SAND2007-4157C

Bifurcation analysis software for large-scale and parallel applications with LOCA, the Library of Continuation Algorithms

Andy Salinger, Eric Phipps

Sandia National Laboratories

Albuquerque, New Mexico, USA

Workshop: Advanced Algorithms and Numerical Software for the Bifurcation Analysis of Dynamical Systems,

Centre de Recherches Mathematiques

July 2, 2007

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

LOCA provides bifurcation analysis tools for large-scale simulation codes

LOCA has these bifurcation analysis capabilities:

- Parameter Continuation
- Bifurcation Tracking
- Constraint Enforcement
- *Space-Time Formulation*
- Linear Stability Analysis (*via* Anasazi code of Thornquist and Lehoucq)

LOCA is C++ and uses Abstract Numerical Algorithms wherever possible, and is designed to work with iterative linear solvers on parallel computers.

- LOCA is very flexible and extensible

LOCA is 1 of 30 packages in the Trilinos Parallel Algorithms framework developed at Sandia National Laboratories, and is interoperable with many of them (preconditioners, linear solvers, sparse matrix, parameter lists, ...)

LOCA does not have:

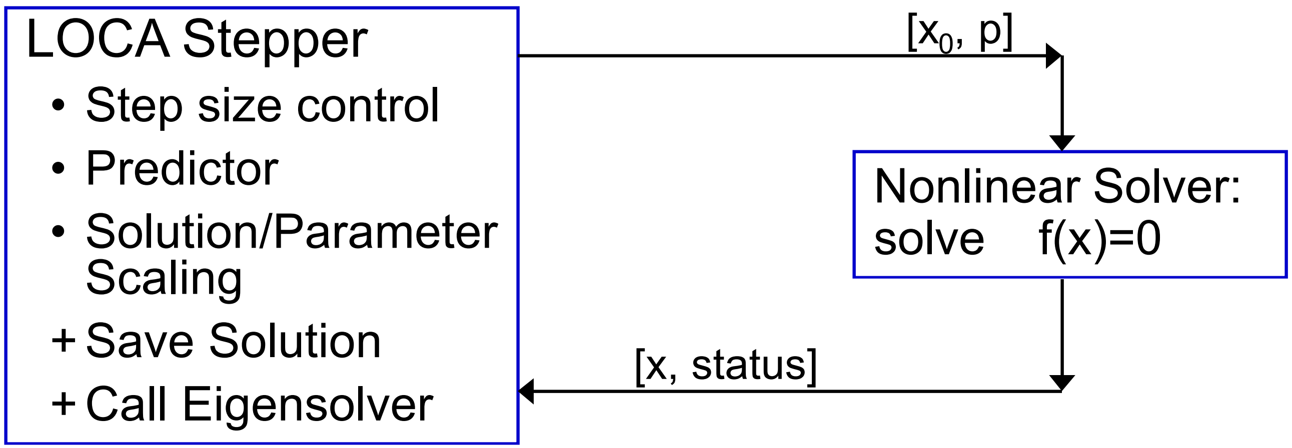
- MATLAB interface
- graphics capabilities
- an executable

http://trilinos.sandia.gov

Sandia National Laboratories

LOCA is a Stepper and a collection of Super-Groups (1 of 3)

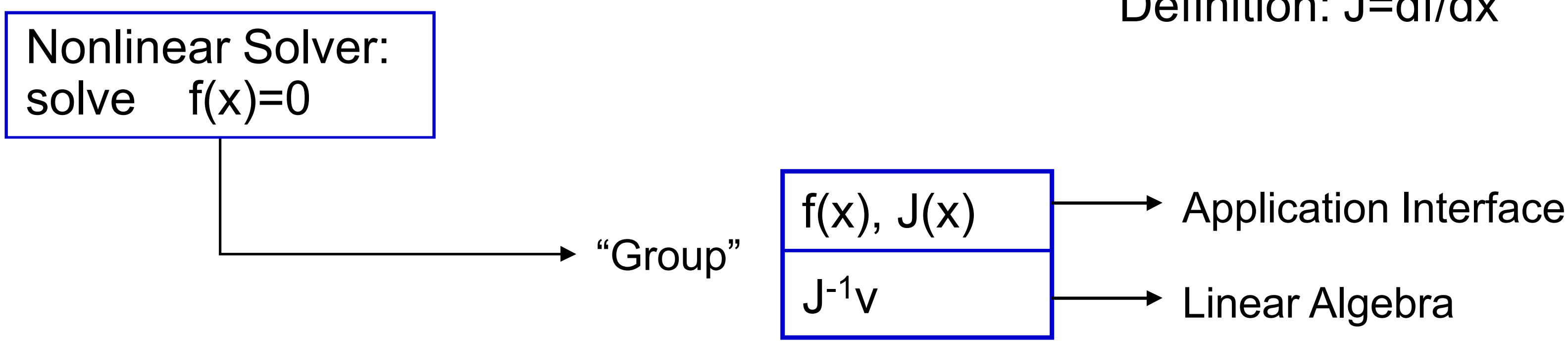
1. Qu'est-ce que c'est Stepper?



LOCA is a Stepper and a collection of Super-Groups (2 of 3)

2. Qu'est-ce que c'est "Group"?

Definition: J=df/dx




```
f = Group.computeF(x);
while (notConverged) {
  Group.computeJacobian(x);
  dx = Group.applyJacobianInverse(f);
  x.update(dx, -1.0);
  f = Group.computeF(x);
  notConverged = status(f, dx);
}
```

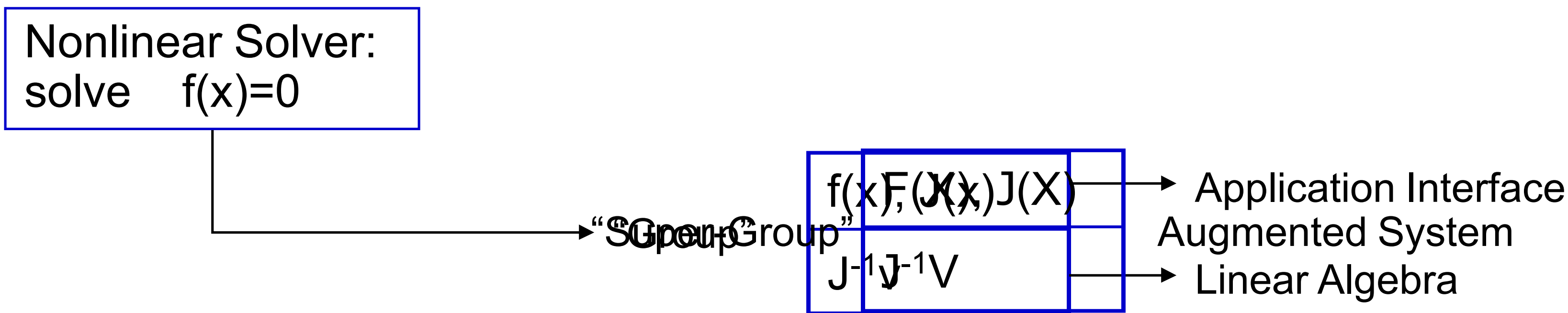
In C++ code: A "Solver" object is constructed with a "Group" object

Abstract Newton solve using a "Group". Note: Algorithm is independent of data structures and solution method.

Sandia National Laboratories

LOCA is a Stepper and a collection of <u>Super-Groups</u> (3 of 3)

3. Qu'est-ce que c'est "Super-Group"?



Ex: Arclength continuation with bordered solve:

$$X = \begin{bmatrix} x \\ p \end{bmatrix} \qquad F(X) = \begin{bmatrix} f(x,p) \\ N(x,p,\Delta s) \end{bmatrix} \qquad J = \begin{bmatrix} \mathrm{J} & f_p \\ N_x^T & N_p \end{bmatrix}$$

$$J^{-1}V = J^{-1}\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} a + \gamma b \\ \gamma \end{bmatrix}$$

where $\quad a = -\mathrm{J}^{-1}v, \quad b = -\mathrm{J}^{-1}f_p, \quad \gamma = -\frac{N_x^T a + w}{N_x^T b + N_p}.$

What Super-Groups are programmed in LOCA already?

Constraint Enforcement (**m** additional equations)

- Arclength Continuation
- Multiparameter Continuation (*Multifario* from Henderson@ibm)
- User-supplied Constraints

Bifurcation Tracking:

- Turning Point [Moore-Spence] (two versions)
- Turning Point [Govaerts' Minimally Augmented]
- Pitchfork Bifurcation
- Hopf Bifurcation [Griewank-Reddien]
- Hopf Bifurcation [Govaerts' Minimally Augmented]
- Phase Transitions

- Space-Time [trajectory or periodic orbit] (BE)

[N+m] x [N+m] solves:

- Augmented
- Bordered
- Householder

$$\begin{bmatrix} \mathrm{J} & \boldsymbol{f_p} \\ \boldsymbol{N_x^T} & \boldsymbol{N_p} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{bmatrix}$$

Sandia National Laboratories

So, what do you need to do to solve a bifurcation problem in LOCA

"Group"

- f(x), J(x) → Application Interface
- $J^{-1}v$ → Linear Algebra

Done for you for:
- LAPACK
- Epetra*

File #1: Interface Class

f(x), J(x), set(p)

File #2: main() or LOCA_Driver()

1. set parameters
   a) LinearAlgebraParams (ILU(2), GMRES(500))
   b) StepperParams
      i. Bifurcation method (pick "SuperGroup")
      ii. Step Size Control (step size, aggressiveness)
      iii. Predictor (secant)
      iv. Eigensolver (spectral transformation, max Iters)
2. construct Interface()
3. construct Group(Interface, LinearAlgebraParams)
4. construct Stepper(Group, StepperParams)
5. Stepper.run()

*Epetra – Distributed-memory sparse matrices and vectors, understood by many preconditioner and linear solver packages

LOCA Example:
1D PDE Chan problem with LAPACK Group

$$\frac{d^2T}{dx^2} + \alpha\left(1 + \frac{T + \frac{1}{2}T^2}{1 + \frac{1}{100}T^2}\right) = 0$$

$$0 \le x \le 1$$

$$T(0) = T(1) = \beta$$

$$\frac{d^2T}{dx^2} \approx \frac{T_{i-1} - 2T_i + T_{i+1}}{\Delta^2}$$

At this point, the presentation will skip to a “demonstration” piece, editing and running the code in Linux. Here is some of the code to be shown – it is Open Source downloadable from trilinos.sandia.gov.

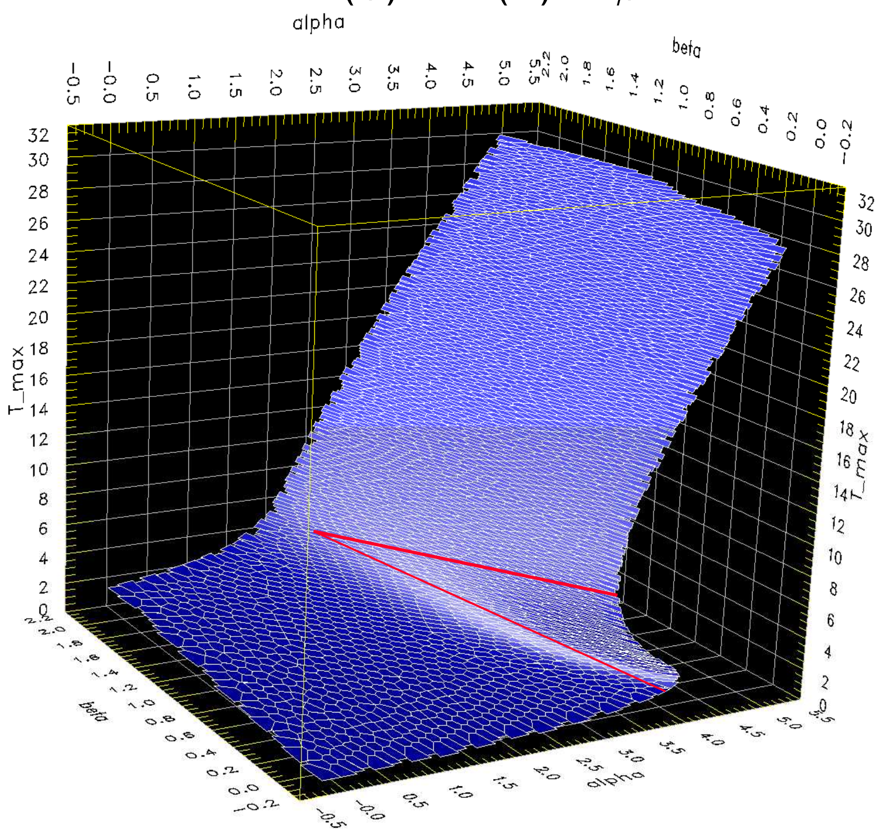
```
bool
ChanProblemInterface::computeF(NOX::LAPACK::Vector& f,
                               const NOX::LAPACK::Vector &x)
{
  f(0) = x(0) - beta;
  f(n-1) = x(n-1) - beta;
  for (int i=1; i<n-1; i++)
    f(i) = (x(i-1) - 2*x(i) + x(i+1))*(n-1)*(n-1)
      + alpha*source_term(x(i));

  return true;
}

bool
ChanProblemInterface::computeJacobian(NOX::LAPACK::Matrix<double>& J,
                                      const NOX::LAPACK::Vector & x)
{
  J(0,0) = 1.0;
  J(n-1,n-1) = 1.0;
  for (int i=1; i<n-1; i++) {
    J(i,i-1) = (n-1)*(n-1);
    J(i,i+1) = J(i,i-1);
    J(i,i) = -2.*J(i,i-1) + alpha*source_deriv(x(i));
  }
  return true;
}
```

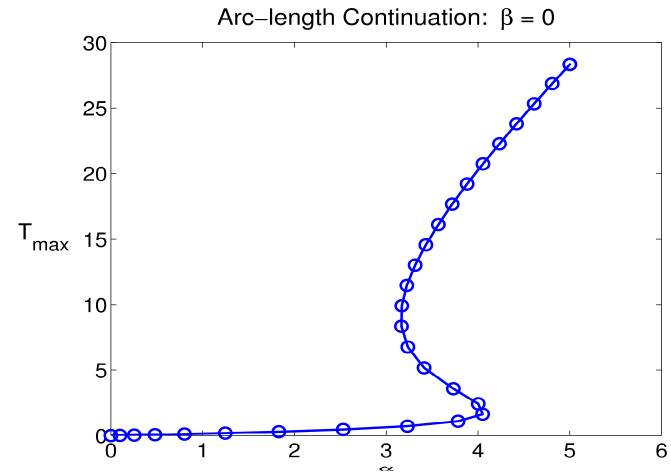
Multi-Parameter Continuation Example: Using *Multifario* code (Mike Henderson, IBM)

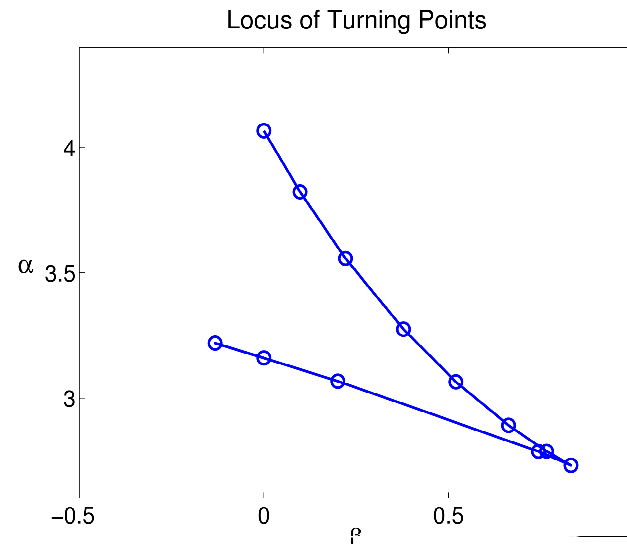
$$\frac{d^2T}{dx^2} + \alpha\left(1 + \frac{T + \frac{1}{2}T^2}{1 + \frac{1}{100}T^2}\right) = 0$$

$$0 \le x \le 1$$

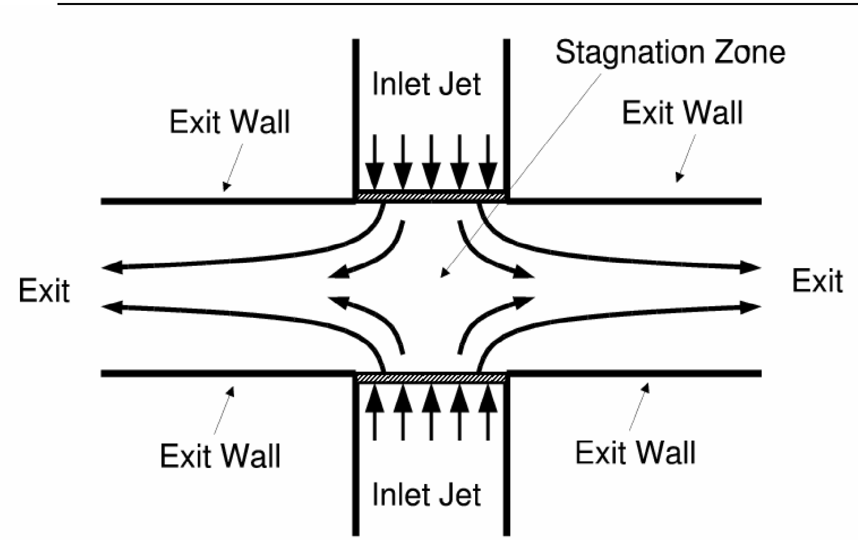
$$T(0) = T(1) = \beta$$



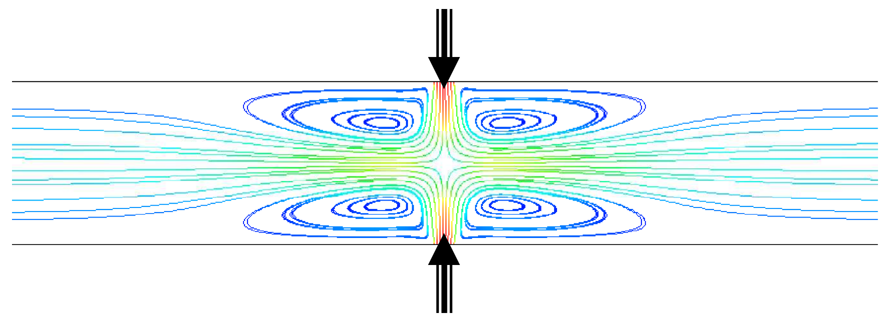


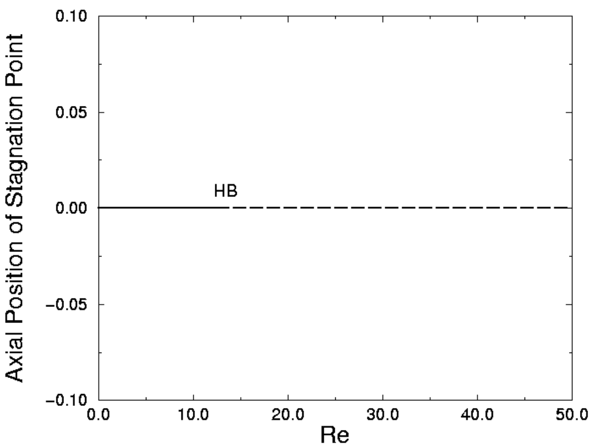


LOCA has been used to analyze 2D and 3D PDE discretizations of over a Million unknowns

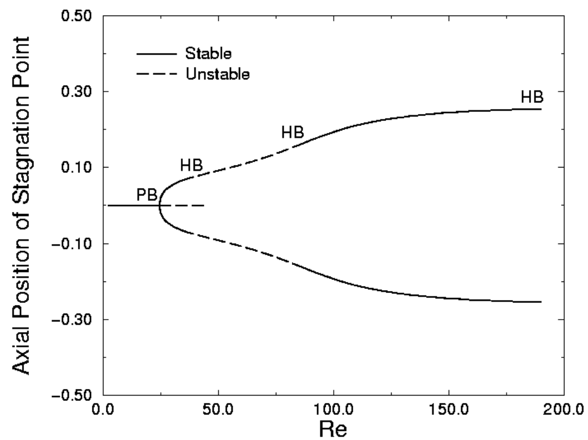


Stability Analysis of Impinging Jets, Pawlowski, Salinger, Shadid, Mountziaris, JFM (2005)

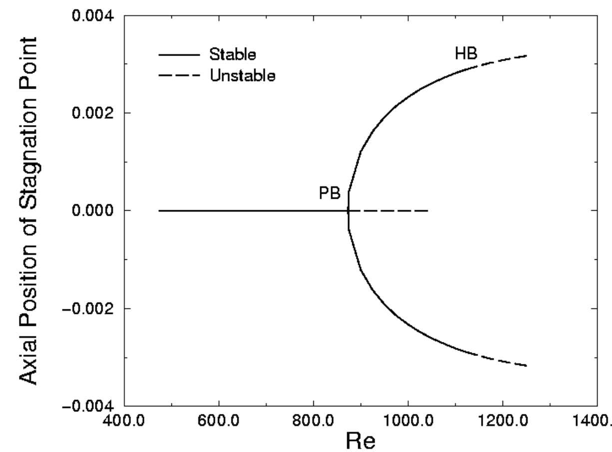




aspect ratio=0.05



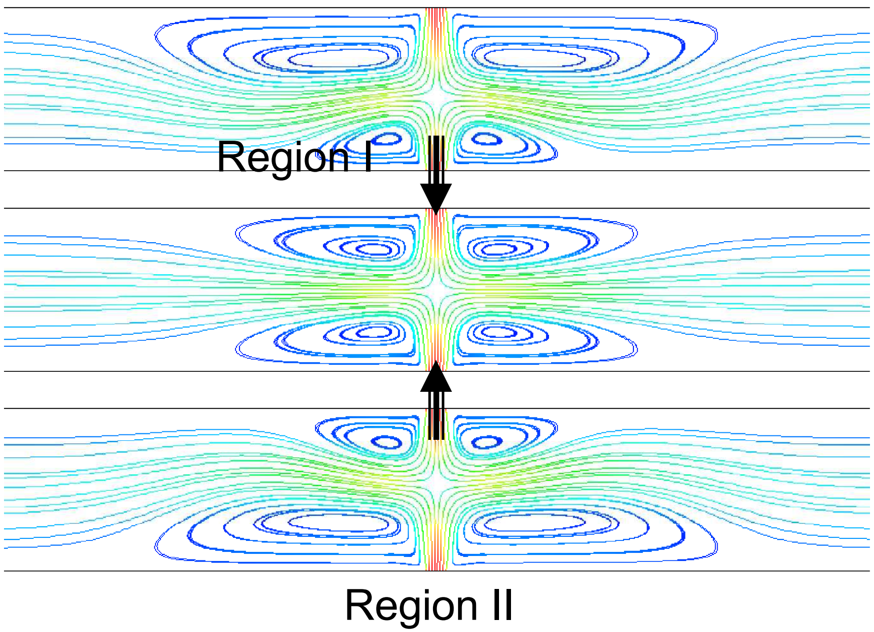
aspect ratio=0.125

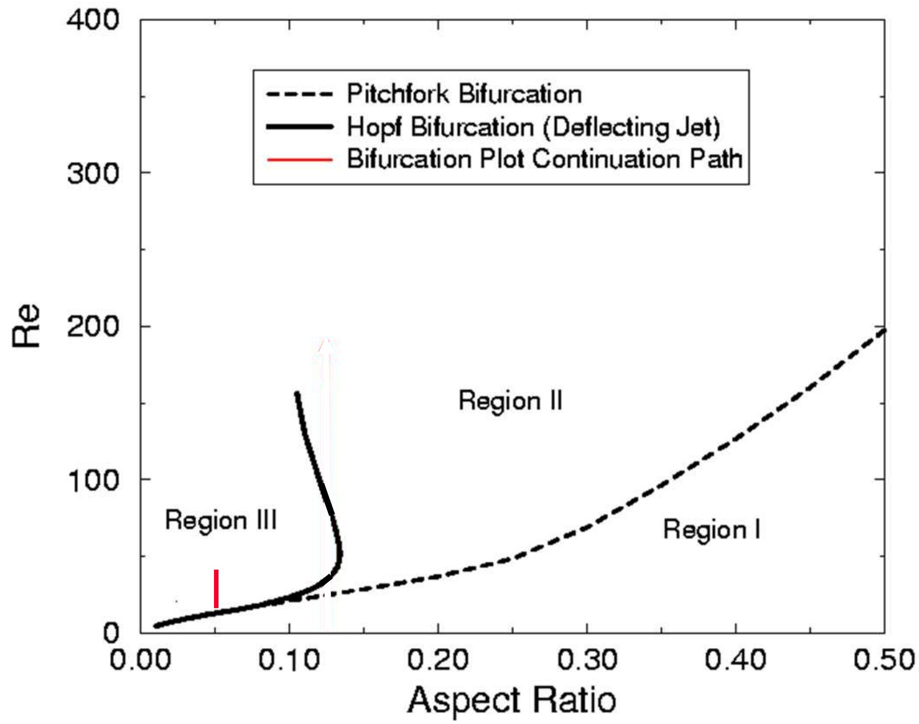


aspect ratio=1.0

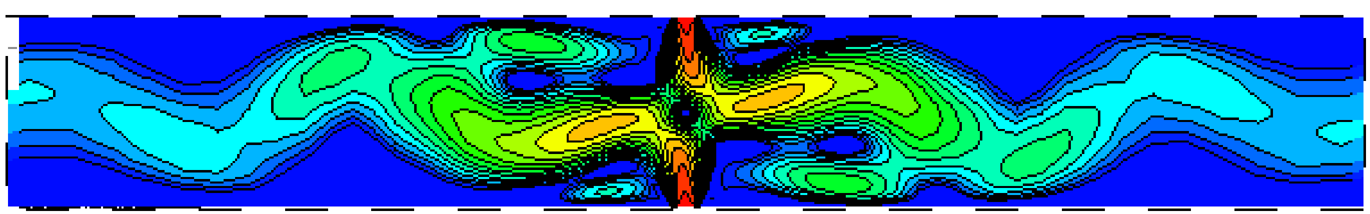
Sandia National Laboratories

Bifurcation tracking routines quickly delineate three flow regimes in 2-parameter space

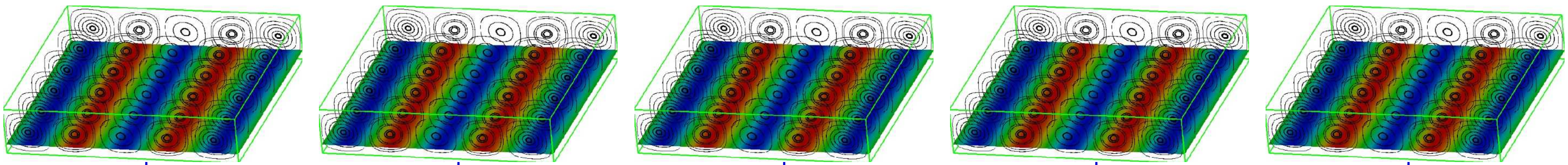




Region III



Current work "4D": General Purpose Space-Time Formulation and Solver with Space x Time Parallelism

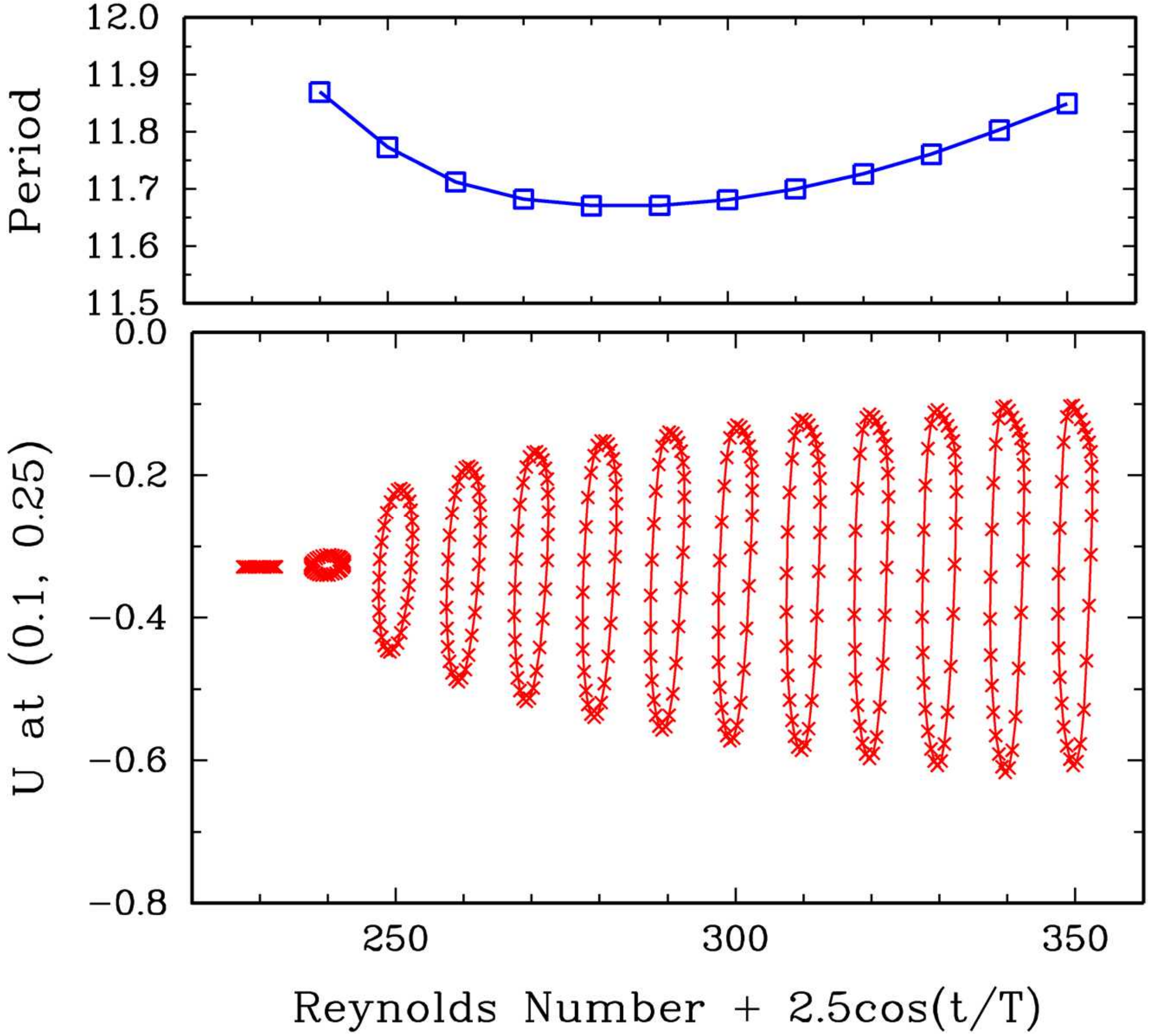


$\mathbf{J}(\mathbf{x}_1), \mathbf{B}(\mathbf{x}_1)$ $\quad \mathbf{J}(\mathbf{x}_2), \mathbf{B}(\mathbf{x}_2)$ $\quad \mathbf{J}(\mathbf{x}_3), \mathbf{B}(\mathbf{x}_3)$ $\quad \mathbf{J}(\mathbf{x}_4), \mathbf{B}(\mathbf{x}_4)$ $\quad \mathbf{J}(\mathbf{x}_5), \mathbf{B}(\mathbf{x}_5)$

$$\begin{vmatrix} (\mathbf{B} - \Delta t\mathbf{J}) & 0 & 0 & 0 & -\mathbf{B} \\ -\mathbf{B} & (\mathbf{B} - \Delta t\mathbf{J}) & 0 & 0 & 0 \\ 0 & -\mathbf{B} & (\mathbf{B} - \Delta t\mathbf{J}) & 0 & 0 \\ 0 & 0 & -\mathbf{B} & (\mathbf{B} - \Delta t\mathbf{J}) & 0 \\ 0 & 0 & 0 & -\mathbf{B} & (\mathbf{B} - \Delta t\mathbf{J}) \end{vmatrix} \begin{vmatrix} \Delta\mathbf{x}_1 \\ \Delta\mathbf{x}_2 \\ \Delta\mathbf{x}_3 \\ \Delta\mathbf{x}_4 \\ \Delta\mathbf{x}_5 \end{vmatrix} = \begin{vmatrix} -\mathbf{g}_1 \\ -\mathbf{g}_2 \\ -\mathbf{g}_3 \\ -\mathbf{g}_4 \\ -\mathbf{g}_5 \end{vmatrix}$$

- Continuation of Trajectories, with end constraints (BVP in time)
- Periodic Orbit Tracking

Periodic Orbit Tracking for Impinging Jet Reactor



Phase condition:
"Newton update is orthogonal to the flow":

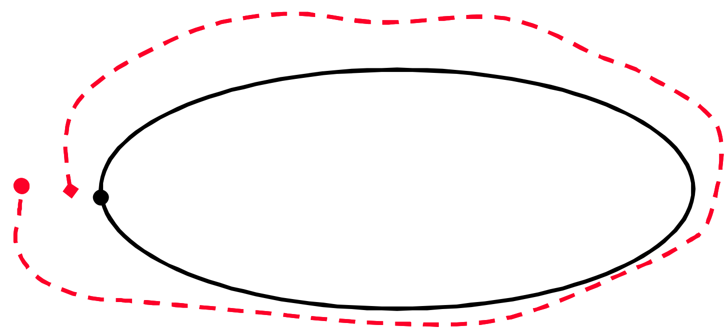
$$M\dot{x} \cdot \Delta x = 0$$

Details:
7K nodes
21K spatial unknowns
30 time steps/period
620K total unknowns
30 Processors
2-4 min / Newton iter

Sandia National Laboratories

Floquet Stability Analysis of Periodic Orbits

Floquet Multipliers:
- Linearize around a periodic solution
- Integrate perturbations through 1 period
- Eigenvalues of this operator are called Floquet Multipliers: σ_i
- Orbits are stable if, forall i: $\|\sigma_i\| < 1$

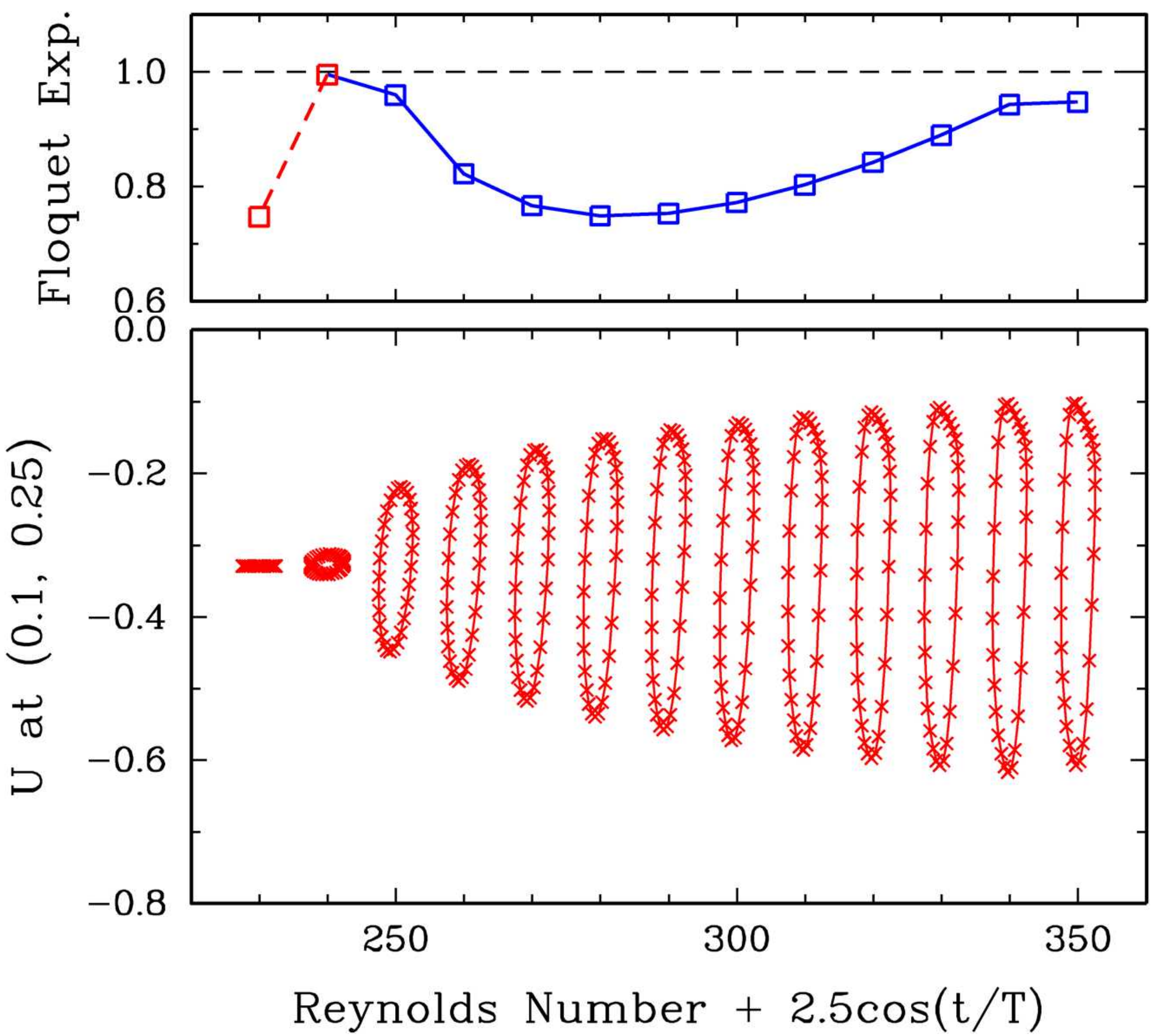


Krylov iteration for Anasazi eigensolver (Thornquist, Baker, Lehoucq):

$$\begin{bmatrix} (\mathbf{B}-\Delta t\mathbf{J}) & 0 & 0 & 0 & 0 \\ -\mathbf{B} & (\mathbf{B}-\Delta t\mathbf{J}) & 0 & 0 & 0 \\ 0 & -\mathbf{B} & (\mathbf{B}-\Delta t\mathbf{J}) & 0 & 0 \\ 0 & 0 & -\mathbf{B} & (\mathbf{B}-\Delta t\mathbf{J}) & 0 \\ 0 & 0 & 0 & -\mathbf{B} & (\mathbf{B}-\Delta t\mathbf{J}) \end{bmatrix} \begin{bmatrix} \dots \\ \dots \\ \dots \\ \dots \\ \mathbf{v}_{i+1} \end{bmatrix} = \begin{bmatrix} \mathbf{B}\mathbf{v}_i \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Sandia National Laboratories

Periodic Orbit Tracking for Impinging Jet Reactor – Floquet Rresults



Details:
- •10-15 Arnoldi iterations to converge leading 2 eigenvalues
- •~1 min / iteration