Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

PETER DEUFLHARD ULRICH NOWAK

MARTIN WEISER

Affine Invariant Adaptive Newton Codes for Discretized PDEs

ZIB-Report 02-33 (October 2002)

Affine Invariant Adaptive Newton Codes for Discretized PDEs*

Peter Deuflhard, Ulrich Nowak, and Martin Weiser

Abstract

The paper deals with three different Newton algorithms that have recently been worked out in the general frame of affine invariance. Of particular interest is their performance in the numerical solution of discretized boundary value problems (BVPs) for nonlinear partial differential equations (PDEs). Exact Newton methods, where the arising linear systems are solved by direct elimination, and inexact Newton methods, where an inner iteration is used instead, are synoptically presented, both in affine invariant convergence theory and in numerical experiments. The three types of algorithms are: (a) affine covariant (formerly just called affine invariant) Newton algorithms, oriented toward the iterative errors, (b) affine contravariant Newton algorithms, based on iterative residual norms, and (c) affine conjugate Newton algorithms for convex optimization problems and discrete nonlinear elliptic PDEs.

AMS MSC 2000: 65H10, 65H20

Keywords: affine invariant Newton methods, global Newton methods, inexact Newton methods, adaptive trust region methods, nonlinear partial differential equations, discretized partial differential equations

---

*Invited paper presented at 'AspenWorld2002, The International Conference for Process Industry Leaders', Washington

Contents

Introduction

The present paper deals with the numerical solution of boundary value problems (BVPs) for nonlinear partial differential equations (PDEs). Typically, the situation in *industrial technology* is characterized by the fact that grid generation is decoupled from the actual solution process. In this setting, nonlinear PDEs arise as a discrete system of nonlinear equations with fixed finite, but usually high dimension n and large sparse Jacobian (n, n)-matrix.

Among the possible numerical approaches to tackle such problems we here focus on affine invariant adaptive Newton methods. The theoretical convergence analysis of such methods is systematically elaborated in a forthcoming monograph [10] – there for nonlinear problems from finite dimension up to infinite dimension in function space. In the present paper we restrict our attention to Newton methods for finite dimension, and deliberately omit function space oriented Newton multigrid methods. We report about exact as well as inexact Newton codes and their comparative performance in solving discretized nonlinear PDEs. For *exact* Newton methods, which require the direct solution of arising linear subsystems, adaptivity shows up through affine invariant trust region (or damping) strategies, which are descendants of former adaptive damping strategies suggested in [6, 3] and realized in the quite popular code family NLEQ [19]. For *inexact* Newton methods, which are combined with some inner iterative solver, adaptivity additionally includes an accuracy matching between inner and outer iteration; the subtle convergence analysis presented in [10] leads to a family of new versions of the former code GIANT [7, 18] whose name is an acronym of **G**lobal **I**nexact **A**ffine invariant **N**ewton **T**echniques.

In Section 1, we characterize three affine invariance concepts and their corresponding Newton codes. *Affine covariance*, which has hitherto simply been called affine invariance (e.g., in [5, 6, 12]), leads to the *error oriented* exact Newton code NLEQ–ERR as well as its inexact counterparts GIANT–CGNE and GIANT–GBIT. *Affine contravariance*, a concept that has first been suggested by Hohmann [17], leads to the *residual based* Newton codes NLEQ–RES and GIANT–GMRES. *Affine conjugacy*, suggested quite recently in [14, 15], applies to *convex optimization* problems; the corresponding exact Newton code is NLEQ–OPT, its inexact variant is GIANT–PCG. In Section 2, the performance of the latter codes is compared at a test set of discretized elliptic PDEs. In Section 3, the performance of the residual based codes NLEQ–RES and GIANT–GMRES versus the error oriented codes NLEQ–ERR, GIANT–CGNE, and GIANT–GBIT is investigated in detail at a common test set due to [18]. In each of the inexact Newton codes left or right preconditioning is indicated by /L or /R, respectively.

1 Affine Invariant Newton Algorithms

Consider a system of n nonlinear equations, say

$$F(x) = 0 \ .$$

Suppose we have a starting guess x^0 of an unknown solution x^* at hand. Then successive linearization leads to the *ordinary* Newton method

$$F'(x^k)\Delta x^k = -F(x^k) \ , \quad x^{k+1} = x^k + \Delta x^k \ , \qquad k = 0, 1, \ldots \ . \tag{1}$$

In this paper we focus on the whole class of nonlinear systems

$$G(y) = AF(By) = 0 \ , \quad y = B^{-1}x \ ,$$

which is generated by pre– and postmultiplication with arbitrary *nonsingular* (n, n)-matrices A and B. The ordinary Newton method applied to $G(y) = 0$ would read

$$G'(y^k)\Delta y^k = -G(y^k) \ , \quad y^{k+1} = y^k + \Delta y^k \ , \qquad k = 0, 1, \ldots \ .$$

With the relation

$$G'(y^k) = AF'(x^k)B$$

and a transformed starting guess $y^0 = B^{-1}x^0$ we immediately obtain

$$y^k = B^{-1}x^k \ , \qquad k = 0, 1, \ldots \ .$$

It is only natural to require that affine invariance should be inherited to both the Newton algorithms and their convergence analysis. The simultaneous inheritance of the full invariance in terms of arbitrary A and B appears to be impossible. However, depending on the problem context, four special invariances may be preserved:

- The iterates are invariant under transformation (by A) of the image space – a property called *affine covariance* (Section 1.1).
- The iterates transform (by B) just as the domain space as a whole – a property called *affine contravariance* (Section 1.2).
- For discretized nonlinear elliptic PDEs, which may be viewed as convex optimization problems, the connection $A = B^T$ is natural – a property called *affine conjugacy* (Section 1.3).
- *Affine similarity* with $A = B^{-1}$, a property that shows up in time dependent problems, is not treated here; the interested reader may, however, look up the recent report [9] on adaptive pseudo-transient continuation methods.

Scaling invariance. The scaling or re–gauging of variables (say, from mm to km) is a special affine transformation given into the hands of the user. In actual computation, this issue deserves careful consideration.

Exact Newton methods. For moderate size up to large systems, we discuss global Newton methods in the form

$$F'(x^k)\Delta x^k = -F(x^k)\ , \quad x^{k+1} = x^k + \lambda_k \Delta x^k\ , \qquad k = 0, 1, \dots\ , \tag{2}$$

with damping factors in the range $0 < \lambda_k \leq 1$ and direct (sparse) elimination of the arising linear systems. For the different affine invariance classes, different damping strategies are suggested.

Inexact Newton methods. For very large systems, we discuss global Newton methods characterized by

$$F'(x^k)\delta x^k = -F(x^k) + r^k\ , \quad x^{k+1} = x^k + \lambda_k \delta x^k\ , \qquad k = 0, 1, \dots\ , \tag{3}$$

where the linear systems are solved by some inner iteration, hence we should more precisely write δx_i^k and r_i^k with some inner iteration index i. Again, the damping factors vary in the range $0 < \lambda_k \leq 1$.

Matching strategies. As an additional adaptivity device, the inner and outer iteration errors need to be controlled such that the deviation between the exact and the inexact Newton method is 'sufficiently small'. Upon combining (2) and (3) we obtain

$$F'(x^k)\left(\delta x^k - \Delta x^k\right) = r^k\ .$$

From this we see that we have several options to 'measure' this deviation: via some norm of the *residuals* r^k or of the *errors* $\delta x^k - \Delta x^k$ – depending on the affine invariance class. Throughout the paper $\|\cdot\|$ will mean the Euclidean vector norm.

Preconditioning. In order to possibly speed up the *inner* iteration within each outer Newton iteration step, preconditioning is often advisable. Such a device will not only influence the convergence rate of the iterative linear solver, but may also change the deviation measure. To fix the notation, we here add the residual equation in the form

$$C_L F'(x^k) C_R C_R^{-1}\left(\delta x^k - \Delta x^k\right) = C_L r^k\ , \tag{4}$$

where the nonsingular matrices C_L and C_R characterize the left and right preconditioner. The issue of left versus right preconditioning will be discussed separately for each of the affine invariance classes.

1.1 Affine covariant Newton algorithms

Affine covariance has originally been the only invariance systematically exploited for adaptive Newton algorithms – formerly called 'affine invariant Newton methods' [5, 6]. In this setting, we keep the domain space of F fixed ($B = I$) and look at the problem class

$$G(x) = AF(x) = 0 \ .$$

As discussed above, the Newton iterates are the same for all A. However, upon revisiting the standard local convergence theorems like the Newton–Kantorovich or the Newton–Mysovskikh theorem, we stumble over assumptions of the kind

$$\|G'(x^0)^{-1}\| \leq \beta(A) \ , \quad \|G'(x) - G'(\bar{x})\| \leq \gamma(A)\|x - \bar{x}\| \ ,$$

which give rise to a local convergence ball with radius

$$\rho(A) \sim \frac{1}{\beta(A)\gamma(A)} \ .$$

Assuming best possible theoretical estimates (hard to get anyway) we obtain

$$\beta(A) \leq \beta(I)\|A^{-1}\| \ , \quad \gamma(A) \leq \gamma(I)\|A\|$$

and therefore, with $\mathrm{cond}(A) = \|A\| \cdot \|A^{-1}\|$, a possible worst case situation

$$\rho(A) \sim \frac{\rho(I)}{\mathrm{cond}(A)} \ . \tag{5}$$

Obviously, by a mean choice of A we may 'shrink the baby' to arbitrarily small size! Fortunately, careful examination of the classical theorems shows that a telescoped Lipschitz condition such as

$$\|F'(x^0)^{-1}\big(F'(x) - F'(\bar{x})\big)\| \leq \omega\|x - \bar{x}\| \ , \tag{6}$$

will do as well. Both this Lipschitz condition and the thus defined Lipschitz constant ω are affine covariant, since

$$\begin{aligned} G'(x^0)^{-1}\big(G'(x) - G'(\bar{x})\big) &= \big(AF'(x^0)\big)^{-1}A\big(F'(x) - F'(\bar{x})\big) \\ &= F'(x^0)^{-1}\big(F'(x) - F'(\bar{x})\big) \ , \end{aligned} \tag{7}$$

so that both sides of (6) are independent of A.

This change of assumption allows a clean affine covariant convergence theory which leads to results in terms of *iterates* $\{x^k\}$, *correction norms* $\|\Delta x^k\|$ or *error norms* $\|x^k - x^*\|$ and eventually to adaptive *error oriented* Newton algorithms [10].

Of course, the formal assumption $B = I$ covers any *fixed scaling* transformation of the type $B = D$. In fact, 'dimensionless' variables

$$y = D^{-1}x \ , D = \mathrm{diag}(\alpha_1, \ldots, \alpha_n) \ , \alpha_i > 0$$

are typically used inside our codes. Whenever the quantities α_i are chosen in some scaling dependent way, then the variable y is scaling invariant.

Exact Newton methods. In the affine covariant setting, a new iterate x^{k+1} is accepted, if the (so–called natural) monotonicity test

$$\Theta_k(\lambda_k) = \frac{\|\overline{\Delta x}^{k+1}\|}{\|\Delta x^k\|} \leq 1 \tag{8}$$

holds, where the simplified Newton corrections $\overline{\Delta x}^{k+1}$ are defined via

$$F'(x^k)\overline{\Delta x}^{k+1} = -F(x^k + \lambda_k \Delta x^k) \ . \tag{9}$$

These additional linear systems are cheap to solve, since the matrix decompositions can be kept from the computation of the ordinary Newton corrections Δx^k. Local convergence analysis [10] shows that

$$\Theta_k(\lambda) \leq 1 - \lambda + \frac{1}{2}\lambda^2 h_k \tag{10}$$

in terms of the affine covariant Kantorovich quantities

$$h_k = \omega \|\Delta x^k\| \ . \tag{11}$$

From the local upper bound (10), the following damping factor would be optimal

$$\lambda_{\mathrm{opt}} = \min\left(1, \frac{1}{h_k}\right) \ . \tag{12}$$

The basic *paradigm* now is to replace the computationally unavailable theoretical Lipschitz constant ω by a computationally available local estimate $[\omega] \leq \omega$, both of them being affine covariant. This gives rise to computational Kantorovich quantities

$$[h_k] = [\omega]\|\Delta x^k\| \leq h_k \tag{13}$$

and, in turn, to estimated locally optimal damping factors

$$[\lambda_{\mathrm{opt}}] = \min\left(1, \frac{1}{[h_k]}\right) \geq \lambda_{\mathrm{opt}} \ . \tag{14}$$

Computational a-priori estimates then lead to some *prediction strategy* for λ_k^0. If the monotonicity test (8) fails for a chosen damping factor, then new a–posteriori estimates lead to a *correction strategy* that involves the theoretically backed selection of 'better' damping factors λ_k^i , $i = 1, \ldots$. The precise formulas in [10] are omitted here, they are a slight modification of the results published in [6] and [3]. If the computational estimates catch at least *one binary digit* of the true theoretical values, then the following *restricted* monotonicity test should hold:

$$\Theta_k(\lambda_k) = \frac{\|\overline{\Delta x}^{k+1}\|}{\|\Delta x^k\|} \leq 1 - \frac{1}{4}\lambda_k \; . \tag{15}$$

This test is used in all our numerical experiments in Section 3.2. Experience shows that the correction strategy is rarely activated; only when a critical point with singular Jacobian is in the neighborhood of the Newton sequence, then repeated reductions may occur, which we then terminate via some threshold condition $\lambda_k \geq \lambda_{\min}$. In our numerical experiments below, the term 'λ-fail' indicates a violation of this criterion.

The just described algorithmic structure is implemented in the affine covariant, error oriented Newton code NLEQ-ERR. Unlike in its quite popular predecessor NLEQ1, we do not apply any intermediate quasi–Newton steps in the present problem setting of discrete PDEs.

Inexact Newton methods. In the affine covariant setting, the inexact Newton method will naturally be combined with the iterative linear solver CGNE, known to *minimize* $\|\delta x_i^k - \Delta x_i^k\|$ over some Krylov space for successive index $i = 0, 1, \ldots$ – for details see, e.g., the textbook of Saad [21]. Therefore we naturally measure the residual error via the characterizing quantity

$$\delta_k = \frac{\|\delta x^k - \Delta x^k\|}{\|\delta x^k\|} \; , \tag{16}$$

wherein we dropped the inner iteration index i in δx_i^k for ease of writing. Within CGNE, the value of $\|\delta x_i^k\|$ will increase and the value of $\|\delta x_i^k - \Delta x^k\|$ will decrease for increasing index i. Hence, we can asymptotically meet any criterion $\delta_k \leq \overline{\delta}$ in terms of a prescribed threshold value $\overline{\delta}$. A technique for the computational estimation of the term $\|\delta x_i^k - \Delta x^k\|$ is given in [10], similar to a suggestion for the PCG case given in [8].

On the basis of the detailed analysis given in [10] we will have to exchange the monotonicity test (8) by the *inexact* monotonicity test

$$\widetilde{\Theta}_k(\lambda_k) = \frac{\|\widetilde{\delta x}^{k+1}\|}{\|\delta x^k\|} \leq 1 \; , \tag{17}$$

where the inexact simplified Newton corrections $\widetilde{\delta x}^{k+1}$ are defined via

$$F'(x^k)\widetilde{\delta x}^{k+1} = \Big(-F(x^k + \lambda_k \delta x^k) + r^k\Big) + \widetilde{r}^{k+1} \tag{18}$$

with $\widetilde{r}^{k+1}$ denoting the inner residual obtained in the course of the corresponding CGNE iterations. From the convergence analysis in [10] we obtain suggested initial values for the inner iterations as

$$\delta x_0^{k+1} = \widetilde{\delta x}^{k+1} \ , \quad \widetilde{\delta x}_0^{k+1} = (1-\lambda_k)\delta x^k \ . \tag{19}$$

This suggestion turned out to be important for the efficiency of the corresponding codes.

As shown in [10], the above damping strategies can be modified such that, for each inner iteration index i, we can obtain affine covariant computational Lipschitz estimates

$$[h_k^\delta]_i = \omega \|\delta x_i^k\| \ ,$$

where the accuracy matching is done via the saturation property

$$[h_k^\delta]_i \leq [h_k^\delta]_{i+1} \leq h_k \ .$$

As soon as the thus estimated Kantorovich quantities are 'accurate enough', the above prediction and correction strategies can be realized.

This combined adaptive matching/damping strategy is implemented in the code GIANT–CGNE. In the ordinary Newton method the code realizes two different local convergence modes (to be selected by the user): a *linear* convergence mode, which aims at a convergence of the kind

$$\widetilde{\Theta}_k \leq \overline{\Theta} < 1 \ , \tag{20}$$

and the standard *quadratic* convergence mode.

Preconditioning. If we apply preconditioning in the form (4), then any choice of C_L only influences the convergence speed of the inner iteration. The choice of C_R, however, affects all adaptivity devices for the outer (Newton) iteration: wherever norms in domain space arise, for example $\|\delta x^k - \Delta x^k\|$, we have to insert preconditioned norms, for example $\|C_R^{-1}\left(\delta x^k - \Delta x^k\right)\|$. For this reason, we only realized left preconditioning indicated as GIANT–CGNE/L – apart from scaling.

Remark. If we replace CGNE by some other iterative solver, known to only *reduce*, but not minimize the residual error, then the above convergence analysis needs to be slightly modified. In [7, 18], the solver GBIT [11], a 'good Broyden' update technique optimized for linear systems, has been implemented in combination with the code GIANT. For reasons of compatibilty, an update of that code is renamed here as GIANT–GBIT/L.

1.2 Affine contravariant Newton algorithms

The door to *affine contravariant* Newton methods has been opened by Hohmann in his dissertation [17], wherein he exploited it for the construction of an adaptive inexact Newton-GMRES method. The affine invariance setting is dual to the preceding one: we keep the image space of F fixed ($A = I$) and consider the problem class

$$G(y) = F(By) \ , \quad y = B^{-1}x \ .$$

Consequently, a common convergence theory for this class will not lead to statements about the Newton iterates $\{x^k\}$, but only about the image space data $F(x^k)$, which are independent of B. Again, the classical assumptions can be telescoped, this time in image space:

$$\big\|\big(F'(\bar{x}) - F'(x)\big)(\bar{x} - x)\big\| \leq \omega \|F'(x)(\bar{x} - x)\|^2 \ . \tag{21}$$

Observe that both sides above are independent of B. A local convergence theorem on the basis of such an affine contravariant Lipschitz condition [10] will lead to results in terms of *residual norms* $\|F(x^k)\|$.

As in the former case, *scaling* deserves careful consideration: here it should be applied to the image space of F, i.e. to the image space components

$$F \rightarrow G = D^{-1}F$$

with appropriately chosen diagonal matrices D. For ease of writing, we will ignore these scaling matrices in the following.

Exact Newton methods. In the affine contravariant setting, a new iterate x^{k+1} is accepted, if the residual monotonicity test (also called standard convergence test)

$$\Theta_k(\lambda_k) = \frac{\|F(x^{k+1})\|}{\|F(x^k)\|} \leq 1 \tag{22}$$

holds. Local convergence analysis [10] here again shows that

$$\Theta_k(\lambda) \leq 1 - \lambda + \frac{1}{2}\lambda^2 h_k \tag{23}$$

in terms of the special Kantorovich quantities

$$h_k = \omega \|F(x^k)\| \ , \tag{24}$$

which leads to the theoretically optimal damping factor

$$\lambda_{\text{opt}} = \min\left(1, \frac{1}{h_k}\right) \ . \tag{25}$$

We again follow the basic *paradigm* and replace the computationally unavailable theoretical Lipschitz constant ω by a computationally available local estimate $[\omega] \leq \omega$, both being affine contravariant. This gives rise to computational Kantorovich quantities

$$[h_k] = [\omega] \|F(x^k)\| \leq h_k \ , \tag{26}$$

which, in turn, lead to estimated locally optimal damping factors

$$[\lambda_{\text{opt}}] = \min \left(1, \frac{1}{[h_k]} \right) \geq \lambda_{\text{opt}} \tag{27}$$

and eventually to some *prediction strategy* for λ_k^0. If the residual monotonicity test fails for an estimated damping factor, then new a–posteriori estimates are available to compute 'better' damping factors $\lambda_k^i \ , i = 1, \ldots$ in the frame of a *correction strategy*. The detailed formulas can be found in [10]. If the computational estimates catch at least *one binary digit* of the true theoretical values, then the following *restricted* monotonicity test should hold:

$$\Theta_k(\lambda_k) = \frac{\|F(x^{k+1})\|}{\|F(x^k)\|} \leq 1 - \frac{1}{4}\lambda_k \ . \tag{28}$$

The latter test is rather similar to the well–known classical Armijo test [1], only the theoretical considerations that led to it are different. This restricted test is used throughout our numerical experiments in Section 3.2.

The just described algorithmic structure is implemented in the affine contravariant, residual based Newton code NLEQ–RES.

Inexact Newton methods. In the affine contravariant setting, the inexact Newton method will naturally be combined with the iterative linear solver GMRES, known to *minimize* the iterative residual norms $\|r_i^k\|, i = 0, 1, \ldots$ – for details see, e.g., the textbook [21]. We naturally measure the deviation from the exact Newton method by quantities

$$\eta_k = \frac{\|r^k\|}{\|F(x^k)\|} \ , \tag{29}$$

wherein we dropped the iteration index i in r_i^k. For increasing i the value of η_k will decrease so that we can asymptotically meet any prescribed error criterion $\eta_k \leq \overline{\eta}$ in term of a given threshold value $\overline{\eta}$. For the thus defined inexact Newton method we obtain the theoretical result [10]

$$\Theta_k(\lambda) \leq 1 - (1 - \eta_k)\lambda + \frac{1}{2}(1 - \eta_k^2)\lambda^2 h_k \ , \tag{30}$$

which gives rise to the optimal damping factor

$$\lambda_{\text{opt}} = \min \left(1, \frac{1}{(1 + \eta_k) h_k} \right) \ , \tag{31}$$

and, in turn, to the computational estimates $[h_k]$, $[\lambda_{\text{opt}}]$ as a basis of modified prediction and correction strategies depending now on the relative residual norms η_k to be controlled adaptively such that they are 'sufficiently small'. As in all of our inexact Newton codes, a linear and a quadratic *local convergence mode* can be chosen by the user.

The just described algorithm is implemented in the affine contravariant, residual based Newton codes GIANT–GMRES. This code performs quite differently from codes based on the suggestions in [4, 2] or their implementation in the code NITSOL [20].

Preconditioning. If we apply preconditioning in the form (4), then the choice of C_R only influences the convergence speed of the inner iteration. The choice of C_L, however, also affects the adaptive control of the outer iteration: wherever norms in image space arise, for example$\|r^k\|$ or $\|F(x^k)\|$, we have to insert preconditioned norms, for example $\|C_L r^k\|$ or $\|C_L F(x^k)\|$. Left and right preconditioning will be indicated as GIANT–GMRES/L, and GIANT–GMRES/R.

1.3 Affine conjugate Newton algorithms

In [14, 15], the term 'affine conjugacy' has been coined and exploited for the construction of adaptive Newton multilevel finite element methods for nonlinear elliptic PDEs. Such PDEs are connected with an underlying convex functional to be minimized. After discretization we have to solve the corresponding discrete convex minimization problem

$$f(x) = \min \ , \quad f : D \subset \mathbb{R}^n \to \mathbb{R} \ ,$$

equivalent to solving the nonlinear equations

$$F(x) = \text{grad} f(x) = f'(x)^T = 0 \ , \qquad x \in D \ .$$

Obviously, the mapping F is a gradient mapping and its Jacobian $F'(x) = f''(x)$ is symmetric and positive semi–definite. In what follows, we will assume (and check within the algorithms to be constructed) that $F'(x)$ is even strictly positive definite so that $F'(x)^{1/2}$ is well–defined – in which case f is strictly convex. Upon transforming the minimization problem to

$$g(y) = f(By) = \min \ , \quad y = B^{-1}x \ ,$$

we arrive at the transformed equations

$$G(y) = B^T F(By) = 0 \ ,$$

and the transformed Jacobian

$$G'(y) = B^T F'(x) B \ , \quad x = By \ .$$

As can be seen, the Jacobian transformation is *conjugate* so that all G' are symmetric and strictly positive definite. Among possible affine conjugate theoretical terms are certainly any functional values $f(x)$. Moreover, since the transformation

$$u, v, x \quad \to \quad \bar{u} = Bu, \bar{v} = Bv, x = By \ ,$$

implies

$$u^T G'(y) v = \bar{u}^T F'(x) \bar{v} \ ,$$

so–called energy products

$$(u, v) = u^T F'(x) v$$

are also seen to be affine conjugate. Such inner products induce (discrete) energy norms like

$$\|F'(x)^{1/2} u\|^2 = (u, u) = u^T F'(x) u \ , \quad u, x \in D \ .$$

Telescoping the classical theoretical assumptions here leads to an affine conjugate Lipschitz condition

$$\|F'(x)^{-1/2}\big(F'(\bar{x}) - F'(x)\big)(\bar{x} - x)\| \leq \omega \|F'(x)^{1/2}(\bar{x} - x)\|^2 \ . \tag{32}$$

Any affine conjugate convergence theorems will therefore lead to results in terms of *functional values* $f(x^k)$ and *energy norms* of corrections

$$\epsilon_k = \|F'(x^k)^{1/2} \Delta x^k\|^2$$

or errors

$$\|F'(x^k)^{1/2}(x^k - x^*)\| \ .$$

By construction, the affine conjugate energy products are *scaling invariant.*

Exact Newton methods. In the affine conjugate setting, we assume that the underlying convex functional f and its gradient mapping F can both be evaluated. Then a new iterate x^{k+1} is accepted, if the functional monotonicity test

$$f(x^{k+1}) \leq f(x^k) \tag{33}$$

holds. Local convergence analysis [10] here shows that

$$f(x^k + \lambda \Delta x^k) \leq f(x^k) - \left(\lambda - \frac{1}{2}\lambda^2 - \frac{1}{6}\lambda^3 h_k\right) \epsilon_k \ , \tag{34}$$

in terms of the special Kantorovich quantities

$$h_k = \omega \|F'(x^k)^{1/2} \Delta x^k\| \ . \tag{35}$$

Here the theoretically optimal damping factor would be

$$\lambda_{\text{opt}} = \frac{2}{1 + \sqrt{1 + 2h_k}} \leq 1 \ . \tag{36}$$

We again apply the *paradigm* to replace the computationally unavailable theoretical Lipschitz constant ω by a computationally available local estimate $[\omega] \leq \omega$, both being affine conjugate. This gives rise to computational Kantorovich quantities

$$[h_k] = [\omega] \|F'(x^k)^{1/2} \Delta x^k\| \leq h_k \tag{37}$$

and, in turn, to estimated locally optimal damping factors

$$[\lambda_{\text{opt}}] = \frac{2}{1 + \sqrt{1 + 2[h_k]}} \geq \lambda_{\text{opt}} \ , \quad [\lambda_{\text{opt}}] \leq 1 \ . \tag{38}$$

In the same way as in the two other affine invariance concepts, a-priori and a-posteriori computational Lipschitz estimates then lead to a prediction and a correction strategy for the choice of the damping factors. The precise formulas in [10] are omitted here.

This algorithmic structure is implemented in the affine conjugate Newton code NLEQ-OPT for convex optimization problems.

Inexact Newton methods. In the affine conjugate setting, the inexact Newton method will naturally be combined with any preconditioned conjugate gradient method, denoted by PCG, known to *minimize* the local energy norms $\|F'(x^k)^{1/2} \left(\delta x_i^k - \Delta x^k\right) \|$ for iteration index $i = 0, 1, \ldots$ independent of the selected preconditioner. Therefore we naturally measure the deviation of exact and inexact Newton method via the characterizing quantities

$$\delta_k = \frac{\|F'(x^k)^{1/2} \left(\delta x^k - \Delta x^k\right) \|}{\|F'(x^k)^{1/2} \delta x^k\|} \ , \tag{39}$$

where again we have dropped the inner iteration index i. For increasing index i the energy norm of the correction $\|F'(x^k)^{1/2} \delta x_i^k\|$ increases and the energy norm of the deviation $\|F'(x^k)^{1/2} \left(\delta x_i^k - \Delta x^k\right) \|$ decreases so that, in total, the value of δ_k decreases. Hence, we can asymptotically meet a prescribed threshold criterion such as $\delta_k \leq \overline{\delta}$. A rather simple computational estimate of the terms $\|F'(x^k)^{1/2} \left(\delta x_i^k - \Delta x^k\right) \|$ is suggested in the paper [8] on the solution of linear elliptic PDEs.

On the basis of the analysis in [14, 15, 10] we obtain modifications of the functional behavior

$$f(x^k + \lambda \delta x^k) \leq f(x^k) - \left(\lambda - \frac{1}{2}\lambda^2 - \frac{1}{6}\lambda^3 h_k^\delta\right) \epsilon_k^\delta \ , \tag{40}$$

wherein we used the notations

$$\begin{aligned} \epsilon_k^\delta &= \|F'(x^k)^{1/2}\delta x^k\|^2 &= \frac{\epsilon_k}{1+\delta_k^2} \ , \\ h_k^\delta &= \omega\|F'(x^k)^{1/2}\delta x^k\| &= \frac{h_k}{(1+\delta_k^2)^{1/2}} \ . \end{aligned} \tag{41}$$

The theoretically optimal damping factors and all further computational estimates are then essentially obtained replacing the h_k by the h_k^δ. We will have to take care that the thus estimated local quantities are 'accurate enough', to be able to realize an efficient prediction and correction strategy for the damping factors.

The just described combined adaptive matching/damping strategy is implemented in the affine conjugate Newton code GIANT–PCG for convex optimization problems. As the other inexact Newton codes, this code also realizes either a *linear* or a *quadratic* local convergence mode – to be chosen by the user.

Preconditioning. The analysis present above is independent of any choice of preconditioner, as long as it does not change the general symmetric positive definite pattern.

2 Newton Codes for Convex Optimization

In this section we want to compare different options within our exact and inexact affine conjugate Newton codes.

2.1 Test set

This test set consists of three discretized nonlinear elliptic PDE boundary value problems in two space dimensions where the discretized functional is also at hand. All discrete PDE problems are obtained by uniform discretization using simple finite difference schemes to obtain the corresponding finite dimensional convex optimization problems. For selected moderate size meshes, certain problem characteristics are arranged synoptically in Table 1. In particular, the column with $M_{\max}$ shows the maximal nonlinearity weight factor, for which the uncontrolled ordinary Newton method converges.

Name	Grid	Dim n	$M_{\max}$
msc	32×32	1024	6.2
elas	32×32	2048	1.0
msnc	32×32	3072	1.9

Table 1: Test set characteristics for special 2D grid. $M_{\max}$ is some nonlinearity weight factor.

Of course, below we will treat much finer meshes with much larger problem dimensions (up to $n \approx 200.000$).

Example: Minimal surface problem over convex domain (msc). Given the domain $\Omega =]0,1[^2$, minimize the surface area

$$\int_\Omega (1 + |\nabla u|^2)^{\frac{1}{2}} \, dx$$

subject to the Dirichlet boundary conditions

$$u(x_1, x_2) = M(x_1 + (1 - 2x_1)x_2) \text{ on } \partial\Omega \, ,$$

The function $u(x) \in \mathbb{R}$ is the vertical position of the surface parameterized over Ω. The scaling parameter M of the boundary conditions allows to vary the 'nonlinearity' of the problem. The initial value u^0 is chosen as the bilinear interpolation of the boundary conditions. Note that the simpler choice of $u^0 = 0$ is incompatible with the boundary conditions and, hence, would introduce an artificial dependence of the initial value on the mesh size.

This problem has a unique well-defined solution depicted in Fig. 1, left.

Example: Simple elastomechanics problem (elas). Given the domain $\Omega =]0,1[^2$, minimize the total energy

$$\int_\Omega \left(\|F\|^2 + (\det F)^{-1} - M(1/2, -1)u \right) \, dx \quad \text{with } F = I + \nabla u \, ,$$

subject to the Dirichlet boundary conditions $u = 0$ on $\{0\} \times [0,1]$. On the remaining boundary part, natural boundary conditions are imposed. The function $u(x) \in \mathbb{R}^2$ is the displacement of an elastic body. The deformation energy is modeled by a particularly simple variant of an Ogden material. The volume force $(1/2, -1)^T$ acting on the body is scaled by M, which can be used to vary the 'nonlinearity' of the problem. The undeformed state

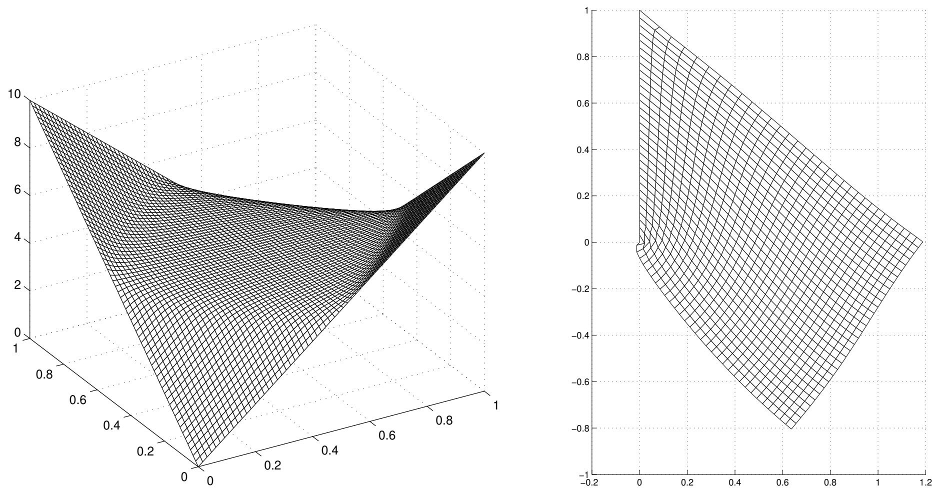

Figure 1: Left: solution of problems **msc** ($h = 1/63, M = 10$). Right: solution of problem **elas** ($h = 1/31, M = 2$).

$u^0 = 0$, compatible with the boundary conditions, is chosen as the initial value.

In this problem, the total energy functional defined via the Ogden material law is not convex on the whole domain of definition. Furtunately, it is convex in a sufficiently large neighborhood of the solution. Therefore, the Newton codes starting at u^0 did not encounter any nonpositive second derivatives.

The locally unique solution is depicted in Fig. 1, right.

Example: Minimal surface problem over nonconvex domain (msnc). Given the domain $\Omega =]0,2[^2 \backslash]1,2[^2$, minimize the surface area

$$\int_\Omega (1 + |\nabla u|^2)^{\frac{1}{2}} \, dx$$

subject to the Dirichlet boundary conditions

$$u = 0 \text{ on } [0,2] \times \{0\} \cup \{0\} \times [0,2] \,, \quad u = M \text{ on } [1,2] \times \{1\} \cup \{1\} \times [1,2] \,.$$

On the remaining boundary parts, $[0,1] \times \{2\} \cup \{2\} \times [0,1]$, homogeneous Neumann boundary conditions $\partial_n u = 0$ are imposed. The function $u(x) \in \mathbb{R}$ is the vertical position of the surface parameterized over Ω. As in problem **msc**, the scaling parameter M allows to vary the 'nonlinearity' of the problem. The initial value u^0 is chosen as the linear interpolation of the Dirichlet boundary conditions on $[0,1] \times [1,2] \cup [1,2] \times [0,1]$ and the bilinear interpolation of the thus defined boundary values on $[0,1]^2$. Again, the simpler choice $u^0 = 0$ would introduce an artificial dependence on the mesh size.

In contrast to **msc** and **elas**, this problem has been deliberately constructed such that the underlying PDE does *not* have a unique solution: indeed, function space multilevel Newton methods are able to undoubtedly detect the nonexistence of a continuous solution instead of incorrectly computing a finite dimensional pseudosolution (cf. [15]).

Nevertheless, each discretization does have a unique solution – see Fig. 2 for $M = 2$ and different mesh sizes h. Of course, this feature of the continuous problem appears in sufficiently fine discretizations. In fact, as shown in [15], the local convergence domain of Newton's method shrinks when $h \to 0$. Hence, we expect to see a clear dependence of the number of Newton iterations on the mesh size. However, since the main effect is highly localized at the corner $(1,1)^T$, this dependence is more clearly visible in the setting of adaptive discretizations.

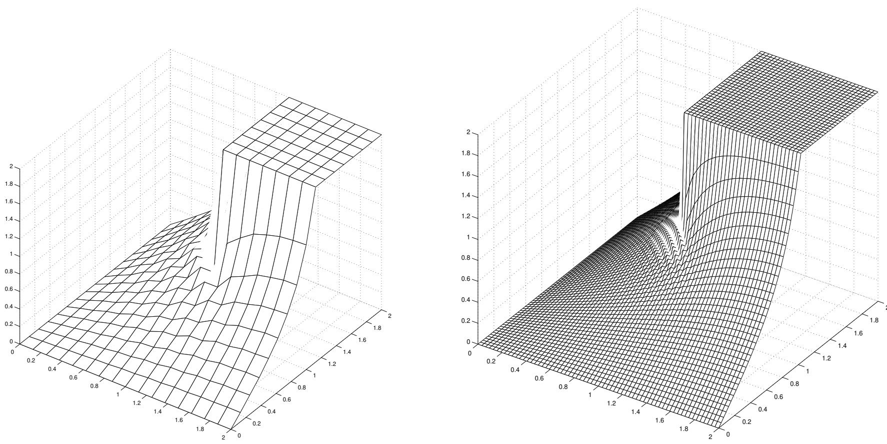

Figure 2: Two different discrete pseudosolutions of problem **msnc**$(M = 2)$ for $h = 2/7$ (left) and $h = 2/31$ (right).

2.2 Numerical experiments

In this section the adaptive Newton methods for convex optimization as sketched in Section 1.3 are compared at the test set described just above. For the exact methods, the sparse solver provided by MATLAB is used. For the inexact methods, a PCG method with energy oriented termination criterion based on [8, 10] is used. As preconditioners we tested both the Jacobi and the incomplete Cholesky preconditioner (ICC) as provided by

MATLAB (with droptol=10^{-3}). A relative energy norm tolerance of 10^{-8} is set for the Newton iteration. Fail exits in the numerical tests occurred as

- λ-fail: 'too small' damping factor $\lambda_k \leq 10^{-4}$ suggested or 'too many' damping factor reductions,
- ITMAX: more than 500 inner iterations in GIANT–PCG, and
- Θ-fail: insufficient contraction rate in linear convergence mode of GIANT–PCG.

Local versus global Newton methods. In Table 2, the numerical solution of problem **msc** is compared for different Newton algorithms and varying weight factor M. Among the local Newton methods we included the simplified Newton method, which keeps the initial Jacobian throughout the iteration – the corresponding convergence analysis is given in [10]. The inexact Newton codes use ICC within PCG and are run in the quadratic convergence mode.

	local			global	
M	simplified	exact	inexact	exact	inexact
2	21	5	5	9	9
5	DIV	7	7	10	9
10	DIV	DIV	DIV	10	10

Table 2: Problems **msc**. Number of iterations versus nonlinearity weight factor M for different Newton algorithms. DIV means divergence of local Newton algorithms.

In this example, the local Newton methods are seen to converge only for the mildly nonlinear case, among which the simplified Newton behaves worst. We observe that exact and inexact Newton methods, both local and global, realize nearly the same number of Newton iterations.

Mesh dependence of different global Newton algorithms. In this experiment we test different Newton algorithms over the whole test set for increasing mesh size. The results are arranged in Table 3.

Asymptotic mesh independence. This well–known feature (cf. [13]) is clearly visible in problem **msc**, but less in problem **elas**, since there the singularity at the origin is not sufficiently well represented on the here selected uniform grids. In problem **msnc**, the number of the Newton iterations increases with decreasing mesh size or increasing dimension of the discrete problem – as expected from [15]. For very fine meshes, failures indicate the underlying non-existence of a PDE solution.

Exact versus inexact Newton algorithms. As apparent from the missing entries in Table 3, inexact Newton codes relying on iterative solvers are able to tackle much larger problems than exact codes that use direct sparse solvers. Both time and, even more pronounced, memory requirements of the direct solver are prohibitively large for the finest meshes. The inexact Newton method is run in the quadratic convergence mode with an incomplete Cholesky preconditioner.

	msc$(M=10)$		**elas**$(M=2)$		**msnc**$(M=2)$	
N	exact	inexact	exact	inexact	exact	inexact
4	9	8	10	9	9	8
8	10	9	10	10	9	9
16	10	9	10	10	10	10
32	10	10	10	10	10	11
64	10	10		11		13
128		10				λ-fail
256		10				ITMAX

Table 3: Iteration numbers of global Newton methods versus mesh size $h = 1/(N-1)$ over test set. The dimension of each discrete problem is given in Table 1.

Different preconditioners. For local Newton algorithms, two different preconditioners are used in combination with either the linear or the quadratic local convergence modes. For the linear mode, a reduction factor of $\bar{\Theta} = 0.5$ is imposed, and, accordingly, the nonlinearity weight factor M is reduced such that the exact Newton method reaches contraction factors well below $\bar{\Theta}$ for most mesh sizes. The worst contraction factor encountered in the exact Newton method is given in the last column of Table 4. Note that for $N = 4$ both **msc** and **elas** lead to $\Theta_{\max} > \bar{\Theta}$ in the exact Newton iteration, so that the Θ-failures in the first rows are consistent with the convergence theory of [10].

As expected, the *Jacobi* preconditioner is insufficient for very fine discretizations and, hence, leads to failures in the PCG convergence. The low quality of the Jacobi preconditioner is also the reason for the Θ-failures for $N > 4$: the PCG error estimator is severely impaired and tends to stop the PCG iteration too early, thus delivering a Newton correction which is not accurate enough to reduce the energy error.

The *incomplete Cholesky* preconditioner is more effective, at least for small up to moderate size meshes. This is the reason why for such discretizations the linear convergence mode is as fast as the quadratic convergence mode: the accuracy of the inexact Newton corrections is far better than requested. For very small mesh sizes, the effectiveness of the incomplete Cholesky preconditioner decreases, and hence the number of Newton iterations in linear convergence modes becomes significantly larger than in quadratic convergence mode.

		quadratic		linear		exact
	N	ICC	Jac	ICC	Jac	$\Theta_{\max}$
msc(M=3.5)	4	7	7	Θ-fail	Θ-fail	0.51
	8	6	6	6	9	0.31
	16	6	6	6	13	0.30
	32	6	7	7	Θ-fail	0.30
	64	6	ITMAX	8	Θ-fail	
	128	6	ITMAX	12	Θ-fail	
	256	6	ITMAX	Θ-fail	ITMAX	
elas(M=0.2)	4	6	6	Θ-fail	Θ-fail	0.61
	8	5	6	6	Θ-fail	0.26
	16	5	6	7	Θ-fail	0.18
	32	5	ITMAX	8	Θ-fail	0.15

Table 4: Iteration numbers of local inexact Newton methods: quadratic versus linear local convergence mode ($\bar{\Theta} = 0.5$).

3 Residual Based versus Error Oriented Newton Codes

There is clear evidence from the numerical solution of nonlinear BVPs for ordinary differential equations (ODEs) that error oriented Newton techniques are preferable over residual based Newton techniques, both in multiple shooting and in collocation methods. The question to be answered in this section is whether this feature carries over to the PDE situation and there, in particular, to inexact Newton methods.

3.1 Common test set

We consider a subset of the test problems presented in [18] and used there for the test of the older code GIANT. All examples are discretized nonlinear

PDE problems in only two space dimensions. This leads to system dimensions n that still permit a direct solution of the arising linear equations – so that exact and inexact Newton codes can be compared. For discretization we used the usual second order, centered finite differences on tensor product grids. Neumann boundary conditions are included by simple one-sided differences, as usual.

Example: Artifical test problem (atp1). This problem comprises the simple scalar PDE

$$\Delta u - (0.9\exp(-q) + 0.1u)(4x^2 + 4y^2 - 4) - g = 0 \ ,$$

where

$$g = \exp(u) - \exp(\exp(-q)) \ \text{ and } q = x^2 + y^2 \ ,$$

and boundary conditions $u|_{\partial\Omega} = 0$ on the domain $\Omega = [-3, 3]^2$.

Its analytical solution is known to be $u(x, y) = \exp(-q)$.

Example: Driven cavity problems (dcp1000, dcp5000). This problem involves the steady stream-function/vorticity equations

$$\Delta\omega + \mathrm{Re}(\psi_x\omega_y - \psi_y\omega_x) = 0 \ , \quad \Delta\psi + \omega = 0 \ ,$$

where ψ is the stream-function and ω the vorticity. For the domain $\Omega = [0, 1]^2$ the following discrete boundary conditions are imposed

$$\begin{aligned}
&\tfrac{\partial\psi}{\partial y}(x, 1) = -16x^2(1 - x)^2 \ , \\
&\omega(x, 0) = -\tfrac{2}{\Delta y^2}\psi(x, \Delta y) \ , \\
&\omega(x, 1) = -\tfrac{2}{\Delta y^2}[\psi(x, 1 - \Delta y) + \Delta y \tfrac{\partial\psi}{\partial y}(x, 1)] \ , \\
&\omega(0, y) = -\tfrac{2}{\Delta x^2}\psi(\Delta x, y) \ , \\
&\omega(1, y) = -\tfrac{2}{\Delta x^2}\psi(1 - \Delta x, y) \ .
\end{aligned}$$

Problems **dcp1000, dcp5000** correspond to Reynolds numbers $\mathrm{Re} = 1000$ and $\mathrm{Re} = 5000$. For both cases the default initial guess is $\psi^0 = \omega^0 = 0$.

As it turned out, the purely residual based Newton strategy was not able to solve these examples. For this reason, we additionally considered problems **dcp1000a** and **dcp5000a** corresponding to the modified initial guesses $\omega^0 = y^2\sin(\pi x)$, $\psi^0 = 0.1\sin(\pi x)\sin(\pi y)$.

Example: Supersonic transport problem (sst2). The four model equations for the chemical species O, O_3, NO, NO_2 are

$$
\begin{aligned}
0 &= D\Delta u_1 + k_{1,1} - k_{1,2}u_1 + k_{1,3}u_2 + k_{1,4}u_4 - k_{1,5}u_1u_2 - k_{1,6}u_1u_4 \ , \\
0 &= D\Delta u_2 + k_{2,1}u_1 - k_{2,2}u_2 + k_{2,3}u_1u_2 - k_{2,4}u_2u_3 \ , \\
0 &= D\Delta u_3 - k_{3,1}u_3 + k_{3,2}u_4 + k_{3,3}u_1u_4 - k_{3,4}u_2u_3 + 800.0 + SST \ , \\
0 &= D\Delta u_4 - k_{4,1}u_4 + k_{4,2}u_2u_3 - k_{4,3}u_1u_4 + 800.0 \ ,
\end{aligned}
$$

where $D = 0.5\cdot 10^{-9}$, $k_{1,1}, \ldots, k_{1,6} = 4\cdot 10^5, 272.443800016, 10^{-4}, 0.007, 3.67\cdot 10^{-16}, 4.13\cdot 10^{-12}$, $k_{2,1}, \ldots, k_{2,4} = 272.4438, 1.00016\cdot 10^{-4}, 3.67\cdot 10^{-16}, 3.57\cdot 10^{-15}$, $k_{3,1}, \ldots, k_{3,4} = 1.6\cdot 10^{-8}, 0.007, 4.1283\cdot 10^{-12}, 3.57\cdot 10^{-15}$, $k_{4,1}, \ldots, k_{4,3} = 7.000016\cdot 10^{-3}, 3.57\cdot 10^{-15}, 4.1283\cdot 10^{-12}$, and

$$
SST = \begin{cases} 3250 & \text{if } \ (x,y) \in [0.5, 0.6]^2 \\ 360 & \text{otherwise.} \end{cases}
$$

The computational domain is the unit square, homogeneous Neumann boundary conditions are imposed. Initial guess is

$$
u_1^0(x,y) = 10^9 \ , u_2^0(x,y) = 10^9 \ , u_3^0(x,y) = 10^{13} \ , u_4^0(x,y) = 10^7 \ .
$$

Again, we consider an alternative initial guess to allow for convergence in the purely residual based Newton schemes:

$$
u_i^0 \to 100(\sin(\pi x)\sin(\pi y))^2 u_i^0 \ .
$$

Characteristics of test set. An overview on size and difficulty of the examples is given in Table 5. In order to characterize the difficulty of the problems we have tried to solve them with an uncontrolled exact ordinary Newton method. The results are given in the last column of Table 5. All failures are due to reaching the prescribed maximum permitted number of Newton (outer) iterations (indicated by OUTMAX, here set to 75).

For the residual based methods, a termination criterion FTOL $= 10^{-8}$ is required – except for the badly scaled problems **sst**, where FTOL is relaxed to 10^{-5}. For the error oriented methods, a relative termination criterion XTOL $= 10^{-8}$ is set.

Our experimental Newton codes are written in standard FORTRAN77 and use the sparse linear algebra package SLAP due to [16, 22] in order to perform the linear iterations (except GBIT). In particular, we use the left or right preconditioned GMRES/L or GMRES/R and the left preconditioned CGNE/L. As preconditioner we take the default incomplete LU (ILU) decomposition from SLAP.

Name	Grid	Dim n	OrdNew
atp1	31×31	961	4
dcp1000	31×31	1922	OUTMAX
dcp1000a	31×31	1922	9
dcp5000	63×63	7983	OUTMAX
dcp5000a	63×63	7983	OUTMAX
sst2	51×51	10404	OUTMAX
sst2a	51×51	10404	OUTMAX

Table 5: Characteristics of used test set

3.2 Comparative performance

We now compare the performance of exact and inexact Newton methods, both residual based and error oriented, in the frame of our common disctretized PDE test set. Failure exits are characterized throughout by

- OUTMAX: the outer (Newton) iteration does not converge within 75 iterations,
- ITMAX: the inner iteration per Newton step does not converge within 2000 iterations,
- λ-fail: the damping strategy suggests some 'too small' value $\lambda_k < 10^{-4}$.

Residual based vs. error oriented exact Newton codes. In all exact Newton codes, the arising linear systems were solved by band mode LU-decomposition and forward/backward substitution. We implemented the following versions:

- NLEQ–RES based on the standard nonlinear residual F,
- NLEQ–RES/L based on the preconditioned residual $C_L F$, and
- NLEQ–ERR oriented towards the local error.

In Table 6 we arrange the comparative results for our common test set of discretized PDEs.

This comparison is really striking: obviously, for discrete PDEs, the error oriented adaptive Newton methods are clearly preferable to the residual based ones. One reason for this occurrence might be that in PDE discretizations the arising discrete Jacobian matrices are bound to be ill–conditioned due to the noncompact PDE operator behind the discretization. The effect is the more significant, the finer the discretization is.

Name	RES	RES/L	ERR
atp1	4 (0)	4 (0)	4 (0)
dcp1000	OUTMAX	10 (5)	8 (4)
dcp1000a	21 (17)	8 (2)	8 (2)
dcp5000	OUTMAX	OUTMAX	11 (7)
dcp5000a	42 (39)	λ-fail	8 (2)
sst2	λ-fail	12 (11)	13 (8)
sst2a	38 (33)	15 (13)	19 (14)

Table 6: Residual based vs. error oriented exact Newton codes. Comparison in terms of Newton steps (in parentheses: damped Newton steps).

Exact vs. inexact residual based Newton codes. For the inner iteration we chose GMRES (abbreviation for **G**eneralized **M**inimal **RES**idual method), the popular gold standard of linear iterative solvers (see, e.g., [21]). As corresponding inexact Newton codes we implemented GIANT–GMRES/R and GIANT–GMRES/L, the first one with right ILU preconditioning, the second one with left ILU preconditioning. The implemented accuracy matching is not yet fully adaptive in the sense of Section 1.2. Rather, in order to eliminate any side effects stemming from this adaptivity device, we simply imposed a threshold criterion $\eta_k \leq 10^{-3}$ throughout. In Table 7 we document the comparative performance.

Name	EX-RES	INX-RES/R		EX-RES/L	INX-RES/L	
atp1	4 (0)	4 (0)	33	4 (0)	4 (0)	33
dcp1000	OUTMAX	OUTMAX		10 (5)	10 (5)	376
dcp1000a	21 (17)	22 (17)	825	8 (2)	8 (1)	314
dcp5000	OUTMAX	OUTMAX		OUTMAX	OUTMAX	
dcp5000a	42 (39)	44 (39)	5056	λ-fail	22 (13)	1343
sst2	λ-fail	16 (10)	1227	12 (11)	15 (12)	259
sst2a	38 (33)	λ-fail		15 (13)	19 (16)	519

Table 7: Exact vs. inexact residual based Newton codes. Comparison in terms of Newton steps (in parentheses: damped steps) and inner iterations.

In terms of Newton iterations, the inexact residual based Newton codes behave very much like their exact counterparts. Erratic discrepancies arise in two cases where a divergent exact scheme becomes convergent (**sst2** + INX-RES/R, **dcp5000a** + INX-RES/L) and in one reverse case (**sst2a** + INX-RES/R). These differences vanish, if extremely restrictive linear tolerances for the inner iteration ($\eta_k \leq 10^{-7}$) are required. We also relaxed the accuracy for the GMRES iteration to $\eta_k \leq 10^{-2}$: in this case, however, a rather unsatisfactory performance of the outer Newton iteration arose.

Exact vs. inexact error oriented Newton codes. For the error oriented inner iterative solvers we chose CGNE (abbreviation for **C**onjugate **G**radient method for **N**ormal equations with **E**rror minimization, see, e.g., [21]) and GBIT (abbreviation for **G**ood **B**royden **IT**erative solver, see [11]). We implemented the inexact Newton codes GIANT–CGNE/L and GIANT–GBIT/L, i.e. both with left ILU preconditioning. Again, as in the residual based case, we have not yet implemented the fully adaptive accuracy matching strategy as presented in Section 1.1. Rather, we implemented the (scaled) error criterion $\delta_k \leq 10^{-3}$. The initial values for the inner iterations were chosen according to the suggestion (19). The obtained results are arranged in Table 8.

Name	EX-ERR	INX-CGNE/L		INX-GBIT/L	
atp1	4 (0)	4 (0)	331	4 (0)	106
dcp1000	8 (4)	ITMAX		9 (4)	4071
dcp1000a	8 (2)	ITMAX		8 (2)	3012
dcp5000	11 (7)	ITMAX		11 (7)	5616
dcp5000a	8 (2)	ITMAX		8 (1)	7586
sst2	13 (8)	13 (8)	28681	13 (8)	1436
sst2a	19 (14)	19 (14)	61380	19 (14)	2440

Table 8: Exact vs. inexact error oriented Newton codes. Comparison in terms of Newton steps (in parentheses: damped steps) and inner iterations.

From these comparisons, we obtain the following two messages:

- The inner iterative linear solver CGNE is clearly less efficient than GBIT.
- The code GIANT–GBIT/L nearly perfectly reproduces the outer iteration pattern of the exact Newton code NLEQ–ERR.

Note that CGNE *minimizes* the inner iterative error over some Krylov space associated with the normal equations, whereas GBIT only *reduces* the inner iterative error – however, over some different Krylov space corresponding to the original equation. One reason for the bad behavior of CGNE might be caused by the fact that preconditioning for the original linear system (as in GBIT) is more effective than for the normal equations (as in CGNE).

Incidentally, we repeated the computations with GIANT–GBIT/L for a relaxed threshold criterion $\delta_k \leq 10^{-2}$. Once again, all test problems were solved, but now with a saving of up to 50 % in the inner iteration and only a slight deterioration in the outer iteration.

Conclusion

In *elliptic* discrete nonlinear PDE BVPs, both the exact and the inexact affine conjugate Newton codes perform efficiently and reliably, in close connection with the associated convergence theory. The inexact Newton code GIANT–PCG with ICC preconditioning seems to be a real competitor to so–called nonlinear PCG methods.

In *general* discrete nonlinear PDE BVPs, our tests give a clear picture for the *exact* Newton codes: the affine covariant, error oriented adaptive versions are preferable to the affine contravariant, residual based versions – in agreement with expectations from the simpler case of ODE BVPs. For the *inexact* Newton codes, however, the message is less clear. On one hand, the (preconditioned) inner iterative solver GMRES turns out to be more efficient than GBIT and much more efficient than CGNE, at least in our problem class. On the other hand, only the error oriented inexact Newton code GIANT–GBIT has been able to solve all test problems. Therefore, knowing that the affine covariant concept is the right one, in principle, further work needs to be done to improve error oriented linear iterative solvers and preconditioners for the special setting of discretized PDE problems.

Acknowledgements. The authors want to thank R. Ehrig for computational assistance. This work has been supported by the newly established DFG Research Center "Mathematics for Key Technologies", Berlin.

References

[1] L. Armijo. Minimization of functions having Lipschitz–continuous first partial derivatives. *Pacific J. Math.*, 204:126–136, 1966.

[2] R.E. Bank and D.J. Rose. Global approximate Newton Methods. *Numer. Math.*, 37:279–295, 1981.

[3] H.G. Bock. Recent Advances in Parameter Identification Techniques for ODE's. In P. Deuflhard and E. Hairer, editors, *Numerical Treat-*

ment of Inverse Problems in Differential and Integral Equations, volume 2 of *Progress in Scientific Computing*, pages 95–121. Birkhäuser, Boston, Basel, Stuttgart, 1983.

[4] R.S. Dembo, S.C. Eisenstat, and T. Steihaug. Inexact Newton Methods. *SIAM J. Numer. Anal.*, 18:400–408, 1982.

[5] P. Deuflhard. A modified Newton method for the solution of ill–conditioned systems of nonlinear equations with applications to multiple shooting. *Numer. Math.*, 22:289–315, 1974.

[6] P. Deuflhard. A relaxation strategy for the modified Newton method. In R. Bulirsch, W. Oettli, and J. Stoer, editors, *Optimization and Optimal Control*, Springer Lecture Notes in Math. 447, pages 38–55. Springer–Verlag, Berlin, Heidelberg, New York, 1981.

[7] P. Deuflhard. Global Inexact Newton Methods for Very Large Scale Nonlinear Problems. *IMPACT Comp. Sci. Eng.*, 3:366–393, 1991.

[8] P. Deuflhard. Cascadic Conjugate Gradient Methods for Elliptic Partial Differential Equations. Algorithm and Numerical Results. In D.E. Keyes and J. Xu, editors, *Domain Decomposition Methods in Scientific and Engineering Computing*, volume 180 of *AMS Series Contemporary Mathematics*, pages 29–42, 1994.

[9] P. Deuflhard. Adaptive Pseudo-transient Continuation for Nonlinear Steady State Probleems. Tech. Rep. 02–14, ZIB, 2002.

[10] P. Deuflhard. *Newton Methods for Nonlinear Problems. Affine Invariance and Adaptive Algorithms.* Springer International, to be finished 2002.

[11] P. Deuflhard, R. Freund, and A. Walter. Fast Secant Methods for the Iterative Solution of Large Nonsymmetric Linear Systems. *IMPACT Comp. Sci. Eng.*, 2:244–276, 1990.

[12] P. Deuflhard and G. Heindl. Affine Invariant Convergence theorems for Newton's Method and Extensions to related Methods. *SIAM J. Numer. Anal.*, 16:1–10, 1979.

[13] P. Deuflhard and F. A. Potra. Asymptotic Mesh Independence of Newton-Galerkin Methods Via a Refined Mysovskikh Theorem. *SIAM J. Numer. Anal.*, 29:1395-1412 (1992).

[14] P. Deuflhard and M. Weiser. Local Inexact Newton Multilevel FEM for Nonlinear Elliptic Problems. In M.-O. Bristeau, G. Etgen, W. Fitzgibbon, J.-L. Lions, J. Periaux, and M. Wheeler, editors, *Computational Science for the 21st Century*, pages 129–138. Wiley-Interscience-Europe, Tours, France, 1997.

[15] P. Deuflhard and M. Weiser. Global Inexact Newton Multilevel FEM for Nonlinear Elliptic Problems. In W. Hackbusch and G. Wittum, editors, *Multigrid Methods*, volume 3 of *Lecture Notes in Computational Science and Engineering*, pages 71–89. Springer, Berlin, Heidelberg, New York, 1998.

[16] A. Greenbaum. Routines for Solving Large Sparse Linear Systems. Lawrence Livermore Nat. Laboratory, Livermore Computing Center, January 1986 Tentacle, p. 15-21

[17] A. Hohmann. *Inexact Gauss Newton Methods for Parameter Dependent Nonlinear Problems.* PhD thesis, Free University of Berlin, 1994.

[18] U. Nowak and L. Weimann. GIANT – A Software Package for the Numerical Solution of Very Large Systems of Highly Nonlinear Equations. Technical Report TR 90–11, Konrad–Zuse–Zentrum Berlin, 1990.

[19] U. Nowak and L. Weimann. A Family of Newton Codes for Systems of Highly Nonlinear Equations. Technical Report TR 91–10, Konrad–Zuse–Zentrum Berlin, 1991.

[20] M. Pernice and H.F. Walker. NITSOL: a Newton iterative solver for nonlinear systems. *SIAM J. Sci. Comp.*, 5:275–297, 1998.

[21] Y. Saad. *Iterative methods for sparse linear systems.* PWS publishing, New York, 1996.

[22] M. Seager: A SLAP for the Masses. Lawrence Livermore Nat. Laboratory Technical Report, UCRL-100267, December 1988