

Jacobian-Free Newton-Krylov (JFNK) Methods for Nonlinear Neutronics/Thermal-Hydraulic Equations

Bryan Herman

December 10, 2011

Outline

1 Introduction

1 Governing Equations

1 Solvers

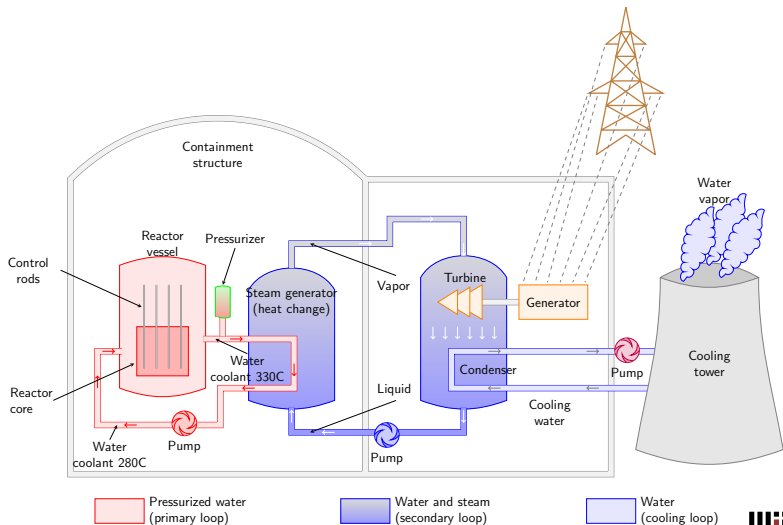
1 Results

1 Conclusions

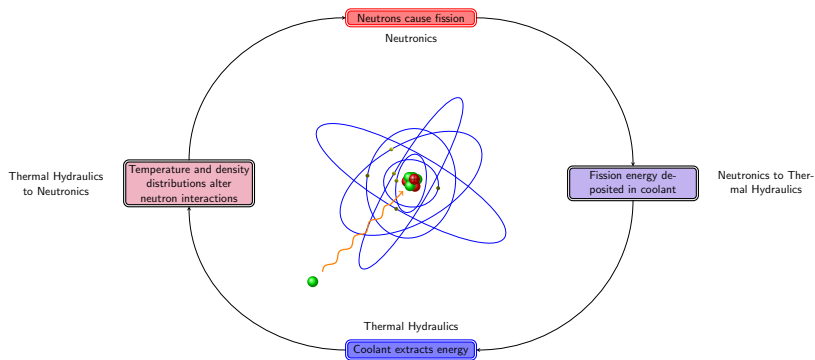
Motivation

- Eventually will be part of thesis work
- JFNK method not currently used in nuclear reactor analysis
- Incorporates a lot of ideas from 2.29 class
- Coupled physics is fun!

Nuclear Reactor Plant



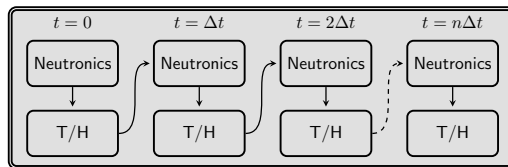
Nuclear Feedback



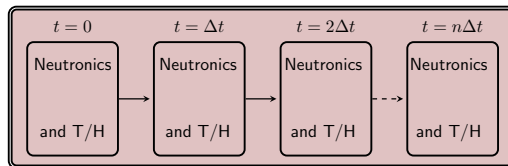
- Fuel Temperature Feedback - $T_f \uparrow$, U-238 Capture \uparrow , Fission Rate \downarrow , Power \downarrow
- Coolant Density Feedback - $\rho \downarrow$, $E_n \uparrow$, Fission Rate \downarrow , Power \downarrow

Common Approach to Coupling: Operator Splitting

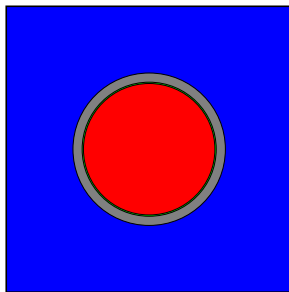
- Solve physics independently and iterate between them



- Fully coupled approach solves the nonlinear physics together

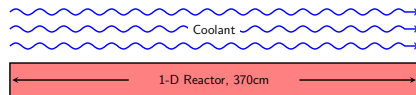


1-D Slab Reactor Geometry



■ Coolant ■ Clad ■ Gas Gap ■ Fuel

Top-view
Fuel Rod Unit-Cell



Side-view (vertical)
1-D Model of Reactor

Neutronics

■ Basic Neutron Conservation:

$$\textit{Change} + \textit{Leakage} + \textit{Interactions} = \textit{Scattering} + \textit{Fission}$$

■ Neutron Transport Equation:

$$\underbrace{\frac{1}{v} \frac{\partial \varphi}{\partial t}}_{\text{time-dependent}} + \underbrace{\boldsymbol{\Omega} \cdot \nabla \varphi(\mathbf{r}, E, \boldsymbol{\Omega}, t)}_{\text{neutron leakage}} + \underbrace{\Sigma_t(\mathbf{r}, E, t) \varphi(\mathbf{r}, E, \boldsymbol{\Omega}, t)}_{\text{interaction of neutrons with medium}} = \underbrace{Q(\mathbf{r}, E, \boldsymbol{\Omega}, t)}_{\text{neutron source}}$$

■ Neutron Diffusion Equation (1-D Energy Integrated)

$$\frac{1}{v} \frac{\partial \phi}{\partial t} - D(x, t) \frac{\partial^2 \phi}{\partial^2 x} + \Sigma_a(x, t) \phi(x, t) = \frac{1 - \beta}{k_{eff}} \nu \Sigma_f(x, t) \phi(x, t) + \lambda_d c(x, t)$$

$$\frac{\partial c}{\partial t} = \frac{\beta}{k_{eff}} \nu \Sigma_f(x, t) \phi(x, t) - \lambda_d c(x, t)$$

Discretization of Neutronics Equations

Assumptions:

- 1 One-dimensional finite volume spatial discretization
- 2 Central difference scheme for diffusion term
- 3 No incoming current of neutrons at boundaries
- 4 Implicit Euler time discretization

- Discretized neutronics equation for interior cell

$$\frac{1}{v} \frac{d\bar{\phi}_i}{dt} - \frac{2}{\Delta x^2} \frac{D_i D_{i-1}}{D_i + D_{i-1}} \bar{\phi}_{i-1} + \left(\frac{2}{\Delta x^2} \frac{D_{i+1} D_i}{D_{i+1} + D_i} + \frac{2}{\Delta x^2} \frac{D_i D_{i-1}}{D_i + D_{i-1}} + \Sigma_{a,i} \right) \bar{\phi}_i - \frac{2}{\Delta x^2} \frac{D_{i+1} D_i}{D_{i+1} + D_i} \bar{\phi}_{i+1} = \frac{1 - \beta}{k_{eff}} \nu \Sigma_{f,i} \bar{\phi}_i + \lambda_d \bar{c}_i$$

- Matrix-form of neutronics equations

$$\bar{\Phi}^{n+1} - \bar{\Phi}^n + v \Delta t (\mathbb{M} \bar{\Phi}^{n+1} - (1 - \beta) \lambda \mathbb{F} \bar{\Phi}^{n+1} - \lambda_d \bar{\mathbf{c}}^{n+1}) = 0$$

- Matrix-form of precursors

$$\bar{\mathbf{c}}^{n+1} - \bar{\mathbf{c}}^n + \Delta t (\lambda_d \bar{\mathbf{c}}^{n+1} - \beta \lambda \mathbb{F} \bar{\Phi}^{n+1}) = 0$$

Thermal Hydraulics

- Energy Equation - single phase fluid and inviscid fluid

$$\frac{\partial(\rho h)}{\partial t} + \nabla \cdot (\rho h \mathbf{u}) = -\nabla \cdot \mathbf{q}'' + q'''$$

- Assuming fissions are a volumetric heat source in 1-D

$$\rho A \frac{\partial h}{\partial t} + \dot{m} \frac{\partial h}{\partial x} = q'$$

- For an incompressible fluid, $dh = c_p dT$

$$\rho A c_p \frac{\partial T}{\partial t} + \dot{m} c_p \frac{\partial T}{\partial x} = q'$$

Discretization of Energy Equation

Assumptions:

- 1 One-dimensional finite volume spatial discretization
- 2 Upwind difference scheme for flux
- 3 Specify inlet conditions and mass flow rate
- 4 Implicit Euler time discretization

■ Spatial discretization

$$\frac{\rho A \Delta x}{\dot{m}} \frac{d\bar{T}_i}{dt} + \bar{T}_i - \bar{T}_{i-1} = \frac{1}{2\dot{m}c_p} (Q_{i-1} + Q_i)$$

■ Matrix-form with time discretization

$$\bar{\mathbf{T}}^{n+1} - \bar{\mathbf{T}}^n + \frac{\dot{m}\Delta t}{\mathcal{P}^{n+1}A\Delta x} (\mathbb{S}\bar{\mathbf{T}}^{n+1} - \mathbb{R}\mathbf{Q}^{n+1}) = 0$$

Note: \mathcal{P} is a vector of cell-averaged coolant densities

Physics Coupling

Neutronics-Thermal Hydraulics

- Neutrons cause fission
- Large portion of fission energy deposited in coolant
- This is represented by

$$\mathbf{Q} = \tilde{c} \mathbb{E} \bar{\Phi} \Delta x$$

where $\mathbb{E} = \text{diag} \{ \kappa \Sigma_f \}$ characterizes energy per fission and \tilde{c} is the flux-to-power normalization constant

Thermal Hydraulics-Neutronics

- Diffusion theory parameters depend on coolant density
- This dependence is determined with a transport theory code
- D , Σ_a , $\nu \Sigma_f$, $\kappa \Sigma_f$ are all affected by coolant density variations
- Data is correlated with a linear regression of the form:

$$\Sigma = \Sigma^{ref} + \frac{\partial \Sigma}{\partial \rho} (\rho - \rho^{ref})$$

The Steady State Eigenvalue Problem

- The steady state equations must be solved first
- Reducing the neutronics equation to steady state form:

$$\mathbb{M}\bar{\Phi} = \lambda\mathbb{F}\bar{\Phi}, \quad \lambda = \frac{1}{k_{eff}}$$

- Eigenvalue, λ , and eigenvector, $\bar{\Phi}$, must be determined
- Flux-to-power normalization constant determined from reactor power:

$$Q_R = \tilde{c} \int_0^L dx \kappa \Sigma_f(x) \phi(x) = \tilde{c} \sum_i \kappa \Sigma_{f,i} \bar{\phi}_i \Delta x = \tilde{c} \kappa \Sigma_f^T \bar{\Phi} \Delta x$$

- λ and \tilde{c} are specified as constants for time-dependent calculations

Newton's Method

Procedure:

```
1: Goal:  $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ 
2: Guess  $\mathbf{x}$ 
3: for  $n = 1, 2, 3, \dots$  do
4:    $\mathbf{r} = \mathbf{F}(\mathbf{x})$ 
5:   if  $\|\mathbf{r}\| < ntol$  then
6:     DONE!
7:   end if
8:    $d\mathbf{x} = -\mathbb{J}^{-1}(\mathbf{x}) \mathbf{F}(\mathbf{x})$ 
9:    $\mathbf{x} = \mathbf{x} + d\mathbf{x}$ 
10: end for
```

Three major tasks:

- 1 Evaluate residual vector in external function
- 2 Evaluate Jacobian in external function (Do we have to?)
- 3 Calculate $d\mathbf{x}$ - Direct or Iterative Solvers?

Krylov Subspace Methods

- A class of iterative methods for sparse systems
- Solves $\mathbb{A}\mathbf{x} = \mathbf{b}$ by projecting a m dimensional problem into a lower dimensional Krylov subspace

$$\mathcal{K}_n(\mathbb{A}, \mathbf{v}) = \text{span} \{ \mathbf{v}, \mathbb{A}\mathbf{v}, \mathbb{A}^2\mathbf{v}, \dots, \mathbb{A}^{n-1}\mathbf{v} \}$$

- Here \mathbb{A} is nonhermitian and we use the GMRES method
- GMRES uses the Arnoldi method to reduce the system to Hessenberg form

$$\mathbb{A}\mathbb{Q} = \mathbb{Q}\mathbb{H}$$

$$\mathbb{H} = \begin{bmatrix} h_{11} & & \cdots & h_{1n} \\ h_{21} & h_{22} & & \\ & \ddots & \ddots & \vdots \\ & & h_{n,n-1} & h_{n,n} \end{bmatrix}$$

Generalized Minimal RESidual Method

- Goal: $\mathbf{x}_* = \mathbb{A}^{-1}\mathbf{b}$
- A step n , \mathbf{x}_* is approximated by $\mathbf{x}_n \in \mathcal{K}_n$ that minimizes the norm of the residual $\mathbf{r}_n = \mathbf{b} - \mathbb{A}\mathbf{x}_n$

Procedure:

```

1:  $q_1 = b / \|b\|$ 
2: for  $n = 1, 2, 3, \dots$  do
3:   Perform step  $n$  of Arnoldi (Creates Hessenberg matrix)
4:   Find  $y$  to minimize  $\|\tilde{\mathbb{H}}_n \mathbf{y} - \|b\| \mathbf{e}_1\|$ 
5:   if  $\|\mathbf{r}\| < \text{ltol}$  then
6:     DONE!
7:   end if
8: end for
9:  $\mathbf{x} = \mathbb{Q}_n \mathbf{y}$ 

```

- Saad et al.[‡] defines a novel method to compute $\tilde{\mathbb{H}}_n$ from $\tilde{\mathbb{H}}_{n-1}$ from Givens rotations

[‡]Yousef Saad and Martin H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. Society for Industrial and Applied Mathematics, 7:856-859, 1986.

Inexact Newton's Method

- Newton-Krylov is an inexact Newton method since the linear step is not determined *exactly*
- In Newton-Krylov framework, two tolerances were defined:
 - 1 Nonlinear tolerance for Newton iteration
 - 2 Linear tolerance for GMRES iteration
- Why have tight linear convergence when nonlinear residual is large?
- Instead, a relative residual tolerance, η , is used

$$\|\mathbb{J}(\mathbf{x}^n) \mathbf{d}\mathbf{x}_m^n + \mathbf{F}(\mathbf{x}^n)\| < \eta \|\mathbf{F}(\mathbf{x}^n)\|$$

- At initial Newton iterations, GMRES will not be converged very tightly
- For the last couple of Newton iterations, convergence may be too tight
∴ limit how small linear tolerance can get

Jacobian-Free Approximation

- Recall a Krylov subspace: $\mathcal{K}_n(\mathbb{A}, \mathbf{v}) = \text{span} \{ \mathbf{v}, \mathbb{A}\mathbf{v}, \mathbb{A}^2\mathbf{v}, \dots, \mathbb{A}^{n-1}\mathbf{v} \}$
- Why create \mathbb{A} when it is only used to multiply a vector?
- Option 1: Perform Jacobian-vector product analytically

$$\mathbb{J}\mathbf{y} = \begin{bmatrix} \mathbb{M} - \lambda\mathbb{F} & -\mathbb{F}\bar{\Phi} \\ -\bar{\Phi}^\top & 0 \end{bmatrix} \begin{bmatrix} y_\phi \\ y_\lambda \end{bmatrix} = \begin{bmatrix} (\mathbb{M} - \lambda\mathbb{F})y_\phi - \mathbb{F}\bar{\Phi}y_\lambda \\ -\bar{\Phi}^\top y_\phi \end{bmatrix}$$

- Option 2: Approximate Jacobian-vector product with finite difference

$$\mathbb{J}\mathbf{y} \approx \frac{\mathbf{F}(\mathbf{x} + \epsilon\mathbf{y}) - \mathbf{F}(\mathbf{x})}{\epsilon}$$

- Advantages: Saves memory and possibly computational time to form Jacobian
- ϵ is the perturbation parameter and is somewhat arbitrary - Mousseau[§] recommends:

$$\epsilon = \frac{\sum_{i=1}^N bx_i}{N \|\mathbf{y}\|_2} \quad b = 1 \times 10^{-8}$$

[§]V.A. Mousseau. Implicitly balanced solution of the two-phase flow equations couple to nonlinear heat conduction. Journal of Computational Physics, 200:104-132, 2004.

Preconditioning

- Want to limit number of GMRES iterations
- Before a calculation, a Jacobian matrix is formed analytically and a zero-fill Incomplete LU (ILU) is performed:

$$\mathbb{R} = \mathbb{L}\mathbb{U} - \mathbb{A}$$

- In ILU, residual matrix \mathbb{R} is constrained to certain conditions
- Zero-fill implies that the number and location of nonzeros is preserved
- Left preconditioning is used in this project:

$$\mathbb{U}^{-1}\mathbb{L}^{-1}\mathbb{A}\mathbf{x} = \mathbb{U}^{-1}\mathbb{L}^{-1}\mathbf{b}$$

Steady State - Neutronics

- Residual Equations

$$\mathbf{F} = \begin{bmatrix} \mathbf{M}\bar{\Phi} - \lambda \mathbf{F}\bar{\Phi} \\ -\frac{1}{2}\bar{\Phi}^\top \bar{\Phi} + \frac{1}{2} \end{bmatrix}$$

- Resulting neutron flux distribution:

- A Newton method cannot guarantee that the fundamental eigenmode is calculated
- Any mode satisfies the nonlinear set of equations
- Here the ratio of the first two eigenvalues is close to unity (dominance ratio)
- Use a few power iterations to get gross flux shape

Steady State - Coupled Neutronics/Thermal Hydraulics [1/2]

Residual Equations

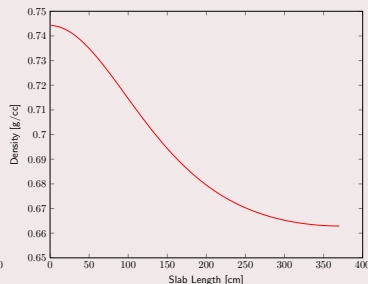
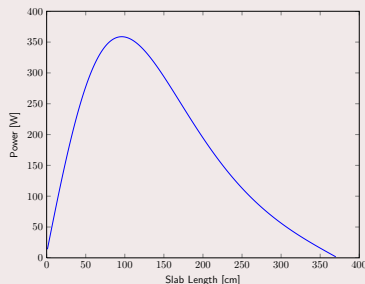
$$\mathbf{F} = \begin{bmatrix} \mathbf{M}\Phi - \lambda\mathbf{F}\Phi \\ Q_R - \tilde{c}\kappa\Sigma_f^T\Phi\Delta x. \\ \mathbf{Q} - \tilde{c}\mathbf{E}\Phi\Delta x \\ \mathbf{S}\mathbf{T} - \mathbf{R}\mathbf{Q} \\ \mathcal{P} - \rho(\mathbf{T}, p) \\ \Sigma_a - \Sigma_a^{ref} - \frac{\partial \Sigma_a}{\partial \rho} [\mathcal{P} - \rho^{ref}] \\ \nu\Sigma_f - \nu\Sigma_f^{ref} - \frac{\partial \nu\Sigma_f}{\partial \rho} [\mathcal{P} - \rho^{ref}] \\ \mathbf{D} - \mathbf{D}^{ref} - \frac{\partial \mathbf{D}}{\partial \rho} [\mathcal{P} - \rho^{ref}], \\ \kappa\Sigma_f - \kappa\Sigma_f^{ref} - \frac{\partial \kappa\Sigma_f}{\partial \rho} [\mathcal{P} - \rho^{ref}] \\ -\frac{1}{2}\Phi^T\Phi + \frac{1}{2} \end{bmatrix}$$

- Neutronics
- Flux normalization
- Energy deposition
- Temperature distribution
- Density distribution
- Absorption xs
- Fission xs
- Diffusion
- Energy deposition xs
- Eigenvalue

- X-Steam MATLAB tables were used to look up density as a function of temperature
- An analytical Jacobian-vector product can be formed for everything except $\frac{\partial \rho}{\partial T}$
- For this block a finite difference approximation must be used

Steady State - Coupled Neutronics/Thermal Hydraulics [2/2]

Results



- Timing results using MATLAB profiler:
 - 7 seconds to converged overall
 - 6+ out of 7 seconds in X-Steam!
 - Solution: fit density range with polynomial - with linear fit ; 0.5 seconds

Transient - Coupled Neutronics/Thermal Hydraulics [1/2]

Residual Equations

$$\mathbf{F} = \begin{bmatrix} \Phi^{n+1} - \Phi^n + v\Delta t [\mathbf{M}\Phi^{n+1} - (1 - \beta)\lambda\mathbf{F}\Phi^{n+1} - \lambda_d\mathbf{c}^{n+1}] \\ \mathbf{c}^{n+1} - \mathbf{c}^n + \Delta t (\lambda_d\mathbf{c}^{n+1} - \beta\lambda\mathbf{F}\Phi^{n+1}) \\ \mathbf{Q} - \tilde{\mathbf{c}}\mathbf{E}\Phi\Delta x \\ \mathbf{T}^{n+1} - \mathbf{T}^n + \frac{\dot{m}\Delta t}{\mathcal{P}^{n+1}A\Delta x} (\mathbf{S}\mathbf{T}^{n+1} - \mathbf{R}\mathbf{Q}^{n+1}) \\ \mathcal{P} - \rho^{ref} - \frac{\partial \rho}{\partial T} (\mathbf{T} - T^{ref}) \\ \Sigma_a - \Sigma_a^{ref} - \frac{\partial \Sigma_a}{\partial \rho} [\mathcal{P} - \rho^{ref}] \\ \nu\Sigma_f - \nu\Sigma_f^{ref} - \frac{\partial \nu\Sigma_f}{\partial \rho} [\mathcal{P} - \rho^{ref}] \\ \mathbf{D} - D^{ref} - \frac{\partial D}{\partial \rho} [\mathcal{P} - \rho^{ref}], \\ \kappa\Sigma_f - \kappa\Sigma_f^{ref} - \frac{\partial \kappa\Sigma_f}{\partial \rho} [\mathcal{P} - \rho^{ref}] \end{bmatrix}$$

- Code time per loop
- Number of GMRES and Newton

Transient - Coupled Neutronics/Thermal Hydraulics [2/2]

Conclusions

- JFNK methods are well suited for solving coupled systems
- Finite difference method for Jacobian-vector product works well
- Write your own steam tables or fit data with polynomial
- Watch out for scaling issues when coupling physics

Future Work

- Rewrite code in a compiled language (Fortran)
- Use PetSc Nonlinear solvers
- Model other important nuclear feedback mechanisms
- Compare JFNK to Operator-splitting methods
- This project can be downloaded from GitHub:

`www.github.com/bhermanmit/JFNK`
`git clone git@github.com:bhermanmit/JFNK`