Newton's Method and Efficient, Robust Variants

Philipp Birken

University of Kassel (SFB/TRR 30)
Soon: University of Lund

October 7th 2013
Efficient solution of large systems of non-linear PDEs in science
Lyon

Outline

1. Introduction

2. Newton's method

3. JFNK

4. Newton and Multigrid

5. The fine print

Outline

1. Introduction
2. Newton's method
3. JFNK
4. Newton and Multigrid
5. The fine print

Flow-Chart for DIRK scheme with Newtonx

Given error tolerance TOL, initial time t_0 and time step size Δt_0

- For $i = 1, ..., s$
  - For $k = 0, 1, ...$ until termination criterion with tolerance $TOL/5$ is satisfied or MAX_NEWTON_ITER has been reached
    - Solve linear system up to certain tolerance
- If MAX_NEWTON_ITER has been reached, but the tolerance test has not been passed, repeat time step with $\Delta t_n = \Delta t_n/4$
- Estimate local error and compute new time step size Δt_{n+1}
- $t_{n+1} = t_n + \Delta t_n$

Outline

1. Introduction

2. Newton's method

3. JFNK

4. Newton and Multigrid

5. The fine print

Illustration of Newton's method

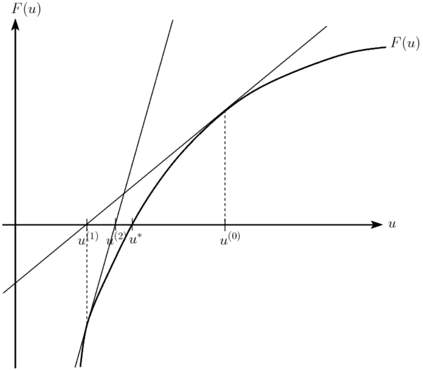

Figure: Illustration of Newton's method in one dimension (left); convergence curve (right)

Newton's method

Assume we solve $\mathbf{F}(\mathbf{x}) = \mathbf{0}$. Given starting value $\mathbf{x}^0$, Newton's method is given by:

$$\left.\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right|_{\mathbf{x}^{(k)}} \Delta \mathbf{x} = -\mathbf{F}(\mathbf{x}^{(k)}),$$
$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \Delta \mathbf{x}, \quad k = 0, \dots.$$

- Uses first order approximation of function
- Local quadratic convergence
- Outside of local ball of convergence no statement possiblex
- Nonlinear system transformed into sequence of linear systems

Standard Assumptions

- Always need Differentiability (at least piecewise)
- For easy convergence theory
  i) $\mathbf{F}(\mathbf{x}) = 0$ has a solution $\mathbf{x}^*$.
  ii) $\mathbf{F}' : \Omega \rightarrow \mathbb{R}^{m \times m}$ is Lipschitz continuous with Lipschitz constant L'.
  iii) $\mathbf{F}'(\mathbf{x}^*)$ is nonsingular.
- Lower requirements technically possible

The linear system

- System matrix: $\mathbf{A} = \left[\frac{\mathbf{I}}{\Delta t} - \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\right]\Big|_{\mathbf{x}^{(k)}}$.
- This matrix is sparse, nonnormal, not diagonally dominant and ill conditioned for reasonable Δt.
- Generally unstructured, but symmetric block-sparsity pattern.
- In FVM/low order FEM context: block sizes small even in 3D.
- For DG/high order FEM: Block sizes depend on degree N and dimension d:
  - $(d+2) \cdot (N+d)!/(N!d!)$
- This makes the design of efficient implicit DG schemes in 3D so difficult, since blocks have couple hundred unknowns

Problems of exact Newton method

Problem 1: The linear systems are solved exactly

- This implies either a direct solver or an iterative one with extremely small tolerance
- But we want to solve nonlinear systems only approximately
- This is way too costly
- But what happens when we do not solve exactly?

Problem 2: The Jacobian has to be assembled

- Typically very costly to do
- Huge amount of storage for 3D problems
- Need way around this

Inexact Newton methods

Let's solve problem 1!

$$\begin{aligned} \left\| \frac{\partial \mathbf{F}(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}^{(k)}}\mathbf{d} + \mathbf{F}(\mathbf{x}^{(k)}) \right\| &\leq \eta_k \|\mathbf{F}(\mathbf{x}_k)\| \\ \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \mathbf{d}, \quad k = 0, 1, ... \end{aligned} \tag{1}$$

Theorem

Let the standard assumptions hold. Then there is δ such that if $\mathbf{x}^{(0)}$ is in a δ-neighborhood of $\mathbf{x}^*$, $\{\eta_k\} \subset [0, \eta]$ with $\eta < \bar{\eta} < 1$, then the inexact Newton iteration (1) converges linearly. Moreover,

- *if $\eta_k \to 0$, the convergence is superlinear and*
- *if $\eta_k \leq K_\eta \|\mathbf{F}(\mathbf{x}^{(k)})\|^p$ for some $K_\eta > 0$ and $p \in [0, 1]$, the convergence is superlinear with order $1 + p$.*

Eisenstat-Walker '96

$$\eta_k^A = \gamma \frac{\|\mathbf{F}(\mathbf{x}^{(k)})\|^2}{\|\mathbf{F}(\mathbf{x}^{(k-1)})\|^2}, \quad \gamma \in (0,1]$$

To obtain quadratic convergence, bound sequence away from zero:

$$\eta_k^B = \min(\eta_{max}, \eta_k^A)$$

Definition of η_k is refined to

$$\eta_k^C = \begin{cases} \eta_{max}, & n = 0, \\ \min(\eta_{max}, \eta_k^A), & n > 0, \gamma\eta_{k-1}^2 \leq 0.1 \\ \min(\eta_{max}, \max(\eta_k^A, \gamma\eta_{k-1}^2)) & n > 0, \gamma\eta_{k-1}^2 > 0.1 \end{cases}$$

to avoid volatile decreases in η_k. To avoid oversolving

$$\eta_k = \min(\eta_{max}, \max(\eta_k^C, 0.5\tau / \|\mathbf{F}(\mathbf{x}^{(k)})\|)). \tag{2}$$

Methods of Newton type

Now, let's solve problem 2!

- Freeze the Jacobian and periodically recompute periodically
- Thus we use approximation

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mathbf{A}(\mathbf{x}^{(k)})^{-1}\mathbf{F}(\mathbf{x}^{(k)})$$

- Meaning we solve a different linear equation system
- Only linear convergence, provided that $\rho(\mathbf{I} - \mathbf{A}(\mathbf{x})^{-1}\frac{\partial \mathbf{F}}{\partial \mathbf{x}})$ is small enough
- This is not a good solution for problem 2!

Outline

1. Introduction

2. Newton's method

3. JFNK

4. Newton and Multigrid

5. The fine print

Jacobian-free Newton-Krylov method

Let's give problem 2 another try!

- Solve iterative systems using Krylov subspace method (e.g. GMRES)
- These do not need Jacobian, only matrix vector products.
- Approximate $\frac{\partial \mathbf{F}}{\partial \mathbf{x}} q$ by finite differences:

$$\mathbf{A}\mathbf{q} = \frac{\partial \mathbf{F}(\mathbf{x}^{(k)})}{\partial \mathbf{x}} \mathbf{q} \approx \frac{\mathbf{F}(\mathbf{x}^{(k)} + \epsilon \mathbf{q}) - \mathbf{F}(\mathbf{x}^{(k)})}{\epsilon}.$$

- More expensive than sparse matrix vector products.
- Immense flexibility
- Since we solve the correct linear equation system, quadratic convergence is retained, thus fast method!

Convergence of JFNK

Theorem

Let the standard assumptions hold. Then there are δ, $\bar{\sigma}$, C_G *such that if* $\mathbf{x}_0$ *is in a* δ*-neighborhood of* $\vec{x}^*$ *and the sequences* $\{\eta_k\}$ *and* $\{\epsilon_k\}$ *satisfy*

$$\sigma_n = \eta_n + C_G \epsilon_n \leq \bar{\sigma},$$

then the Jacobian free Newton-GMRES iteration converges linearly. Moreover,

- *if* $\sigma_k \to 0$*, the convergence is superlinear and*
- *if* $\sigma_k \leq K_\eta \|\mathbf{F}(\mathbf{x}_k)\|^p$ *for some* $K_\eta > 0$ *and* $p \in [0,1]$*, the convergence is superlinear with order* $1+p$*.*

Final Flow-Chart for DIRK scheme

Given error tolerance TOL, initial time t_0 and time step size Δt_0

- For $i = 1, ..., s$
  - For $k = 0, 1, ...$ until termination criterion with tolerance $TOL/5$ is satisfied or MAX_NEWTON_ITER has been reached
    - Determine Eisenstat-Walker relative tolerance
    - Solve linear system using preconditioned GMRES
- If MAX_NEWTON_ITER has been reached, but the tolerance test has not been passed, repeat time step with $\Delta t_n = \Delta t_n/4$
- Estimate local error and compute new time step size Δt_{n+1}
- $t_{n+1} = t_n + \Delta t_n$

Note: Puts additional bound on time step via nonlinear solver

Efficiency of different Newton schemes

JFNK-FT	JFNK-EW	Newton-type-FT	Newton-type-EW
3.0	4.1	0.7	0.9

Table: Upper bounds of convergence radius for Shu vortex problem in terms of CFL numbers. FT stands for a fixed tolerance of 10^{-2}, EW for Eisenstat-Walker.

	JFNK-FT	JFNK-EW	Newton-type-FT	Newton-type-EW
Iter.	32,793	6,821	45,531	10,101
CPU	17,394	4,566	19,920	10,672

Table: Comparison of efficiency of different Newton variants. SDIRK2, $TOL = 10^{-2}$

Newton convergence: Unsteady TEMPO

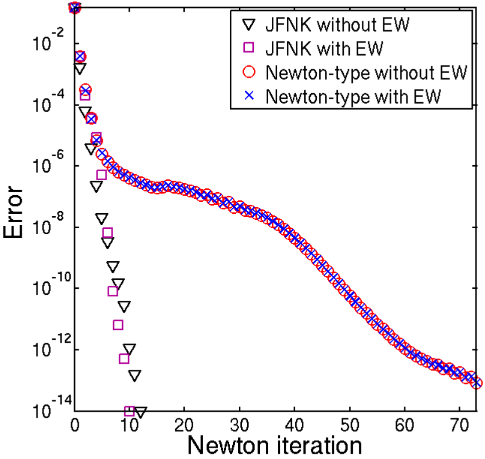

Figure: Convergence of different Newton schemes for one time step during an unsteady calculation

Outline

1. Introduction
2. Newton's method
3. JFNK
4. Newton and Multigrid
5. The fine print

Newton convergence: TEMPO for steady states

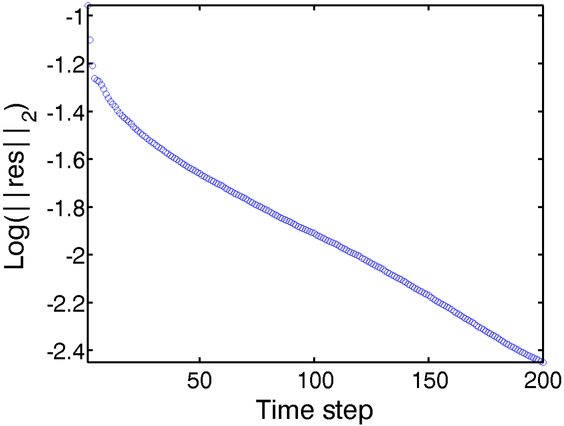

Figure: Convergence to steady state of a damped scheme of Newton-type

Multigrid convergence: UFLO103 on structured grid

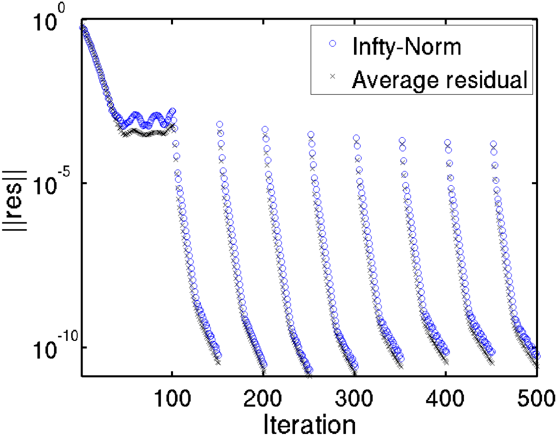

Figure: UFLO103 convergence, first 100 steps using the steady state solver, then 50 iteration of dual time stepping per time step

Multigrid convergence: DLR TAU

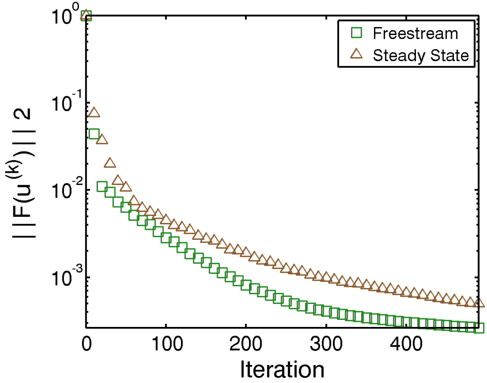

Figure: DLR TAU dual time stepping for two different systems.

FAS in TAU converges to something else. Wrong BC on coarse grids?

Newton and Multigrid

Steady States

- Multigrid fast
- Newton slow

Unsteady problems

- Here we expect significantly faster convergence
- Multigrid marginally, if at all, faster, not fast on unstructured grids
- Newton significantly faster than for steady state

Newton and Multigrid

Multigrid

- Dual time stepping typically implies reusing the steady state algorithm without changes
- Thus less than optimal multigrid convergence
- Redesign multigrid (see B., *Optimizing Runge-Kutta smoothers for unsteady flow problems*, ETNA, 2012)

Newton

- Very good scheme: Jacobian-Free Newton-GMRES with good parallel preconditioner
- Multigrid candidate for preconditioner (see above)

Compare *Solving nonlinear systems inside implicit time integration schemes for unsteady viscous flows*, P. Birken, pp. 57-71 in R. Ansorge, H. Bijl, A. Meister, T. Sonar (editors), Numerics of Nonlinear Hyperbolic Conservation Laws, Notes on Numerical Fluid Mechanics, Springer

Outline

1. Introduction
2. Newton's method
3. JFNK
4. Newton and Multigrid
5. The fine print

Choice of initial guess

- Good choice of intial guess important
- Without that, we might have divergence
- With it, we could be close to the solution and need very few steps
- In unsteady time stepping, last value often good choice
- But: For huge time steps, this might not be in radius of convergence (see flow chart)
- Extrapolation of old values can help
- Stage value predictors other possibility

Globalization

- Problem: Local convergence
- Solution globalization strategy via line search:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda \Delta \mathbf{x}^{(k)}, \quad \lambda \in [0,1]$$

- Popular choice: Armijo line search

$$\|\mathbf{F}(\mathbf{x}^{(k+1)}\| < (1-\alpha\lambda)\|\mathbf{F}(\mathbf{x}^{(k)}\|, \quad 0 < \alpha << 1$$

- For example, $\alpha = 10^{-4}$ is a suitable choice.
- Search for suitable λ done in trivial way by starting with 1 and dividing by 2 if condition not satisfied.
- This method is linearly convergent
- When close to solution, full steps with $\lambda = 1$ can be taken and original convergence rate recovered

Implementation Issues

- JFNK approach extremely flexible (think change of space discretization)
- Very modular, thus high reusability
- Avoids storing the Jacobian
- Need vector data structure at some point
- Might be good to have solution in that instead of grid based structure
- Parallelization hinges on space discretization and Krylov solver
- Software: PETSc, SUNDIALS, TRILINOS, TEMPO (soon)

Preconditioning

- Convergence of Krylov subspace methods (GMRES) depends strongly on matrix.
- Transform problem to get better convergence.
- Use right preconditioning, since residual is unchanged (inexact JFNK)

$$\mathbf{APy} = \mathbf{b}, \quad \mathbf{y} = \mathbf{P}^{-1}\Delta\mathbf{x}$$

- In a JFNK scheme, we do not compute the matrix a priori.
- Need to compute all parts of matrix needed for preconditioner.
- This is where things get difficult for the first time

Summary

- Newton locally quadratic convergent
- Problem 1: Need to solve equation exactly
- Solved by inexact Newton scheme
- Choosing proper termination criteria key to efficiency!
- Problem 2: Need to store and compute Jacobian
- Solved by JFNK approach
- Multigrid is main competitor
- Can be used inside a Newton scheme
- Preconditioning is a key issue.

Literature

- C. T. Kelley. Iterative Methods for Linear and Nonlinear Equations. SIAM, Philadelphia, PA, 1995
- J. E. Dennis and R. B. Schnabel. Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Classics in Applied Mathematics. SIAM, Philadelphia, 1996
- P. Deuflhard. Newton Methods. Springer, 2004
- E. Hairer and G. Wanner. Solving Ordinary Differential Equations II. Springer, Berlin, 2nd edition, 2004
- B., Numerical methods for the unsteady compressible Navier-Stokes equations, Habilitation Thesis, 2012, University of Kassel