M&C 2017 - International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering, Jeju, Korea, April 16-20, 2017, on USB (2017)

A quadratic depletion coupling scheme with adaptive stepsize control in CASMO5

Joshua Hykes and Rodolfo Ferrer

Studsvik Scandpower, Inc., 1070 Riverwalk Dr. Suite 150, Idaho Falls, ID
joshua.hykes@studsvik.com, rodolfo.ferrer@studsvik.com

Abstract - *Building on recent advances in depletion coupling algorithms, we examine the adaptive time step P_2FC_2 method, a higher-order predictor-corrector coupling scheme to solve the neutron transport-Bateman system. P_2FC_2 approximates the transition matrix A(t) as a quadratic polynomial in time, interpolating the polynomial through the values at previous time steps. The quadratic expansion provides a good fit for the observed shape of various elements of A(t) for most of the depletion (with the exception of the first few depletion steps during the initial accumulation of fission product poisons). The P_2FC_2 method is combined with an adaptive time step algorithm that controls the local truncation error of the computed nuclide number densities. The error estimators are based on the difference between the predictor and corrector number densities. For the error estimators examined, the predicted error is mostly within a factor of ten of the true local error, which is sufficient accuracy to perform step size adjustment. The P_2FC_2 method with fixed time steps shows substantial reduction in the nuclide number density and eigenvalue errors. Adding the adaptive time step algorithm reduces the number of statepoints needed by about 20% (with the particular error goals specified here). However, this reduction is counteracted by an increase in runtime of the burnup calculation, which in CASMO5 is a small but non-negligible fraction of the total runtime. While most of this work is concerned with coupling schemes with one flux solution per time step, a method with two flux solutions per step, FP_2FC_2, is briefly considered as well, and shown to have slightly better accuracy, although with double the number of computationally-expensive transport solves.*

I. INTRODUCTION

We are currently evaluating the use of a quadratic depletion coupling scheme with adaptive stepsize control in CASMO5. This builds on recent work on depletion methods. Ref. [1] shows the potential for accuracy improvements with linear and quadratic interpolation of the transition matrix, in contrast with the conventional approach of assuming it constant during the solution of the Bateman equations.

Addressing the issue of proper stepsize selection in Ref. [2], Walter and Manera demonstrate adaptive stepsize control for 2D lattice depletion calculations. As they note, traditionally lattice physics depletions have been performed with pre-determined depletion schedules. This is the approach taken in CASMO5 [3] (ignoring the conditional step refinement that is applied based on the presence of gadolinium burnable absorber). These depletion schedules were tuned to produce accurate solutions for representative lattices. Sometimes the schedules have differed based on characteristics of the lattice, such as having different schedules for PWRs and BWRs. The disadvantage of the schedule is that the steps must be sufficiently fine for the limiting lattice case, but likely this will result in more detail (and effort) than is needed for many lattices.

Combining ideas in Refs. [1] and [2], we have implemented a quadratic adaptive stepsize algorithm into a developmental version of CASMO5. This paper describes our experiences with the method, especially where they differ from these previous works. A major difference is that CASMO5 tracks the fuel and burnable absorber depletion separately [4]. Thus, we have focused this work on the depletion of cases free of gadolinium burnable absorber, since gadolinium triggers a refined stepsize schedule.

II. THEORY

1. Depletion algorithms

The Bateman isotopic depletion equation is

$$\frac{d}{dt}\vec{n}(t) = A(t)\,\vec{n}(t) \tag{1}$$

where $\vec{n}(t)$ is the vector of nuclide number densities in a particular fuel region and $A(t)$ is a matrix whose element A_{ij} specifies the probability of transition of nuclide j to nuclide i due to neutron capture, fission, decay, etc. Conventionally, the transition matrix $A(t)$ is assumed to be a constant A_0 during a single transport-depletion step. CASMO5 currently uses a predictor-corrector (PC) algorithm for the coupling of neutron transport and isotopic depletion, which we shall denote P_0FC_0L. The algorithm is:

- P_0 Starting with the final number densities from the previous step $\vec{n}^f_{i-1}$, compute the predictor number densities $\vec{n}^p_i$ for time i using one-group fluxes $\vec{\phi}_{i-1}$ and cross sections $\vec{\sigma}_{i-1}$ from the previous time $i-1$.
- F Compute the one-group fluxes $\vec{\phi}_i$ and cross sections $\vec{\sigma}_i$ for time i using the predictor number densities $\vec{n}^p_i$ for time i.

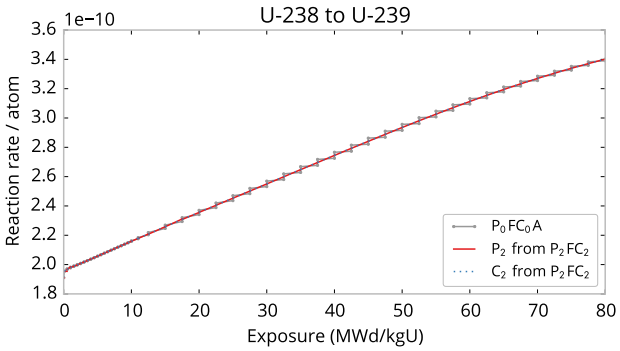


Fig. 1: Shape of $A_{ij}(t)$ for U-238(n,γ)U-239 transition.

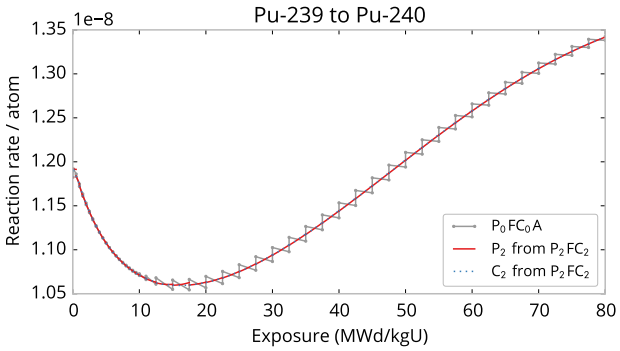


Fig. 2: Shape of $A_{ij}(t)$ for Pu-239(n,γ)Pu-240 transition.

C$_0$ With initial condition $\vec{n}^f_{i-1}$, compute the corrector number densities $\vec{n}^c_i$ for time i using one-group fluxes $\vec{\phi}_i$ and cross sections $\vec{\sigma}_i$ from the current time i.

L Compute the final number densities $\vec{n}^f_i = (\vec{n}^p_i + \vec{n}^c_i)/2$

The mnemonics are P for predictor, C for corrector, F for flux transport calculation, and L for local extrapolation. The subscripts on P and C represent the order of the polynomial approximation of the transition matrix $A(t)$ with respect to time. The notation is a modification of the PECLE notation used to identify predictor-corrector methods for solving numerical ODEs [5, chapter 4]. In this default P$_0$FC$_0$L scheme, only one flux calculation is performed per statepoint. In the notation of Ref. [6], CASMO5 uses the Ce/BE method.

Ref. [1] describes a LE/QI depletion coupling scheme that approximates $A(t)$ as linear for the predictor step and quadratic for the corrector. This would be FP$_1$FC$_2$ in the present notation. A quadratic function in t seems to be a good representation of the behavior of the transition matrix $A(t)$; see Figures 1–4. We have implemented a similar scheme in CASMO5, namely P$_2$FC$_2$, where each element $A_{ij}(t)$ in the transition matrix $A(t)$ is approximated as

$$A_{ij}(t) = A_{ij,2}t^2 + A_{ij,1}t + A_{ij,0} \quad . \tag{2}$$

Before each P$_2$ and C$_2$ step, the coefficients in Eq. 2 are updated so that the polynomial interpolates the last three values

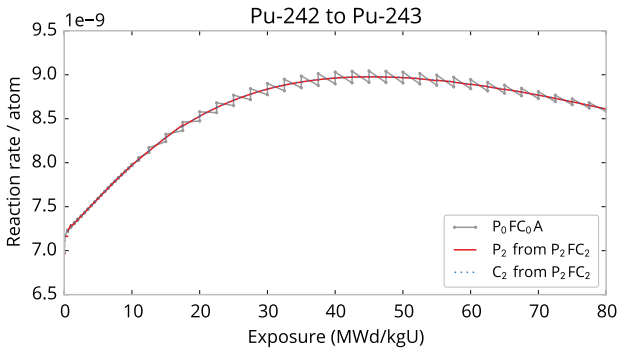


Fig. 3: Shape of $A_{ij}(t)$ for Pu-242(n,γ)Pu-243 transition.

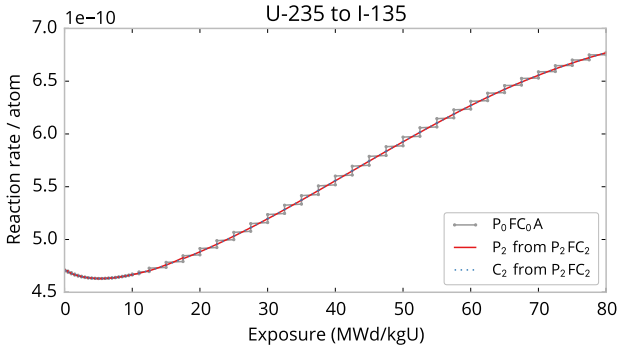


Fig. 4: Shape of $A_{ij}(t)$ for U-235 to I-135 fission yield.

of $A(t)$. This algorithm is similar to the existing CASMO5 algorithm, especially in that it uses one flux transport solve per step. The differences from the existing CASMO5 algorithm are the quadratic polynomial approximation of $A(t)$ and the lack of the averaging final step.

Since P_2FC_2 relies on previous steps to form the interpolant of $A(t)$, a different method is necessary for the first steps. In the results below, we have used the P_0FC_1 method for the first two steps. Unless a very fine first step is used, the initial accumulation of fission products creates a kink in $A(t)$, making interpolation or extrapolation between steps inaccurate [7]. Therefore, we use P_0FC_1 for the first two steps.

By default, CASMO5 performs one flux calculation per statepoint. For comparison, we have included limited results below for the FP_2FC_2 method. This method is similar to P_2FC_2, but each predictor step is preceded by a flux calculation.

A. Flux normalization

As demonstrated in Ref. [8], significant accuracy gains can be realized by renormalizing the flux to match the specified power throughout the step. (Contrary to the statement in Ref. [8], CASMO has performed flux renormalization to maintain constant power during a depletion step since at least CASMO-3 v4.4 [9, 10]. A CASMO4 developer coauthored Ref. [11], which describes a related normalization method in section 6.3.12.) With the traditional piecewise constant A_0 approximation, CASMO5 computes the integrated fission energy at step i by multiplying the constant one-group fission cross sections $\sigma_{f,ijk}$, the energy release per fission κ_{ijk}, the time step length Δt, the one-group flux ϕ_{ik}, the region volume V_k, and the average nuclide number densities $\bar{n}_{ijk}$,

$$E_i = g_i \, \Delta t_i \sum_{j,k} \bar{n}_{ijk} \, \sigma_{f,ijk} \, V_k \, \phi_{ik} \, \kappa_{ijk} \qquad , \tag{3}$$

where the nuclides are indexed by j and the fuel regions by k. The desired energy per step is met by applying fixed point iteration on the flux normalization parameter g, repeating the depletion of the heavy nuclides until the normalization parameter converges within the tolerance criterion. For the current depletion coupling scheme P_0FC_0L, fixed point iteration converges in just a few iterations (usually within one or two iterations, almost always within 5). However for certain steps in P_2FC_2, we observed that simple fixed point iteration required more than 20 iterations to converge. A damped fixed point iteration resolved this difficulty,

$$g_{l+1} = \omega g^*_{l+1} + (1 - \omega) g_l \qquad . \tag{4}$$

We used $\omega = 0.7$.

Since $A(t)$ is not constant in P_2FC_2, we cannot compute the integrated fissions using the average number density,

$$f_j = \Delta t \, \bar{n}_j \, \sigma_{f,j} \, \phi \qquad . \tag{5}$$

Instead, we integrate the time-varying quantities as

$$f_j = \int dt \; n_j(t) \; \sigma_{f,j}(t) \; \phi(t) \qquad . \tag{6}$$

The time-dependent fission cross section is approximated using a similar quadratic interpolation as for $A(t)$. The integrals are computed using a numerical ODE solver, as described next.

B. Numerically integrating the ODEs

To solve the Bateman equations with a time-dependent $A(t)$, we have used DVODE [12], a numerical ODE solver for stiff (and non-stiff) systems. See Refs. [13, 14, 15, 16] for previous applications of ODE solvers to solve the Bateman equations. At present, we have decided to use DVODE to avoid the substeps that would be necessary with CASMO5's default CRAM solver, as suggested in Ref. [6]. DVODE is performing internal substeps, but beyond updating the derivative function and Jacobian routines, the interface to DVODE is the same as for the constant A_0 case.

In addition to solving for the final nuclide number densities at each step, DVODE also computes the integrated fissions, which are necessary to deplete at a constant power. These integrated quantities are computed simultaneously with the number densities by expanding the system of equations as described in Ref. [17]. The system of equations is thus

$$\frac{d}{dt} \begin{bmatrix} \vec{n} \\ \vec{f} \end{bmatrix} = \begin{bmatrix} A(t) & 0 \\ \Sigma_f(t) & 0 \end{bmatrix} \begin{bmatrix} \vec{n} \\ \vec{f} \end{bmatrix} \qquad , \tag{7}$$

where $\vec{f}$ is the vector of the integrated fissions. $\Sigma_f(t)$ is a diagonal matrix with entries corresponding to the one-group microscopic fission cross sections times flux, using a similar quadratic interpolation as for $A(t)$.

C. Timing considerations

Ref. [16] provides several techniques to substantially reduce the runtime of DVODE for solving the Bateman equations. In review, for stiff problems the methods in DVODE require the Jacobian. Although it can be computed using a finite-difference formula, DVODE's authors suggest providing it explicitly if possible. The Jacobian of Eq. 7 is the matrix on the right-hand side. This full, dense matrix can be provided to DVODE to achieve accurate results, but the calculation time is long. Fortunately, the authors of DVODE recommend that an approximate Jacobian is often sufficient, potentially with many less non-zero entries. In addition to the suggestions in Ref. [16], we have experimented with zeroing various blocks of the full Jacobian and have achieved significant efficiency gains. Our shortest runtimes were achieved by passing the Jacobian as a tridiagonal matrix. This ignores the integrated fission block $\Sigma_f(t)$, the fission product yield block in $A(t)$, and also many of the decay and neutron-induced transitions. (While these entries are approximated as zero in the Jacobian,

they are still accounted for because they are included in the evaluation of the derivative function $\dot{y} = f(t, y)$.)

For the 320 nuclide system (both actinides and fission products) we consider here, CRAM-14 for one composition requires approximately 0.003 s. With a variable $A(t)$, the DVODE solver takes about 0.015 s. (These timings were performed in serial on a 2.8 GHz AMD Opteron 6348.) Our DVODE performance is comparable to the 0.02 s time for a 200-nuclide, constant A_0 system reported in Ref. [16].

In contrast to the Monte Carlo transport code considered in Ref. [7], in CASMO5 the burnup calculation consumes a small but non-negligible portion of the runtime in a typical depletion calculation. For example, from a set of 900 single-assembly test cases, 85% of cases spent between 0 and 12% of the runtime in the burnup calculation. Further, 37% of the cases fell between 4.8 and 12%. The median usage was 5.2%. This timing was conducted using CASMO5's default CRAM solver for the solution of the Bateman equations.

For the default P_0FC_0L coupling scheme, a 5% usage for the burnup calculation is quite satisfactory. However, when using a higher-order method (such as P_2FC_2), a one-step CRAM solution cannot capture the variation of $A(t)$ throughout the step. Thus, Ref. [6] uses a substep approach, where several CRAM substeps are taken within each step, adjusting the constant A_0 matrix to correspond to the average value of $A(t)$ in each substep. There are two disadvantages to this approach. First, the substepping introduces another source of truncation error, which can be limited by increasing the number of substeps. Ref. [6] used five substeps for the quadratic corrector step. Second, since the runtime in the burnup solver is roughly proportional to the number of linear system solves (required by CRAM), we expect the runtime of N substeps of CRAM to be N times longer than the runtime of single-step CRAM. If the average single-step CRAM burnup calculation uses 5% of the total runtime, we would expect a 5 substep CRAM solution to add 20% to the total runtime.

In the current P_2FC_2 implementation, we have used the DVODE solver instead of CRAM. Currently the DVODE solver is 2.3–5.0 (depending on the case or computer) times slower than the single-step CRAM solver, which implies that it is competitive with five-substep CRAM. While both the CRAM and DVODE (plus accompanying routines) solvers have been optimized for runtime, further optimizations for both solvers are surely possible. Thus, the runtime figures here are not intended as a final verdict, but rather evidence that DVODE has the potential to be competitive with substep CRAM.

In the results that follow, we used an absolute tolerance of 10^{-30} atoms/barn-cm and a relative tolerance of 10^{-7} in the DVODE Bateman solver. These tolerances are different than those used for the coupled transport-depletion solver.

2. Error estimators

Before defining the error estimators, it is helpful to draw the distinction between local and global truncation error. The local truncation error at time i is the error incurred for one step of the numerical method,

$$\vec{\epsilon}_i^{\,\text{local}} = \left| \vec{n}_i^{\,\text{approx}} - \vec{n}_i^{\,\text{exact}} \right| \qquad , \tag{8}$$

where both solutions on the right are computed from the same values at the previous step, $\vec{n}_{i-1}$. The global truncation error is similar, but instead of starting with the same values for the previous step, they use the same values at the initial condition. At the first step, the local and global truncation errors are the same. Thereafter they diverge. Control of the local error should limit the global error, but the exact correspondence between the two is not straightforward. We seek an estimate of the local truncation error.

As observed in Ref. [2], the predictor-corrector method is amenable to computing an error estimator since lower-accuracy and higher-accuracy solutions are available. The error estimator used in Ref. [2] was the one-group neutron flux difference between predictor and final values. However, since CASMO5 performs only one flux calculation per time step, this is not a viable option in CASMO5. Instead, we use a similar estimator based on the nuclide number densities. The P_0FC_0L estimator at step i is

$$\vec{\epsilon}_i = \left| \vec{n}_i^{\,f} - \vec{n}_i^{\,p} \right| \tag{9}$$

Note that this is an estimate of the error in the low-order, predictor number densities, not the corrector or final number densities. For the P_2FC_2 method, a similar estimator can be formed, by using the corrector number densities $\vec{n}^c$ in place of $\vec{n}^f$.

3. Adaptive stepsize algorithm

We implemented the "elementary adaptive stepsize algorithm" from Ref. [2] rather than the control-theory based algorithm. This algorithm is designed to reduce the local truncation error such that

$$\epsilon_j \le \tau_a + \tau_r n_{i,j}^f, \qquad j = 1, \ldots, n_j \text{ nuclides.} \tag{10}$$

The update formula for the stepsize we used was

$$h_{i+1} = h_i \max\left[\min\left(\gamma x^{1/(k+1)}, \rho_2\right), \rho_1\right] \tag{11}$$

$$x = \min_j \frac{\tau_a + \tau_r n_{i,j}^f}{\epsilon_j} \tag{12}$$

where

j = the nuclide index,
k = the order of the predictor method (1, 3),
τ_a = the absolute error tolerance (2e-7 atoms/(barn·cm)),
τ_r = the relative error tolerance (2e-4, 5e-5),
γ = a safety factor (0.9, 0.8),
ρ_1 = lower multiplicative limit on stepsize change (0.1),
ρ_2 = upper multiplicative limit on stepsize change (5.0, 1.2).

The values in parenthesis are the values used in the results section below. Where two values are listed, the first is for P_0FC_0L and the second is for P_2FC_2. The parameters are chosen conservatively for P_2FC_2 to prevent a breakdown in the quadratic interpolation caused by rapidly changing step sizes. The error tolerances were chosen to roughly match the eigenvalue error in the current CASMO5 depletion schedule.

If $x > 1$, the error is below the specified tolerance specified in Eq. 10 and the step is accepted. Otherwise, the step is rejected and repeated with a reduced step length. Whether the stepsize is accepted or rejected, the stepsize is changed according to Eq. 12. The safety factor $\gamma < 1$ limits the number of failed steps by shortening each step.

Since the error estimator is for the predictor number densities $\vec{n}_i^p$, it would be more consistent to carry forward the predictor number densities rather than the final number densities $\vec{n}_i^f$. However, we expect the final number densities $\vec{n}_i^f$ to be more accurate and thus use them instead.

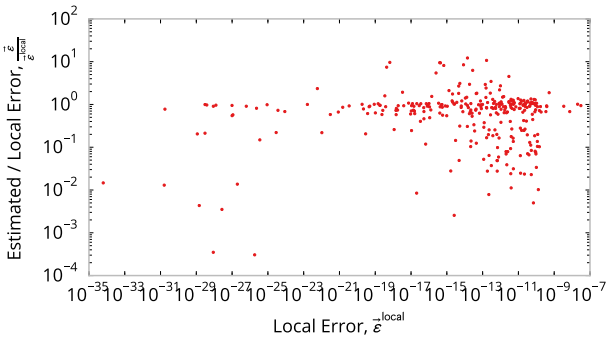
A. Memory storage considerations

For P_0FC_0L, the A_0 matrix is formed at each step and then deallocated at the end of the step. The main additional storage burden for P_2FC_2 is the coefficients of the quadratic polynomial for $A(t)$ for each depletion region, so that we need to store three sets of matrices of the size of A_0 per region. Sparse matrix storage can reduce the size needed to store each A. For the adaptive time stepping algorithm, we also store an old copy of $A(t)$ so that it can be restored if a step needs to be repeated, which means that we need six sets of matrices with the same size and sparsity of A_0 per region. For the 2D single-assembly calculations in CASMO5 performed in this study, the additional storage (using mostly banded-diagonal sparse matrix format) required approximately 50 MB, which is about 20% of the default memory usage. However, for a large multi-assembly or 3D configuration (or if thousands of nuclides are tracked instead of the hundreds that CASMO5 tracks), the additional storage might become prohibitive.

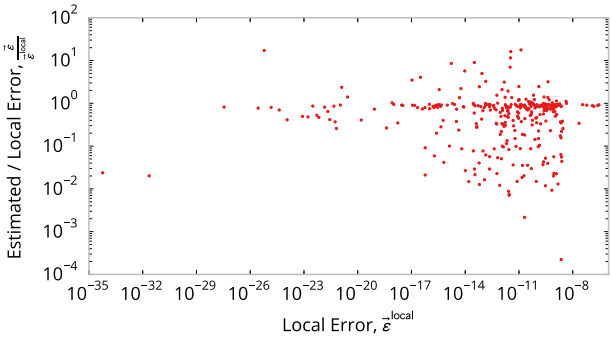
III. RESULTS

1. Error estimators

To gain confidence that the adaptive step size algorithm will work, we first need to verify that the error estimators



(a) Step from 0.5 to 1.0 MWd/kgU.



(b) Step from 15 to 17.5 MWd/kgU.

Fig. 5: Accuracy of P_0FC_0L error estimator.

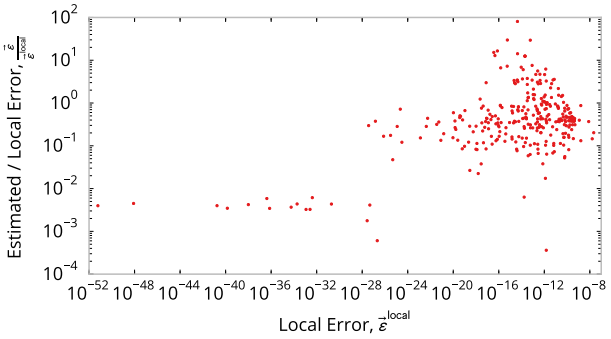
are accurate. Thus, we have compared the accuracies of the P_0FC_0L and P_2FC_2 error estimators for two steps in a pincell depletion calculation. The step were from 0.5 to 1 and 15 to 17.5 MWd/kgU. Figures 5 and 6 show the ratio of the estimated to true local error. The desired result is that all points would equal 1.0, that is, that the predictions are perfect.

For both estimators, the majority of the points are within the range (0.1, 10), where the error estimators are accurate within a factor of 10. The estimators perform better at the later depletion step, especially for the P_2FC_2 estimator. This is presumably because the kinks that appear in $\vec{n}(t)$ and $A(t)$ early in the depletion have smoothed out by 15 MWd/kgU.

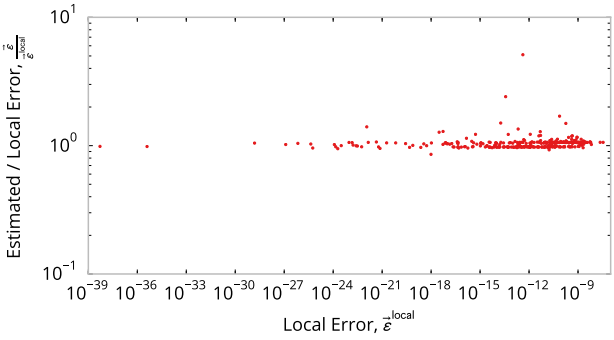
Note that inaccuracies for lower number-density nuclides are not a difficulty because of the absolute error tolerance included in the adaptive stepsize algorithm (τ_a).

2. Adaptive stepsize, quadratic-expansion depletion calculations

A PWR and two BWR lattice depletions are presented here to evaluate the performance of the methods under consideration: scheduled and adaptive versions of both P_0FC_0L and P_2FC_2.



(a) Step from 0.5 to 1.0 MWd/kgU.



(b) Step from 15 to 17.5 MWd/kgU.

Fig. 6: Accuracy of P_2FC_2 error estimator.

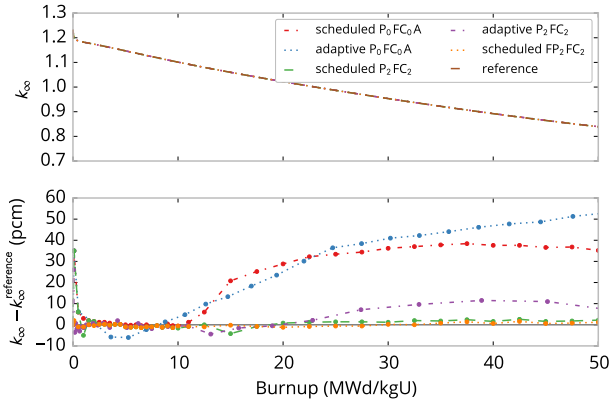


Fig. 7: Eigenvalue and eigenvalue error for 17×17 PWR lattice depletion.

A. 17×17 PWR

The first case is a 17×17 PWR lattice with a uniform enrichment of 4.1%. This is a less strenuous test for the depletion solver. The eigenvalues and errors are presented in Figure 7. Six separate depletion calculations are performed:

- depletions with the default schedule using P_0FC_0L, P_2FC_2, and FP_2FC_2,
- depletions with the adaptive stepsize algorithm P_0FC_0L and P_2FC_2, and
- a fine-step depletion for reference.

The number of depletion steps to reach 50 MWd/kgU is given in Table I. The improvement in accuracy after 10 MWd/kgU is substantial for P_2FC_2 as compared to the default P_0FC_0L.

The error tolerances have been tuned to reproduce a similar level of error in the lattice eigenvalue, and indeed, the P_0FC_0L "scheduled" and "adaptive" results agree within about 10–20 pcm.

B. 10×10 BWR natural-U blanket segment

The second depletion case is a 10×10 BWR with natural uranium (U-235 at 0.71% "enrichment"). This is a more challenging depletion calculation because of the reactivity peak at about 2 MWd/kgU caused by the accumulation of plutonium. Figure 8 shows the lattice infinite multiplication factor, and Figures 9 and 10 show the U-235 and Pu-239 number densities in one of the corner pins. The number of depletion steps to reach 50 MWd/kgU is given in Table I.

The adaptive stepsize algorithm smooths the error. The large gradients in the "scheduled" error are smoothed by the adaptive algorithm.

C. 8×8 BWR with gadolinium burnable absorber

Figure 11 shows the eigenvalue results for the BWR lattice with 10 pins containing gadolinium. The quadratic gadolinium solver in CASMO5 is not currently compatible with the

Case	Scheduled	Adaptive P_0FC_0L	Adaptive P_2FC_2	Reference
PWR	39	24	31	128
BWR	39	31	32	128

TABLE I: Number of depletion steps needed to deplete to 50 MWd/kgU.

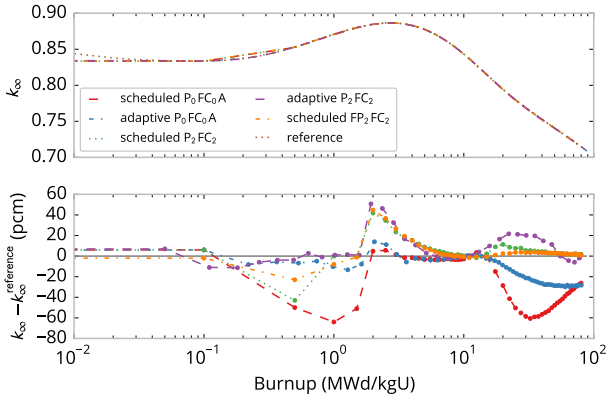


Fig. 8: Eigenvalue and eigenvalue error for depletion of a natural uranium BWR lattice from the assembly blanket.

adaptive stepsize algorithm, so only the scheduled depletion results are given. Furthermore, during the gadolinium burnout period (from 0 to 20 MWd/kgU), the higher-order coupling schemes perform poorly as compared to the default scheme. However, when the quadratic gadolinium solver shuts off around 20 MWd/kgU, the performance of the higher-order P_2FC_2 and FP_2FC_2 schemes improves considerably.

IV. CONCLUSIONS

As expected, the quadratic P_2FC_2 method has substantially better accuracy than the default P_0FC_0L method, es-

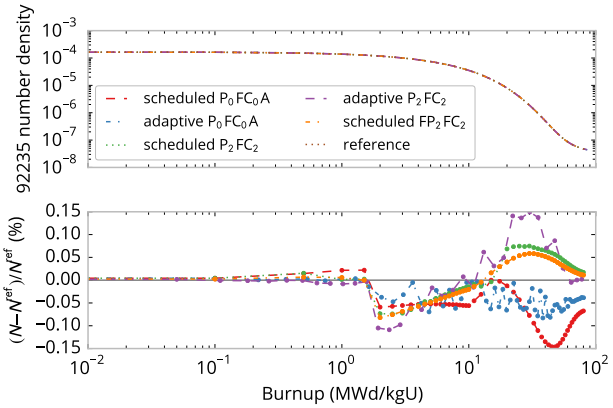


Fig. 9: ^{235}U predictor number densities for corner pin in natural uranium BWR lattice from blanket.

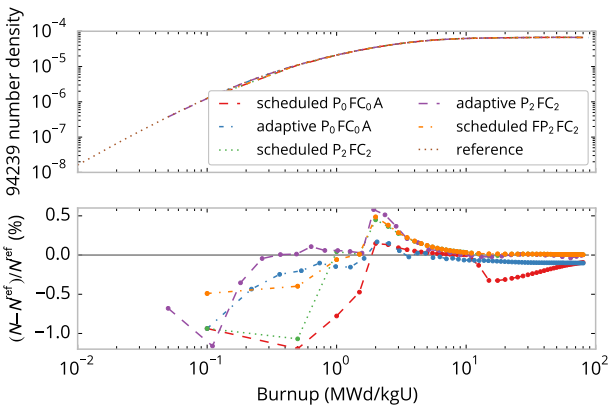


Fig. 10: ^{239}Pu predictor number densities for corner pin in natural uranium BWR lattice from blanket.

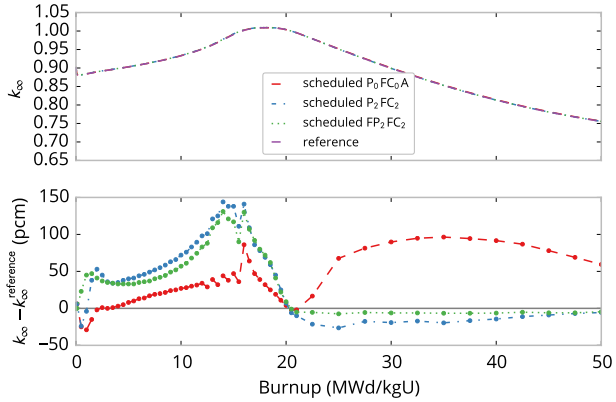


Fig. 11: Eigenvalue and eigenvalue error for depletion of a BWR lattice with Gd burnable absorber.

pecially from the middle to end of the depletion, where the eigenvalue error was reduced by a factor of 3 to 6. In addition, the reduction in the number of statepoints through the use of adaptive stepsize control is appealing. P_2FC_2 with adaptive stepsize control has the potential to reduce CASMO5 runtime while maintaining or improving the solution accuracy, although the net runtime effect is still unknown due to uncertainties in the P_2FC_2 solver.

Several open questions remain:

- What is the net runtime change for adaptive P_2FC_2 versus P_0FC_0L with one-step CRAM, the current CASMO5 default? The P_2FC_2 routines can likely be further optimized to decrease runtime. However, there are other competing effects. For example, the longer steps allowed by the adaptive P_2FC_2 method can increase the number of eigenvalue iterations, since the initial flux guess coming from the previous statepoint is further from the new state.

- Currently adaptive P_0FC_0L and P_2FC_2 are incompatible with the quadratic gadolinium solver. While it seems that these two could coexist, we need to implement the mechanics of such cooperation.

- CASMO5 requires certain exposures (such as 10, 20, 30, ... MWd/kgU) to be present in depletions so that it can perform branch calculations at those points. Any adaptive time step algorithm in CASMO5 should respect those requirements.

- Numerical ODE solvers have methods to automatically choose a good starting step size. This could be a useful addition to an adaptive time-step depletion algorithm so that the user does not need to select the first step size.

- In addition to using variable step size, numerical ODE solvers can also automatically switch the order of the method [5, 12]. Initially, or during periods of steep changes, a lower-order method is used. Then when the solution becomes smoother, a higher-order method is selected. This could be useful in the depletion algorithms, first because it would allow the algorithm to start without needing special logic for the first step or two. The first step would be taken with a low-order method, and then the order could be subsequently increased automatically. Second, while the improved accuracy of P_2FC_2 seems clear after ~15 MWd/kgU, it may be that a low-order method would be a better choice early in the depletion.

REFERENCES

1. A. ISOTALO, "Comparison of Neutronics-Depletion Coupling Schemes for Burnup Calculations," *Nuclear Science and Engineering*, **179**, 434–459 (2015).
2. D. J. WALTER and A. MANERA, "Adaptive burnup stepsize selection using control theory for 2-D lattice depletion simulations," *Progress in Nuclear Energy*, **88**, 218 – 230 (2016).
3. J. RHODES, K. SMITH, and D. LEE, "CASMO5 Development and Applications," in "PHYSOR 2006," Vancouver, BC (2006).
4. D. LEE, J. RHODES, and K. SMITH, "Quadratic Depletion Method for Gadolinium Isotopes in CASMO5," *Nuclear Science and Engineering*, **174**, 79–86 (2013).
5. J. LAMBERT, *Numerical methods for ordinary differential systems: the initial value problem*, John Wiley & Sons, Inc. (1991).
6. A. ISOTALO, "Comparison of Neutronics-Depletion Coupling Schemes for Burnup Calculations–Continued Study," *Nuclear Science and Engineering*, **180**, 286–300 (2015).
7. A. ISOTALO and P. AARNIO, "Higher order methods for burnup calculations with Bateman solutions," *Annals of Nuclear Energy*, **38**, *9*, 1987 – 1995 (2011).
8. A. ISOTALO, G. DAVIDSON, T. PANDYA, W. WIESELQUIST, and S. JOHNSON, "Flux renormalization in constant power burnup calculations," *Annals of Nuclear Energy*, **96**, 148 – 157 (2016).
9. M. EDENIUS, H. HÄGGBLOM, and B. H. FORSSÉN, "CASMO3 Methodology," Tech. Rep. STUDSVIK/NFA-89/2 (1989).
10. D. KNOTT, B. H. FORSSÉN, and M. EDENIUS, "CASMO4 Methodology," Tech. Rep. STUDSVIK/SOA-95/2 (1995).
11. D. KNOTT and A. YAMAMOTO, *Lattice Physics Computations*, Springer US, Boston, MA, pp. 913–1239 (2010).
12. P. BROWN, G. BYRNE, and A. HINDMARSH, "VODE: a variable-coefficient ODE solver," *SIAM Journal on Scientific and Statistical Computing*, **10**, *5*, 1038–1051 (1989).
13. D. C. CARPENTER, "A comparison of constant power depletion algorithms," in "M&C 2009," Saratoga Springs, NY (2009).
14. J. HYKES and R. FERRER, "Solving the Bateman Equations in CASMO5 Using Implicit ODE Numerical Methods for Stiff Systems," in "M&C 2013," Sun Valley, ID (2013).
15. J.-C. C. SUBLET, J. W. EASTWOOD, and J. G. MORGAN, "The FISPACT-II User Manual," Tech. Rep. CCFE-R(11)11 (2014).
16. D. P. GRIESHEIMER, D. C. CARPENTER, and M. H. STEDRY, "Techniques for practical Monte Carlo reactor depletion calculatons," in "PHYSOR 2016," Sun Valley, ID (2016).
17. A. ISOTALO, "Calculating Time-Integral Quantities in Depletion Calculations," *Nuclear Science and Engineering*, **183**, 421–429 (2016).