Lecture 19: The Inexact Newton-Krylov Algorithm

COMP5930M Scientific Computation

Today

Newton's method + iterative linear solver

- We now have 2 levels of iteration
  - An outer iteration for the nonlinear system: Newton's Method
  - An inner iteration for the linear system at each nonlinear step: Preconditioned Krylov-subspace iterations
- It is important to consider the effect of approximate solution of the (inner) linear system on the (outer) nonlinear system solution

Convergence control

- Iterative solution of the linear system is approximate
  - We monitor the residual norm $R_i = ||\mathbf{r}_i||$
- The Newton iteration is also approximate
  - We monitor the nonlinear function norm $F_k = ||\mathbf{F}(\mathbf{U}_k)||$
- While F_k is large we do not require R_i to be completely converged
  - We require sufficient accuracy
- We do not have this flexibility with direct solution

Inexact Newton method

Exact Newton iteration: find $\boldsymbol{\delta}_k$ s.t.

$$\begin{aligned} J(\mathbf{x}_k)\boldsymbol{\delta}_k &= -\mathbf{F}(\mathbf{x}_k) \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha\boldsymbol{\delta}_k \end{aligned}$$

where $\alpha > 0$ is found through a line-search procedure

Inexact Newton iteration: for given $0 \leq \eta_k < 1$, find $\boldsymbol{\delta}_k$ s.t.

$$\begin{aligned} \|\mathbf{J}(\mathbf{x}_k)\boldsymbol{\delta}_k + \mathbf{F}(\mathbf{x}_k)\| &\leq \eta_k\|\mathbf{F}(\mathbf{x}_k)\| \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \boldsymbol{\delta}_k \end{aligned}$$

Inexact Newton method

Exact Newton iteration: find $\boldsymbol{\delta}_k$ s.t.

$$\begin{aligned} J(\mathbf{x}_k)\boldsymbol{\delta}_k &= -\mathbf{F}(\mathbf{x}_k) \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha\boldsymbol{\delta}_k \end{aligned}$$

where $\alpha > 0$ is found through a line-search procedure

Inexact Newton iteration: for given $0 \leq \eta_k < 1$, find $\boldsymbol{\delta}_k$ s.t.

$$\begin{aligned} \|\mathbf{J}(\mathbf{x}_k)\boldsymbol{\delta}_k + \mathbf{F}(\mathbf{x}_k)\| &\leq \eta_k\|\mathbf{F}(\mathbf{x}_k)\| \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \boldsymbol{\delta}_k \end{aligned}$$

2

Inexact Newton method

Exact Newton iteration: find $\boldsymbol{\delta}_k$ s.t.

$$\begin{aligned} \mathbf{J}(x_k)\boldsymbol{\delta}_k &= -\mathbf{F}(\mathbf{x}_k) \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha\boldsymbol{\delta}_k \end{aligned}$$

where $\alpha > 0$ is found through a line-search procedure

Inexact Newton iteration: for given $0 \leq \eta_k < 1$, find $\boldsymbol{\delta}_k$ s.t.

$$\begin{aligned} \|\mathbf{r}_k\| &\leq \eta_k\|\mathbf{F}(\mathbf{x}_k)\| \qquad \text{(linear residual)} \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \boldsymbol{\delta}_k \end{aligned}$$

Theorem: With some mild assumptions, the inexact Newton iteration converges linearly. If $\eta_k \to 0$, convergence is superlinear.

Inexact Newton method

Exact Newton iteration: find $\boldsymbol{\delta}_k$ s.t.

$$\begin{aligned} \mathbf{J}(x_k)\boldsymbol{\delta}_k &= -\mathbf{F}(\mathbf{x}_k) \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha\boldsymbol{\delta}_k \end{aligned}$$

where $\alpha > 0$ is found through a line-search procedure

Inexact Newton iteration: for given $0 \le \eta_k < 1$, find $\boldsymbol{\delta}_k$ s.t.

$$\begin{aligned} \|\mathbf{r}_k\| &\le \eta_k \|\mathbf{F}(\mathbf{x}_k)\| \qquad \text{(linear residual)} \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \boldsymbol{\delta}_k \end{aligned}$$

Theorem: With some mild assumptions, the inexact Newton iteration converges linearly. If $\eta_k \to 0$, convergence is superlinear.

Implementation

At inexact Newton iteration step k, the linear system is solved to the tolerance $\|\mathbf{r}_k\| \leq \eta_k \|\mathbf{F}(\mathbf{x}_k)\|$ using iterative linear solvers.

Standard choices of η_k:

- Eisenstat-Walker: $\eta_k = C\frac{\|\mathbf{F}(x_k)\|^2}{\|\mathbf{F}(x_{k-1})\|^2}$, where $0 < C \leq 1$
- Kelley:

$$\eta_k = \min\left\{\eta_{\max}, \max\left(\eta_k^{\mathrm{safe}}, C\frac{\|\mathbf{F}(x_k)\|^2}{\|\mathbf{F}(x_{k-1})\|^2}\right)\right\}$$

where $\eta_{\max} < 1$ and η_k^{safe} is chosen to ensure that the η_k do not approach 0 too rapidly

(for details, see Kelley CT. Solving nonlinear equations with Newton's method. SIAM, 2003)

Implementation

At inexact Newton iteration step k, the linear system is solved to the tolerance $\|\mathbf{r}_k\| \leq \eta_k \|\mathbf{F}(\mathbf{x}_k)\|$ using iterative linear solvers.

Standard choices of η_k:

- Eisenstat-Walker: $\eta_k = C \frac{\|\mathbf{F}(x_k)\|^2}{\|\mathbf{F}(x_{k-1})\|^2}$, where $0 < C \leq 1$
- Kelley:

3

$$\eta_k = \min \left\{ \eta_{\max}, \max \left(\eta_k^{\text{safe}}, C \frac{\|\mathbf{F}(x_k)\|^2}{\|\mathbf{F}(x_{k-1})\|^2} \right) \right\}$$

where $\eta_{\max} < 1$ and η_k^{safe} is chosen to ensure that the η_k do not approach 0 too rapidly

(for details, see Kelley CT. Solving nonlinear equations with Newton's method. SIAM, 2003)

The Inexact Newton-Krylov Algorithm

- $k = 0$
- Specify or compute $\mathbf{U}_0$
- Compute $\mathbf{F}_0 = \mathbf{F}(\mathbf{U}_0)$
- While $|\mathbf{F}_k| > Tol$
  - Compute $\mathbf{J}_k = \mathbf{J}(\mathbf{U}_k)$
  - Assemble preconditioner matrix $\mathbf{M}_k$
  - Solve $\mathbf{J}_k\delta_k = -\mathbf{F}_k$ to tolerance $tol = \eta_k\|\mathbf{F}_k\|$ using Krylov-subspace iterative method and preconditioner $\mathbf{M}_k$
  - Update $\mathbf{U}_{k+1} = \mathbf{U}_k + \delta_k$
  - Compute $\mathbf{F}_{k+1} = \mathbf{F}(\mathbf{U}_{k+1})$
  - $k = k + 1$

4

Convergence of residuals: exact vs. inexact Newton

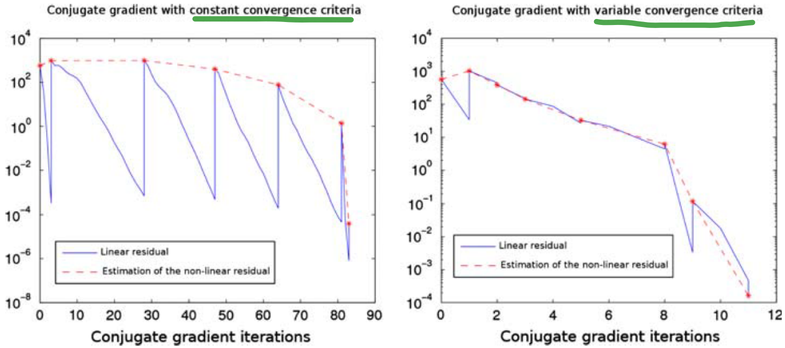

Figure 1. Illustration of the oversolving phenomenon.

Tardieu N and Cheignon E. A Newton-Krylov method for solid mechanics. Eur J Comput Mech 21(3-6), 2012.

Summary

The combination of:

- Newton's method;
- inexact solution of the linear system;
- preconditioned Krylov-subspace iterative techniques;
- matrix-permutation algorithms;
- approximate matrix factoring algorithms;

enables the efficient solution of very large, sparse, highly-nonlinear systems

5

Case study: Porous-medium equation

$u(x, y)$ represents concentration of some property

$$\frac{\partial}{\partial x}\left(g(u)\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(g(u)\frac{\partial u}{\partial y}\right) = 0$$

where

$$g(u) = 1 + \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2$$

Defined on a spatial region Ω with boundary conditions
$u = U_1(x, y)$ on boundary $\partial\Omega$

Spatial approximation

- A linear triangular finite element method is used (equivalent to FDM in 1-D)
- The resulting Jacobian is sparse and symmetric positive definite and can be approximated analytically

Numerical method

- A Newton-Krylov solution strategy
- Preconditioned Conjugate Gradient iterations for the Jacobian system
- A state-of-the-art multigrid preconditioner[1] was tested for the CG iterations

[1] Recall idea of using multiple levels of discrete grids to find initial conditions, can also be used here to generate efficient preconditioners

Timing results

Equations N	Newton iterations	Sparse direct Time (s)	CG iterations Time (s)
9^2	10	0.9	0.1
17^2	12	21.6	0.6
33^2	13	382.6	2.7
65^2	15	-	12.7
129^2	17	-	57.1
257^2	19	-	261.6