Pseudo-Transient Continuation

C. T. Kelley
NC State University
tim_kelley@ncsu.edu
Supported by ARO, NSF, DOEd, DOE

Purdue, October 10, 2013
Outline

Collaborators

Nonlinear Equations and Newton’s Method
  Integration to Steady State
  Implementation

Pseudo-Transient Continuation (Ψtc )
  CFD Application
  Nonlinear Reaction-Diffusion

Constrained Ψtc (if I talk fast)
  Inverse Singular Value Problem

Conclusions
Moral of Talk

You can see a lot just by listening.

Y. Berra
Collaborators

- Todd Coffey, Katie Fowler, Jill Reese, Corey Winton, Moody Chu
- (CFD) Scott McRae, Jeff McMullan, Paul Orkwis
- (Theory) David Keyes, Liqun Qi, Li-Zhi Liao, X-L Luo, H-W Tam, Moody Chu
- (Hydrology) Casey Miller, Chris Kees, Matthew Farthing
- (Mechanics) Rich Lehoucq, Michael Gee
Objective: Integrate to Steady State

Given an initial value problem

\[ u_t = -F(u), \quad u(0) = u_0 \]

find \( u^* = \lim_{t \to \inf} u(t) \).

Assume \( u^* \) exists, then the obvious thing to do is

Solve \( F(u) = 0 \)
Newton’s method

Problem: solve $F(u) = 0$

$F : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is Lipschitz continuously differentiable.

Newton’s method

$$u_+ = u_c + s.$$ 

Compute the step $s$ by solving the linearized problem

$$F'(u_c)s = -F(u_c)$$

$F'(u_c)$ is the Jacobian matrix

$$F'_{ij} = \frac{\partial f_i}{\partial x_j}$$
Implementation

Inexact formulation:

\[ \| F'(u_c)s + F(u_c) \| \leq \eta_c \| F(u_c) \|. \]

\( \eta = 0 \) for direct solvers + analytic Jacobians.

\( \eta \) hides

- iterative linear solvers
- approximations of \( F' \) like finite differences, different physics, low-order schemes, . . .
Convergence for smooth $F$

If $F(u^*) = 0$, $F'(u^*)$ is nonsingular, and $u_c$ is close to $u^*$

$$
\|u_+ - u^*\| = O(\eta_c \|u_c - u^*\| + \|u_c - u^*\|^2)
$$

For less smooth $F$ ...
But what if $u_0$ is far from $u^*$?

Armijo Rule: Find the least integer $m \geq 0$ such that

$$\|F(u_c + 2^{-m}s)\| \leq (1 - \alpha 2^{-m}) \|F(u_c)\|$$

- $m = 0$ is Newton’s method.
- Make it fancy by replacing $2^{-m}$.
- $\alpha = 10^{-4}$ is standard.
Theory

If $F$ is smooth and you get $s$ with a direct solve or GMRES then either

- **BAD**: the iteration is unbounded, i.e. $\limsup \|u_n\| = \infty$,
- **BAD**: the derivatives tend to singularity, i.e. 
  $\limsup \|F'(u_n)^{-1}\| = \infty$, or
- **GOOD**: the iteration converges to a solution $u^*$ in the terminal phase, $m = 0$, and

$$\|u_{n+1} - u^*\| = O(\eta_n \|u_n - u^*\| + \|u_n - u^*\|^2).$$

Bottom line: you get an answer or an easy-to-detect failure.
Theory

If $F$ is smooth and you get $s$ with a direct solve or GMRES then either

- **BAD**: the iteration is unbounded, i.e. $\limsup \|u_n\| = \infty$,
- **BAD**: the derivatives tend to singularity, i.e. $\limsup \|F'(u_n)^{-1}\| = \infty$, or
- **GOOD**: the iteration converges to a solution $u^*$ in the terminal phase, $m = 0$, and

$$\|u_{n+1} - u^*\| = O(\eta_n \|u_n - u^*\| + \|u_n - u^*\|^2).$$

Bottom line: you get an answer or an easy-to-detect failure. *Newton's method works great except when it doesn't.*
What’s wrong with Newton?

- Stagnation at singularity of $F'$ really happens.
  - steady flow $\rightarrow$ shocks in CFD
- Non-physical results
  - fires go out
  - negative concentrations
- Nonsmooth nonlinearities
  - are not uncommon: flux limiters, constitutive laws
  - globalization is harder
  - finite diff directional derivatives may be wrong

Ψtc is one way to fix some of these things.
Steady-state Solutions

Enforce dynamics by solving

$$\frac{du}{dt} = -F(u), \, u(0) = u_0,$$

to obtain $u(t)$. $F(u)$ contains

- the nonlinearity,
- boundary conditions, and
- spatial derivatives.

Define the right answer as the steady-state solution:

$$u^* = \lim_{t \to \infty} u(t).$$
What can go wrong?

If \( u_0 \) is separated from \( u^* \) by

- complex features like shocks,
- stiff transient behavior, or
- unstable equilibria,

then the Newton-Armijo iteration can

- **stagnate** at a singular Jacobian, or
- find a solution of \( F(u) = 0 \) that is **not the one you want**.
A Questionable Idea

One way to guarantee that you get $u^*$ is

- Find a high-quality temporal integration code.
- Set the error tolerances to very small values.
- Integrate the PDE to steady state.
  - Continue in time until $u(t)$ isn’t changing much.
- Then apply Newton to make sure you have it right.

Good news: Even fixes problems for some non-smooth $F$.

Problem: you may not live to see the results.
Integrate

$$\frac{du}{dt} = -F(u)$$

to steady state in a stable way with increasing time steps. Equation for $\Psi_{tc}$ Newton step:

$$\left(\delta^{-1}_c I + F'(u_c)\right) s = -F(u_c),$$

or

$$\| (\delta^{-1}_c I + F'(u_c)) s + F(u_c) \| \leq \eta_c \| F(u_c) \|.$$
ψtc as an Integrator

- Low accuracy PECE integration
  - Trivial predictor
  - Backward Euler corrector + one Newton iteration
  - 1st order Rosenbrock method
    High order possible, Luo, K, Liao, Tam 06
- Begin with small “time step” δ. Resolve transients.
- Grow the “time step” near $u^*$. Turn into Newton.
Grow the time step with switched evolution relaxation (SER)

\[ \delta_n = \min(\delta_0 \| F(u_0) \| / \| F(u_n) \|, \delta_{\text{max}}). \]

If \( \delta_{\text{max}} = \infty \) then \( \delta_n = \delta_{n-1} \| F(u_{n-1}) \| / \| F(u_n) \| \).

Alternative with no theory (SER-B):

\[ \delta_n = \delta_{n-1} / \| u_n - u_{n-1} \| \]
Temporal Truncation Error (TTE)

Estimate local truncation error by

\[ \tau = \frac{\delta_n^2 (u)''(t_n)}{2} \]

and approximate \((u)''\) by

\[ \frac{2}{\delta_{n-1} + \delta_{n-2}} \left[ \frac{(u_i)_n - (u_i)_{n-1}}{\delta_{n-1}} \frac{((u_i)_n - (u_i)_{n-1}) - ((u_i)_{n-1} - (u_i)_{n-2})}{\delta_{n-2}} \right] \]

Adjust step so that \(\tau = .75\).
PTC Convergence: SER

- If $F$ is smooth enough (LIP),
- $u^* = \lim_{t \to \infty} u(t)$ exists,
- $u^*$ is dynamically stable, and
- $\delta_0$ sufficiently small

then $u_n \to u^*$ and you get the local convergence rates for Newton you deserve.
Three phase iteration:

- Small $\delta$, inaccurate $u$; it’s Euler’s method (elementary)
- Small $\delta$, good $u$; grow $\delta$ and make $u$ no worse (hard)
- Big $\delta$, good $u$; it’s Newton (no surprise)
Euler Equations: Unknowns density, velocity, energy.

\[ \nabla \cdot (\rho \mathbf{v}) = 0 \]
\[ \nabla \cdot (\rho \mathbf{v} \mathbf{v} + p I) = 0 \]
\[ \nabla \cdot ((\rho e + p) \mathbf{v}) = 0 \]

Ideal gas law \( p = \rho (\gamma - 1)(e - |\mathbf{v}|^2/2) \), where \( \gamma \) is the ratio of specific heats.
But $F$ is not smooth!

Typical Euler equation approach

- Discretize with 2nd order scheme with slope limiter.
  Slope limiters can be nonsmooth, but Lipschitz continuous.
- Use Jacobian of a (smooth) 1st order scheme.

Modified method: $u_+ = u_c + s$ where

$$\| (\delta_c^{-1} I + J_c) s + F(u_c) \| \leq \eta_c \| F(u_c) \|,$$

and $J_c$ is the Jacobian of the smooth, low-order discretization.
Example: Flow through a nozzle
Stagnation with Newton-Armijo
Success with $\Psi_{tc}$
Nonlinear Reaction-Diffusion: Fowler-K, 2005

\[-u_{zz} + \lambda \max(0, u)^p = 0\]

\[z \in (0, 1), \ u(0) = u(1) = 0,\]

where \(p \in (0, 1)\).

Reformulate as a DAE to make the nonlinearity Lipschitz. Let

\[v = \begin{cases} u^p & \text{if } u \geq 0 \\ u & \text{if } u < 0 \end{cases}\]
Set \( x = (u, v)^T \) and solve

\[
F(x) = \begin{pmatrix}
  f(u, v) \\
  g(u, v)
\end{pmatrix} = \begin{pmatrix}
  -u_{zz} + \lambda \max(0, v) \\
  u - \omega(v)
\end{pmatrix} = 0,
\]

The nonlinearity is

\[
\omega(v) = \begin{cases} 
  v^{1/p} & \text{if } v \geq 0 \\
  v & \text{if } v < 0
\end{cases}
\]
DAE Dynamics

Semi-explicit index-one differential-algebraic equation (DAE)

\[ D \begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} u' \\ 0 \end{pmatrix} = - \begin{pmatrix} f(u, v) \\ g(u, v) \end{pmatrix} = -F(x), \quad x(0) = x_0, \]
Why not ODE dynamics?

Original time-dependent problem is

\[ u_t = u_{zz} - \lambda \max(0, u)^p. \]

Applying \( \Psi_{tc} \) to

\[ v_t = u - \omega(v) \]

rather than using \( u - \omega(v) = 0 \) as an algebraic constraint

▶ adds non-physical time dependence,
▶ changes the problem, and
▶ doesn’t work.
Parameters

- $p = .1$ and $\lambda = 200$. Leads to "dead core".
- $\delta_0 = 1.0$, $\delta_{max} = 10^6$.
- Spatial mesh size $\delta_z = 1/2048$; discrete Laplacian $L_{\delta_z}$
- Terminate nonlinear iteration when either

$$\|F(x_n)\|/\|F(x_0)\| < 10^{-13} \text{ or } \|s_n\| < 10^{-10}.$$  

Step is an accurate estimate of error (semismoothness).
Solution
Analytic $\partial F$

\[
F(x) = \left( \begin{array}{c} f(u, v) \\ g(u, v) \end{array} \right)
\]

\[
= \left( \begin{array}{c} -L\delta_z u \\ u - v - \max(0, v^{1/p}) \end{array} \right) + \left( \begin{array}{c} \lambda \\ 1 \end{array} \right) \max(0, v).
\]

Since

\[
\partial \max(0, v) = \begin{cases} 
0, & \text{if } v < 0 \\
[0, 1], & \text{if } v = 0 \\
1, & \text{if } v > 0,
\end{cases}
\]

we get ...
\[ \partial F = \begin{pmatrix} -L \delta_z & 0 \\ 1 & -1 - \frac{1}{p} \max(0, v^{(1-p)/p}) \end{pmatrix} + \begin{pmatrix} 0 & \lambda \\ 0 & 1 \end{pmatrix} \partial \max(0, v). \]
Convergence
\[
\frac{du}{dt} = -F(u), \quad u(0) = u_0 \in \Omega.
\]

\(u(t) \in \Omega, \ F(u) \in T(u)\) (tangent to \(\Omega\)).

Examples:

- \(\Omega\) has interior: bound constrained optimization
- \(\Omega\) smooth manifold: inverse eigen/singular value problems

Problem: \(\Psi_{tc}\) will drift away from \(\Omega\).
Projected $\Psi_{tc}$

$$u_+ = P(u_c - (\delta_c^{-1} I + H(u_c))^{-1} F(u_c))$$

where

- $P$ is map-to-nearest $R^N \rightarrow \Omega$
  \[ \|P'(u)\| = 1 \text{ for } u \in \Omega. \]
- $H(u_c)$ makes Newton-like method fast.
General Method for Constraints

$F$ Lipschitz (no smoothness assumptions)

$$u_+ = \mathcal{P}(u_c - (\delta^{-1} I + H(u_c))^{-1} F(u_c)),$$

where $H$ is an approximate Jacobian.

Theory: $H$ bounded, other assumptions imply $u_n \rightarrow u^*$ and

$$u_{n+1} = u_{n+1}^N + O(\delta_n^{-1} + \eta_n)\|u_n - u^*\|$$

where

$$u_{n+1}^N = u_n - H(u_n)^{-1} F(u_n)$$

which is as fast as the underlying method.
What are those other assumptions?

- \( u(t) \rightarrow u^* \)
- \( \delta_0 \) is sufficiently small.
- \( \|P'(u)\| = 1 \) or Lip const of \( P = 1 \)
- \( u^* \) is dynamically stable
- \( H(u) \) is uniformly well-conditioned near \( \{u(t) | t \geq 0\} \)
- \( u_+ = u_c - H(u_c)^{-1}F(u_c) \) is rapidly locally convergent near \( u^* \)
Chu, 92 . . .

Find \( c \in \mathbb{R}^N \) so that the \( M \times N \) matrix

\[
B(c) = B_0 + \sum_{k=1}^{N} c_k B_k
\]

has prescribed singular values \( \{\sigma_i\}_{i=1}^{N} \).

Data: Frobenius orthogonal \( \{B_i\}_{i=0}^{N}, \{\sigma_i\}_{i=1}^{N} \).
Formulation

Least squares problem

\[
\min F(U, V) \equiv \|R(U, V)\|_F^2
\]

where

\[
R(U, V) = U\Sigma V^T - B_0 - \sum_{k=1}^{N} < U\Sigma V^T, B_k >_F B_k
\]

**Manifold constraints:**  \( U \) is orthogonal \( M \times M \) and \( V \) is orthogonal \( N \times N \)
Dynamic Formulation

\[ \Omega = \left\{ \begin{pmatrix} U \\ V \end{pmatrix} \in R^{M \times M} \oplus R^{N \times N} \mid U \text{ and } V \text{ orthogonal} \right\} \]

Projected gradient:

\[ g(U, V) = \frac{1}{2} \left( (R(U, V)V\Sigma^T U^T - U\Sigma V^T R(U, V)^T)U \\ (R(U, V)^T U\Sigma V^T - V\Sigma^T U^T R(U, V))^T V \right). \]

ODE:

\[ \dot{u} = \begin{pmatrix} \dot{U} \\ \dot{V} \end{pmatrix} = -F(u) \equiv -g(U, V). \]
Projection onto $\Omega$

Higham 86, 04
Projection of square matrix onto orthogonal matrices
\[ A \rightarrow U_P. \]
where $A = U_P H_P$ is the polar decomposition.
Compute $U_P$ via the SVD $A = U \Sigma V^T$
\[ U_P = UV^T. \]
Projection of
\[ w = \begin{pmatrix} A \\ B \end{pmatrix} \]
onto $\Omega$ is
\[ P(w) = \begin{pmatrix} U_P^A \\ U_P^B \end{pmatrix}. \]
The local method

Given $u \in \Omega$ let $P_T(u) = \mathcal{P}'(u)$ be the projection onto the tangent space to $\Omega$ at $u$. Let

$$H = (I - P_T(u)) + P_T(u)F'(u)P_T(u)$$

Locally (very locally) superlinearly convergent if $\Omega$ is OK near $u^*$. 

Inverse Singular Value Problem
Conclusions

- $\Psi_{tc}$ computes steady-state solutions.
  - Can succeed when traditional methods fail.
  - It is not a general nonlinear solver!
- Works on some manifolds.
- Theory and practice for many problems
  - ODEs, DAEs
  - Nonsmooth $F$
  - Inverse eigen/singular value problems.
- Explicit methods for gradient flows (Liao+K)
It’s over

*It ain’t over ’till it’s over.*

Y. Berra