Newton-Krylov Methods for Expensive Nonlinear Function Evaluations

Rebecca Wasyk[1]

In collaboration with: Peter N. Brown[2], Homer Walker[1], and Carol Woodward[2]

[1]Mathematical Sciences Dept., Worcester Polytechnic Institute [2]Center for Applied Scientific Computing, LLNL

The Problem

Find u^* such that $F(u^*) = 0$.

$u^* \in \Re^n$ **n is large**

***F* is nonlinear, and expensive to evaluate**

Introduction

Newton-Krylov methods are useful for solving large scale nonlinear problems. Key features include:

- Fast (quadratic) local convergence to a solution
- Do not require storage of the system Jacobian
  - more storage space available
  - able to solve bigger problems

Algorithm

u^k = current approx. to u^*

Form linear model for F:

$$\underbrace{F(\vec{u}^{k+1})}_{\text{want this } = 0} = F(\vec{u}^k) + \underbrace{J(\vec{u}^k)}_{\text{system Jacobian}} \underbrace{(\vec{u}^{k+1} - u^k)}_{=\vec{s}} \Rightarrow$$

$$J(\vec{u}^k)\vec{s} = -F(\vec{u}^k)$$

Approximate:

$$J(\vec{u}^k)\vec{s} \approx \frac{F(\vec{u}^k + \tau\vec{s}) - F(\vec{u}^k)}{\tau}$$

- Approximation does not change convergence rate
- $F(\vec{u}^k)$ can be saved, so only computed once
- $F(\vec{u}^k + \tau\vec{s})$ is calculated with each linear iteration
  - Very costly if F is expensive to evaluate!!

Solve the linear model for $\vec{s}$ using Krylov method (iterative)

Update:

$$\vec{u}^{k+1} = \vec{u}^k + \vec{s}$$

$$\vec{u}^k = \vec{u}^{k+1}$$

Is $F(\vec{u}^k) \approx 0$?

Yes → End

No → u^k = current approx. to u^*

Modifying for Expensive Nonlinear Functions

Replace $F(\vec{u}_k + \tau\vec{s})$ with $\tilde{F}(\vec{u}_k + \tau\vec{s})$ in the difference quotient only, where $F \approx \tilde{F}$, but $\tilde{F}$ is cheaper to evaluate.

- Theory shows careful choice of $\tilde{F}$ will not slow convergence
- Number of function evaluations dominated by calls made for difference quotient
  - Calls will be made to less expensive $\tilde{F}$
  - Expect less run time

Testing the Modified Method

We consider the system below which models predator-prey interaction for a number of species (ns):

$$F_i(\vec{u}) = \nabla \cdot D(u_i)\nabla u_i + f_i(x, y, \vec{u}) = 0, \; i = 1 \ldots ns$$

We solve for $\vec{u}$, which represents the species concentration over the domain, with u_i the concentration of the ith species, and $\hat{u}$ the approximation to u^* from the previous iteration.

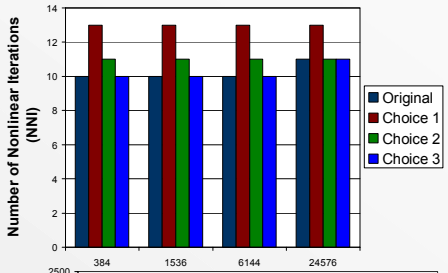
Three choices are made for $\tilde{F}$, by changing how $D(u_i) = d_i + 0.5u_i^2$ is computed:

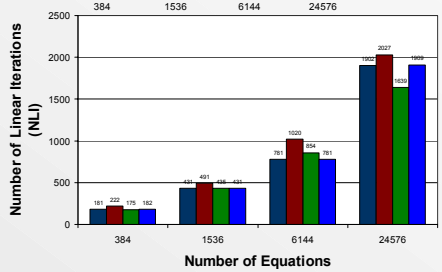
1. $\tilde{D} = D(\hat{u}_i)$ (cheapest, theory predicts poor performance)
2. $\tilde{D} = D(u_i, \hat{u}_i) = d_i + 0.5\hat{u}_i u_i$
3. $\tilde{D} = D(\hat{u}_i) + D'(\hat{u}_i)(u_i - \hat{u}_i)$ (most expensive, theory shows this choice will preserve convergence rate)

Preliminary Results

In the following figures, note:

- NNI (NLI) for **Choice 3** ≈ NNI (NLI) for **Original**
- Cheapest estimate, **Choice 1**, did poorly in comparison
- Interesting results for Choice 2 merit more investigation





Discussion

- Initial testing indicates modified difference quotient retains the local convergence properties, for at least some choices of approximating functions
- System function used for testing was not expensive to evaluate, so no direct evidence that this modification will decrease runtime
- Future tests on problems with more significant nonlinearities are planned to explore the issue of runtime savings

This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

UCRLxx-xxxxxxx