used.

Parameter	Thermal reactor	Fast reactor
Fuel radius R_{fuel} (m)	0.0041	0.00348
Gap radius R_{gap} (m)	0.00411	n/a
Clad radius R_{clad} (m)	0.00468	0.004
Active core height H (m)	4	0.8
Coolant speed, u (m/s)	5	5
Inlet temperature, T^{in} (°C)	290	355
Hydraulic area per fuel pin, A_{flow} (m^2)	8.99516 E−5	5.281 E−5
Nominal power density conversion factor Ω_{pow} (W/m^3)	3.53E8	4.77E8
Mean generation time (s)	1 E−4	1 E−5
Delayed neutron fractions (pcm)	650	9, 87, 70, 140, 60, 55
Precursors time constants (1/s)	0.08	0.0124, 0.0305, 0.111, 0.301, 1.14, 3.01
Doppler reactivity coefficient, α_{Doppler} (pcm/(C^d))	−150	−0.8841
Coolant reactivity coefficient, α_{cool} (pcm/C)	−7	+0.1263
Gap conductance, h_{gap} (W/m^2 C)	1E4	n/a

exchange at the clad-water interface from the Dittus Boelter correlation, decay heat coefficients from (ANS-5.1, 1973). The Doppler reactivity was modeled using a square root law (exponent $d = (1/2)$). Additional data can be found in Table A.1.

The PRKE test case representative of a fast sodium-cooled reactor was modeled using the sodium thermodynamical properties from (Fink and Leibowitz, 1995), the fuel was taken to be in metal form (fuel and clad properties from (Palmiotti, 2006), no gap is present in between the fuel and the clad). The radial and axial expansion reactivity coefficients were merged with the fuel Doppler coefficient for simplicity since the purpose of this work was to demonstrate the potential gain from using fully implicit coupling for reactor transient analysis. The Doppler reactivity was modeled using a linear law (exponent $d = 1$). The Nu number used to determined the convective heat exchange coefficient in sodium is from (Todreas and Kazimi, 1990). The decay heat data was unchanged. Additional data can be found in Table A.1.

The 1D model used the neutronic and thermo-hydraulic data from (Ivanov et al., 1999). An average hydraulic channel and an average fuel pin are modeled. The neutronic data used for this 1D model is axial composition #20 (Table 2.4.2 of (Ivanov et al., 1999)). This composition is made of 24 axial layers of fuel and 2 axial layers of reflector. Each layer is described by 2-g cross sections that are tabulated as a function of fuel temperature and moderator density. Values in between the tabulated points are obtained by bilinear interpolation. A rodded version of composition #20 is given in Ivanov et al. (1999) to account for the control rod insertion in that composition. For our 1D calculation, that rodded composition was too absorbing (skewing too much the axial power profile) and we modified the rodded fuel assembly composition by using half of the unrodded composition and half of the original rodded composition.

References

Akdeniz, B., Ivanov, K.N., Olson A.M., 2006. Boiling Water Reactor Turbine Trip (TT) Benchmark—vol. III: Summary Results of Exercise 2. OECD 2006. NEA/NSC/DOC(2006)23. ISBN 92-64-02331-3.

ANS-5.1, October 1973. Decay Energy Release Rates Following Shutdown of Uranium-Fueled Thermal Reactors. Draft ANS-5.1/N18.6.

Carlos, A.F., Park, K.C., Charbel, F., 2001. Partitioned analysis of coupled mechanical systems. Advances in Computational Methods for Fluid-Structure Interaction and Coupled Problems 190 (24–25), 3247–3270.

Curtis, C., Ober, John, N., Shadid, 2004. Studies on the accuracy of time-integration methods for the radiation-diffusion equations. Journal of Computational Physics 195 (April (2)), 743–772.

Dormand, J., 1996. Numerical Methods for Differential Equations: A Computational Approach. CRC Press.

Duderstadt, J.J., Hamilton, L.J., 1976. Nuclear Reactor Analysis. Wiley, New York.

Fink, J.K., Leibowitz, L., January 1995. Thermodynamic and transport properties of sodium liquid and vapor, Technical Report. Argonne National Laboratory, ANL/RE-95/2.

Gjesdal, T., 2007. Implicit–explicit methods based on strong stability preserving multistep time discretizations. Applied Numerical Mathematics 57 (August (8)).

Gustafsson, K., 1991. Control theoretic techniques for stepsize selection in explicit Runge–Kutta methods. ACM TOMS 17, 533–554.

Gustafsson, K., 1994. Control theoretic techniques for stepsize selection in implicit Runge–Kutta methods. ACM TOMS 20, 496–517.

Hairer, E., Wanner, G., 1996. Solving Ordinary Differential Equations: II. Stiff and Differential Algebraic Problems. Springer, Berlin.

Hairer, E., Norsett, S.P., Wanner, G., 1987. Solving Ordinary Differential Equations: I. Nonstiff Problems. Springer, Berlin.

Housiadas, C., 2002. Lumped parameters analysis of coupled kinetics and thermal-hydraulics for small reactors. Annals of Nuclear Energy 29, 1315–1325.

Ivanov, K.N., Beam, T., Baratta, A.,April 1999. Pressurised Water Reactor Main Steam Line Break (MSLB) Benchmark, vol. I: Final Specifications. OECD NEA/NSC/DOC(99)8.

Kaps, P., Rentrop, P., 1979. Generalized Runge–Kutta methods of order four with step size control for stiff ordinary differential equations. Numerische Mathematik. 33 (1), 55–68.

Knoll, D.A., Keyes, D.E., 2004. Jacobian-free Newton–Krylov methods: a survey of approaches and applications. Journal of Computational Physics 193 (2), 357–397.

Knoll, D.A., Chacon, L., Margolin, L.G., Mousseau, V.A., 2003. On balanced approximations for time integration of multiple time scale systems. Journal of Computational Physics 185 (March (2)), 583–611.

Lowrie, R.B., 2004. A comparison of implicit time integration methods for nonlinear relaxation and diffusion. Journal of Computational Physics 196 (2), 566–590.

Mousseau, V.A., Knoll, D.A., Rider, W.J., 2000. Physics-Based Preconditioning and the Newton–Krylov Method for Non-equilibrium Radiation Diffusion. Journal of computational physics 160 (February (2)), 743–765.

Mousseau, V.A., 2004. Implicitly balanced solution of the two-phase flow equations coupled to nonlinear heat conduction. Journal of computational physics 200 (October (1)), 104–132.

Olsen, C.S., Miller R.L., 1979. MATPRO, A Handbook of Material Properties for Use in the Analysis of Light Water Reactor Fuel Rod Behavior. NUREG/CR-0497.

Palmiotti, G., July 2006. Private communication, Argonne National Laboratory.

Ragusa, J.C., April 2001. Contribution à l'élaboration d'un simulateur de réacteur nucléaire utilisant une modélisation 3D de la neutonique du cœur. Ph.D. thesis. National Polytechnic Institute of Grenoble.

Stoer, J., Bulirsch, R., 1993. Introduction to Numerical Analysis, 2nd ed. Springer-Verlag, New York.

Taylor, B., Todorova, N.K., Ivanov, K.N., October 2002. Pressurised Water Reactor Main Steam Line Break (MSLB) Benchmark, vol. III: Results of Phase 2 on 3-D Core Boundary Conditions Model. OECD NEA/NSC/DOC(2002)12.

Todreas, N.E., Kazimi, M.S., 1990. Nuclear Systems: I. Thermal Hydraulic Fundamentals. Taylor & Francis.

Watts, H., 1984. A step size control in ordinary differential equation solvers. Society for Computer Simulation 1 (May), 15–25.

Woodruff, W.L., 1984. A kinetics and thermal-hydraulics capability for the analysis of research reactors. Nuclear Technology 64, 196–206.