Newton's Method for Monte-Carlo Based Residuals

C. T. Kelley
NC State University
tim_kelley@ncsu.edu
Joint with Jeff Willert, Dana Knoll, Ryosuke Park, Xiaojun Chen

Supported by CASL, LANL, ARO, NSF

CUNY, New York, April 25, 2013

Outline

1 Collaborators

2 Newton's Method

3 Newton-MC

4 JFNK-MC

5 Neutron Transport Equation

6 Results

7 Conclusions

Collaborators

- Jeff Willert, NCSU
- Dana Knoll, Ryosuke Park, LANL
- Xiaojun Chen, Hong Kong Poly

Notation

Objective: find a solution of

$$F(u) = 0$$

where $F : R^N \to R^N$.
We write $F = (f_1, \ldots, f_N)^T$. The Jacobian matrix F' is

$$(F')_{ij} = \partial f_i / \partial u_j$$

Newton's Method

Transition from current point u_c to new one u_+.

$$u_+ = u_c - F'(u_c)^{-1} F(u_c).$$

Interpretation: u_+ is the root of the local linear model at u_c

$$L_c(u) = F(u_c) + F'(u_c)(u - u_c)$$

Convergence Theory for Exact Linear Solves

Standard Assumptions (SA):

- $F(u^*) = 0$
- $F'(u^*)$ is nonsingular.
- $F'(u)$ is Lipschitz continuous with Lipschitz constant γ

$$\|F'(u) - F'(u')\| \leq \gamma \|u - u'\|$$

Convergence Theory: I

Theorem: SA + Good Data ($\|e_0\|$ small) implies that

$$\|e_+\| = O(\|e_c\|^2).$$

Here $e = u - u^*$.
So the iteration converges.

Errors in Function and Jacobian

Theorem: SA + Good Data + Small Perturbations imply that if

$$u_+ = u_c + s$$

$$\|(F'(u_c) + \Delta(u_c))^{-1}s + (F(u_c) + \epsilon(u_c))\| \leq \eta_c \|(F(u_c) + \epsilon(u_c))\|$$

then

$$\|e_+\| = O(\|e_c\|^2 + (\|\Delta(u_c)\| + \eta_c)\|e_c\| + \|\epsilon(u_c)\|).$$

So the iteration converges up to the resolution of the residual F.

The problem here is different

Evaluations of F, F', ... have Monte Carlo (MC) components, so

- You have a variance estimate, but ...
- No bounds on errors in functions/Jacobians
- Severe error accumulation in methods like matrix-free Newton-GMRES

MC-based Data for Newton: Notation

- N_{MC} (N^J_{MC}) is the number of trails for the function (Jacobian or Jacobian-vector product).
- $\tilde{F}(u, N_{MC})$ an outcome for the residual $F(u)$.
- $J(u, N^J_{MC})$ an outcome for the Jacobian $F'(u)$.
- $J_p(u, v, N^J_{MC})$ an outcome for the product $F'(u)v$.

Consistency: Matrix-Based Methods

There are functions c_F and c_J such that for $\delta > 0$

$$Prob \left(\| F(u) - \tilde{F}(u, N_{MC}) \| > \frac{c_F(\delta)}{\sqrt{N_{MC}}} \right) < \delta,$$

and

$$Prob \left(\| F'(u) - J(u, N_{MC}^J) \| > \frac{c_J(\delta)}{\sqrt{N_{MC}^J}} \right) < \delta.$$

Algorithmic Idea

- Increase N_{MC} as the iteration progresses
  to refine quality of solution.
- Leave N_{MC}^J alone.
  Speedup in nonlinear iteration not worth it.

Will this iteration converge?

Algorithmic Idea

- Increase N_{MC} as the iteration progresses to refine quality of solution.
- Leave N_{MC}^{J} alone.
  Speedup in nonlinear iteration not worth it.

Will this iteration converge? Probably!

Newton-MC Algorithm

Newton-MC$(u, N_{MC}, N^J_{MC}, N_{inc}, \eta, \tau_r, \tau_a)$

Evaluate $R_{MC} = \tilde{F}(u, N_{MC})$; $\tau \leftarrow \tau_r \|R_{MC}\| + \tau_a$.
while $\|R_{MC}\| > \tau$ **do**
  Compute $J(u, N^J_{MC})$
  Find s which satisfies
  $\|J(u, N^J_{MC})s + \tilde{F}(u, N_{MC})\| \le \eta \|\tilde{F}(u, N_{MC})\|$.
  $u \leftarrow u + s$
  Evaluate $R_{MC} = \tilde{F}(u, N_{MC})$;
  $N_{MC} \leftarrow N_{inc} N_{MC}$
end while

Tracking Theorem: Assumptions

- Standard Assumptions for Newton
- Exact (no MC) Newton iteration $\{x_n^N\}$
- Initial iterate good enough and η small enough so that $\|e_{n+1}^N\| \le r_N \|e_n^N\|$

We do not try to drive $\eta_n \to 0$. Not practical with an MC-based residual.

Tracking Theorem

Given integer K, $\omega \in (0,1)$, and $r \in (r_N, 1)$, there are N_{MC}, N^J_{MC}, and N_{inc}, such that with probability $(1-\omega)$ for all $1 \le n \le K$, the iteration produced by Algorithm **Newton-MC** satisfies

$$\|e_n\| \le r^n \|e_0\|,$$

and there is K_F (depending only on F and u^*) such that

$$\|F(u_n)\| \le K_F r^n \|F(u_0)\|.$$

Tracking is not Convergence

- You only see a high probability of error/residual reduction,
- and only for a fixed (K) number of iterations.
- You have to manage N_{MC} to get even that.

And it gets stranger ...

Consistency: Matrix-Free Krylov Methods

Residual Consistency

$$Prob\left(\|F(u) - \tilde{F}(u, N_{MC})\| > \frac{c_F(\delta)}{\sqrt{N_{MC}}}\right) < \delta,$$

plus

$$Prob\left(\|F'(u)v - J_p(u, v, N^J_{MC})\| > \frac{c_{Jv}(\delta)}{\sqrt{N^J_{MC}}}\right) < \delta.$$

Bad Idea for Algorithm

Terminate inner iteration when the Monte Carlo Inexact Newton Condition (INC-MC) holds:

$$\|J_p(u, s, N_{MC}^J) + \tilde{F}(u_c, N_{MC})\| \leq \eta \|\tilde{F}(u_c, N_{MC})\|,$$

and obtain a nice tracking theorem.
Just like normal inexact Newton. What could go wrong?

But there's a problem

- You call the mat-vec with every Krylov iteration.
  There is no Jacobian behind that mat-vec.
- The accuracy is low.
- Errors accumulate rapidly.
- This is a big deal (Simoncini-Szyld 03–07)
  - Their solution: very high accuracy for early Krylovs
  - We can't do that; MC simulation too costly.

But there's a problem

- You call the mat-vec with every Krylov iteration.
  There is no Jacobian behind that mat-vec.
- The accuracy is low.
- Errors accumulate rapidly.
- This is a big deal (Simoncini-Szyld 03–07)
  - Their solution: very high accuracy for early Krylovs
  - We can't do that; MC simulation too costly.
- One solution:

But there's a problem

- You call the mat-vec with every Krylov iteration.
  There is no Jacobian behind that mat-vec.
- The accuracy is low.
- Errors accumulate rapidly.
- This is a big deal (Simoncini-Szyld 03–07)
  - Their solution: very high accuracy for early Krylovs
  - We can't do that; MC simulation too costly.
- One solution: Take Youngman's Method,

But there's a problem

- You call the mat-vec with every Krylov iteration.
  There is no Jacobian behind that mat-vec.
- The accuracy is low.
- Errors accumulate rapidly.
- This is a big deal (Simoncini-Szyld 03–07)
  - Their solution: very high accuracy for early Krylovs
  - We can't do that; MC simulation too costly.
- One solution: Take Youngman's Method, please.

But there's a problem

- You call the mat-vec with every Krylov iteration.
  There is no Jacobian behind that mat-vec.
- The accuracy is low.
- Errors accumulate rapidly.
- This is a big deal (Simoncini-Szyld 03–07)
  - Their solution: very high accuracy for early Krylovs
  - We can't do that; MC simulation too costly.
- One solution: Take Youngman's Method, please.
  Doctor, it hurts when I take too many Krylovs.

So don't take too many Krylovs!

H. Youngman

Limitations of Youngman's Method

- Need a good preconditioner $\rightarrow$ low Krylov count is ok.
  - You don't know how low, in general.
  - Our example has the compactness properties you want.
- Must impose a hard limit on Krylov count and hope it's enough.
- Theory is cleanest with GMRES as the Krylov solver.

Jacobian-Free Newton-Krylov-MC (JFNK-MC)

JFNK-MC$(u, N_{MC}, N^J_{MC}, N_{inc}, \eta, K_L, K_R, \tau_r, \tau_a)$

Evaluate $R_{MC} = \tilde{F}(u, N_{MC})$; $\tau \leftarrow \tau_r \|R_{MC}\| + \tau_a$.
while $\|R_{MC}\| > \tau$ **do**
  Use GMRES(K_L) with at most $K_R - 1$ restarts to find s.
  Terminate the Krylov iteration when INC-MC holds or after $K_L \times K_R$ linear iterations.
  $u \leftarrow u + s$
  $N_{MC} \leftarrow N_{inc} * N_{MC}$;
  Evaluate $R_{MC} = \tilde{F}(u, N_{MC})$
end while

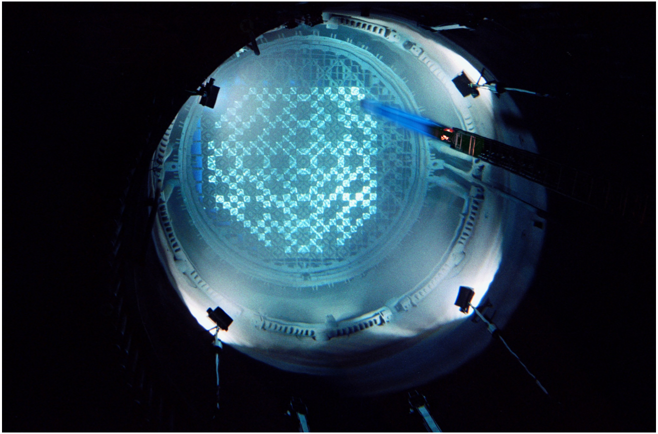
Tracking Theorem for JFNK-MC

Assumptions:

- Assumptions for first tracking theorem, PLUS
- At most K_L Krylovs needed to get INC for exact iteration.
- No breakdown (premature termination) in GMRES iteration.

Conclusion: Same.

Region of Interest: Heterogeneous



From CASL Image Gallery

Neutron Transport Equation: Ia

Notation:

- $D \subset R^3$, $r = (x, y, z)^T \in D$
- $\hat{\Omega} \in S^2$
- $\psi(r, E, \hat{\Omega})$ angular flux of Neutrons at
  - space: location r
  - momentum: direction $\hat{\Omega}$ and energy $E \geq 0$

Boundary conditions on incoming flux.

Neutron Transport Equation: Ib

This is a high-dimensional problem:

- One space dimension is 3D (space+direction+energy)
- Two space dimensions is 5D (space+ $2 \times$ direction + energy)
- Three space dimensions is 6D (space+$2 \times$ direction+energy)

Why does one space dimension lead to a continuum of directions? Stay tuned.

MC vs Deterministic

- Solutions in 3D are non-smooth
  - Fine grids (in 6D!) are needed to resolve the solution.
  - Angular grids produce artifacts in vacuum regions.
  - Poorly scaled problems (shielding) can't be done well at all.
- MC addresses these things, but . . .
  - MC is slow
  - MC needs tuning for do its best.
- Hybrid methods seem to solve some of the problems.

Neutron Transport Equation: II

Simplifications: round 1

- No time dependence.
  Solve time-independent problems to integrate in time.
- Neglect fission sources.
  Methods don't change much if we put them in.

Neutron Transport Equation: III

$$\hat{\Omega} \cdot \nabla \psi(r, E, \hat{\Omega}) + \Sigma_t(r, E)\psi(r, E, \hat{\Omega})$$

$$= \frac{1}{4\pi} \int_{S^2} d\hat{\Omega}' \int_0^\infty dE' \Sigma_s(r, E' \to E, \hat{\Omega}' \to \hat{\Omega}) \psi(r, E', \hat{\Omega}')$$

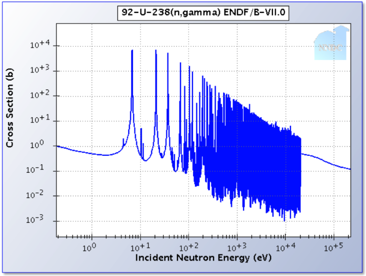
$$+ q(r, E, \hat{\Omega})/4\pi,$$

Quantity of interest: scalar flux

$$\phi(r, E) = \int_{S^2} \psi(r, E, \hat{\Omega}') d\hat{\Omega}'$$

Neutron Transport Equation: III

Σ_t and Σ_s are transmission and scattering cross sections. They are ugly!



From CASL Image Gallery

Solution Methods: two of many

- S^N: Faster and easier to analyze
  - quadrature in angle,
  - differences/elements/volumes in space,
  - piecewise constant in Energy (multi-group approximation). Yet another problem with deterministic computing.
  - Solve with standard linear (or nonlinear!!) solvers.
- Monte-Carlo (MC): Slower with better results
  - No meshes in angle/energy
  - Neutron ensemble randomly scatters via scattering/energy cross sections
  - Accumulate fluxes (and other angular moments) at the end.

More Simplification

This is too much, so we simplify again to

- Monoenergetic (no E)
- Isotropic (no $\hat{\Omega}' \to \hat{\Omega}$)
- One space dimension

You can learn a lot from even this simple case.

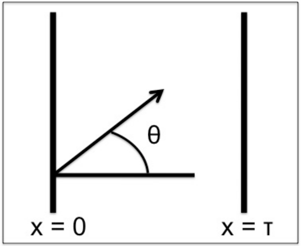
Transport Equation in 1-D

$$\mu \frac{\partial \psi}{\partial x}(x, \mu) + \psi(x, \mu) = \frac{c(x)}{2} \int_{-1}^{1} \psi(x, \mu') \, d\mu' + q(x)/2,$$

Boundary Conditions:

$$\psi(0, \mu) = \psi_l(\mu), \mu > 0; \psi(\tau, \mu) = \psi_r(\mu), \mu < 0.$$

$\mu = \cos(\theta)$



Nonlinear Diffusion Acceleration (NDA): Step 1

Nonlinear compact fixed point problem for ϕ.
Transport sweep: Given input ϕ^{LO}, solve the high-order problem

$$\mu \frac{\partial \psi}{\partial x}(x, \mu) + \psi(x, \mu) = \frac{c(x)}{2} \phi^{LO}(x) + q(x)/2$$

Then compute the high-order flux

$$\phi^{HO}(x) = \int_{-1}^{1} \psi(x, \mu') \, d\mu'$$

and current

$$J^{HO}(x) = \int_{-1}^{1} \mu' \psi(x, \mu') \, d\mu' \equiv M_1 \psi$$

computing ψ only once for both

Nonlinear Diffusion Acceleration (NDA): Step 2

Define

$$\hat{D} = \frac{J^{HO} + \frac{1}{3}\frac{d\phi^{HO}}{dx}}{\phi^{HO}}.$$

The low-order nonlinear equation for ϕ^{LO} is $\mathcal{F}(\phi^{LO}) = 0$, where

$$\mathcal{F}(\phi) = \frac{d}{dx}\left[\frac{-1}{3}\frac{d\phi}{dx}\right] + (1-c)\,\phi + \frac{d}{dx}\left[\hat{D}(\phi^{HO}, J^{HO})\phi\right]$$

with boundary conditions $\phi^{LO} = \phi^{HO}$.

Equivalent to original formulation!

Preconditioner

Preconditioner inverts

$$Lw = \frac{d}{dx}\left[\frac{-1}{3}\frac{dw}{dx}\right] + (1-c)\,w + \hat{D}(\phi^{HO}, J^{HO})\frac{dw}{dx}$$

with correct boundary conditions to solve linearized problem for the Newton step.
This linearized problem is a compact fixed point problem.

NDA Hybrid Method

- Solve the high-order problem with Monte Carlo for ϕ and J
  - better results + higher cost
  - BUT we are asking MC to solve a fixed source problem
    no scattering is good + GPU Friendly.
- Solve the low-order problem in a standard way
- Resolve a few details

Analytic Jacobian-Vector Product

If we plan to replace the transport sweep with a MC simulation, then

- A finite difference Jacobian is a problem
  - Accuracy is roughly square root of function accuracy at best
  - MC accuracy will be low and expressed probabilistically
- So the only hope for us is an analytic Jacobian-vector product:
  - chain rule + linear dependence of ϕ^{HO} and J^{HO} on ϕ^{LO}.
  - cost of mat-vec is one extra MC transport sweep.

Example

Physical parameters:

$$c = .99; \tau = 10; q = .5;$$

Old Way: Beat GMRES to death and

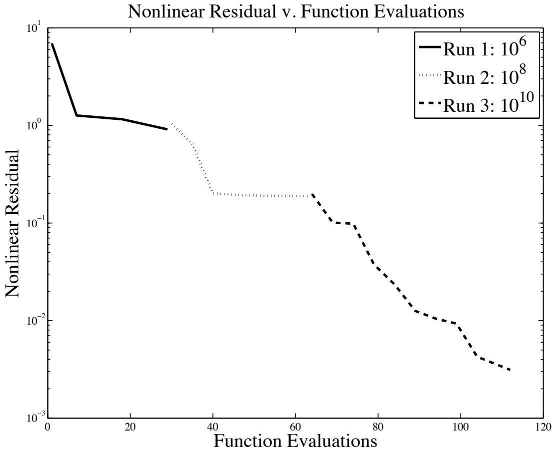
$$N_{MC} = 10^6, 10^8, 10^{10}$$

New Way:

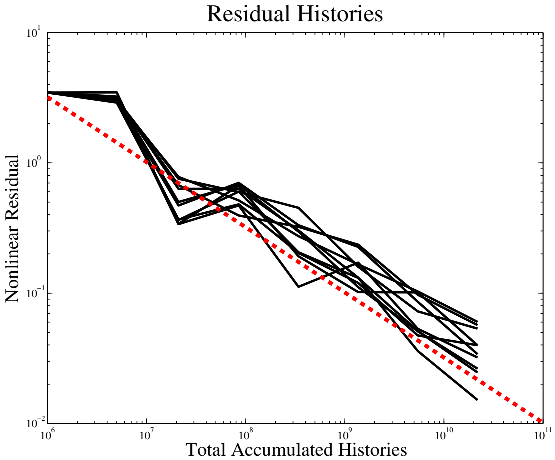
$$N_{MC} \leftarrow N_{MC}/2$$

with each iteration

Iteration History: Old Way



Iteration History: New Way $K_L = 10, K_R = 1, \eta = .01$



Conclusions

- Tracking theorems for Newton-MC
- Watch out for linear solver errors
- Examples from neutronics simulations