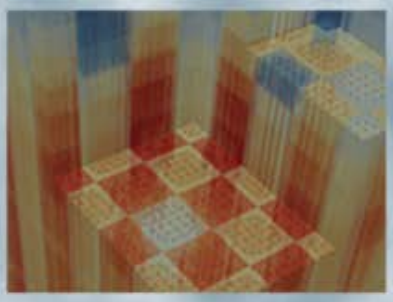
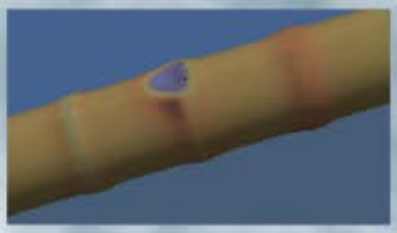
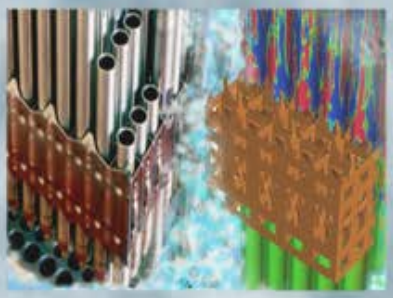
CASL-U-2015-0257-000

Anderson Acceleration for Tiamat







Alexander Toth, Tim Kelley
North Carolina State University

Roger Pawlowki
Sandia National Laboratory

July 7, 2015

U.S. DEPARTMENT OF ENERGY | Nuclear Energy

CASL-U-2015-0257-000

Anderson Acceleration for Tiamat

Alex Toth – North Carolina State University

With: Tim Kelley (NCSU), Roger Pawlowski (Sandia)

Introduction

- Tiamat is code coupling being developed in CASL as a tool for analysis of pellet-cladding interaction (PCI).
- Couples together the Bison-CASL fuel performance code with the COBRA-TF thermal hydraulics code and either MPACT or Insilico neutronics (we utilize Insilico) to identify fuel rods and assemblies of interest for further PCI calculations.
- Tiamat has previously used Picard iteration for this coupling, which is attractive primarily for its simplicity of implementation, though it has several drawbacks:
  - Relatively slow rate of convergence.
  - Necessity of ad hoc chosen damping factors to obtain acceptable performance.
- Anderson acceleration is an alternative solution method which has been seen to be faster converging and more numerically robust than Picard in several fields.
- In this work, we implement Anderson acceleration in Tiamat and evaluate its potential as an alternative to Picard iteration.

Tiamat Code Coupling

Participating codes:

- COBRA-TF (CTF) – Thermal hydraulic simulation code developed at PSU. Models coolant conditions using two-fluid, three-field representation of two-phase flow.
- Insilico – Neutronics solver from the VERA code suite. Uses XSProc to generate multigroup cross sections and solves S_n or SP_n form of the transport equation.
- Bison-CASL – Single-rod fuel performance code. Models mechanical, thermal, and chemical properties of an individual fuel rod.
- Data Transfer Kit (DTK) – Handles transfers of data between codes, accounting for parallel communication, unit conversions, coordinate orientation, etc.

Bison-CASL is transient and must model the heating of a fuel rod from zero power to full power. This is performed in Tiamat according to the following process:

1. Obtain an estimate of hot full power (HFP) conditions. Several ways to do this:
   a. Perform one or more iterations of stand-alone CTF+Insilico.
   b. Start from data saved in a restart file.
   c. A combination of a. and b.
2. Model transition from cold zero power (CZP) to hot zero power (HZP) in Bison. Bison is run transient for 100 seconds, linearly increasing clad surface temperature from 20 to 293 C.
3. Model transition from HZP to HFP in Bison. Bison is run transient for 48 hours, linearly increasing clad surface temperature and power from HZP conditions to the estimated HFP conditions.
4. Solve the fully-coupled problem at HFP for one or more time steps.

- To this point, the fully-coupled HFP problem has been solved by Picard iteration (also known as fixed-point iteration or successive substitution).
- Utilizes two variations of Picard iteration:
  - Block Jacobi: Alternate between phase of solving of all applications and phase of transferring updated coupling data.
  - Block Gauss-Seidel: Solve individual applications sequentially and transfer updated coupling data as soon as it is obtained.

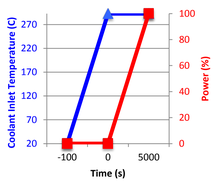


Figure 1: Bison-CASL ramp to HFP

Anderson Acceleration

- Acceleration method for solving the fixed-point problem u = G(u).
- Stores at most m (user specified number) previous function evaluations and computes new iterate as the linear combination of these with minimal linearized residual:

$$u_{n+1} = (1-\beta)\sum_{j=0}^{m_n}\alpha_j^{(n)}u_{n-m_n+j} + \beta\sum_{j=0}^{m_n}\alpha_j^{(n)}G(u_{n-m_n+j})$$

where $m_n = \min\{m,n\}$, and $\{\alpha_j^{(n)}\}$ solves the constrained least-squares problem:

$$\min_{\{\alpha_j\}}\left\|\sum_{j=0}^{m_n}\alpha_j[G(u_{n-m_n+j}) - u_{n-m_n+j}]\right\|, \quad \text{subject to} \quad \sum_{j=0}^{m_n}\alpha_j = 1$$

- The algorithm with any particular m is called Anderson-m (Anderson-0 is Picard), and β is called the mixing parameter.
- No derivative information or linear iterations required, giving an advantage over Newton-like methods.
- Assuming G is relatively expensive, the dominant computational cost per iteration is the evaluation of G (same as Picard).
- Main added cost over Picard is storage of additional vectors; Anderson-m requires the storage of at most 2(m+1) vectors.
- Theoretical background:
  - Anderson acceleration may be viewed as a sort of quasi-Newton method [Fang and Saad, 2009].
  - Anderson acceleration is "essentially equivalent" to GMRES iteration for linear problems [Walker and Ni, 2011].
  - Anderson acceleration is locally r-linearly convergent if G is nonlinear and contractive near the solution [Toth and Kelley, 2015].

Fixed-Point Map Definitions

- To implement Anderson acceleration for this problem, we need to define the fixed-point map G to which it will be applied.
- G is formulated in such a way that Picard iteration can be written as $u_{n+1} = G(u_n)$.
- We pose the problem as a fixed-point problem in terms of transferred data.
- Tiamat includes 5 data transfers:
  - CTF to Bison: Surface clad temperatures T_c from CTF are transferred to Bison.
  - Bison to CTF: Heat flux q'' from Bison is transferred to CTF.
  - CTF to Insilico: Coolant temperatures T_w and densities ρ_w from CTF are transferred to Insilico
  - Bison to Insilico: Fuel temperatures T_f from Bison are transferred to Insilico.
  - Insilico to Bison: Power density q from Insilico is transferred to Bison.
- Indicate source/target data by a superscript (or subscript for heat flux) indicating the code: B for Bison, C for CTF, and I for Insilico.
- In this work, we consider Anderson acceleration applied to two fixed-point maps:

Gauss-Seidel Map

- Given T_f^I, T_w^I, ρ_w^I, and T_c^B.
- Using, T_f^I, T_w^I, and ρ_w^I solve neutronics.
- Transfer $q^I \to q^B$.
- Using T_c^B and q^B, solve Bison.
- Transfer $q_B'' \to q_C''$ and $T_f^B \to \bar{T}_f^I$.
- Using q_C'' solve CTF.
- Transfer $T_w^C \to \bar{T}_w^I, \rho_w^C \to \bar{\rho}_w^I$, and $T_c^C \to \bar{T}_c^B$.

$$G_{GS}\begin{pmatrix} T_f^I \\ T_w^I \\ \rho_w^I \\ T_c^B \end{pmatrix} = \begin{pmatrix} \bar{T}_f^I \\ \bar{T}_w^I \\ \bar{\rho}_w^I \\ \bar{T}_c^B \end{pmatrix}$$

Intermediate Map

- Given $T_f^I, T_w^I, \rho_w^I, T_c^B$, and q^B.
- Using, T_f^I, T_w^I, and ρ_w^I solve neutronics.
- Using T_c^B and q^B, solve Bison.
- Transfer $q_B'' \to q_C''$ and $T_f^B \to \bar{T}_f^I$.
- Transfer $q^I \to \bar{q}^B$.
- Using q_C'' solve CTF.
- Transfer $T_w^C \to \bar{T}_w^I, \rho_w^C \to \bar{\rho}_w^I$, and $T_c^C \to \bar{T}_c^B$.

$$G_{INT}\begin{pmatrix} T_f^I \\ T_w^I \\ \rho_w^I \\ T_c^B \\ q^B \end{pmatrix} = \begin{pmatrix} \bar{T}_f^I \\ \bar{T}_w^I \\ \bar{\rho}_w^I \\ \bar{T}_c^B \\ \bar{q}^B \end{pmatrix}$$

- The Intermediate map falls in-between block Gauss-Seidel and block Jacobi.
- In current form of Tiamat, applications exist in independent processor space (Figure 2).
  - For the Gauss-Seidel map, applications must be solved in series.
  - For the Intermediate map, Insilico and Bison are solved simultaneously.
- The Intermediate map couples applications more loosely, so higher iteration counts should be expected. More efficient processor utilization may still result in lower run times.

Numerical damping of certain fields is necessary to obtain convergence for Picard iteration.

- Given damping factor $0<\alpha\leq 1$, set the component of G corresponding to the field x to the weighted sum $(1-\alpha)x + \alpha\bar{x}$ rather than $\bar{x}$.
- We apply a separate damping to temperatures, densities, and powers.
- The mixing parameter in Anderson applies the same damping to each exposed field.

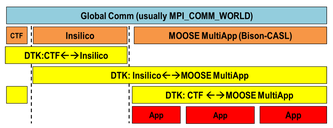


Figure 2: MPI communicator layers in Tiamat

Scaling of Variables

Problem: The residual vector being exposed to Anderson acceleration consists of physical quantities which may be on vastly different orders of magnitude.

- Larger quantities may dominate the solution of the least-squares problem.
- Poor linear combination coefficients for smaller magnitude quantities may slow overall rate of convergence.

Solution: Scale the variables that are exposed to Anderson acceleration, i.e. instead of solving u = G(u), consider the scaled variables v = Mu, where M is a diagonal scaling matrix, and solve the scaled fixed-point problem $v = MG(M^{-1}v)$.

For the Gauss-Seidel map, the exposed transfers are temperatures and densities, and we scale these by their initial value, i.e. $M_{ii} = u_{0,i}^{-1}$ where u_0 is the initial iterate.

For the Intermediate map, power density is another exposed quantity, and these can be zero (e.g. in non-fuel pins). For this map, temperatures and densities are scaled as above, and powers are scaled by the local average power within the owning processor (or the scaling factor is set to 1 if the local average power is 0).

Solver Implementation

- We utilize the Anderson acceleration solver in the Trilinos nonlinear solver package NOX.
- Tiamat heavily leverages the PIKE (Physics Integration Kernels) software package for solvers, observers, status tests, etc., so a wrapper class was created which implements the NOX solver as a PIKE solver.
- The NOX solver is used to converge the data transferred between applications, but PIKE determines global convergence of the coupled system. The following criteria are required to be satisfied:
  - CTF: Change in max coolant and clad temperatures less than some tolerance.
  - Bison: Change in the max fuel temperature for each rod less than some tolerance.
  - Insilico: L2 norm of relative power change and absolute change in the eigenvalue less than some tolerance.

Single Fuel Rod Results

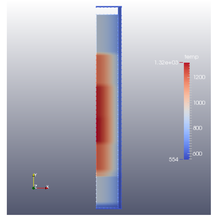


Figure 3: Final temperature distrubtion in the fuel rod, computed by Bison-CASL

First consider some parametric studies for a problem consisting of a single fuel rod.

- Fuel is 3.1% enriched UO_2.
- The moderator contains 1300 ppm dissolved boron.
- This problem includes 3 spacer grids, and a relatively coarse axial mesh consisting of 9 regions in the fuel.
- All tests in this work use the XSProc 8 group test library to generate cross sections for Insilico.
- For these tests, temperature tolerance is 1K, power tolerance is 1e-3 and eigenvalue tolerance is 2 pcm.
- These tests are run of 4 processors: 2 for Insilico, 1 for Bison-CASL, and 1 for CTF.
- For Picard tests damping refers to a power damping and for Anderson it is the mixing parameter.

Figures 4 and 5 show a comparison of run times for Picard and Anderson with several storage depths over a variety of damping/mixing parameters:

- Anderson converges for each test considered, whereas Picard fails to converge without sufficient damping.
- Anderson run times can increase for small damping factors, but performance is otherwise consistently competitive with or better than optimally damped Picard.
- Performance of Anderson suffers as storage depth is increased above 2.
- Bison solves dominate timings, no gains from solving Insilico and Bison simultaneously.

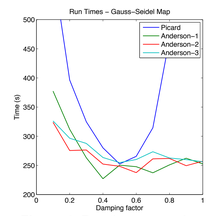


Figure 4: Run time comparison: Gauss-Seidel map

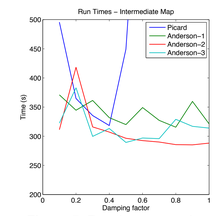


Figure 5: Run time comparison: Intermediate map

Tables 1 and 2 show the effect varying power level has on the iterations:

- At each power level, eigenvalues differ by less than 2 pcm, which is less than the tolerance for global convergence.
- For each test at 150% power, the iteration fails as CTF does not converge in the first step in the coupled HFP solve.
- Gauss-Seidel tests are insensitive to the power level, but for the Intermediate map, iteration counts increase with power, somewhat less so for Anderson than Picard.

Table 1: Power variation for the Gauss-Seidel map, damping/mixing factor = 0.5

	Picard		Anderson-2	
Power	Eigenvalue	Iterations	Eigenvalue	Iterations
25%	1.2057739092	6	1.2057631718	6
50%	1.2019279876	7	1.2019219774	6
75%	1.1982826604	6	1.1982741575	6
100%	1.1946961900	6	1.1946833864	6
125%	1.1911857865	6	1.1911770197	6
150%	CTF DNC		CTF DNC	

Table 2: Power variation for the Intermediate map, damping/mixing factor = 0.4

	Picard		Anderson-2	
Power	Eigenvalue	Iterations	Eigenvalue	Iterations
25%	1.2057703126	8	1.2057693978	7
50%	1.2019180590	9	1.2019205968	11
75%	1.1982627453	10	1.1982716393	10
100%	1.1946874863	12	1.1946820355	11
125%	1.1911663355	15	1.1911712479	10
150%	CTF DNC		CTF DNC	

3x3 Mini-Assembly Results

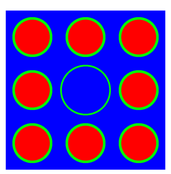


Figure 6: 3x3 mini-assembly geometry

Consider a 3x3 array consisting of 8 3.1% UO_2 fuel rods around a central guide tube (Figure 6).

- CTF+Insilico sub-cycled for 3 iterations before ramp to HFP. Provides a very good initial iterate for coupled HFP solve.
- Tolerances: 1K for temperatures, 1e-3 for power, and 5 pcm for the eigenvalue
- Tests are run on 16 processors: 9 for Insilico, 6 for Bison, and 1 for CTF.

Run times/iteration counts for these tests given in Table 3:

- Like single rod tests, Picard timings strongly dependent on damping level, but Anderson performs competitively with optimally damped Picard over wide range of parameters.
- Best time is obtained for Anderson with Intermediate map, because Insilico and Bison solves are relatively balanced (about 9 seconds for Insilico and 25 seconds for Bison)
- For problems like this where Picard converges very quickly, Anderson likely won't reduce iteration counts as Picard and Anderson are the same for the first two iterations.

Table 3: Run times (and iteration counts in parentheses) for 3x3 mini-assembly tests

	Gauss-Seidel Map		Intermediate Map	
Damping/Mixing Factor	Picard	Anderson-2	Picard	Anderson-2
0.2	790.0(9)	622.7(5)	814.7(9)	561.9(6)
0.4	514.9(4)	513.2(4)	524.2(5)	522.6(5)
0.5	517.0(4)	526.1(4)	522.1(5)	482.9(5)
0.6	485.1(4)	516.5(4)	611.7(7)	476.5(5)
0.8	802.2(11)	563.8(5)	DNC	527.2(6)
1.0	DNC	566.3(5)	DNC	724.6(9)

17x17 Assembly Results

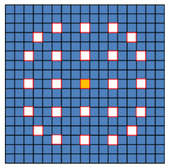


Figure 7: P6 lattice

For the last set of tests, we consider the VERA benchmark progression problem P6.

- This problem consists of a single 17x17 assembly at HFP with 264 fuel rods, 24 guide tubes, and one central instrument tube (Figure 7).
- These tests use restart data for the HPF approximation, no CTF+Insilico sub-cycling.
- Tolerances: 1K for temperatures, 1e-4 for power, and 2 pcm for the eigenvalue
- These tests were run on 64 processors: 36 for Insilico, 27 for Bison-CASL, and 1 for CTF.

Table 4: Iteration statistics for P6, damping factor = 0.5

	k_{eff}	Iterations	Time(s)
Picard (Gauss-Seidel map)	1.2030101	7	9608
Anderson-2 (Gauss-Seidel map)	1.2030117	5	7930
Picard (Intermediate map)	1.2030075	11	11660
Anderson-2 (Intermediate map)	1.2030097	10	10680

Table 4 shows a comparison of methods to solve this problem with equal damping factors:

- Very good agreement in the solution (less than 1 pcm difference in the eigenvalues).
- For both maps, Anderson provides an improvement in iteration count over Picard.
- The Insilico solve is relatively inexpensive for this problem due to cheap cross section processing, and as a result the Intermediate map does comparatively poorly.

Table 5: Anderson-2 for Intermediate map, varying damping

	Power Damping Factor			
	0.25	0.5	0.75	1.0
Iterations	14	10	10	10
Time(s)	12830	10680	10650	10170

Table 5 shows run times for Anderson-2 with the Intermediate map at various damping levels:

- Anderson again converges at each damping level.
- Convergence slows if the damping factor is made too small, but performance is again consistent over a wide range of parameters.

Figure 8 shows the agreement between Picard and Anderson for several assembly-averaged physical quantities:

- For each quantity, maximum relative difference is on the order of 1e-4, which is the power convergence tolerance.

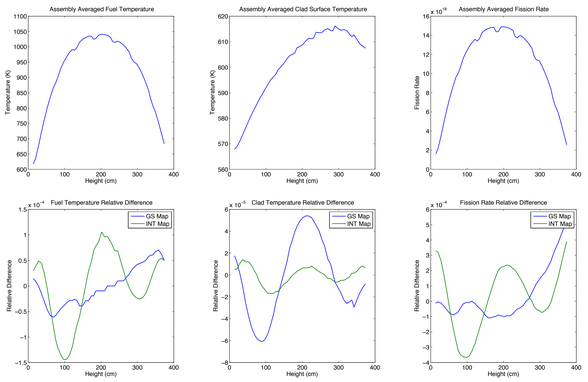


Figure 8: Assembly-averaged solutions from Picard iteration, and relative difference between Anderson acceleration solutions and the Picard solution.

Conclusions

- Anderson acceleration has been successfully integrated into the Tiamat coupling between Bison-CASL, CTF, and Insilico.
- Comparison with Picard iteration shows that the methods achieve roughly the same level of accuracy in the solution.
- Parametric studies on a variety of test problem sizes display advantages of utilizing Anderson acceleration as an alternative solution method to Picard iteration.
  - Good performance of Picard depends strongly on correct choice of damping factors.
  - Anderson displays performance competitive with or better than optimally damped Picard over a wide range of parameters.
  - Despite significant processor idle time, tests with the Gauss-Seidel map generally display best timings.

Acknowledgements

This research used resources of the Oak Ridge National Laboratory and the CASL Program which is supported by the Department of Energy's Office of Science DE-AC05-00OR222725.

CASL-U-2015-0257-000