NEWTON'S METHOD FOR MONTE CARLO-BASED RESIDUALS[*]

JEFFREY WILLERT[†], XIAOJUN CHEN[‡], AND C. T. KELLEY[§]

Abstract. We analyze the behavior of inexact Newton methods for problems where the nonlinear residual, Jacobian, and Jacobian-vector products are the outputs of Monte Carlo simulations. We propose algorithms which account for the randomness in the iteration, develop theory for the behavior of these algorithms, and illustrate the results with an example from neutronics.

Key words. Newton's method, Monte Carlo simulation, JFNK methods, Neutron transport

AMS subject classifications. 45L10, 65H10, 82D75.

1. Introduction.

We consider the solution of systems of nonlinear equations

$$F(u) = 0, \tag{1.1}$$

when the residual, $F(u)$, Jacobian, and Jacobian-vector products are not computed directly, but are instead approximated with a Monte Carlo (MC) simulation using a number of trials which one may vary during an inexact Newton iteration. Such problems arise, for example, in neutronics [17, 26], and we will use an example from [26] in § 5 as an example in this paper. We propose and analyze an inexact Newton method and show how the MC error affects the iteration. The theory will provide guidance in managing the number of trials in the MC simulation as the iteration progresses.

The results in this paper are very different from results which consider deterministic errors in residuals, Jacobians, and Jacobian-vector products [4–7, 13–16] some of which we review in § 2. An important feature of these previous papers is that the errors have upper bounds which can be used in the analysis. The errors in the problems considered here do not have upper bounds, but rather, small variances. This leads to significant differences in both the theory and the algorithms.

The randomness implies that one cannot prove asymptotic convergence results without letting the number of trials increase very rapidly, which is an impractical approach. Hence we prove results about how well an iteration based on MC approximations tracks a finite part of the idealized iteration using the true function and Jacobian. Therefore we use the term "tracking" instead of convergence.

We will consider local theory in this paper, so we assume that the standard assumptions for local quadratic convergence [9, 13] hold for the function F and the initial iterate u_0. We will assume

[*] Revision of April 15, 2015. This work has been been partially supported by the Consortium for Advanced Simulation of Light Water Reactors (www.casl.gov), an Energy Innovation Hub (http://www.energy.gov/hubs) for Modeling and Simulation of Nuclear Reactors under U.S. Department of Energy Contract No. DE-AC05-00OR22725, National Science Foundation Grants DMS-1406349, and CDI-0941253, Los Alamos National Laboratory contract No. 172092-1, and Army Research Office Grant W911NF-11-1-0367, and by Hong Kong Research Grant Council PolyU5003/11p.

[†] Theoretical Division, MS B216, Los Alamos National Laboratory, Los Alamos, NM 87545, USA. (jaw@lanl.gov).

[‡] Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China (maxjchen@polyu.edu.hk).

[§] North Carolina State University, Department of Mathematics, Box 8205, Raleigh, NC 27695-8205, USA (Tim_Kelley@ncsu.edu).

that the errors in the MC simulations behave like those from a Monte Carlo method for computing integrals [21]. So, when we ask for the residual F, the Jacobian F', or a Jacobian-vector product, the output of the simulation is centered at the correct value with a variance inversely proportional to the square root of the number of trials. This is not the same as saying the error is inversely proportional to the square root of the number of trials, which is a case one can understand with existing theory (see [13] and § 2).

The results in this paper explain the behavior of the algorithm reported in [26] and improve that algorithm by better management of the number of MC trials. The algorithm in [26] modifies a Jacobian-free Newton-Krylov (JFNK) iteration by testing for random errors which, for example, cause the norm of the nonlinear residual to fail to decrease after a Newton step or a linear iteration for an inexact Newton step to fail to converge. The new approach in this paper increases the number of MC trials with every nonlinear iteration.

In § 2 we will establish notation and formally state our assumptions. Then we will review some of the local theory for Newton's method. In § 3 we will state the assumptions on the MC simulations and connect those assumptions to the concept of consistency from stochastic optimization [21].

In § 4 we will state and prove two tracking theorems. Theorem 4.1 is for the idealized case where we can approximate residuals and Jacobians equally well. We can directly apply the results in § 2 to this case because we can explicitly estimate the moduli of continuity of a single inexact Newton iteration as a function of the residual and Jacobian (see Theorem 2.2). Theorem 4.2 is for the particular JFNK method we used in [26], which uses GMRES as the linear solver and MC approximations of the Jacobian-vector product. In this case we do not have an explicit expression of the continuity properties of the iteration as a function of the residual and sequence of Jacobian-vector products in the linear iteration.

Finally we illustrate the results by solving one of the problems from [26] with several variations of the algorithm.

2. Nonlinear Solver Preliminaries.

We begin by setting the notation for nonlinear equations and reviewing the local convergence theory for Newton's method. We give the estimate from [13] on the effects of errors in the function and Jacobian evaluations on inexact Newton methods. The results in this section are either known [9, 13, 18] or direct consequences of known results, but our use of them is new, so we present them in some detail.

We will let $\| \cdot \|$ denote any weighted inner product norm on R^N. We will use the inner product only in our discussion of Newton-Krylov methods, Newton-GMRES in particular, in § 4.2. Elsewhere the norm could be any norm. We will also use $\| \cdot \|$ to denote the matrix norm induced by the vector norm and $\kappa(\cdot)$ to denote the condition number relative to the norm.

2.1. Convergence of Newton's Method.

We will let $u^* \in R^N$ be a solution of (1.1) at which the standard assumptions for local quadratic convergence of Newton's method hold. These assumptions are

Assumption 2.1.

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

Here F' is the Jacobian of F and the final assumption is that

$$\|F'(u) - F'(v)\| \le \gamma \|u - v\| \tag{2.1}$$

for all u, v sufficiently near u^*. Now let

$$\rho \in \left(0, \frac{1}{2\gamma \|F'(u^*)^{-1}\|}\right) \tag{2.2}$$

be such that (2.1) holds for all u, v in the set

$$\mathcal{B}(\rho) = \{z \,|\, \|z - u^*\| \le \rho\}. \tag{2.3}$$

As is standard [9, 13] we will describe iterative methods in terms of the transition from u_c, the current approximation to u^*, to a new approximation, u_+. Newton's method is

$$u_+ = u_c - F'(u_c)^{-1} F(u_c). \tag{2.4}$$

Theorem 2.1 is taken from several results in [13]. Beyond the convergence result, we also include some estimates that we will use in the rest of the paper. In the statement, and in the rest of the paper, we use the standard notation

$$e = u - u^*.$$

THEOREM 2.1. *Assume that the standard assumptions hold. Let $u_c \in \mathcal{B}(\rho)$. Then*

- *$F'(u_c)$ is nonsingular and*
$$\|F'(u_c)^{-1}\| \le 2\|F'(u^*)^{-1}\|. \tag{2.5}$$
-
$$\frac{3}{4\|F'(u^*)^{-1}\|}\|e_c\| \le \|F(u_c)\| \le \frac{5\|F'(u^*)\|}{4}\|e_c\|. \tag{2.6}$$
- *The Newton iterate $u_+ \in \mathcal{B}(\rho)$ satisfies*
$$\|e_+\| \le \|e_c\|/2 \text{ and } \|e_+\| \le \gamma \|F'(u^*)^{-1}\| \|e_c\|^2.$$

We must deal with errors in both the function and Jacobian and with an inexact solution of the linear equation for the Newton step. The iteration of interest is

$$u_+ = u_c + s, \tag{2.7}$$

where for some $0 \le \eta_c < 1$,

$$\|J_c s + \tilde{F}_c\| < \eta_c \|\tilde{F}_c\|, \tag{2.8}$$

$$J_c = F'(u_c) + \Delta_c, \text{ and } \tilde{F}_c = F(u_c) + \epsilon_c. \tag{2.9}$$

The condition on the step (2.8) is analogous to the classic inexact Newton condition [8, 13]

$$\|F'(u_c)s + F(u_c)\| < \eta_c \|F(u_c)\|. \tag{2.10}$$

Theorem 2.2, taken from [13] summarizes the effects of all of the above deviations from (2.4).

We depart from convention here by making the inequalities in (2.8) and (2.10) strict. We will only need this in the proof of Corollary 2.5. This change does not alter the standard convergence theory or analysis.

THEOREM 2.2. *Let the assumptions of Theorem 2.1 hold. Assume that* $\eta_c < 1$ *and*

$$\|\Delta_c\| \leq \frac{1}{4\|F'(u^*)^{-1}\|}. \tag{2.11}$$

Then J_c *is nonsingular and* u_+*, as defined by* (2.7)*, satisfies*

$$\|e_+\| \leq \gamma\|F'(u^*)^{-1}\|\|e_c\|^2 + (C_J\|\Delta_c\| + C_I\eta_c)\|e_c\| + C_F\|\epsilon_c\|, \tag{2.12}$$

where

$$C_J = 6\|F'(u^*)^{-1}\|,\ C_I = 5\kappa(F'(u^*)),\ \textit{and}\ C_F = 8\|F'(u^*)^{-1}\|. \tag{2.13}$$

We have expressed our convergence results in terms of a general norm. Inexact Newton methods can be formulated in the context of the norm

$$\|\cdot\|_* = \|F'(u^*)\cdot\|$$

[8,10]. With this choice of norm, any $\eta \in [0,1)$ will lead to a locally q-linearly convergent iteration. We have elected to use a general norm $\|\cdot\|$. Among our reasons for this are that (1) most solvers either use the ℓ^2 norm or allow the user to select a norm and (2) the $\|\cdot\|_*$ norm is not the norm associated with the scalar product in a Krylov linear solver.

2.2. A Tracking Theorem for Inexact Newton Methods.

Theorem 2.2 quantifies explicitly how the inexact Newton iteration depends continuously on the residual and Jacobian. That continuity will be critical to the results in this paper. The algorithms we propose in this paper manage the errors in the Jacobian and the residual as the iteration progresses and attempt to track the performance of a pure Newton iteration. To that end we let $\{u_n^\nu\}$ be the sequence of Newton iterations starting with $u_0 \in \mathcal{B}(\rho)$. Theorem 2.1 is applicable and hence (2.5) holds. One may either attempt to manage the errors so that superlinear convergence is preserved or, as we advocate in this paper, preserve q-linear convergence. Our reason for this choice is that rapidly increasing the accuracy via Monte Carlo trials could be prohibitively expensive. We will discuss the alternative of preserving superlinear convergence later in this section.

Theorem 2.1 implies that

$$\|e_{n+1}^\nu\| \leq r_{Newton}\|e_n^\nu\|, \tag{2.14}$$

where

$$r_{Newton} = \|e_0\|\|F'(u^*)^{-1}\|\gamma \leq \rho\|F'(u^*)^{-1}\|\gamma \leq 1/2,$$

by the choice of ρ. Note that r_{Newton} depends on the quality of the initial iterate.

The proof of our tracking theorems will depend on two corollaries of Theorem 2.2. The first, Corollary 2.3, requires a bound on the error in the Jacobian. Corollary 2.5 is more specific and directed at an iteration which uses GMRES as the linear solver with approximate Jacobian-vector products.

COROLLARY 2.3. *Let the assumptions of Theorem 2.2 hold. Let an integer* $K \in [0,\infty]$ *and* $r \in (r_{Newton}, 1)$ *be given. There there are* ϵ_0*,* $\bar{\eta}$*, and* $\bar{\Delta}$ *so that for all* $u_0 \in \mathcal{B}(\rho)$ *the sequence*

$$u_{n+1} = u_n + s_n,\ 0 \leq n \leq K$$

from the iteration (2.7)–(2.9) *satisfies*

$$\|e_n\| \le r^n \|e_0\| \tag{2.15}$$

if

$$\eta_n \in [0, \bar{\eta}], \|J_n - F'(u_n)\| \le \bar{\Delta}, \text{ and } \|\tilde{F}_n - F(u_n)\| \le \epsilon_0 r^n \tag{2.16}$$

for all $0 \le n \le K-1$.

Proof. We begin with the error-free case where $J_n = F'(u_n)$ and $\tilde{F}_n = F(u_n)$ (*i. e.* $\Delta = 0$ and $\epsilon = 0$). Our target will be an iteration $\{u_n^\iota\}$ that converges q-linearly with q-factor $r_\eta \in (r_{Newton}, r)$, *i. e.*

$$\|e_{n+1}^\iota\| \le r_\eta \|e_c^\iota\|. \tag{2.17}$$

Since $\Delta = 0$ and $\epsilon = 0$ in the error-free case, we can apply Theorem 2.2 and can combine (2.12) and (2.14) to obtain

$$\|e_{n+1}^\iota\| \le r_{Newton}\|e_n^\iota\| + C_I \eta_n \|e_n^\iota\|. \tag{2.18}$$

We will have (2.17) if we pick η_n so that

$$\|e_{n+1}^\iota\| \le r_{Newton}\|e_n^\iota\| + C_I \eta_n \|e_n^\iota\| \le r_\eta \|e_n^\iota\|.$$

This requires that

$$\eta_n \le \bar{\eta} \equiv \frac{r_\eta - r_{Newton}}{C_I} = \frac{r_\eta - r_{Newton}}{5\kappa(F'(u^*))}, \tag{2.19}$$

for all n. We will manage η in the following sections by insisting on (2.19).

While one could manage the sequence of Jacobian errors $\{\Delta_n\}$ and the forcing terms simultaneously, we will not do that because we control them in different ways. We chose to manage the forcing term first because we can do that independently of the methods we use to approximate the residual and Jacobian. Now suppose $\epsilon_n = 0$ for all n. In that case we can obtain a q-factor $r_\Delta \in (r_\eta, r)$ by requiring that

$$C_J \|\Delta_n\| + r_\eta \le r_\Delta.$$

Hence we will require that

$$\|\Delta_n\| \le \bar{\Delta} \equiv \frac{r_\Delta - r_\eta}{C_J} = \frac{r_\Delta - r_\eta}{6\|F'(u^*)^{-1}\|}. \tag{2.20}$$

So, if the errors in the residual are zero, we can obtain q-linear convergence with a q-factor that is as close to r_{Newton} as we like. To move beyond that to superlinear or quadratic convergence, we would have to let $\Delta_n \to 0$ and $\eta_n \to 0$. We do not think that is practical in the MC setting of this paper.

Finally we consider reduction in the residual error. Convergence requires that the residual errors $\{\epsilon_n\}$ converge to zero. A q-linear convergence estimate would require that

$$\epsilon_n = O(\|e_n\|) = O(\|F(u_n)\|),$$

with a sufficiently small constant in the O-term. An r-linear convergence estimate would need

$$\epsilon_n = O(r^n)$$

for some $r \in (0,1)$. We will take the latter approach and seek r-linear convergence with an r-factor r, *i. e.*

$$\|e_n\| \leq r^n \|e_0\|. \tag{2.21}$$

We can use Theorem 2.2 again to obtain (2.21). Beginning with $u_0 \in \mathcal{B}(\rho)$ and requiring that (2.19) and (2.20) hold, we will obtain (2.21) as well as $\{u_n\} \subset \mathcal{B}(\rho)$ if

$$\|e_{n+1}\| \leq r_\Delta \|e_0\| r^n + C_F \|\epsilon_n\| \leq \|e_0\| r^{n+1}, \tag{2.22}$$

which is satisfied if

$$\|\epsilon_n\| \leq \epsilon_0 r^n, \tag{2.23}$$

where

$$\epsilon_0 \leq \|e_0\| \frac{r - r_\Delta}{C_F} = \|e_0\| \frac{r - r_\Delta}{8\|F'(u^*)^{-1}\|}. \tag{2.24}$$

□

Our plan for the analysis in § 4.1 is to require (2.19) and then to manage the number of MC trials to force (2.20) and (2.23) to hold with high probability, and thereby obtain (2.21). The difficulty is that one can only do this for a finite subsequence of the iteration (so $K < \infty$), as we will see in the proof of Theorem 4.1.

If one wanted to track superlinear convergence, one would have to reduce the errors in residuals superlinearly and drive the Jacobian error to zero. For example, if one demanded that

$$\lim_{n \to \infty} \frac{\|\epsilon_{n+1}\|}{\|\epsilon_n\|} = 0 \text{ and } \lim_{n \to \infty} \|\Delta_n\| = 0,$$

then one could easily extend the analysis above to show that the iteration was superlinearly convergent if u_0 were sufficiently near u^*. This is, in our opinion, too costly if residuals, Jacobians, and Jacobian-vector products are approximated with MC simulations.

2.3. A Tracking Theorem for JFNK Methods. In this section we consider a matrix-free method. By this we mean that we compute only approximate Jacobian-vector products within the linear solver, and do not apply the linear solver to an approximate Jacobian matrix. The details differ from the analysis in § 2.2 because we cannot consider the error in an approximate Jacobian directly, but must instead analyze the sequence of approximate matrix-vector products within the inner iteration. We will consider only Newton-GMRES in this section. The analysis applies to other Krylov methods, such as conjugate gradient, which are continuous in their data. The results in this section are significantly more detailed than those in § 6.2.1 of [13].

Suppose one can only approximate a Jacobian-vector product and not a complete Jacobian. One example of this situation is using a finite-difference Jacobian-vector product in a Newton-Krylov method [13]. For definiteness, we will use GMRES [20] throughout this paper. We will denote the approximate Jacobian-vector by

$$J_p(u,v) \approx F'(u)v.$$

The difference in the iteration from Corollary 2.3 is only that the inexact Newton condition is realized by a Newton-GMRES iteration with $J_p(u,v)$ used for all the Jacobian-vector products. In the case of finite-difference Jacobian-vector products

$$J_p(u,v) = \frac{F(u+hv) - F(u)}{h}$$

for a properly chosen difference increment h [13].

The effect of the approximate Jacobian-vector product is well understood [4, 13] for a finite-difference directional derivative, and the results in this section apply directly to that case. As is the case for a finite-difference directional derivative, one must scale the direction to obtain good results. For a finite difference approximation [13] we choose the difference increment proportional to $\|u\|/\|v\|$ if $v \neq 0$. In this paper, the approximation of a Jacobian-vector product is via a Monte Carlo simulation, and there is no difference increment. For example, in the Monte Carlo case we consider in § 5 we only evaluate $J_p(u, v)$ for unit vectors v and the define the approximation in the general case by

$$J_p(u, \alpha v) = \alpha J_p(u, v), \tag{2.25}$$

for all scalars α, vectors u, and unit vectors v. One could envision a case where one should use vectors v of the size of $\|u\|$, as one does in the finite difference case, but that would depend on the nature of the Monte Carlo approximation, and is not appropriate for the example in § 5. We will use (2.25) in the analysis that follows.

We now restrict our attention to problems which are well-conditioned enough for the number of Jacobian-vector products per nonlinear iteration in a Newton-GMRES iteration to be uniformly bounded for the entire nonlinear iteration. So, we will impose a limit K_L on the size of the Krylov subspace, *i. e.* we will use GMRES(K_L) as the linear solver, and limit the number of restarts to K_R. We must also keep in mind that GMRES tests the termination criterion (2.10) indirectly within the linear iteration. We must now look into the continuity of the linear iteration in the entire set of Jacobian-vector products.

Let A be a nonsingular matrix. We will denote the output of GMRES(K_L) applied to the linear system $Ax = b$ with at most K_R restarts, relative residual tolerance η, and initial iterate $x_0 = 0$ by

$$x = \mathrm{GMRES}(A, b, K_L, K_R, \eta).$$

Here A may represent either multiplication by the matrix A or application of an approximate matrix-vector product. Because the initial iterate is $x_0 = 0$, one can see from the algorithmic description for GMRES that

$$\mathrm{GMRES}(A, \alpha b, K_L, K_R, \eta) = \alpha \mathrm{GMRES}(A, b, K_L, K_R, \eta), \tag{2.26}$$

for any α. The most important consequence of this is that if two matrices (or the related history of matrix-vector products) are close, then the outputs of the iteration are relatively close. To make this precise, suppose that we compare an "exact" implementation GMRES($A, \alpha b, K_L, K_R, \eta$) with GMRES($A_p, \alpha b, K_L, K_R, \eta$), where A_p is an approximate matrix-vector product function. We state the result as a lemma.

Lemma 2.4. *Suppose that*

$$A_p(\alpha v) = \alpha A_p(v), \text{ for all } \alpha \geq 0, v \in R^N. \tag{2.27}$$

and that

$$\|Aw - A_p(w)\| \leq \Delta_p \tag{2.28}$$

for all unit vectors w, then

$$\| GMRES(A, \alpha b, K_L, K_R, \eta) - GMRES(A_p, \alpha b, K_L, K_R, \eta)\| = \|b\| |\alpha| o(1), \tag{2.29}$$

as $\Delta_p \to 0$. Moreover, if ρ is the computed residual on termination of the iteration with A and ρ^p the residual of the iteration with A_p, then

$$|\rho - \rho^p| = \|b\| o(1) \tag{2.30}$$

as $\Delta_p \to 0$.

Proof. The proof follows from the algorithmic description of a GMRES iteration [13, 20], which implies that the GMRES iteration is continuous in the matrix and right-hand side, and the observation [4, 13] that GMRES with an approximate matrix-vector product $A_p(v)$ is equivalent to GMRES applied to the matrix $\tilde{A}$ whose products with the Krylov vectors $\{v_k\}$ are $A_p(v_j)$. It follows from (2.28) that $\|\tilde{A} - A\| = O(\Delta_p)$. Then the continuity of the GMRES iteration, (2.28), and (2.27) imply (2.29) and (2.30). □

We remark that if the exact iteration terminates prematurely with a "happy breakdown", then (2.29) should be taken to mean that the approximate iteration remains close to the converged result of the exact iteration.

The method of interest in this section replaces $s = -F'(u_c)^{-1} F(u_c)$ in a Newton iteration with:

$$s = GMRES(J_p, -\tilde{F}_c, K_L, K_R, \eta). \tag{2.31}$$

We will compare this with an idealized error-free inexact Newton method

$$s = GMRES(F'(u_c), -F(u_c), K_L, K_R, \eta). \tag{2.32}$$

or the step with an approximate residual and an error-free Jacobian-vector product,

$$s = GMRES(F'(u_c), -\tilde{F}_c, K_L, K_R, \eta). \tag{2.33}$$

We will assume that the iteration in the error-free case converges sufficiently rapidly.

Assumption 2.2. *There are $r_{GMRES} \in (r_{Newton}, 1)$ and $\eta_{GMRES} \in (0, 1)$, such that for any $\eta \in (0, \eta_{GMRES})$ and $u_0 \in \mathcal{B}(\rho)$ the sequence $\{u_n^G\}$ of Newton-GMRES iterations using* (2.32) *as the linear solver converges q-linearly with q-factor at most r_{GMRES}. Moreover, the internal GMRES solver does not break down.*

We will need this in the second part of the proof of Corollary 2.5. We are also implicitly assuming that there is no loss of orthogonality within the GMRES iteration. This assumption is needed to guarantee that the residual computed internally in a GMRES implementation is the same as the actual residual. This is true in exact arithmetic, of course. One can nearly realize this in practice by either using Householder reflections to maintain orthogonality [25] or orthogonalizing twice within each GMRES iteration [19].

Corollary 2.5. *Let Assumption 2.2 and the assumptions of Theorem 2.2 hold. Let an integer $K \in [0, \infty)$, $r \in (r_{GMRES}, 1)$, and $u_0 \in \mathcal{B}(\rho)$ be given. Assume the approximate Jacobian-vector product J_p satisfies* (2.25).

Then there are ϵ_0, $\bar{\eta}$, and $\bar{\Delta}_p$ so that the sequence

$$u_{n+1} = u_n + s_n, \; 0 \le n \le K$$

from the iteration (2.7)–(2.9)*, with the Jacobian-vector products approximated by $J_p(u, v)$ satisfies*

$$\|e_n\| \le r^n \|e_0\|, \; \text{for } 0 \le n \le K, \tag{2.34}$$

if

$$\|F'(u)v - J_p(u,v)\| \le \Delta_p \tag{2.35}$$

for all $u \in \mathcal{B}(\rho)$ and unit vectors v, and

$$\|\tilde{F}_n - F(u_n)\| \le \epsilon_0 r^n, \text{ for } 0 \le n \le K.$$

Proof. The proof follows directly from Lemma 2.4. The lemma states that if Δ_p is sufficiently small then the linear iteration GMRES$(J_p, -\tilde{F}_c, K_L, K_R, \bar{\eta})$ will remain close to those of GMRES$(F'(u_c), -\tilde{F}_c, K_L, K_R, \bar{\eta})$ and the steps will be near enough so that

$$\|F'(u_c)s + \tilde{F}_c\| \le \frac{\bar{\eta} + \eta_{GMRES}}{2}$$

which is sufficient for (2.34) to hold. This completes the proof. □

3. Monte Carlo Approximations and Consistency.

In this section we formalize our assumptions on the accuracy of the function, Jacobian, and Jacobian-vector approximations. We then prove a consistency result to explain in what sense the approximate equations satisfy the hypotheses of the Kantorovich theorem [12, 13]. We defer the statement and proof of a tracking theorem for a specific algorithm until the next section.

3.1. Notation and Accuracy Assumptions.

We will approximate each function, Jacobian, and Jacobian-vector product with a randomized simulation using a variable number of trials. Our notation will be

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

We will refer to the approximations as Monte Carlo approximations because that was the setting in [26]. We assume that the evaluations of $\tilde{F}$, J, and J_p are independent.

Recall that $\mathcal{B}(\rho)$ (see (2.2), (2.1), and (2.3)) is a set of initial iterates from which Newton's method converges. For consistency and the tracking theorems we will require

Assumption 3.1. *There are functions c_F and c_J and an open set $\mathcal{B}'$ which contains $\mathcal{B}(\rho)$ such that, for all $u \in \mathcal{B}'$ and $\delta > 0$*

$$Prob\left(\|F(u) - \tilde{F}(u, N_{MC})\| > \frac{c_F(\delta)}{\sqrt{N_{MC}}}\right) < \delta, \tag{3.1}$$

and

$$Prob\left(\|F'(u) - J(u, N_{MC}^J)\| > \frac{c_J(\delta)}{\sqrt{N_{MC}^J}}\right) < \delta. \tag{3.2}$$

For the matrix-free implementation we will replace (3.2) by a similar assumption on Jacobian-vector products.

ASSUMPTION 3.2. *There is a function c_{Jv} and an open set $\mathcal{B}'$ which contains $\mathcal{B}(\rho)$ such that for all $u \in \mathcal{B}'$, unit vectors $v \in R^N$, and $\delta > 0$,* (3.1) *holds and*

$$Prob\left(\|F'(u)v - J_p(u, v, N^J_{MC})\| > \frac{c_{Jv}(\delta)}{\sqrt{N^J_{MC}}}\right) < \delta. \tag{3.3}$$

These assumptions are very weak. One way to think of them is that the function, Jacobian, and Jacobian-vector products are the outputs of experiments, *i. e.* vector or matrix-valued random variables. Hence one cannot say that $\tilde{F}$, J, or J_p inherit any continuity properties from F. Our tracking results cannot talk about asymptotic convergence rates, but only describe how an iteration based on $\tilde{F}$, J, or J_p tracks an idealized iteration for F itself.

3.2. Consistency Results.

Consistency results for sequences of nonlinear problems typically use the Kantorovich theorem [12, 13] to show that the approximate problems have solutions and that those solutions converge to the solution of the limiting problem. We perform a similar analysis here, but that analysis is complicated by the MC evaluation of the approximations.

If a sequence of functions $\{F_N\}$ converges to F pointwise and the Jacobians F'_N are uniformly Lipschitz continuous and well-conditioned in a neighborhood of u^*, the Kantorovich theorem implies that $F_N(u) = 0$ has a unique solution near u^* and that these solutions converge to u^*. In the present case, however, our assumptions do not imply any continuity properties of F_N or F'_N.

Theorem 3.1 connects the standard assumptions, which F satisfies, with the approximations.

THEOREM 3.1. *Assume that the standard assumptions and Assumption 3.1 hold. Then for any $\epsilon > 0$, $\omega \in (0, 1)$, and $u, v \in \mathcal{B}(\rho)$ there is N^*_{MC} such that, for all $N^J_{MC}, N_{MC} \geq N^*_{MC}$,*

$$\|\tilde{F}(u^*, N_{MC})\| \leq \epsilon, \tag{3.4}$$

$$\|J^{-1}(u, N^J_{MC})\| \leq 4\|F'(u^*)^{-1}\|, \tag{3.5}$$

and

$$\|J(u, N^J_{MC}) - J(v, N^J_{MC})\| \leq \gamma\|u - v\| + \epsilon, \tag{3.6}$$

with probability at least $1 - \omega$.

Proof. Let $\epsilon > 0$ and $\omega \in (0, 1)$ be given. Let N^*_{MC} be large enough so that

$$\frac{c_F(\delta)}{\sqrt{N^*_{MC}}} \leq \epsilon \tag{3.7}$$

and

$$\frac{c_J(\delta)}{\sqrt{N^*_{MC}}} \leq \min\left(\frac{1}{2\|F'(u)^{-1}\|}, \frac{\epsilon}{2}\right). \tag{3.8}$$

Now let

$$\delta \leq 1 - \sqrt{1 - \omega}, \tag{3.9}$$

and $N_{MC} \geq N^*_{MC}$. Since $\delta \leq \omega$, equation (3.1) from Assumption 3.1 and (3.7) imply (3.4) with probability no less than $1 - \delta \geq 1 - \omega$.

Let $N^J_{MC} \geq N^*_{MC}$. (3.8) implies that, with probability $1 - \delta \geq 1 - \omega$

$$\|J(u, N^J_{MC})^{-1}\| \leq 2\|F'(u)^{-1}\|.$$

This and (2.5) imply (3.5) since $u \in \mathcal{B}(\rho)$.

Since $N^J_{MC} \geq N^*_{MC}$, Lipschitz continuity of F' and (3.8) imply, with probability at least

$$(1-\delta)^2 \geq 1 - \omega$$

that

$$\|J(u, N^J_{MC}) - J(v, N^J_{MC})\| \leq \|F'(u) - F'(v)\| + \epsilon \leq \gamma\|u - v\| + \epsilon.$$

This completes the proof. □

4. Algorithms and Tracking Theorems.

In this section we show how the MC approximations of functions, Jacobians, and Jacobian-vector products change the Newton iteration. The theoretical results will then guide the algorithmic discussion. The algorithms manage the errors in the function, Jacobian, and linear solver by increasing the number of MC trials as the iteration progresses.

Our results differ from those for problems with deterministic errors because we cannot compare a point in $\mathcal{B}$ to a root of $\tilde{F}$. In fact, there are no roots of $\tilde{F}$ because $\tilde{F}$ does not return the same value for successive calls with the same inputs. Hence we can only assert that, with high probability, the inexact Newton sequence which uses the approximations tracks the sequence with the exact functions F and F' for a given finite number of iterations.

The JFNK algorithm proposed in [26] tested for stagnation by looking for a decrease in the residual norm. If the residual norm failed to decrease, then the algorithm increased N_{MC}. GMRES was the linear solver. We will discuss a version of that algorithm later in § 4.2. Before that, in § 4.1, we will consider the case where one uses MC simulations to approximate the entire Jacobian matrix.

Our tracking results may remind the reader of mesh-independence theorems (see [1, 2, 11, 16], for example), where one compares a Newton iteration for an infinite-dimensional problem with one for a discretization as the mesh is refined. We will sketch a version of such an analysis in § 2.2 to illustrate the kind of result we seek in this case.

The main tracking results are in § 4.1 and § 4.2.

4.1. Tracking with MC Residual and Jacobian Approximations.

We begin with the case where the residual and Jacobian come from MC simulations, and we compute the matrix-vector product used within the inner GMRES iteration as

$$J(u_c, N^J_{MC})v$$

rather than with a MC matrix-vector product

$$J_p(u_c, v, N^J_{MC}).$$

In this case Theorem 2.2 and the ideas in § 2.2 can be applied directly. This is simpler than the case where the matrix-vector product is an MC simulation (see § 4.2) because estimating the error in the linear solver becomes significantly more complex both in theory and in practice [22–24].

We must explicitly show how the number of MC trials fits into the iteration:

$$u_+ = u_c + s, \text{ where } \|J(u_c, N_{MC}^J)s + \tilde{F}(u_c, N_{MC})\| \le \eta_n \|\tilde{F}(u_c, N_{MC})\|. \tag{4.1}$$

Note that the number of trials N_{MC} for the function need not (and, as we shall see, should not) be the same as the number N_{MC}^J for the Jacobian. The reason for that, as one can see from Theorem 2.2, is that the forcing terms and Jacobian errors influence the rate of convergence, but not the accuracy of the iteration.

Theorem 2.2, of course, does not apply with certainty if one uses MC residuals and Jacobian-vector products. We can, however, use Assumption 3.1 and Corollary 2.3 to adjust N_{MC} and N_{MC}^J so that a finite number of Newton iterates are approximated sufficiently well with high probability.

Our primary goal will be tracking r-linear convergence and obtaining (2.21). For a given finite $K > 0$, the analysis will show that we can, if the algorithmic parameters are chosen correctly, obtain (2.21) for the first K iterations.

We will assume that the hypotheses of Corollary 2.3 hold and that $\eta_n \le \bar{\eta}$, where $\bar{\eta}$ is defined by (2.19). While we could make the limiting convergence as fast as q-quadratic by decreasing η_n and increasing the number of MC trials very rapidly, the work required to capture that convergence rate with the MC-based iteration is far too much. Hence we will fix N_{MC}^J in a way that will enable us to bound the Jacobian error Δ_n with high probability for the first K iterations.

We must increase N_{MC} as the iteration progresses to obtain the tracking results we want. To track r-linear convergence we can increase N_{MC} by a factor of at least r^{-2} with each iteration.

The iteration is

$$u_{n+1} = u_n + s, \text{ where,}$$

$$\|J(u_n, N_{MC}^J)s + \tilde{F}(u_n, N_{MC}^n)\| \le \eta_n \|\tilde{F}(u_n, N_{MC}^n)\|.$$

We formalize this idea in Algorithm **Newton-MC**. The inputs are an initial iterate u, N_{MC}^J, an upper bound $\hat{\eta}$ for the forcing term η, an initial value of N_{MC}, the factor N_{inc} by which N_{MC} will increase with each iteration, and relative and absolute termination parameters. Based on Corollary 2.3 and its proof we will require that

$$0 \le \hat{\eta} \le \bar{\eta} \tag{4.2}$$

where $\bar{\eta}$ is defined by (2.19).

```
**Newton-MC**$(u, N_{MC}, N_{MC}^J, N_{inc}, \hat{\eta}, \tau_r, \tau_a)$
  Evaluate $R_{MC} = \tilde{F}(u, N_{MC})$; $\tau \leftarrow \tau_r \|R_{MC}\| + \tau_a$.
  **while** $\|R_{MC}\| > \tau$ **do**
    Compute $J(u, N_{MC}^J)$
    Find $s$ which satisfies $\|J(u, N_{MC}^J)s + \tilde{F}(u, N_{MC})\| \le \eta \|R_{MC}\|$ with $0 \le \eta \le \hat{\eta}$
    $u \leftarrow u + s$
    Evaluate $R_{MC} = \tilde{F}(u, N_{MC})$;
    $N_{MC} \leftarrow N_{inc} N_{MC}$
  **end while**
```

The tracking result will follow from Corollary 2.3. We will use the terminology from § 2.2.

THEOREM 4.1. *Let the assumptions of Theorems 2.1 and 2.2 hold. Let a positive integer K, $r \in (r_{Newton}, 1)$ and $\omega \in (0,1)$ be given. Then there are $\hat{\eta}$, 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*

$$\|e_n\| \le r^n \|e_0\|, \tag{4.3}$$

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

$$\|F(u_n)\| \le K_F r^n \|F(u_0)\|. \tag{4.4}$$

Proof. We will prove (4.3). After that, (4.4) will follow from (2.6) and

$$K_F = \frac{5\kappa(F'(u^*))}{3}.$$

We will use Corollary 2.3. Let $\bar{\eta}$ be the bound from the corollary and let $0 \le \hat{\eta} \le \bar{\eta}$. Now let $\bar{\Delta}$ be as in Corollary 2.3 and set

$$\delta = 1 - (1-\omega)^{1/2K}. \tag{4.5}$$

If we require that

$$N^J_{MC} \ge \left(\frac{c_J(\delta)}{\bar{\Delta}}\right)^2. \tag{4.6}$$

then the choice of N^J_{MC} implies that

$$\|F'(u_n) - J(u_n, N^J_{MC})\| \le \bar{\Delta} \tag{4.7}$$

for $0 \le n \le K-1$ with probability no smaller than

$$(1-\delta)^K = \sqrt{1-\omega},$$

provided $u_n \in \mathcal{B}(\rho)$, which will follow from our completion of the proof.

The next step is to manage N_{MC} and N_{inc}. Let ϵ_0 be as in Corollary 2.3 and let

$$\epsilon_n = \epsilon_0 r^n. \tag{4.8}$$

Let N^0_{MC} be the initial value of N_{MC} and assume

$$N^0_{MC} \ge \left(\frac{c_F(\delta)}{\epsilon_0}\right)^2. \tag{4.9}$$

Then, with probability at least $1-\delta$

$$\|F(u_0) - \tilde{F}(u_0, N^0_{MC})\| \le \epsilon_0.$$

This completes the proof if $N_{inc} \ge r^{-2}$ as then,

$$\|F(u_n) - \tilde{F}(u_n, N^n_{MC})\| \le \epsilon_0 r^n, \tag{4.10}$$

for all $0 \le n \le K-1$. Hence, with probability no less than $(1-\delta)^K$, $u_n \in \mathcal{B}(\rho)$ for $0 \le n \le K$ by (4.3), (4.7), and Corollary 2.3.

We complete the proof by noting that both (4.10) and (4.7) hold with probability no less than

$$(1-\delta)^{2K} = (1-\omega).$$

□

4.2. Matrix-Free Newton-GMRES Solvers. In this section we apply Corollary 2.5 in the context of MC approximations to the residual and Jacobian-vector product. We implement a GMRES solver for the linear equation for the nth Newton step by using the approximate function $\tilde{F}(u_n, N_{MC}^n)$ for the right hand side and approximating the Jacobian-vector product $F'(u_n)v$ with $J_p(u, v, N_{MC}^J)$. As was the case in § 4.1, we must manage the number of trials for the residual computation N_{MC}^n as the iteration progresses, but the number of trials for the Jacobian-vector product need not be increased.

Low accuracy matrix-vector products and their effects on GMRES have been studied previously [22–24]. Those papers show the errors in the early matrix-vector products can accumulate and lead to severe loss of accuracy in the solution which GMRES returns. The accumulation of errors is especially problematic if we take many GMRES iterations, as one might for a poorly conditioned problem or when one wants a significant reduction in the residual norm.

In the context of a Newton-GMRES iteration, this loss of accuracy may result in the nonlinear residual norm's not decreasing from one Newton-GMRES iteration to the next, and is especially likely to be the cause of such residual norm behavior when the iterates are near the solution. We saw such behavior in the computations reported in [26] and had to apply a line-search [3, 9, 13] to avoid increasing N_{MC} too rapidly. The authors of [22–24] recommend that one use higher accuracy matrix-vector products early in the Krylov iteration, but that is not practical in the problems we consider here.

The meaning of Assumption 2.2 is that the number of Krylov iterations for each Newton iteration is uniformly bounded. In practice, in view of the effects of error propagation, that bound should be small, as it is in our examples. This bound enables one to prove a tracking theorem because then one can derive a bound on the number of MC residuals and Jacobian-vector products that one will need for K nonlinear iterations.

To state the tracking result, Theorem 4.2, we will formulate a JFNK iteration that explicitly bounds the number of Krylov iterations and restarts. While such an iteration will not be a general purpose method, it will be effective on sufficiently well-conditioned problems. To that end, as we said above, we constrain the number K_L of GMRES vectors we are willing to store, *i. e.* to use GMRES(K_L) rather than full GMRES, and limit the number of restarts to K_R. Algorithm **JFNK-MC** combines these limits with the increments in N_{MC} from Algorithm **Newton-MC**.

```
**JFNK-MC**($u, N_{MC}, N_{MC}^J, N_{inc}, \eta, K_L, K_R, \tau_r, \tau_a, I_{max}$)
  Evaluate $R_{MC} = \tilde{F}(u, N_{MC})$; $\tau \leftarrow \tau_r \|R_{MC}\| + \tau_a$.
  **while** $\|R_{MC}\| > \tau$ **do**
    $s = GMRES(J_p(u_c, \cdot, N_{MC}^J), -\tilde{F}_c, K_L, K_R, \eta)$
    $u \leftarrow u + s$
    $N_{MC} \leftarrow N_{inc} * N_{MC}$;
    Evaluate $R_{MC} = \tilde{F}(u, N_{MC})$
  **end while**
```

Theorem 4.2. *Assume that the assumptions of Theorem 2.2 and Corollary 2.5 hold. Let* $u_0 \in \mathcal{B}$, *a positive integer* K, $r \in (r_{GMRES}, 1)$ *and* $\omega \in (0, 1)$ *be given. Then there are* N_{MC}, N_{MC}^J, *and* N_{inc}, *such that with probability* $(1 - \omega)$ *the iteration produced by Algorithm* **JFNK-MC** *satisfies* (4.3) *and* (4.4) *for* $0 \le n \le K$.

Proof. The ideas in the proof are similar to that of Theorem 4.1. We use continuity of the iteration in its data and the fact that we do at most $(K_L K_R + 1)K$ MC residual or Jacobian-vector product evaluations. The difference is the type of data. Here the nonlinear iterations are functions of K residual evaluations and at most $K_L K_R$ Jacobian vector products. The continuity implies that if the residuals and Jacobian-vector products are sufficiently accurate, then at most one additional Jacobian-vector product after the final restart will be needed to satisfy the inexact Newton condition (2.10).

So let

$$\delta = 1 - (1 - \omega)^{1/(K[K_L K_R + 1])}. \tag{4.11}$$

Let Δ_p and ϵ_0 come from Corollary 2.5 and choose N_{MC}^0 to satisfy (4.9), $N_{inc} \geq r^{-2}$, and

$$N_{MC}^J \geq \left(\frac{c_{Jv}(\delta)}{\Delta_p} \right)^2. \tag{4.12}$$

For $1 \leq k \leq K$ we will show that, with probability $(1 - \delta)^{m_k}$ that

- $u_k \in \mathcal{B}(\rho)$, so we can attempt the next iteration and
- $$\|F'(u_k)v_j - J(u_k, v_j, N_{MC}^J)\| \leq \bar{\Delta}_p \|\tilde{F}_k\|, \tag{4.13}$$

  so the next GMRES iteration will be accurate enough to invoke (2.25) and Lemma 2.4 and take the next Newton step.

We will proceed by induction. For $k = 0$ $u_0 \in \mathcal{B}(\rho)$ by assumption. From that we conclude that, with probability $(1-\delta)$, (4.10) holds. To take the next Newton step we need (4.13) to hold for all of the Krylov vectors. Since there are at most $K_L K_R$ such vectors, our choice of N_{MC}^J implies that (4.13) holds for all the Krylov vectors with probability $(1 - \delta)^{K_L K_R}$. Hence the assumptions of Corollary 2.5 hold for $K = 1$ and $u_1 \in \mathcal{B}(\rho)$ with probability $(1 - \delta)^{K_L K_R + 1}$.

Continuing the induction, we see that if $u_k \in \mathcal{B}(\rho)$ with probability $(1 - \delta)^{m_k}$, then we can take the next inexact Newton step and apply Corollary 2.5 if (4.13) holds. Since (4.13) holds for all the Krylov vectors, we have

$$m_{k+1} = m_k + K_L K_R + 1.$$

Therefore, setting $k = K - 1$ we have that the assumptions of Corollary 2.5 hold with probability no less than

$$(1 - \delta)^{K(K_L K_R + 1)} = 1 - \omega.$$

□

5. Numerical Results.

In this section we apply Algorithm `JFNK-MC` with a variety of choices of the algorithmic parameters η, K_L and K_R to an example from [26]. One should keep in mind that restarting GMRES(K_L) has a very low incremental cost if, as is the case here, N_{MC}^J is fixed a low value while N_{MC} increases throughout the iteration. One conclusion from the testing here is that restarting once ($K_R = 2$) does no harm, helps one keep both η large and K_L small, and can improve the results in some cases. On the other hand, a value of η that is too small combined with a limit K_L that is too large can lead to the types of errors that were analyzed in [22–24] in the linear iteration.

As an example we use the nonlinear system for the nonlinear diffusion acceleration (NDA) of the equation for neutron transport in one space dimension. We refer the reader to [17, 26] for the motivation for and the derivation of these equations. We will describe the continuous form of the equations and not discuss the details of the discretizations.

The linear equation for the Newton step requires preconditioning before the assumptions of the theory in § 4 hold. We will describe that preconditioner below and explain how it compactifies the linearized operator.

The NDA equation is for a "low-order" flux $\phi \in C[0, L]$. The coefficients and boundary conditions for the low-order equation depend on a "high-order" equation, which we will solve with a MC approximation.

The low-order equation is

$$\frac{d}{dx}\left[-\frac{1}{3\Sigma_t}\frac{d\phi}{dx} + \hat{D}^{HO}\phi\right] + (\Sigma_t - \Sigma_s)\phi = q(x). \tag{5.1}$$

In (5.1), Σ_t and Σ_s are transmission and scattering cross sections and q is a source term.

The coefficient $\hat{D}$ depends on the solution of the high-order equation

$$\mu\frac{\partial\psi}{\partial x} + \Sigma_t\psi(x, \mu) = \frac{1}{2}\left[\Sigma_s\phi(x) + q(x)\right], \tag{5.2}$$

where $\mu \in [-1, 1]$ is the angular variable. The boundary conditions for the high-order equation are the incoming fluxes $\psi(0, \mu)$ and $\psi(L, -\mu)$ for $\mu > 0$.

We compute $\hat{D}$ using the high-order flux

$$\phi^{HO}(x) = \int_{-1}^{1}\psi(x, \mu')d\mu', \tag{5.3}$$

and high-order current

$$J^{HO}(x) = \int_{-1}^{1}\psi(x, \mu')\mu' d\mu', \tag{5.4}$$

with the formula

$$\hat{D} = \frac{J^{HO} + \frac{1}{3\Sigma_t}\frac{d\phi^{HO}}{dx}}{\phi^{HO}}. \tag{5.5}$$

We can represent this problem as a nonlinear system of equations by writing

$$F(\phi) = \frac{d}{dx}\left[-\frac{1}{3\Sigma_t}\frac{d\phi}{dx} + \hat{D}^{HO}(\phi)\phi\right] + (\Sigma_t - \Sigma_s)\phi - q. \tag{5.6}$$

We write $\hat{D}^{HO}(\phi)$ to demonstrate the dependence of $\hat{D}^{HO}$ on ϕ as is seen in (5.5), in which ϕ^{HO} and J^{HO} are recovered from the solution to (5.2). Now, we employ a Newton-GMRES algorithm to solve $F(\phi) = 0$. This algorithm for solving the transport equation is known as JFNK-NDA(MC) when the computation of $\hat{D}$ employs a Monte Carlo simulation.

Within our Newton-GMRES algorithm there are several parameters which we may change in order to tune the performance of JFNK-NDA(MC). First, we may change K_L the maximum number of Krylov vectors allowed per linear iteration. We used $K_L = 5, 10, 20$ in our testing. Secondly, we may change the forcing term, η, which for these tests, takes on values .1 and .001. Lastly, we also look at the possibility of restarting GMRES ($K_R = 1$ or $K_R = 2$). We will see that restarting GMRES(K_L) once ($K_R = 2$) can reduce the storage requirements, as compared to doubling K_L and not restarting, while not degrading the performance of the algorithm.

Table 5.1: Problem Data

Parameter	Value
Σ_t	10
Σ_s	9.9
τ	1
q	.5
Spatial Cells	50

We will consider a single test problem which is representative, in general, of problems for which JFNK-NDA(MC) has been employed. In this computation we fix $N_{MC}^J = 10^6$ and use the zero function as the initial iterate. We present the problem data in Table 5.1.

In each of the following figures we employ the same structure. On the y-axis we plot the nonlinear residual, on the x-axis we plot the cumulative number of realizations (particle histories) for residual and Jacobian-vector products. We plot the results of ten simulations along with a dashed line demonstrating a rate of residual decrease that tracks $\frac{1}{\sqrt{N_{MC}}}$. We initialized $N_{MC} = N_{MC}^J = 10^6$, held N_{MC}^J constant for the entire iteration, and increased N_{MC} by a factor of $N_{inc} = 2$ after each nonlinear iteration.

We configured the solver to respond to a failure of the linear solver to satisfy the inexact Newton condition by accepting the step anyway and continuing the nonlinear iteration. The experiments show that there is little change in performance if one saves storage, while keeping the number of Jacobian-vector products the same, by setting $K_R = 2$ and reducing K_L by a factor of two.

We begin with a tight ($\eta = .001$) tolerance on the linear solver and a limit of 20 Jacobian-vector projects. In Figures 5.1 and 5.2 the overall performance of the two nonlinear iterations is the same. In this set of experiments the linear solver failed seven times over the ten simulations for $K_L = 10$ and $K_R = 2$ and never failed for $K_L = 20$ and $K_R = 1$. One would expect that the larger dimension for the Krylov subspace would lead to fewer failures. However, the failures of the linear solver did not affect the overall performance of the nonlinear iteration.

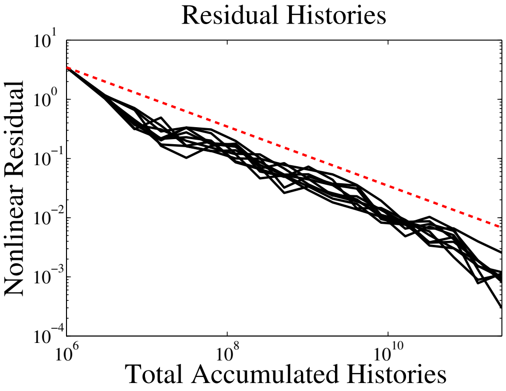

Fig. 5.1: $K_L = 10$, $\eta = .001$, $K_R = 2$

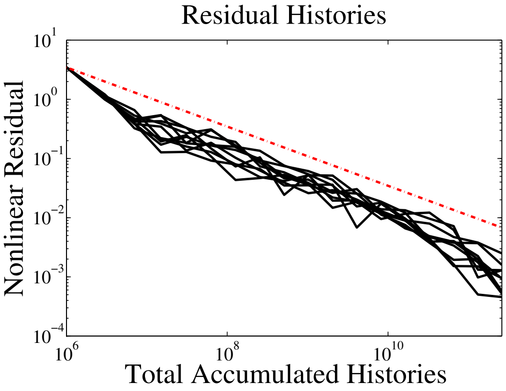

Fig. 5.2: $K_L = 20$, $\eta = .001$, $K_R = 1$

Next we let $\eta = .1$ with 10 Jacobian-vector products. We report the convergence results in Figures 5.3 and 5.4. For the ten simulations and the case $K_R = 5, K_L = 2$, the linear solver failed to converge a total of 102 times for the first pass. After restart, again for the entire suite of ten simulations, we recorded 66 failures. Larger forcing terms need fewer Krylov iterations for each Newton step, but could require more nonlinear iterations. However, when one measures cost in terms of the accumulated particle histories the cost of the entire iteration is roughly the same as in the $\eta = .001$ case.

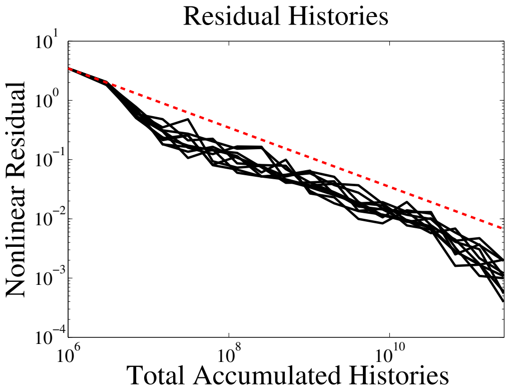

Fig. 5.3: $K_L = 5$, $\eta = .1$, $K_R = 2$

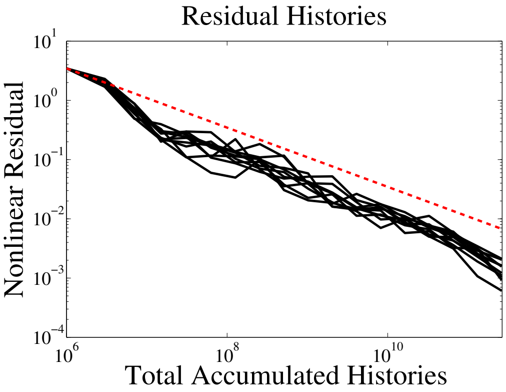

Fig. 5.4: $K_L = 10$, $\eta = .1$, $K_R = 1$

All of the figures in this section demonstrate that we can track r-linear convergence by properly increasing the number of particles (realizations) per function evaluation.

6. Conclusions. In this paper we propose and analyze an inexact Newton algorithm for problems in which residuals, Jacobians, and Jacobian-vector products are approximated by a Monte Carlo simulation. For such problems, one may think of a call to a residual as performing an experiment which does not give reproducible results. We prove results that show how the iteration tracks an idealized inexact Newton iteration based on exact residuals, Jacobians, and Jacobian-vector products.

We report on a set of numerical experiments which illustrate the analysis and show how the theory can provide guidance for an efficient implementation.

Acknowledgments. The authors are grateful for the efforts of two very knowledgeable and thorough referees.

REFERENCES

[1] E L Allgower and K Böhmer, *Application of the mesh independence principle to mesh refinement strategies*, SIAM J. Numer. Anal., 24 (1987), pp. 1335–1351.

[2] E L Allgower, K Böhmer, F A Potra, and W C Rheinboldt, *A mesh-independence principle for operator equations and their discretizations*, SIAM J. Numer. Anal., 23 (1986), pp. 160–169.

[3] L Armijo, *Minimization of functions having Lipschitz-continuous first partial derivatives*, Pacific J. Math., 16 (1966), pp. 1–3.

[4] P N Brown, *A local convergence theory for combined inexact–newton/ finite–difference projection methods*, SIAM J. Numer. Anal., 24 (1987), pp. 407–434.

[5] P. N. Brown and Y. Saad, *Convergence theory of nonlinear Newton-Krylov algorithms*, SIAM J. Optim., 4 (1994), pp. 297–330.

[6] P. N. Brown, H. F. Walker, R. Wasyk, and C. S. Woodward, *On using approximate finite differences in matrix-free Newton-Krylov methods*, SIAM J. Numer. Anal., 46 (2008), pp. 1892–1911.

[7] R. G. Carter, *Numerical experience with a class of algorithms for nonlinear optimization using inexact function and gradient information*, SIAM J. Sci. Comp., 14 (1993), pp. 368–388.

[8] R S Dembo, S C Eisenstat, and T Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[9] J E Dennis and R B Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, no. 16 in Classics in Applied Mathematics, SIAM, Philadelphia, 1996.

[10] P. Deuflhard, *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*, vol. 35 of Computational Mathematics, Springer, Berlin, 2004.

[11] W. R. Ferng and C. T. Kelley, *Mesh independence of matrix-free methods for path following*, SIAM J. Sci. Comp., 21 (2000), pp. 1835–1850.

[12] L.V. Kantorovich and G.P. Akilov, *Functional Analysis*, Pergamon Press, New York, second ed., 1982.

[13] C T Kelley, *Iterative Methods for Linear and Nonlinear Equations*, no. 16 in Frontiers in Applied Mathematics, SIAM, Philadelphia, 1995.

[14] ———, *Solving Nonlinear Equations with Newton's Method*, no. 1 in Fundamentals of Algorithms, SIAM, Philadelphia, 2003.

[15] C. T. Kelley and E. W. Sachs, *Fast algorithms for compact fixed point problems with inexact function evaluations*, SIAM J. Sci. Stat. Comp., 12 (1991), pp. 725–742.

[16] ———, *Mesh independence of Newton-like methods for infinite dimensional problems*, J. Int. Eq. Appl., 3 (1991), pp. 549–573.

[17] D A Knoll, H Park, and K Smith, *Application of the Jacobian-free Newton-Krylov method to nonlinear acceleration of transport source iteration in slab geometry*, Nuclear Sci. Eng., 167 (2011), pp. 122–132.

[18] J M Ortega and W C Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[19] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice Hall, Englewood Cliffs, 1980.

[20] Y Saad and M H Schultz, *GMRES a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comp., 7 (1986), pp. 856–869.

[21] A Shapiro, D Dentcheva, and A Ruszczyński, *Lectures on Stochastic Programming*, MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2009.

[22] Valeria Simoncini and DB Szyld, *Flexible inner-outer Kylov subspace methods*, SIAM J. Numer. Anal., 40 (2003), pp. 2219–2239.

[23] Valeria Simoncini and Daniel B Szyld, *Theory of inexact Krylov subspace methods and applications to scientific computing*, SIAM J. Sci. Comp., 25 (2003), pp. 454–477.

[24] Valeria Simoncini and Daniel B. Szyld, *Recent computational developments in Krylov subspace methods for linear systems*, Numer. Linear Algebra Appl., 14 (2007), pp. 1–59.

[25] H. F. Walker, *Implementation of the GMRES method using Householder transformations*, SIAM J. Sci. Stat. Comp., 9 (1989), pp. 815–825.

[26] Jeff Willert, C. T. Kelley, D. A. Knoll, and H. K. Park, *Hybrid deterministic/Monte Carlo neutronics*, SIAM J. Sci. Comp., 35 (2013), pp. S62–S83.