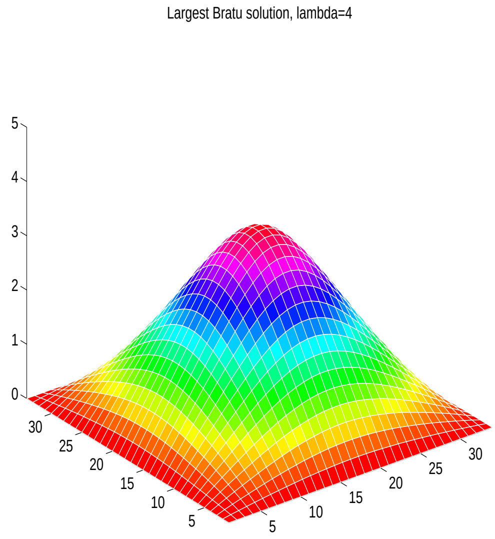



Universiteit Utrecht

Department
of
Mathematics


# Accelerated Inexact Newton Schemes for Large Systems of Nonlinear Equations

by


**Diederik R. Fokkema, Gerard L. G. Sleijpen and Henk A. Van der Vorst**



Preprint

nr. 918

July, 1995

# ACCELERATED INEXACT NEWTON SCHEMES FOR LARGE SYSTEMS OF NONLINEAR EQUATIONS

DIEDERIK R. FOKKEMA*, GERARD L. G. SLEIJPEN*, AND HENK A. VAN DER VORST*

**Abstract.** Classical iteration methods for linear systems, such as Jacobi Iteration, can be accelerated considerably by Krylov subspace methods like GMRES. In this paper, we describe how Inexact Newton methods for nonlinear problems can be accelerated in a similar way and how this leads to a general framework that includes many well-known techniques for solving linear and nonlinear systems, as well as new ones. Inexact Newton methods are frequently used in practice to avoid the expensive exact solution of the large linear system arising in the (possibly also inexact) linearization step of Newton's process. Our framework includes acceleration techniques for the "linear steps" as well as for the "nonlinear steps" in Newton's process. The described class of methods, the **AIN** (Accelerated Inexact Newton) methods, contains methods like GMRES and GMRESR for linear systems, Arnoldi and Jacobi-Davidson for linear eigenproblems, and many variants of Newton's method, like Damped Newton, for general nonlinear problems. As numerical experiments suggest, the **AIN** approach may be useful for the construction of efficient schemes for solving nonlinear problems.

**Key words.** Nonlinear problems, Newton's method, Inexact Newton, Iterative methods.

**AMS subject classification.** 65H10.

## 1. Introduction.

Our goal in this paper is twofold. A number of iterative solvers for linear systems of equations, such as FOM [23], GMRES [26], GCR [31], Flexible GMRES [25], GMRESR [29] and GCRO [7], are in structure very similar to iterative methods for linear eigenproblems, like shift and invert Arnoldi [1, 24], Davidson [6, 24], and Jacobi-Davidson [28]. We will show that all these algorithms can be viewed as instances of an Accelerated Inexact Newton (**AIN**) scheme (cf. ALG. 3), when applied to either linear equations or to linear eigenproblems. This observation may help us in the design and analysis of algorithms by "transporting" algorithmic approaches from one application area to another. Moreover, our aim is to identify efficient **AIN** schemes for nonlinear problems as well, and we will show how we can learn from the algorithms for linear problems.

To be more specific, we will be interested in the numerical approximation of the solution $u$ of the nonlinear equation

$$F(u) = 0, \tag{1}$$

where $F$ is some smooth (nonlinear) map from a domain in $\mathbb{R}^n$ (or $\mathbb{C}^n$) that contains the solution $u$, into $\mathbb{R}^n$ (or $\mathbb{C}^n$), where $n$ is typically large.

Some special types of systems of equations will play an important motivating role in this paper.

The first type is the linear system of equations

$$Ax = b, \tag{2}$$

*Mathematical Institute, Utrecht University, P. O. Box 80.010, NL-3508 TA Utrecht, The Netherlands. E-mail: fokkema@math.ruu.nl, sleijpen@math.ruu.nl, vorst@math.ruu.nl.

where $A$ is a nonsingular matrix and $b$, $x$ are vectors of appropriate size; $A$ and $b$ are given, $x$ is unknown. The dimension $n$ of the problem is typically large and $A$ is often sparse. With $u = x$ and $F(u) := b - Au$, (2) is equivalent to (1). This type will serve as the main source of inspiration for our ideas.

The second type concerns the generalized linear eigenproblem

$$Av = \lambda B v. \tag{3}$$

With $u = (v, \lambda)$ we have that, for normalized $v$, with $F(u) := Av - \lambda B v$, equation (3) is equivalent to (1). This type is an example of a mildly nonlinear system and will serve as an illustration for the similarity between various algorithms, seen as instances of **AIN** (see Section 6).

However, the **AIN** schemes, that we will discuss, will be applicable to more general nonlinear problems, like, for instance, equations that arise from discretizing nonlinear partial differential equations of the form

$$-\nabla(a\nabla u) + b\,g(u)\,\nabla u + c\,h(u) = f \text{ on } \Omega, \tag{4}$$

where $\Omega$ is a domain in $\mathbb{R}^2$ or $\mathbb{R}^3$, $a, b \in C^1(\bar{\Omega})$, $c \in C(\bar{\Omega})$, $h, g \in C^1(\mathbb{R})$, and $f \in L^2(\Omega)$, and $u$ satisfies suitable boundary conditions. An example of (4) is, for instance

$$-\Delta u - \lambda e^u = 0 \text{ on } \Omega, \tag{5}$$

where $\Omega$ is some domain in $\mathbb{R}^2$ and $u = 0$ on $\partial\Omega$ (see also Section 8).

Guided by the known approaches for the linear system (cf. [25, 29, 7]) and the eigenproblem (cf. [28, 27]) we will define accelerated Inexact Newton schemes for more general nonlinear systems. This leads to a combination of Krylov subspace methods for Inexact Newton (cf. [16, 4] and also [8]) with acceleration techniques (as in [2]) and offers us an overwhelming choice of techniques for further improving the efficiency of Newton type methods. As a side-effect this leads to a surprisingly simple framework for the identification of many well-known methods for linear–, eigen–, and nonlinear problems.

Our numerical experiments for nonlinear problems, like problem (5), serve as an illustration for the usefulness of our approach.

The rest of this paper is organized as follows. In Section 2 we briefly review the ideas behind the Inexact Newton method. In Section 3 we introduce the Accelerated Inexact Newton methods (**AIN**). We will examine how iterative methods for linear problems are accelerated and we will distinguish between a Galerkin approach and a Minimal Residual approach. These concepts are then extended to the nonlinear case. In Section 4 we make some comments on the implementation of **AIN** schemes. In Section 5 we show how many well-known iterative methods for linear problems fit in the **AIN** framework. In Section 6 and Section 7 we consider instances of **AIN** for the mildly nonlinear generalized eigenproblem and for more general nonlinear problems. In Section 8 we present our numerical results and some concluding remarks are in Section 9.

1. Set $k = -1$ and choose an initial approximation $u_0$.
2. Repeat until $u_k$ is accurate enough:
   (a) $k = k + 1$
   (b) Compute the residual $r_k = F(u_k)$.
   (c) Compute an approximation $J_k$ for the Jacobian $F'(u_k)$.
   (d) *Solve the correction equation (approximately).* Compute an (approximate) solution $p_k$ of the correction equation
   $$J_k p = -r_k.$$
   (e) *Update.* Compute the new approximation: $u_{k+1} = u_k + p_k$.

ALGORITHM 1: Inexact Newton

**2. Inexact Newton methods.** Newton type methods are very popular for solving systems of nonlinear equations as, for instance, represented by (4). If $u_k$ is the approximate solution at iteration number $k$, Newton's method requires for the next approximate solution of (1), the evaluation of the Jacobian $J_k := F'(u_k)$ and the solution of the *correction equation*

$$J_k p = -r_k, \text{ where } r_k := F(u_k). \tag{6}$$

Unfortunately, it may be very expensive, or even practically impossible to determine the Jacobian and/or to solve the correction equation exactly, especially for larger systems.

In such situations one aims for an approximate solution of the correction equation, possibly with an approximation for the Jacobian (see, e.g., [8]). ALG. 1 is an algorithmical representation of the resulting Inexact Newton scheme.

If, for instance, Krylov subspace methods are used for the approximate solution of the correction equation (6), then only directional derivatives are required (cf. [8, 4]); there is no need to evaluate the Jacobian explicitly. If $v$ is a given vector then the vector $F'(u)v$ can be approximated using the fact that

$$\frac{F(u + \varepsilon v) - F(u)}{\varepsilon} \to F'(u)v \text{ for } \varepsilon \to 0.$$

The combination of a Krylov subspace method with directional derivatives combines the steps 2c and 2d in ALG. 1.

For an initial guess $u_0$ sufficiently close to a solution, Newton's method has asymptotically at least quadratic convergence behavior. However, this quadratic convergence is usually lost if one uses inexact variants and often the convergence is not much faster than linear. In the next section we make suggestions how this (linear) speed of convergence may be improved.

NOTE: It is our aim to restore, as much as possible, the asymptotic convergence behavior of exact Newton and we do not address the question of global convergence.

1. Set $k = -1$ and choose an initial approximation $x_0$.
2. Repeat until $x_k$ is accurate enough:
   (a) $k = k + 1$
   (b) $r_k = b - Ax_k$
   (c) $p_k = \mathrm{diag}(A)^{-1} r_k$
   (d) $x_{k+1} = x_k + p_k$

ALGORITHM 2: Jacobi Iteration

**3. Accelerating Inexact Newton methods.** Newton's method is a one step method, that is, in each step, Newton's method updates the approximate solution with information from the previous step only. However, in the computational process, subspaces have been built gradually, that contain useful information concerning the problem. This information may be exploited to improve the current approximate solution, and this is what we propose to do. More precisely, we will consider alternative update strategies for step 2e of the Inexact Newton algorithm ALG. 1.

**3.1. Acceleration in the linear case.** The linear system $Ax = b$ can be written as $F(x) := b - Ax = 0$, and $-A$ is the Jacobian of $F$. When the approximate solution $p_k$ is computed as $p_k = M^{-1} r_k$, where $M$ is some preconditioning matrix (approximating $A$), then the Inexact Newton algorithm, ALG. 1, reduces to a standard Richardson-type iteration process for the splitting $A = M - R$. For instance, the choice $M = D$, where $D = \mathrm{diag}(A)$ leads to Jacobi iteration (see ALG. 2).

One may improve the convergence behavior of standard iterations schemes by

- using more sophisticated preconditioners $M$, and/or
- applying acceleration techniques in the update step.

Different preconditioners and different acceleration techniques lead to different algorithms, some of which are well-known.

Examples of iterations schemes that use more sophisticated preconditioners are, for instance, Gauss-Seidel iteration, where $M = L + D$, with $L$ is the strict lower triangular part of $A$ and $D = \mathrm{diag}(A)$ and SOR where $M = \omega L + D$, and $\omega$ is a relaxation parameter.

Examples of iterations schemes that use acceleration techniques are algorithms that take their updates to the approximate solution as a linear combination of previous directions $p_j$. Preferable updates $\tilde{p}_k := \sum_{j \le k} \gamma_j p_j$ are those for which $b - Ax_{k+1}$, where $x_{k+1} = x_k + \tilde{p}_k$, is minimal in some sense: e.g., $\|b - Ax_{k+1}\|_2$ is minimal, as in GMRES [26] and GCR [31], or $b - Ax_{k+1}$ is orthogonal to the $p_j$ for $j \le k$, as in FOM or GENCG [23], or $b - Ax_{k+1}$ is "quasi-minimal", as in Bi-CG [17], and QMR [11].

Of course the distinction between preconditioning and acceleration is not a clear one. Acceleration techniques with a limited number of steps can be seen as a kind of dynamic preconditioning as opposed to the static preconditioning with fixed $M$. In this view one is again free to choose an acceleration technique. Examples of such iteration schemes are Flexible GMRES [25], GMRESR [29] and GCRO [7].

All these accelerated iteration schemes for linear problems construct approximations $x_{k+1} = x_0 + V_k y_k$, $V_k := [p_1, p_2, \ldots, p_k]$, with $y_k$ the solution of a *smaller* or an *easier* projected problem. For example, GMRES computes $y_k$ such that $\|b - A(x_k + V_k y_k)\|_2$ is minimal, or equivalently $(AV_k)^*(b - A(x_k + V_k y_k)) = 0$; FOM computes $y_k$ such that $V_k^*(b - A(x_k + V_k y_k)) = 0$, whereas Bi-CG and QMR compute $y_k$ as the solution of a larger tri-diagonal problem obtained with oblique projections.

For stability (and efficiency) reasons one usually constructs another basis for the span of $V_k$ with certain orthogonality properties, depending on the selected approach.

**3.2. Acceleration in the nonlinear case.** We are interested in methods for finding a zero of a general nonlinear mapping $F$. For the linear case, the methods mentioned above are, apart from the computation of the residual, essentially a mix of two components: (1) the computation of a new search direction (which involves the residual), and (2) the update of the approximation (which involves the current search direction and possibly previous search directions, and the solution $y_k$ of a projected problem). The first component may be interpreted as preconditioning, while the second component is the acceleration.

Looking more carefully on how $y_k$ in the linear case is computed, we can distinguish between two approaches based on two different conditions. With $G_k(y) := F(x_k + V_k y)$, the FOM and the other "oblique" approaches lead to methods that compute $y$ such that (for appropriate $W_k$) $W_k^* G_k(y) = 0$ (a *Galerkin* condition), whereas the GMRES approach leads to methods that compute $y$ such that $\|G_k(y)\|_2$ is minimal (a *Minimal Residual* condition).

From these observations for the linear case we now can formulate iteration schemes for the nonlinear case.

The Inexact Newton iteration can be accelerated in a similar way as the standard linear iteration. This acceleration can be accomplished by updating the solution by a correction $\tilde{p}_k$ in the subspace spanned by all correction directions $p_j$ $(j \le k)$.

To be more precise, the update $\tilde{p}_k$ for the approximate solution is given by $\tilde{p}_k = V_k y$, where $V_k = [v_1, v_2, \ldots, v_k]$ for some basis $(v_j)$ of the *search space* $\mathcal{V}_k$ spanned by $p_1, p_2, \ldots, p_k$. Furthermore, with $G_k(y) = F(u_k + V_k y)$, we propose to determine $y$ by

- a *Galerkin* condition on $G_k(y)$: $y$ is a solution of

$$W_k^* G_k(y) = 0, \tag{7}$$

  where $W_k$ is some matrix of the same dimensions as $V_k$,
- or a *Minimal Residual* (MR) condition on $G_k(y)$: $y$ is a solution of

$$\min_y \|G_k(y)\|_2, \tag{8}$$

- or a mix of both, a *Restricted Minimal Residual* (RMR) condition on $G_k(y)$: $y$ is a solution of

$$\min_y \|W_k^* G_k(y)\|_2. \tag{9}$$

Equation (7) generalizes the FOM approach, while equation (8) generalizes the GMRES approach.

Solving (7) means that the component of the residual $r_{k+1}$ in the subspace $\mathcal{W}_k$ (spanned by $W_k$) vanishes. For $W_k$ one may choose, for instance, $W_k = V_k$ (as in

FOM), or $W_k = [W_{k-1}, w_k]$ where $w_k$ is the component of $J_k p_k$ orthogonal to $W_{k-1}$, (as in GMRES: $w_k = (I - W_{k-1}W_{k-1}^*)J_k p_k$). For linear equations the Minimal Residual and the Galerkin approach coincide for the last choice.

As is known from the linear case, a complication of the Galerkin approach is that equation (7) may have no solution, which means that this approach may lead to breakdown of the method. In order to circumvent this shortcoming to some extend we have formulated the Restricted Minimal Residual approach (9). Compared to (7) this formulation is also attractive for another reason: one can apply standard Gauss-Newton [9] schemes for solving general nonlinear least squares problems to it. One might argue that a drawback of a Gauss-Newton scheme is that it may converge slowly (or not at all). However, for least squares problems with zero residual solutions, the asymptotic speed of convergence of a Gauss-Newton method is that of Newton's method. This means, that if the Galerkin problem (7) has a solution, a Gauss-Newton scheme applied to (9) will find it fast and efficient (see also Section 8).

Note that equations (7)–(9) represent nonlinear problems in only $k$ variables, which may be much easier to solve than the original problem. If these smaller nonlinear problems can be formulated cheaply, then the costs for an update step may be considered as being relatively small.

Note also that since equations (7)–(9) are nonlinear, they may have more than one solution. This fact may be exploited to steer the computational process to a specific preferable solution of the original problem.

**Accelerated Inexact Newton.** For the Galerkin approach, step 2e in the Inexact Newton's algorithm, ALG. 1, is replaced by four steps in which

- the search subspace $\mathcal{V}_{k-1}$ is expanded by an approximate "Newton correction" and a suitable basis is constructed for this subspace,
- a shadow space $\mathcal{W}_k$ is selected on which we project the original problem,
- the projected problem (7) is solved,
- and the solution is updated.

This is represented, by the steps 3e–3h in ALG. 3. The Minimal Residual approach and the Restricted Minimal Residual approach can be represented in a similar way.

## 4. Computational considerations.

In this section we make some comments on implementation details that mainly focus on limiting computational work and memory space.

### 4.1. Restart.

For small $k$, problems (7)–(9) are of small dimension and may often be solved at relatively low computational costs (e.g., by some variant of Newton's method).

For larger $k$ they may become a serious problem in itself. In such a situation, one may wish to restrict the subspaces $\mathcal{V}$ and $\mathcal{W}$ to subspaces of smaller dimension (see ALG. 3, step 3i). Such an approach limits the computational costs per iteration, but it may also have a negative effect on the speed of convergence.

For example, the simplest choice, restricting the search subspace to a 1-dim. subspace leads to Damped Inexact Newton methods, where, for instance, the damping parameter $\alpha$ is the solution of $\min_\alpha \|G_k(\alpha)\|_2$, where $G_k(\alpha) = F(u_k + p_k\alpha)$.

1. Set $k = -1$ and choose an initial approximation $u_0$.
2. $V = [\,]$, $W = [\,]$.
3. Repeat until $u_k$ is accurate enough:
   - (a) $k = k + 1$
   - (b) Compute the residual $r_k = F(u_k)$.
   - (c) Compute an approximation $J_k$ for the Jacobian $F'(u_k)$.
   - (d) *Solve the correction equation (approximately).* Compute an (approximate) solution $p_k$ for the correction equation
     $$J_k p = -r_k.$$
   - (e) *Expand the search space.* Select a $v_k$ in the span$(V_{k-1}, p_k)$ that is linearly independent of $V_{k-1}$ and update $V_k = [V_{k-1}, v_k]$.
   - (f) *Expand the shadow space.* Select a $w_k$ that is linearly independent of $W_{k-1}$ and update $W_k = [W_{k-1}, w_k]$.
   - (g) *Solve the projected problem.* Compute nontrivial solutions $y$ of the projected system
     $$W_k^* G_k(y) = 0.$$
   - (h) *Update.* Select a $y_k$ (from the set of solutions $y$) and update the approximation: $u_{k+1} = u_k + V_k y_k$.
   - (i) *Restart.* $\ell = \dim(\mathrm{span}(V_k))$. If $\ell$ is too large, select an $\ell' < \ell$, select $\ell \times \ell'$ matrices $R_V$ and $R_W$ and compute $V_k = V_k R_V$, $W_k = W_k R_W$ (i.e., take suitable combinations of the columns of $V_k$ and $W_k$).

ALGORITHM 3: Accelerated Inexact Newton

Of course, a complete restart is also feasible, say after each $m$th step (cf. step 3i of ALG. 3):

3i If $\dim(\mathrm{span}(V_k)) > m$ then $V_k = [\,]$ and $W_k = [\,]$.

The disadvantage of a complete restart is that we have to rebuild subspace information again. Usually it leads to a slower speed of convergence.

It seems like an open door to suggest that parts of subspaces better be retained at restart, but in practical situations it is very difficult to predict what those parts should be. A meaningful choice would depend on spectral properties of the Jacobian as well as on the current approximation. When solving linear equations with GMRESR [29], good results have been reported in [30] when selecting a number of the first and the last columns (cf. step 3i of ALG. 3); e.g.,

3i If $\dim(\mathrm{span}(V_k)) > 10$ then $V_k = V_k R_V$ and $W_k = W_k R_W$, with
$$R_V = R_W = [e_1, \ldots, e_5, e_7, \ldots, e_{11}].$$

In [7], a variant of GMRESR, called GCRO, is proposed, which implements another choice. For subspaces of dimension $l+m$, the first $l$ columns are retained, together with a combination of the last $m$ columns. This combination is taken such that the approximate solution, induced by a minimal residual condition, is the same for both the subspaces of dimension $l+m$ and $l+1$. To be more specific, if $u_{k+1} = u_k + V_k y_k$, where $y_k$ solves $\min_y \|F(u_k + V_k y\|_2$, then $V_k$ is replaced by $[V_{k_l}, V_{k_m} y_{k_m}]$ (denoting $V_k = [V_{k_l}, V_{k_m}]$ and $y_k = [y_{k_l}, y_{k_m}]$).

**4.2. Update.** In the update step (step 3h of ALG. 3), a solution $y_k$ of the projected problem has to be selected from the set of solutions $y$. Selection may be necessary since many nonlinear problems have more than one solution. Sometimes, this may be the reason for poor convergence of (Inexact) Newton: the sequence of approximate solutions "wavers" between different exact solutions. For larger search subspaces, the search subspace may contain good approximations for more than one solution. This may be exploited to steer the sequence of approximate solution to the wanted solution $u_{k+1}$ and it may help to avoid wavering convergence behavior.

The selection of $y_k$ should be based on additional properties of the solution $u_{k+1}$. For instance, we may look for the solution largest in norm, or as in the case of eigenvalue problems, for a solution of which one component is close to some specific value (for instance, if one is interested in eigenvalues close to, say, 0, the Ritz vector with Ritz value closest to 0 will be chosen).

**4.3. The projected problem.** Even though problems of small dimension can be solved with relatively low computational costs, step 3g in ALG. 3 is not necessarily inexpensive. The projected problem is embedded in the large subspace and it may require quite some computational effort to represent the problem in a small subspace (to which $y$ belongs) of dimension $\ell := \dim(\text{span}(V_k))$ ($y \in \mathbb{C}^\ell$). In the case of linear equations (or linear eigenvalue problems) the computation of an $\ell \times \ell$-matrix as $W_k^* A V_k$ requires $\ell^2$ inner products. For this type of problems, and for many others as well, one may save on the computational costs by re-using information from previous iterations.

**4.4. Expanding the search subspace.** The **AIN** algorithm breaks down if the search subspace is not expanded. This happens when $p_k$ belongs to the span of $V_{k-1}$ (or, in finite precision arithmetic, if the angle between $p_k$ and this subspace is very small). Similar as for GMRES, one may then replace $p_k$ by $J_k v_\ell$, where $v_\ell$ is the last column vector of the matrix $V_{k-1}$.

With approximate solution of the correction equation, a breakdown will also occur if the new residual $r_k$ is equal to the previous residual $r_{k-1}$. We will have such a situation if $y_{k-1} = 0$. Then, instead of modifying the expansion process in iteration number $k$, one may also take measures in iteration number $k-1$ in order to avoid $y_{k-1} = 0$. In [29] a few steps by LSQR are suggested when the linear solver is a Krylov subspace method: $p_k = J_{k-1}^T r_{k-1}$ may already cure the stagnation.

**5. How linear solvers fit in the AIN framework.** In this section we will show how some well-known iterative methods for the solution of linear systems fit in the **AIN** framework. The methods that follow from specific choices in **AIN** are equivalent to well-known methods only in the sense that, at least in exact arithmetic, they produce the same basis vectors for the search spaces, the same approximate solutions, and the same Newton corrections (in the same sense as in which GMRES and ORTHODIR are equivalent).

With $u = x$, $F(u) := b - Au$, the linear equation (2) is equivalent to the one in (1) and $J_k = -A$. In this section, $M$ denotes a preconditioning matrix for $A$ (i.e., for a vector $v$, $M^{-1}v$ is easy to compute and approximates $A^{-1}v$).

**5.1. GCR.** With the choice, $p_k = M^{-1}r_k$, $w_k = Av_k$, and $v_k$ such that $w_k \perp \text{span}(W_{k-1})$, the **AIN** algorithm ALG. 3 (without restart) is equivalent to preconditioned GCR [31].

**5.2. FOM and GMRES.** The choice $p_k = M^{-1}r_k$ and $v_k$ such that $v_k \perp \text{span}(V_{k-1})$, gives algorithms that are related to FOM and GMRES [26]. With the additional choice $w_k = v_k$, **AIN** algorithm ALG. 3 is just FOM, while the choice $w_k = Av_k$ gives an algorithm that is equivalent to GMRES.

**5.3. GMRESR.** Taking $w_k = Av_k$, and $v_k$ such that $w_k$ is perpendicular to $\text{span}(W_{k-1})$ as in GCR and taking $p_k$ as an approximate solution of the equation $Ap = r_k$, **AIN** algorithm ALG. 3 is equivalent to the GMRESR algorithms [29]. One might compute $p_k$ by a few steps of GMRES, for instance.

## 6. AIN schemes for mildly nonlinear problems.

In this section we will discuss numerical methods for the iterative solution of the generalized eigenproblem (3). We will show that they also fit in the general framework of the **AIN** ALG. 3 methods.

As already mentioned these **AIN** methods consist of two parts. In one part an approximate solution of the correction equation (cf. step 3d in ALG. 3) is used to extend the search space. In the other part a solution of the projected problem (cf. step 3g in ALG. 3) is used to construct an update for the approximate solution.

We will start with the derivation of a more suitable form for the (Newton) correction equation for the generalized eigenproblem. After that, we will make some comments on how to solve the projected problem.

**The correction equation.** In order to avoid some of the complications that go with complex differentiation, we will mainly focus on the numerical computation of eigenvectors with a fixed component in some given direction (rather then on the computation of eigenvectors with a fixed norm).

First, let $\tilde{u}$ be a fixed vector with a nontrivial component in the direction of the desired eigenvector $x$. We want to compute approximations $u_k$ for $x$ with a normalized component in the $\tilde{u}$-direction: $(u, \tilde{u}) = (x, \tilde{u}) = 1$. We will select $\vartheta_k$ such that the residual $r_k := (A - \vartheta_k B)u_k$ is orthogonal to $w$, where $w$ is another fixed nontrivial vector, i.e., the approximate eigenvalue $\vartheta_k$ is given by $\vartheta_k := w^*Au_k/w^*Bu_k$.

Consider the map $F$ given by

$$F(u) := (A - \vartheta B)u, \text{ where } \vartheta := \frac{w^*Au}{w^*Bu},$$

and $u$ belongs to the hyper-plane $\{y \in \mathbb{C}^n \mid (y, \tilde{u}) = 1\}$. The Jacobian $J_k = F'(u_k)$ is then given by

$$J_k = \left(I - \frac{Bu_k w^*}{w^*Bu_k}\right)(A - \vartheta_k B)_{|\tilde{u}^\perp},$$

and the correction equation reads as

$$p \perp \tilde{u}, \text{ such that } \left(I - \frac{Bu_k w^*}{w^* B u_k}\right)(A - \vartheta_k B)p = -r_k. \tag{10}$$

Since $r_k \perp w$, this equation is equivalent to

$$\begin{bmatrix} A - \vartheta_k B & Bu_k \\ \tilde{u}^* & 0 \end{bmatrix} \begin{bmatrix} p \\ \varepsilon \end{bmatrix} = \begin{bmatrix} -r_k \\ 0 \end{bmatrix}, \tag{11}$$

that is, $p$ is the solution of (10) if and only if $p$ is the solution of (11).

**The projected problem.** For the Generalized eigenvalue problem we are in the fortunate position that all the solutions of problems of moderate size can be computed by standard methods such as for instance the QZ [19] method. However, before we can apply these methods we have to reformulate the projected problem because of the exceptional position of $u_k$ in $W^* F(u_k + V_k y)$.

The key to this reformulation is the observation that in the methods we consider the affine subspace $u_k + \mathrm{span}(V_k)$ is equal to $V_k$ because $V_k$ contains $u_k$ by itself. Now, as an alternative to step 3g in ALG. 3, we may also compute all the solutions $y$ of

$$W_k^* F(V_k y) = 0.$$

This problem can now be solved by for instance the QZ method, and after selecting $y_k$ a new approximation $u_{k+1}$ is given by $u_{k+1} := V_k y_k$.

**6.1. Arnoldi's method.** We consider the simplified case where $B = I$, i.e., the standard eigenproblem. If we do only one step of a Krylov subspace method (Krylov dimension 1) for the solution of the correction equation (10), then we obtain for the correction $p_k$:

$$p_k = -\alpha r_k.$$

Hence, $p_k = -\alpha(A - \vartheta_k I)u_k$. Note that this may be a poor (very) approximation, because, in general, $r_k \not\perp \tilde{u}$. The approximate eigenvector $u_k$ belongs to the search subspace $\mathrm{span}(V_{k-1})$ and expanding the search subspace by the component of $p_k$ orthogonal to $\mathrm{span}(V_{k-1})$ is equivalent to expanding this space with the orthogonal component of $Au_k$, which would be the "expansion" vector in Arnoldi's method. Hence, the search subspace is precisely the Krylov subspace generated by $A$ and $u_0$. Apparently, Arnoldi's method is an **AIN** method (with a "very inexact Newton step") without restart.

The choice $W_k = V_k$ corresponds to the standard one in Arnoldi and produces $\vartheta$'s that are called Ritz values, while the choice $W_k = AV_k$ leads to Harmonic Ritz values [22].

**6.2. Davidson's method.** As in Arnoldi's method, Davidson's method [6] also carries out only one step of a Krylov subspace method for the solution of the correction equation. However, in contrast to Arnoldi's method, Davidson also incorporates a preconditioner.

He suggests to solve (10) approximately by $p_k$ with

$$p_k = -M^{-1} r_k,$$

where $M$ is the inverse of the diagonal of $A - \vartheta_k B$. Other choices have been suggested as well (cf. e.g., [5, 21]). Because of the preconditioner, even if $B = I$, the search space is not simply the Krylov subspace generated by $A$ and $u_0$. This may lead to an advantage of Davidson's method over Arnoldi's method.

For none of the choices of the preconditioner, proper care has been taken of the projections (see (10)): the preconditioner should approximate the inverse of the projected matrix (see (10)) as a map from $\tilde{u}^\perp$ onto $w^\perp$ rather than of $A - \vartheta_k B$.

However, if $M$ is the diagonal of $A - \vartheta B$, and we choose $\tilde{u}$ and $w$ equal to the same, arbitrary standard basis vector (as Davidson does [6]) then

$$\left(I - \frac{B u_k w^*}{w^* B u_k}\right) M \left(I - \frac{u_k w^*}{w^* u_k}\right) p = -r_k,$$

whenever $w \perp r_k$ and $p$ solves $Mp = -r_k$. Note that $p \perp w$, because $M$ is diagonal and $r_k \perp w$. Therefore, for this particular choice of $w$ (and $\tilde{u}$), the diagonal $M$ may be expected to be a good preconditioner for the correction equation (including the projections) in the cases where $M$ is a good preconditioner for $A - \vartheta_k B$. Observe that this argument does not hold for non-diagonal preconditioners $M$.

**6.3. Jacobi-Davidson.** Davidson methods with a non-diagonal preconditioner do not take care properly of the projections in the correction equation (10). This observation was made in [28], and a new algorithm was proposed for eigenproblems by including the projections in the Davidson scheme. In addition, these modified schemes allow for more general approximate solutions $p_k$ than $p_k = -M^{-1} r_k$. For instance, the use of $\ell$ steps of a preconditioned Krylov subspace method for the correction equation is suggested, leading to Arnoldi type of methods in which the variable polynomial preconditioning is determined efficiently and the projections have been included correctly. The new methods have been called *Jacobi-Davidson* methods (Jacobi took proper care of the projections, but did not build a search subspace as Davidson did (see [28] for details and further references)).

The analysis and results in [3, 27] show that these Jacobi-Davidson methods can also be effective for solving generalized eigenproblems, even without any matrix inversion.

The Jacobi-Davidson methods allow for a variety of choices that may improve efficiency of the steps and speed of convergence and are good examples of **AIN** methods in which the projected problem (7) is used to steer the computation.

For an extensive discussion, we refer to [27].

**7. AIN schemes for general nonlinear problems.** In this section we summarize some iterative methods for the solution of nonlinear problems that have been proposed by different authors, and we show how these methods fit in the **AIN** framework.

Brown and Saad [4] describe a family of methods for solving nonlinear problems. They refer to these methods as *nonlinear Krylov subspace projection methods.* Their modifications to Newton's method are intended to enhance robustness and are heavily influenced by ideas presented in [9]. One of their methods is a variant of Damped Inexact Newton, in which they approximate the solution of the correction equation by a few steps of Arnoldi or GMRES and determine the damping parameter $\alpha$ by a "linesearch backtracking technique". So this is just another **AIN** scheme, with a special 1-dimensional subspace acceleration. They also propose a model trust region approach, where they take their update to the approximation from the Krylov subspace $\widetilde{V}_m$ generated by $m$ steps of (preconditioned) Arnoldi or GMRES as $p_k = \widetilde{V}_m \tilde{y}_k$, where $\tilde{y}_k$ is the point on the dogleg curve for which $\|\tilde{y}_k\|_2 = \tau$, the trust region size: $\tilde{y}_k$ is an approximation for $\min_y \|F(u_k + \widetilde{V}_m \tilde{y})\|_2$. This could be considered as a block version of the previous method.

In [2] Axelsson and Chronopoulos propose two nonlinear versions of a (truncated) Generalized Conjugate Gradient type of method. Both methods fit in the **AIN** framework. The first method, NGCG, is a Minimal Residual **AIN** method with $p_k = -r_k$ and $V_k$ orthonormal; in other words the correction equation is not solved. The second method, NNGCG, differs from NGCG in that $p_k$ is now computed as an approximate solution (by some method) of the correction equation (6), where the accuracy is such that $\|F(u_k) - F'(u_k)p_k\| \leq \rho_k \|F(u_k)\|$, for some non-increasing sequence $(\rho_j), 0 \leq \rho_j < 1$ (see also [8]). So the method NNGCG is a Minimal Residual **AIN** method. It can be viewed as generalization of GMRESR [29]. Under certain conditions on the map $F$ they prove global convergence.

In [15], Kaporin and Axelsson propose a class of nonlinear equation solvers, GNKS, in which the ideas presented in [4] and [2] are combined. There, the direction vectors $p_k$ are obtained as linear combinations of the columns of $\widetilde{V}_m$ and $V_k$. To be more precise, $p_k = [\widetilde{V}_m, V_k] y_k$, where $y_k$ solves $\min_y \|F(u_k + [\widetilde{V}_m, V_k]y)\|_2$. This problem is then solved by a special Gauss-Newton iteration scheme, which avoids excessive computational work, by taking into account the acute angle between $r_k$ and $J_k p_k$, and the rate of convergence. The method generalizes GCRO [7].

**8. Numerical Experiments.** In this section we test several **AIN** schemes and present results of numerical experiments on three different nonlinear problems. For tests and test results with methods for linear– and eigenproblems we refer to their references. The purpose of this presentation is to show that acceleration may be useful also in the nonlinear case. By useful, we mean that additional computational cost is compensated for by faster convergence.

Different **AIN** schemes distinguish themselves by the way they (approximately) solve the correction equation and the projected problem (cf. Section 3.2 and 7). Out of the overwhelming variety of choices we have selected a few possible combinations, some of which lead to **AIN** schemes that are equivalent to already proposed methods and some of which lead to new methods. We compare the following (existing) *Minimal Residual* **AIN** schemes:

- LINESEARCH, the backtracking linesearch technique [4, pp. 458];
- DOGLEG, the model trust region approach as proposed in [4, pp. 462];
- NNGCG, a variant of the method proposed in [2], solving (8) by the Levenberg-Marquardt algorithm [20];

- GNKS, the method proposed in [15];

and the (new) *Restricted Minimal Residual* **AIN** schemes:

- RMR A, choosing $W_k = V_k$; and
- RMR B, choosing $W_k = [W_{k-1}, w_k]$, where $w_k$ is the component of $J_k p_k$ orthogonal to $W_{k-1}$.

For these last two schemes, the minimization problem (9) was solved by the Gauss-Newton variant described in [15]. The necessary subspaces for the direction $p_k$ or the projected problem were obtained by 10 steps of GMRES, or (in the third example) also by at most 50 iterations of the generalized CGS variant CGS2 [10].

In all cases the exact Jacobian was used. Furthermore, we used orthonormal matrices $V_k$ and $W_k$, obtained from a modified Gram-Schmidt process and restricted to the last 10 columns in an attempt to save computational work. The computations were done on a Sun Sparc 20 in double precision and the iterations were stopped when $\|r_k\|_2 \leq 10^{-6}$. A method *failed*, either when the convergence was too slow, i.e., when $|\, \|r_{k+1}\|_2 - \|r_k\|_2 \,| < 10^{-6} \|r_{k+1}\|_2$, or when to number of nonlinear iterations (per step) exceeded 200.

Since the computational cost of the methods is approximately proportional to the costs of the number of function evaluations and matrix multiplications, the following counters are given in the tables:

- NI, the number of nonlinear iterations;
- FE, the number of function evaluations;
- MV, the number of multiplications by the Jacobian;
- PRE, the number of applications of the preconditioner;
- TOTAL, the sum of FE, MV and PRE.

**8.1. A 1D Burgers' Equation.** As a first test problem we consider the following 1D Burgers' Equation [14]

$$
\begin{aligned}
&\frac{\partial u}{\partial t} + \sin(2u)\frac{\partial u}{\partial x} = \mu \frac{\partial^2 u}{\partial x^2}, \\
&u(x,0) = g(x), \quad x \in \Omega, \\
&u(x,t) = \psi(x,t), \quad x \in \partial\Omega,
\end{aligned}
\tag{12}
$$

where $\Omega = [0, 1]$.

We discretized the spatial variable $x$ with finite differences with 64 grid points and for the time derivative we used

$$\theta \dot{u}_{n+1} + (1-\theta)\dot{u}_n = \frac{u_{n+1} - u_n}{\Delta t},$$

with $\theta = \frac{2}{3}$ and $\Delta t = 10^{-2}$. $u_n$ denotes the solution at time $t_n = t_0 + n\Delta t$. For this test the solution $u_n$ was computed for $n = 1, 2, \ldots, 30$ and as an initial guess to $u_{n+1}$ we took $u_n$. No preconditioning was used.

In table TAB. 1 we show the results for problem (12) with

$$g(x) = \pi - 2\pi x, \quad \psi(x,t) = g(x), \text{ and } \mu = 10^{-2}.$$

| Method | NI | FE | MV | TOTAL |
|---|---|---|---|---|
| LINESEARCH | 594 | 1818 | 4853 | 6671 |
| DOGLEG | 644 | 3559 | 6140 | 9699 |
| NNGCG | 229 | 4426 | 11769 | 16195 |
| GNKS | 187 | 852 | 7067 | 7919 |
| RMR A | 230 | 926 | 4471 | 5397 |
| RMR B | 236 | 856 | 4606 | 5462 |

TABLE 1: Results for Burgers' Equation

A plot of the solutions $u_1, u_2, \ldots, u_{30}$ is given in FIG. 1. The table shows the cumulative value of the counters for each method after completing the computation of $u_{30}$.

If we look at the number of nonlinear iterations (NI), we see that acceleration indeed reduces this number. However, in the case of GNKS this does not result in less work, because the number of matrix multiplications (MV) increases too much. Here both the Galerkin approaches RMR A and RMR B are less expensive than all the other methods. RMR A being the winner.

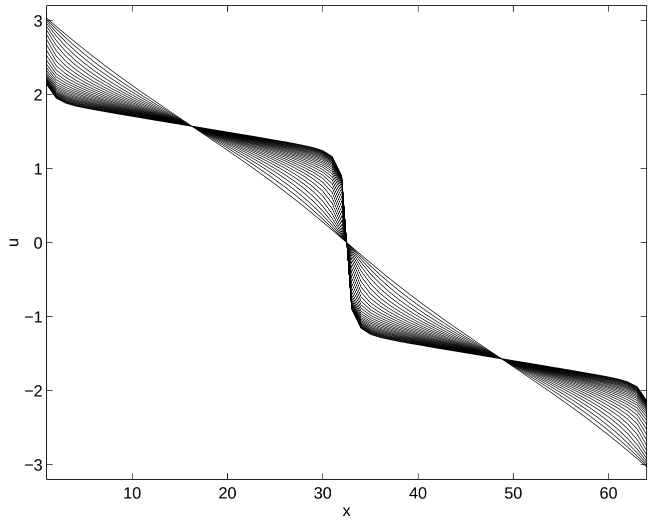


FIGURE 1: Solution of Burgers' Equation

**8.2. The Bratu problem.** As a second test problem we consider the Bratu problem [12, 4]. We seek a solution $(u, \lambda)$ of the nonlinear boundary value problem:

$$-\Delta u - \lambda e^u = 0 \text{ in } \Omega, \quad u = 0 \text{ on } \partial\Omega. \tag{13}$$

For $\Omega$ we took the unit square and we discretized with finite differences on a $31 \times 31$ regular grid. It is known, cf. [12], that there exist a critical value $\lambda^*$ such that for $0 < \lambda < \lambda^*$, problem (13) has two solutions and for $\lambda > \lambda^*$ problem (13) has no solutions. In order to locate this critical value we use the arc length continuation method as described in [12, section 2.3 and 2.4]. Problem (13) is replaced by a problem

| Method | NI | FE | MV | PRE | TOTAL |
|---|---|---|---|---|---|
| LINESEARCH | 391 | 1013 | 3732 | 3421 | 8166 |
| DOGLEG | 381 | 2664 | 3010 | 3010 | 8684 |
| NNGCG | 361 | 1297 | 4243 | 3091 | 8631 |
| GNKS | 358 | 1056 | 6896 | 2780 | 10732 |
| RMR A | 389 | 539 | 4005 | 3399 | 7943 |
| RMR B | 361 | 414 | 3806 | 3091 | 7311 |

TABLE 2: Results for the Bratu Problem, solved by the arc length continuation method.

| Method | NI | FE | MV | PRE | TOTAL |
|---|---|---|---|---|---|
| LINESEARCH | 29 | 85 | 336 | 308 | 729 |
| DOGLEG | 27 | 151 | 494 | 260 | 905 |
| NNGCG | 9 | 49 | 196 | 88 | 333 |
| GNKS | 38 | 119 | 1806 | 370 | 2295 |
| RMR A | 6 | 13 | 77 | 55 | 145 |
| RMR B | 6 | 12 | 79 | 55 | 146 |

TABLE 3: Single solve of the Bratu Problem, $u_0 = 0, \lambda = 6.8$.

of the form

$$\begin{cases} F(u_s, \lambda(s)) = 0, \\ \ell(u_s, \lambda(s), s) = 0, \end{cases}$$

where $\ell$, a scalar valued function, is chosen such that $s$ is some arc length on the solution branch and $u_s$ is the solution of (13) for $\lambda = \lambda(s)$. We preconditioned GMRES by ILU(0) [18] of the discretized Laplace operator $\Delta$.

The first table TAB. 2 shows the results after a full continuation run: starting from the smallest solution $(u, \lambda)$ with $\lambda = 1$ the solution branch is followed along the (discretized) arc with $s_n = s_0 + n\Delta s$ for step-size $\Delta s = 1$ and $n = 1, 2, \ldots, 80$. Again we see that acceleration may be useful, in spite of the fact that there is little room for it, because on the average approximately only 4.5 Newton iterations where necessary to compute the solution per continuation step. In this example RMR B performs better than RMR A.

Table TAB. 3 shows the results for the case where we solve (13) for fixed $\lambda = 6.8$ (near the critical value). In this case Galerkin acceleration is even more useful and the differences are more pronounced.

The sup norm of the solution for the different values of $\lambda$ are plotted in FIG. 2. The two solutions at $\lambda \approx 4$ along the diagonal of the unit square are shown in FIG. 3.

**8.3. The Driven Cavity Problem.** In this Section we present test results for the classical *driven cavity* problem from incompressible fluid flow. We follow closely

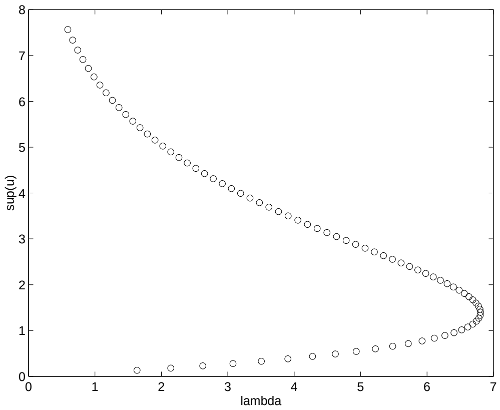


FIGURE 2: Sup norms of the solution $u$ along the arc.

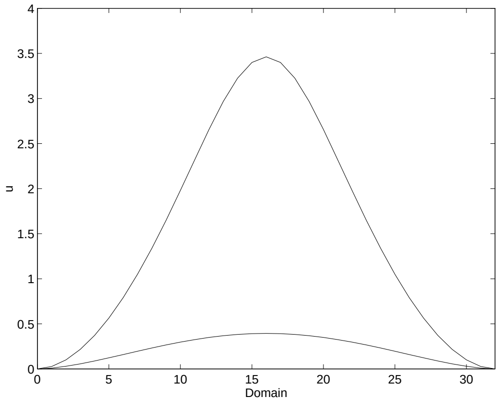


FIGURE 3: Solutions $u$ at $\lambda \approx 4$ along the diagonal of the domain.

the presentations in [12, 4]. In stream function-vorticity formulation the equations are

$$
\begin{aligned}
&\nu\Delta\omega + (\psi_{x_2}\omega_{x_1} - \psi_{x_1}\omega_{x_2}) = 0 \text{ in } \Omega,\\
&-\Delta\psi = \omega \text{ in } \Omega,\\
&\psi = 0 \text{ on } \partial\Omega,\\
&\frac{\partial\psi}{\partial n}(x_1, x_2)_{|\partial\Omega} = \begin{cases} 1 & \text{if } x_2 = 1,\\ 0 & \text{if } 0 \le x_2 < 1,\end{cases}
\end{aligned}
$$

where $\Omega$ is the unit square and the viscosity $\nu$ is the reciprocal of the Reynolds number $Re$. In terms of $\psi$ alone this can be written as

$$\nu\Delta^2\psi + (\psi_{x_2}(\Delta\psi)_{x_1} - \psi_{x_1}(\Delta\psi)_{x_2}) = 0 \text{ in } \Omega,$$

subject to the same boundary conditions. This equation was discretized with finite differences on a $25 \times 25$ grid, see FIG. 4. The grid lines are distributed as the roots of the Chebychev polynomial of degree 25. As preconditioner we used the Modified ILU(2) [13] decomposition of the biharmonic operator $\Delta^2$. Starting from the solution for $Re = 0$, we computed several solutions, using the the arc length continuation method (cf. the previous example, and [12]) with step sizes $\Delta s = 100$ for $0 \le Re \le 1400$, $\Delta s = 200$ for $1400 < Re \le 2600$, and $\Delta s = 400$ for $Re = 3000$.

TAB. 4 shows the results of this test when using 10 steps of GMRES and CGS2 [10] for the correction equation. In the case of CGS2 we approximately solved the correction equation to a relative residual norm precision of $2^{-k}$, where $k$ is the current Newton step [8], with a maximum of 50 steps. Clearly, the methods using (the basis produced by) 10 steps of GMRES perform very poorly for this example. Only GNKS is able to complete the full continuation run, but requires a large number of Newton steps. If we look at the results for the **AIN** schemes for which CGS2 is used, we see that, except for the LINESEARCH method, these methods perform much better. The Restricted Minimal Residual methods are again the most efficient ones.

| GMRES | | | | | |
|---|---|---|---|---|---|
| Method | NI | FE | MV | PRE | TOTAL |
| LINESEARCH | fails at $Re = 400$ after a TOTAL of 545. | | | | |
| DOGLEG | fails at $Re = 100$ after a TOTAL of 113. | | | | |
| NNGCG | fails at $Re = 2200$ after a TOTAL of 19875 | | | | |
| GNKS | 641 | 2315 | 30206 | 6210 | 38731 |
| RMR A | fails at $Re = 2000$ after a TOTAL of 13078 | | | | |
| RMR B | fails at $Re = 800$ after a TOTAL of 7728 | | | | |
| **CGS2** | | | | | |
| Method | NI | FE | MV | PRE | TOTAL |
| LINESEARCH | fails at $Re = 1300$ after a TOTAL of 4342. | | | | |
| NNGCG | 141 | 555 | 7003 | 6119 | 13677 |
| RMR A | 137 | 297 | 6266 | 5969 | 12532 |
| RMR B | 137 | 167 | 6277 | 5937 | 12381 |

TABLE 4: Results for the Driven Cavity problem, solved by the arc length continuation method for $Re = 100, \ldots, 3000$.

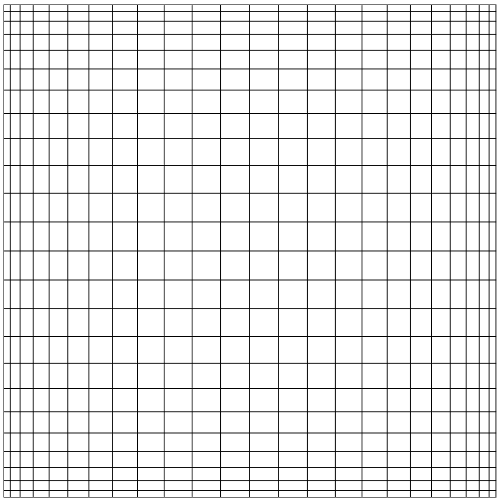

FIGURE 4: Grid for the Driven Cavity problem, $(25 \times 25)$.

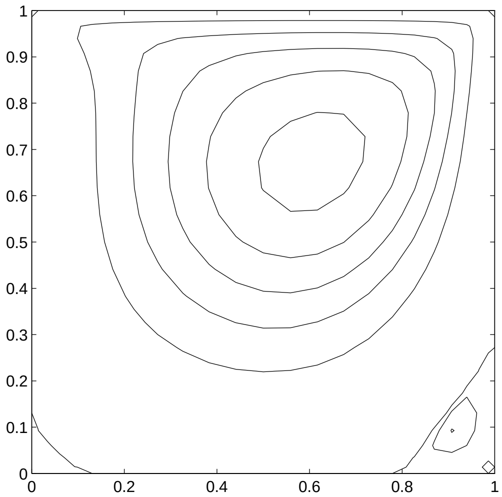


FIGURE 5: Stream lines of the Driven Cavity problem, $Re = 100$.

This test also reveals a possible practical drawback of methods like DOGLEG and GNKS. These methods exploit an affine subspace to find a suitable update for the approximation. This may fail, when the problem is hard or when the preconditioner is not good enough. In that case the dimension of the affine subspace must be large, which may be, because of storage requirements and computational overhead, not feasible. For the schemes that use approximate solutions of the correction equation, delivered by some arbitrary, iterative method, e.g., CGS2, one can easily adapt the precision, which leaves more freedom.

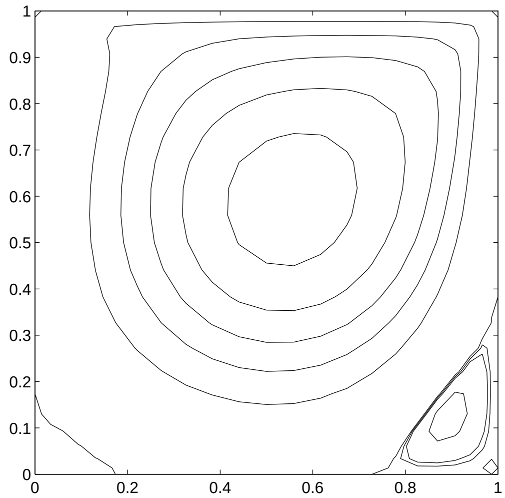


FIGURE 6: Stream lines of the Driven Cavity problem, $Re = 400$.

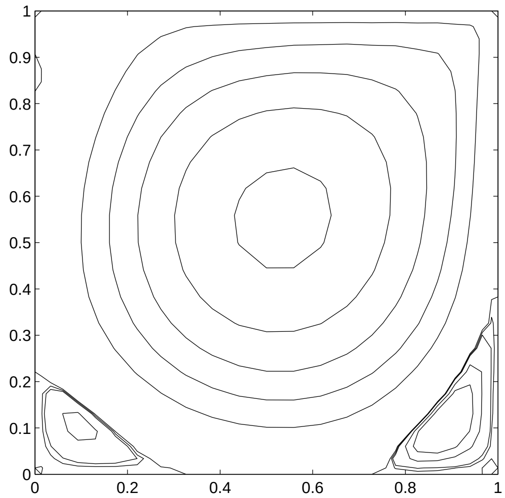


FIGURE 7: Stream lines of the Driven Cavity problem, $Re = 1600$.

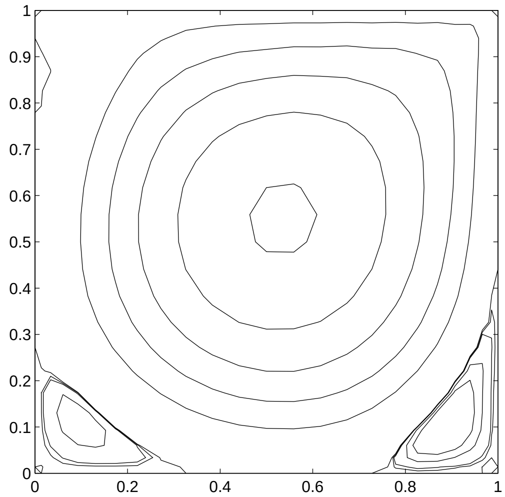


FIGURE 8: Stream lines of the Driven Cavity problem, $Re = 2000$.

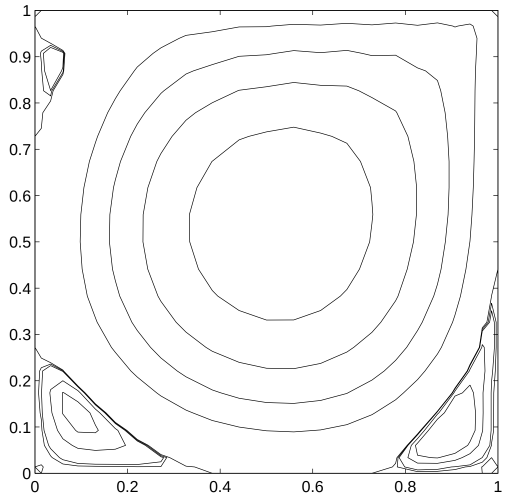


FIGURE 9: Stream lines of the Driven Cavity problem, $Re = 3000$.

Plots of the stream lines for the values

$$\psi = -\,0.12,\ -0.1,\ -0.08,\ -0.06,\ -0.04,\ -0.02,$$
$$0.0,\ 0.0025,\ 0.001,\ 0.0005,\ 0.0001,\ 0.00005$$

(cf. [12]) are given in FIG. 5–9. The plots show virtually the same solutions as in [12].

**9. Conclusions.** We have shown how the classical Newton iteration scheme for nonlinear problems can be accelerated in a similar way as standard Richardson-type iteration schemes for linear equations. This leads to the **AIN** framework in which many well-known iterative methods for linear–, eigen–, and general nonlinear problems fit. From this framework an overwhelming number of possible iterations schemes can be formulated. We have selected a few and shown by numerical experiments that especially the Restricted Minimal Residual methods can be very useful for further reducing computational costs.

## REFERENCES


[1] W. E. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.

[2] O. Axelsson and A. T. Chronopoulos, *On nonlinear generalized conjugate gradient methods*, Numer. Math., 69 (1994), pp. 1–16.

[3] J. G. L. Booten, H. A. Van der Vorst, P. M. Meijer, and H. J. J. te Riele, *A preconditioned Jacobi-Davidson method for solving large generalized eigenvalue problems*, Report NM-R9414, Dept. Num. Math., CWI, Amsterdam, 1994.

[4] P. N. Brown and Y. Saad, *Hybrid Krylov methods for nonlinear systems of equations*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 450–481.

[5] M. Crouzeix, B. Philippe, and M. Sadkane, *The Davidson method*, SIAM J. Sci. Comput., 15 (1994), pp. 62–76.

[6] E. R. Davidson, *The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real symmetric matrices*, J. Comp. Phys., 17 (1975), pp. 87–94.

[7] E. De Sturler and D. R. Fokkema, *Nested Krylov methods and preserving the orthogonality*, in Sixth Copper Mountain Conference on Multigrid Methods, N. D. Melson, T. A. Manteuffel, and S. F. McCormick, eds., vol. Part 1 of NASA Conference Publication 3324, NASA, 1993, pp. 111–126.

[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, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, New Yersey 07632, 1983.

[10] D. R. Fokkema, G. L. G. Sleijpen, and H. A. Van der Vorst, *Generalized conjugate gradient squared*, Preprint 851, Department of Mathematics, Utrecht University, Utrecht, the Netherlands, May 1994. To appear in J. Comput. Appl. Math..

[11] R. W. Freund and N. M. Nachtigal, *QMR: A quasi minimal residual method for non-Hermitian linear systems*, Numer. Math., 60 (1991), pp. 315–339.

[12] R. Glowinski, H. B. Keller, and L. Reinhart, *Continuation-conjugate gradient methods for the least squares solution of nonlinear boundary value problems*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 793–832.

[13] I. Gustafsson, *A class of first order factorizations methods*, BIT, 18 (1978), pp. 142–156.

[14] E. Hopf, *The partial differential equation $u_t + uu_x = \mu u_{xx}$*, Communications Pure Applied Mathematics, 3 (1950), pp. 201–230.

[15] I. E. Kaporin and O. Axelsson, *On a class of nonlinear equation solvers based on the residual norm reduction over a sequence of affine subspaces*, SIAM J. Sci. Comput., 16 (1995).

[16] T. Kerkhoven and Y. Saad, *Acceleration techniques for decoupling algorithms in semiconductor simulation*, Report UIUCDCS-R-87-1363, Dep. of Comp. Sci., University of Illinois, Urbana, 1987.

[17] C. Lanczos, *Solution of systems of linear equations by minimized iteration*, J. Res. Nat. Bur. Standards, 49 (1952), pp. 33–53.

[18] J. A. Meijerink and H. A. Van der Vorst, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp., 31 (1977), pp. 148–162.

[19] C. B. Moler and G. W. Stewart, *An algorithm for generalized matrix eigenvalue problems*, SIAM J. Numer. Anal., 10 (1973), pp. 241–256.

[20] J. J. Moré, *The Levenberg-Marquardt algorithm: Implementation and theory*, in Numerical Analysis, G. A. Watson, ed., vol. 630 of Lecture Notes in Mathematics, Springer-Verlag, Berlin, Heidelberg, New York, 1977, ch. 105-116.

[21] R. B. Morgan, *Generalizations of Davidson's method for computing eigenvalues of large non-symmetric matrices*, J. Comput. Phys., 101 (1992), pp. 287–291.

[22] C. C. Paige, B. N. Parlett, and H. A. Van der Vorst, *Approximate solutions and eigenvalue bounds from Krylov subspaces*, Num. Lin. Alg. Appl., 2 (1995), pp. 115–134.

[23] Y. Saad, *Krylov subspace method for solving large unsymmetric linear systems*, Math. Comp., 37 (1981), pp. 105–126.

[24] ———, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, 1992.

[25] ———, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.

[26] Y. Saad and M. H. Schultz, *GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[27] G. L. G. Sleijpen, J. G. L. Booten, D. R. Fokkema, and H. A. Van der Vorst, *Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems: Part I*, Preprint 923, Dept. Math., University Utrecht, Utrecht, The Netherlands, 1995.

[28] G. L. G. Sleijpen and H. A. Van der Vorst, *A Jacobi-Davidson iteration method for linear eigenvalue problems*, Preprint 856, Dept. Math., University Utrecht, Utrecht, 1994. Revised verion, May 1995: To appear in SIAM J. Matrix Anal. Appl.

[29] H. A. Van der Vorst and C. Vuik, *GMRESR: A family of nested GMRES methods*, Num. Lin. Alg. Appl., 1 (1994), pp. 369–386.

[30] C. Vuik, *Further experiences with GMRESR*, Tech. Report 92-12, Faculty of Technical Mathematics and Informatics, Delft University of Technology, Delft, The Netherlands, 1992.

[31] D. M. Young and K. C. Jea, *Generalized conjugate-gradient acceleration of nonsymmetrizable iterative methods*, Linear Algebra Appl., 34 (1980), pp. 159–194.