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

Bryan Herman

December 8, 2011

Massachusetts Institute of Technology

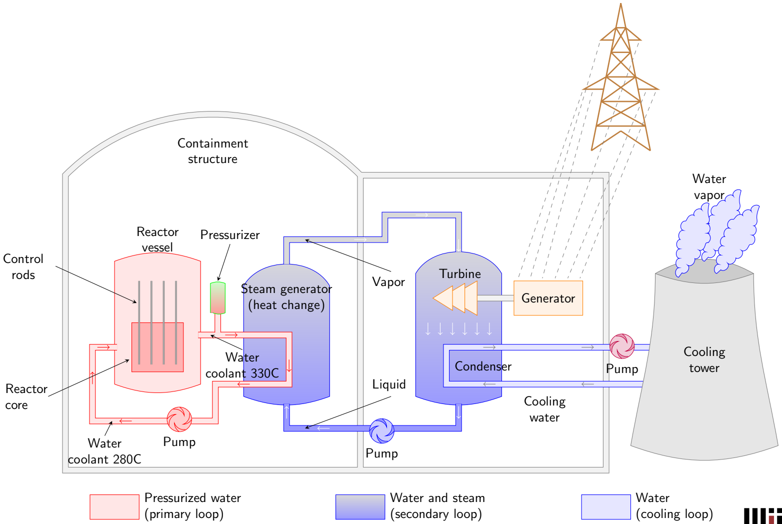
# Outline

# Motivation

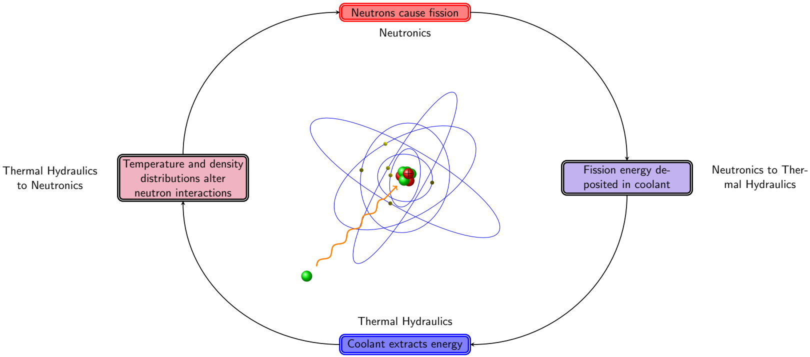
- Eventually will be part of thesis work
- JFNK method not currently used in nuclear reactor analysis
- Incorporates a lot of ideas from 2.29 class
- Coupled physics is fun!

Massachusetts Institute of Technology

# Nuclear Reactor Plant



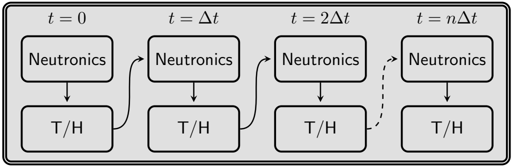
# Nuclear Feedback



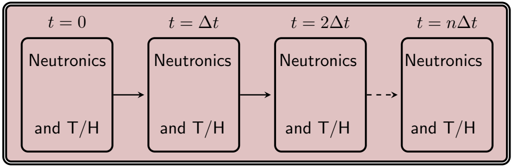

- Fuel Temperature Feedback - $T_f \uparrow$, U-238 Capture $\uparrow$, Fission Rate $\downarrow$, Power $\downarrow$
- Coolant Density Feedback - $\rho \downarrow$, $E_n \uparrow$, Fission Rate $\downarrow$, Power $\downarrow$

# Common Approach to Coupling: Operator Splitting

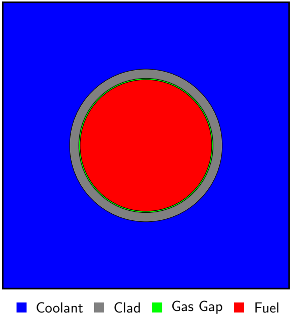
- Solve physics independently and iterate between them



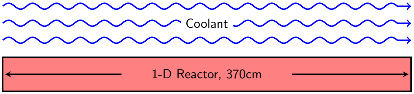

- Fully coupled approach solves the nonlinear physics together



# 1-D Slab Reactor Geometry




Top-view
Fuel Rod Unit-Cell




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

# Neutronics

- Basic Neutron Conservation:

$$Change + Leakage + Interactions = Scattering + Fission$$

- Neutron Transport Equation:

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

- Neutron Diffusion Equation (1-D Energy Integrated)

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

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

Massachusetts Institute of Technology

# 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

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

- Matrix-form of precursors

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

# Thermal Hydraulics

- Energy Equation - single phase fluid and inviscid fluid

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

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

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

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

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

Massachusetts Institute of Technology

# 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

$$\frac{\rho A \Delta x}{w} \frac{d\bar{T}_i}{dt} + \bar{T}_i - \bar{T}_{i-1} = \frac{1}{2wc_p} \left( Q_{i-1} + Q_i \right)$$

- Matrix-form with time discretization

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

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

Massachusetts Institute of Technology

# Physics Coupling

## Neutronics-Thermal Hydraulics

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

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

where $\mathbb{E} = \mathrm{diag}\left\{\kappa\Sigma_f\right\}$ 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
- $D$, $\Sigma_a$, $\nu\Sigma_f$, $\kappa\Sigma_f$ are all affected by coolant density variations
- Data is correlated with a linear regression of the form:

$$\mathbf{\Sigma} = \mathbf{\Sigma}^{ref} + \frac{\partial \Sigma}{\partial \rho}\left(\mathcal{P} - \rho^{ref}\right)$$

sachusetts
tute of
nology

## The Steady State Eigenvalue Problem

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

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

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

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

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

Massachusetts Institute of Technology

# Newton's Method

**Procedure:**

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

Three major tasks:

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

Massachusetts Institute of Technology

# 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

$$\mathcal{K}_n\left(\mathbb{A}, \mathbf{v}\right) = \operatorname{span}\left\{\mathbf{v}, \mathbb{A}\mathbf{v}, \mathbb{A}^2\mathbf{v}, \ldots, \mathbb{A}^{n-1}\mathbf{v}\right\}$$

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

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

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

Massachusetts Institute of Technology

# 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: $q_1 = b/\left\|b\right\|$
2: **for** $n = 1, 2, 3, \ldots$ **do**
3: Perform step $n$ of Arnoldi (Creates Hessenberg matrix)
4: Find $y$ to minimize $\left\|\widetilde{\mathbb{H}}_n\mathbf{y} - \left\|\mathbf{b}\right\|\mathbf{e}_1\right\|$
5: **if** $\left\|\mathbf{r}\right\| < ltol$ **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 $\widetilde{\mathbb{H}}_n$ from $\widetilde{\mathbb{H}}_{n-1}$ from Givens rotations

[‡] Youcef 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, $\eta$, is used

$$\left\| \mathbb{J}\left(\mathbf{x}^{n}\right) \mathbf{dx}_{m}^{n} + \mathbf{F}\left(\mathbf{x}^{n}\right) \right\| < \eta \left\| \mathbf{F}\left(\mathbf{x}^{n}\right) \right\|$$

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

Massachusetts Institute of Technology

# Jacobian-Free Approximation

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

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

- 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
- $\epsilon$ is the perturbation parameter and is somewhat arbitrary - Mousseau[§] recommends:

$$\epsilon = \frac{\sum_{i=1}^{N} b x_i}{N\,\|\mathbf{y}\|_2}$$

[§] 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:

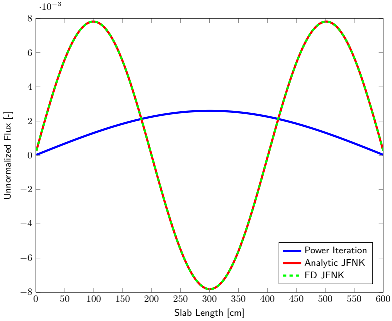
$$\mathbb{U}^{-1}\mathbb{L}^{-1}\mathbb{A}\mathbf{x} = \mathbb{U}^{-1}\mathbb{L}^{-1}\mathbf{b}$$

# Steady State - Neutronics

- Residual Equations

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

- 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

Massachusetts Institute of Technology

# Conclusions