SIAM J. Numer. Anal.
Vol. 47, No. 6, pp. 4660–4679

ON THE SIMILARITIES BETWEEN THE QUASI-NEWTON INVERSE LEAST SQUARES METHOD AND GMRES*

ROB HAELTERMAN†, JORIS DEGROOTE‡, DIRK VAN HEULE†, AND JAN VIERENDEELS‡

Abstract. We show how one of the best-known Krylov subspace methods, the generalized minimal residual method (GMRes), can be interpreted as a quasi-Newton method and how the quasi-Newton inverse least squares method (QN-ILS) relates to Krylov subspace methods in general and to GMRes in particular when applied to linear systems. We also show that we can modify QN-ILS in order to make it analytically equivalent to GMRes, without the need for extra matrix-vector products.

Key words. quasi-Newton method, iterative method, linear algebra, least squares, generalized minimal residual method, rank-one update, secant method

AMS subject classifications. 65F10, 90C53, 90C56

DOI. 10.1137/090750354

1. Introduction.

In this paper we start from a linear system of equations given by

$$K(p) = A_K p - b = 0, \tag{1.1}$$

where $b, p \in \mathbb{R}^{n\times 1}$ and $A_K \in \mathbb{R}^{n\times n}$ is nonsingular. K is an affine operator $\mathbb{R}^{n\times 1} \to \mathbb{R}^{n\times 1}$. We will write the solution of (1.1) as p^*.

We assume that we have no knowledge of A_K or b whatsoever but are able to form either $A_K x$ or $A_K x - b$ for all $x \in \mathbb{R}^{n\times 1}$. We also assume that it is impossible to form either $A_K^T x$ or $A_K^T x - b$.

We will solve (1.1) in an iterative way, starting from an initial value p_o, using the following formulation of the quasi-Newton methods:

$$p_{s+1} = p_s - \hat{M}'_s r_s, \tag{1.2}$$

where the residual r_s is defined as $A_K p_s - b$ and $\hat{M}'_s$ serves as an approximation for the inverse of the Jacobian of K, i.e., A_K^{-1}. If we write $e = p - p^*$, $e_s = p_s - p^*$, and so on, for the errors, then the resulting error, resp. residual, equation is

$$e_{s+1} = e_s - \hat{M}'_s A_K e_s, \tag{1.3a}$$

$$r_{s+1} = r_s - A_K \hat{M}'_s r_s. \tag{1.3b}$$

We count the performance of a method by the number of times a matrix-vector product is computed as we assume that this forms the most expensive part of any algorithm.

The methods that we investigate are the generalized minimal residual method (GMRes) [14] and the quasi-Newton inverse least squares method (QN-ILS) [6].

*Received by the editors February 20, 2009; accepted for publication (in revised form) November 30, 2009; published electronically February 19, 2010.
http://www.siam.org/journals/sinum/47-6/75035.html

†Department of Mathematics, Royal Military Academy, Renaissancelaan 30, B-1000 Brussels, Belgium (Robby.Haelterman@rma.ac.be, Dirk.Van.Heule@rma.ac.be).

‡Department of Flow, Heat and Combustion Mechanics, Ghent University, St.-Pietersnieuwstraat 41, B-9000 Gent, Belgium (Joris.Degroote@uGent.be, Jan.Vierendeels@uGent.be).

Cătinaş [4] has shown in an abstract framework that inexact Newton methods (like GMRes) are equivalent to quasi-Newton methods. In this paper we will show the relationship between the two methods under consideration in more detail. Note in this context that an investigation with respect to the similarities between two other quasi-Newton methods—the Eirola-Nevanlinna method and Broyden's method—and GMRes has been performed by Vuik and van der Vorst in [19].

The paper is organized as follows: In section 2 we give some introductory theorems and definitions; in section 3 we give a brief overview of the GMRes method and its most important properties and interpret it as a quasi-Newton method; in section 4 we define QN-ILS, give some of its properties, and show how it relates to Krylov subspace methods; in section 5 we show that we can modify QN-ILS such that it becomes analytically equivalent to GMRes, without the need for extra matrix-vector products. In section 6 we test the different algorithms on three numerical problems.

2. Introductory definitions and theorems.

All matrix norms that are used are natural matrix norms, unless otherwise stated. $\langle \cdot, \cdot \rangle$ denotes the standard scalar product between vectors.

Most of the definitions and proofs of theorems in this section can be found in [13].

Definition 2.1. *If $P^2 = P$ $(P \in \mathbb{R}^{n\times n})$ we say that P is a projection matrix.*

Some properties of a projection matrix are [13] the following:

- P defines a projection onto $\mathcal{R}(P)$ parallel to $\mathcal{N}(P)$, where $\mathcal{R}(\cdot)$ denotes the range and $\mathcal{N}(\cdot)$ the kernel or null space.
- If P is a projection matrix, then so is $(I - P)$.
- $\mathcal{N}(P) = \mathcal{R}(I - P)$.
- $\mathcal{N}(P) \oplus \mathcal{R}(P) = \mathbb{R}^{n\times 1}$ where $\oplus$ denotes the direct sum of two subspaces.
- Let P be of rank m, $\{v_1, \ldots, v_m\}$ a basis for $\mathcal{R}(P)$, and $\{w_1, \ldots, w_m\}$ a basis for $(\mathcal{N}(P))^\perp$, where $(\cdot)^\perp$ defines the orthogonal complement of a subspace. If $V = [v_1|\ldots|v_m], W = [w_1|\ldots|w_m]$, then $P = V(W^TV)^{-1}W^T$.

Definition 2.2 (see [13]). *Let $\mathcal{Y}_m$ and $\mathcal{Z}_m$ be two subspaces of $\mathbb{R}^{n\times 1}$ of dimension $m \le n$.*

A projection method for solving the linear system given in (1.1) *is a method in which an approximate solution $\tilde{p}$ is found in an affine subspace $p_o + \mathcal{Y}_m$, where p_o is an initial guess, and in which a Petrov–Galerkin condition is imposed:*

$$\tilde{r} = A_K\tilde{p} - b \perp \mathcal{Z}_m.$$

We call $p_o + \mathcal{Y}_m$ the search subspace and $\mathcal{Z}_m$ the subspace of constraints.

If $\mathcal{Y}_m = \mathcal{Z}_m$, we say the projection method is orthogonal; otherwise, we say it is oblique.

If $\{v_1, \ldots, v_m\}$ is a basis for $\mathcal{Z}_m$, $\{w_1, \ldots, w_m\}$ is a basis for $\mathcal{Y}_m$, and $V = [v_1|\ldots|v_m], W = [w_1|\ldots|w_m]$, then a projection method (Definition 2.2) will result in the approximation $\tilde{p}$ given by

$$\tilde{p} = p_o - W(V^TA_KW)^{-1}V^Tr_o, \tag{2.1}$$

if V^TA_KW is nonsingular [13].

V^TA_KW is guaranteed to be nonsingular if one of the following conditions holds [13]:

1. A_K is positive definite and $\mathcal{Z}_m = \mathcal{Y}_m$;
2. A_K is nonsingular and $\mathcal{Z}_m = A_K\mathcal{Y}_m$.

DEFINITION 2.3. *We define a Krylov subspace of dimension s generated by $C \in \mathbb{R}^{n\times n}$ and $v \in \mathbb{R}^{n\times 1}$ as*

$$\mathcal{K}_s\{C; v\} = \operatorname{span}\{v, Cv, C^2v, \ldots, C^{s-1}v\}.$$

DEFINITION 2.4. *A Krylov subspace method to solve* (1.1) *is a projection method where for the mth iterate p_m we have $\mathcal{Y}_m = \mathcal{K}_m\{A_K; r_o\}$* [13].

The choice of $\mathcal{Z}_m$ (Definition 2.2) will define the particular Krylov subspace method.

THEOREM 2.5. *Let $A_K \in \mathbb{R}^{n\times n}$ be an arbitrary matrix and assume that we have $\mathcal{Z}_m = A_K\mathcal{Y}_m$. Then a vector $\tilde{p}$ is the result of an oblique projection method onto $\mathcal{Y}_m$ orthogonally to $\mathcal{Z}_m$ with the starting iterate p_o if and only if it minimizes the Euclidean norm of the residual vector $\tilde{r} = A_K\tilde{p} - b$ over $p \in p_o + \mathcal{Y}_m$.*

THEOREM 2.6. *Let $A_K \in \mathbb{R}^{n\times n}$ be an arbitrary matrix. Let $\tilde{p}$ be the approximate solution obtained from a projection process onto $\mathcal{Y}_m$ orthogonally to $\mathcal{Z}_m = A_K\mathcal{Y}_m$ and let $\tilde{r} = A_K\tilde{p} - b$ be the associated residual vector. Then*

$$\tilde{r} = (I - P)r_o,$$

where P denotes the orthogonal projection matrix onto the subspace $\mathcal{Z}_m$.

The class of methods respecting Theorem 2.6 is called *residual projection methods.*

LEMMA 2.7. *Let $V \in \mathbb{R}^{n\times s}$ be a matrix with column-rank s, and then*

$$V(V^TV)^{-1}V^T = \mathcal{L}\mathcal{L}^T, \tag{2.2}$$

with $\mathcal{L} = [\bar{L}_1|\bar{L}_2|\ \ldots\ |\bar{L}_s]$, where $\{\bar{L}_k\}_{k=1}^s$ is an orthonormal basis for the range of V.

Proof. If $\{\bar{L}_k\}_{k=1}^s$ is an orthonormal basis for the range of V, then there exists a nonsingular matrix $Q \in \mathbb{R}^{s\times s} : V = \mathcal{L}Q$. We can then rewrite (2.2) as follows:

$$\begin{aligned} V(V^TV)^{-1}V^T &= (\mathcal{L}Q)((\mathcal{L}Q)^T(\mathcal{L}Q))^{-1}(\mathcal{L}Q)^T \\ &= (\mathcal{L}Q)(Q^T\mathcal{L}^T\mathcal{L}Q)^{-1}(\mathcal{L}Q)^T. \end{aligned}$$

As $\mathcal{L}$ is an orthogonal matrix, this reduces to

$$\begin{aligned} V(V^TV)^{-1}V^T &= (\mathcal{L}Q)(Q^TQ)^{-1}(\mathcal{L}Q)^T \\ &= \mathcal{L}QQ^{-1}(Q^T)^{-1}Q^T\mathcal{L}^T \\ &= \mathcal{L}\mathcal{L}^T, \end{aligned}$$

which completes the proof. ▯

Note that $V(V^TV)^{-1}V^T = \mathcal{L}\mathcal{L}^T$ is an orthogonal projection matrix on the range of V.

LEMMA 2.8.

$$\forall \mathcal{A} \vec{\subset} \mathbb{R}^{n\times 1} : \|QP_{\mathcal{A}}\| = \sup_{x\in\mathcal{A}, \|x\|=1} \|Qx\|, \tag{2.3}$$

where $P_{\mathcal{A}}$ is an orthogonal projection matrix on $\mathcal{A}$ and $Q \in \mathbb{R}^{n\times n}$.

Proof. We first prove that

$$\|QP_{\mathcal{A}}\| \geq \sup_{x\in\mathcal{A}, \|x\|=1} \|Qx\|. \tag{2.4}$$

As $\{x \in \mathcal{A}\ ;\ \|x\| = 1\} \subset \{x \in \mathbb{R}^{n\times 1}\ ;\ \|x\| = 1\}$ and as $P_{\mathcal{A}}x = x$ $\forall x \in \mathcal{A}$, we have

$$\sup_{x\in\mathcal{A},\|x\|=1} \|Qx\| = \sup_{x\in\mathcal{A},\|x\|=1} \|QP_{\mathcal{A}}x\| \leq \sup_{x\in\mathbb{R}^{n\times 1},\|x\|=1} \|QP_{\mathcal{A}}x\| = \|QP_{\mathcal{A}}\|.$$

We now prove that

$$\|QP_{\mathcal{A}}\| \leq \sup_{x\in\mathcal{A},\|x\|=1} \|Qx\|. \tag{2.5}$$

$\forall x \in \mathbb{R}^{n\times 1}$ for which $\|x\| = 1$ and $P_{\mathcal{A}}x \neq 0$ there exists a $x_{\mathcal{A}} \in \mathcal{A}$ such that $\|x_{\mathcal{A}}\| = 1$ and $\|QP_{\mathcal{A}}x\| \leq \|QP_{\mathcal{A}}x_{\mathcal{A}}\| = \|Qx_{\mathcal{A}}\|$.

To show this, it suffices to take $x_{\mathcal{A}} = kP_{\mathcal{A}}x$ with

$$k = \frac{1}{\|P_{\mathcal{A}}x\|} \geq 1.$$

It follows that $\|x_{\mathcal{A}}\| = k\|P_{\mathcal{A}}x\| = 1$ and $\|Qx_{\mathcal{A}}\| = k\|QP_{\mathcal{A}}x\| \geq \|QP_{\mathcal{A}}x\|$.

Consequently we have

$$\|QP_{\mathcal{A}}x\| \leq \sup_{y\in\mathcal{A},\|y\|=1} \|Qy\|. \tag{2.6}$$

Equation (2.6) is also valid when $P_{\mathcal{A}}x = 0$, and hence

$$\|QP_{\mathcal{A}}\| = \sup_{x\in\mathbb{R}^{n\times 1},\|x\|=1} \|QP_{\mathcal{A}}x\| \leq \sup_{x\in\mathcal{A},\|x\|=1} \|Qx\|.$$

Combining (2.4) and (2.5) we obtain (2.3). □

LEMMA 2.9. *If $\mathcal{A}\vec{\subset}\mathcal{B}\vec{\subset}\mathbb{R}^{n\times 1}$, then $\|QP_{\mathcal{A}}\| \leq \|QP_{\mathcal{B}}\|$, where $P_{\mathcal{A}}$, resp. $P_{\mathcal{B}}$, is an orthogonal projection matrix on $\mathcal{A}$, resp. $\mathcal{B}$, and $Q \in \mathbb{R}^{n\times n}$.*

Proof. Using Lemma 2.8 we obtain

$$\|QP_{\mathcal{A}}\| = \sup_{x\in\mathcal{A},\|x\|=1} \|Qx\| \leq \sup_{x\in\mathcal{B},\|x\|=1} \|Qx\| = \|QP_{\mathcal{B}}\|. \qquad \square$$

THEOREM 2.10. *Let $T_1, T_2 \in \mathbb{R}^{n\times n}$. Let $\{X_s\}_{s\in[1,n]}$ be an arbitrary sequence of vectors ($X_s \in \mathbb{R}^{n\times 1}$) that are linearly independent, let $\{V_s\}_{s\in[1,n]}$ be a sequence defined by $V_s = [X_1|\ X_2|\ \ldots|\ X_s]$, and let $\{J_s\}_{s\in[1,n]}$ be a sequence defined by*

$$J_s = T_1V_s(V_s^TV_s)^{-1}V_s^T - T_2;$$

then

$$\|J_{s+1} - (T_1 - T_2)\| \leq \|J_s - (T_1 - T_2)\|,$$

$s = 1, 2, \ldots.$

Proof. We first note that the rank of V_s is s and that for $s = n$ we will have

$$J_n = T_1V_n(V_n^TV_n)^{-1}V_n^T - T_2 = T_1 - T_2,$$

and hence $\|J_n - (T_1 - T_2)\| = 0$.

For $s < n$ Lemma 2.7 gives us

$$J_s = T_1\mathcal{L}_s\mathcal{L}_s^T - T_2,$$

where $\mathcal{L}_s = [\bar{L}_1|\bar{L}_2|\ \ldots\ |\bar{L}_s]$, with $\{\bar{L}_k\}_{k=1}^s$ an orthonormal basis for the range of V_s.

Hence

$$\begin{aligned} J_s - (T_1 - T_2) &= T_1(\mathcal{L}_s\mathcal{L}_s^T - I), \\ J_{s+1} - (T_1 - T_2) &= T_1(\mathcal{L}_{s+1}\mathcal{L}_{s+1}^T - I). \end{aligned}$$

Introducing P_s, resp. P_{s+1}, as the orthogonal projection matrix on $(\mathcal{R}(V_s))^{\perp}$, resp. $(\mathcal{R}(V_{s+1}))^{\perp}$, we obtain

$$\begin{aligned} J_s - (T_1 - T_2) &= T_1 P_s, \\ J_{s+1} - (T_1 - T_2) &= T_1 P_{s+1}. \end{aligned}$$

As $(\mathcal{R}(V_{s+1}))\,^{\perp}\, \vec{\subset}(\mathcal{R}(V_s))^{\perp}$ we can use Lemma 2.9 to obtain

$$\begin{aligned} \|T_1 P_{s+1}\| &\leq \|T_1 P_s\|, \\ \|J_{s+1} - (T_1 - T_2)\| &\leq \|J_s - (T_1 - T_2)\|, \end{aligned}$$

which completes our proof. □

3. The generalized minimal residual method. GMRes is a Krylov subspace method where at the mth iterate we have $\mathcal{Y}_m = \mathcal{K}_m\{A_K; r_o\}$, $\mathcal{Z}_m = A_K\mathcal{Y}_m$. It is thus an oblique projection method (Definition 2.2).

We will analyze GMRes from an analytical point of view, meaning that we will present it only in its most basic form (Algorithm 3.1). Implementations that are numerically more stable can be found in most textbooks dealing with Krylov methods, e.g., [13, 16].

In the GMRes method it is assumed that for every $x \in \mathbb{R}^{n\times 1}$ we are able to form $A_K x$.

Algorithm 3.1. *GMRes.*

```
1. Take a starting value $p_o$;
   $r_o = A_K p_o - b$;
   Set $s = 1$.
2. Loop until sufficiently converged:
   a. Compute $A_K^s r_o$.
   b. Find coefficients $\{\bar{\omega}_{i,s}\}_{i=1}^{i=s}$ that minimize $\|r_s\|_2$ with $r_s = r_o + \sum_{i=1}^{s} \omega_{i,s} A_K^i r_o$.
   c. Optionally: $p_s = p_o + \sum_{i=1}^{s} \bar{\omega}_{i,s} A_K^{i-1} r_o$.
   d. Set $s = s + 1$.
```

Note that the computation of p_s in step 2.c is optional and is normally performed only when the algorithm has sufficiently converged.

In other words, in the GMRes method we construct

$$p_s = p_o + [r_o\ A_K r_o \ldots A_K^{s-1} r_o][\bar{\omega}_{1,s}\ \bar{\omega}_{2,s} \ldots \bar{\omega}_{s,s}]^T$$

such that

$$r_s = r_o + [A_K r_o\ A_K^2 r_o \ldots A_K^s r_o][\bar{\omega}_{1,s}\ \bar{\omega}_{2,s} \ldots \bar{\omega}_{s,s}]^T$$

is minimal in the Euclidean norm.

We introduce $V_s^{GM} = [A_K r_o|\ A_K^2 r_o| \ldots |A_K^s r_o]$ and $W_s^{GM} = [r_o|A_K r_o| \ldots |A_K^{s-1} r_o] = A_K^{-1} V_s^{GM}$.

Using Lemma 2.7 we see that

$$V_s^{GM}\left((V_s^{GM})^T V_s^{GM}\right)^{-1}(V_s^{GM})^T = \mathcal{L}_s^{GM}(\mathcal{L}_s^{GM})^T \tag{3.1}$$

and

$$W_s^{GM}\left((V_s^{GM})^T V_s^{GM}\right)^{-1}(V_s^{GM})^T = A_K^{-1}\mathcal{L}_s^{GM}(\mathcal{L}_s^{GM})^T, \tag{3.2}$$

where the columns of $\mathcal{L}_s^{GM}$ form an orthonormal basis for the range of V_s^{GM} as shown in Lemma 2.7 and $\mathcal{L}_s^{GM}(\mathcal{L}_s^{GM})^T$ acts as an orthogonal projection matrix on the range of V_s^{GM}.

As for GMRes we have $\mathcal{Z}_m = A_K\mathcal{Y}_m$, and we can use Theorem 2.6 to obtain

$$r_s = \left(I - V_s^{GM}\left((V_s^{GM})^T V_s^{GM}\right)^{-1}(V_s^{GM})^T\right) r_o$$

and

$$p_s = p_o - W_s^{GM}\left((V_s^{GM})^T V_s^{GM}\right)^{-1}(V_s^{GM})^T r_o. \tag{3.3}$$

Using Theorem 2.10, with $T_1 = A_K^{-1}$ and T_2 equal to the zero matrix, we see that for $s \to n$ (3.2) converges monotonically to A_K^{-1}; similarly, (3.1) converges monotonically to I.

GMRes can thus be written as

$$\begin{aligned} p_{s+1} &= p_o - A_K^{-1}\mathcal{L}_s^{GM}(\mathcal{L}_s^{GM})^T r_o,\\ A_K(p_{s+1}-p_o) &= -\mathcal{L}_s^{GM}(\mathcal{L}_s^{GM})^T r_o,\\ A_K(p_{s+1}-p_o) - A_K(p_s-p_o) &= -\mathcal{L}_s^{GM}(\mathcal{L}_s^{GM})^T r_o - A_K(p_s-p_o),\\ A_K(p_{s+1}-p_s) &= -\mathcal{L}_s^{GM}(\mathcal{L}_s^{GM})^T r_o - A_K(e_s-e_o),\\ &= -\mathcal{L}_s^{GM}(\mathcal{L}_s^{GM})^T r_o - (r_s-r_o). \end{aligned}$$

As $(r_s - r_o) \in \mathcal{R}(V_s^{GM})$ and hence $\mathcal{L}_s^{GM}(\mathcal{L}_s^{GM})^T(r_s-r_o) = (r_s-r_o)$, this becomes

$$A_K(p_{s+1}-p_s) = -\mathcal{L}_s^{GM}(\mathcal{L}_s^{GM})^T r_s,$$

and we can thus write GMRes as

$$p_{s+1} = p_s - \underbrace{A_K^{-1}\mathcal{L}_s^{GM}(\mathcal{L}_s^{GM})^T}_{\hat{M}_s'} r_s, \tag{3.4}$$

where $\hat{M}_s'$ can be seen as an approximation to A_K^{-1}. Equation (3.4) corresponds to the form of (1.2), and hence GMRes can be considered as a quasi-Newton method applied to (1.1).

Also note that (3.3) corresponds to (2.1) and that the approximate Jacobian is guaranteed to be nonsingular due to the properties of a projection method.

4. The quasi-Newton inverse least squares method.

4.1. Construction of the algorithm. QN-ILS [6] is a variant of the quasi-Newton least squares method (QN-LS) [9]. Both were originally developed as nonlinear solvers, which still reflects in some of the notations that are used.

While we construct an approximation of the Jacobian of K in the QN-LS method, we construct an approximation of the inverse of that Jacobian in the QN-ILS method.

It is assumed that we can form $K(x) = A_K x - b$ for every $x \in \mathbb{R}^{n\times 1}$ (and not $A_K x$) and we introduce $H(x) = A_H x - b$ where $A_H = A_K + I$. The method can be described as follows.

If we define $M : \mathbb{R}^{n\times 1} \to \mathbb{R}^{n\times 1} : w \mapsto M(w)$ such that $M^{-1} = K$, then $(K'(M(w)))^{-1} = M'(w)$.

If we define $\mathcal{I}$ as the identical function $\mathcal{I} : \mathbb{R}^{n\times 1} \to \mathbb{R}^{n\times 1} : x \mapsto \mathcal{I}(x) = x$, then $H - K = \mathcal{I}$.

Furthermore, let $G(w) = H(K^{-1}(w))$, then

$$\begin{aligned}
G(w) - \mathcal{I}(w) &= H(K^{-1}(w)) - \mathcal{I}(w) \\
&= H(K^{-1}(w)) - K(K^{-1}(w)) \\
&= (H - K)(K^{-1}(w)) \\
&= \mathcal{I}(K^{-1}(w)) \\
&= K^{-1}(w).
\end{aligned}$$

It follows that $(K^{-1})'(w) = G'(w) - I = (K'(K^{-1}(w)))^{-1}$.

Hence, to approximate $(K'(K^{-1}(w)))^{-1}$ we can use the approximation $\hat{G}_s$ of $G'(w)$, using the same technique described in [17], which is given below.

$$\begin{aligned}
\Delta w_i^s &= w_s - w_i \ \ (i = \tilde{\mathfrak{n}}, \dots, s-1), \\
V_s^{wG} &= [\Delta w_{s-1}^s \ \Delta w_{s-2}^s \ \dots \ \Delta w_{\tilde{\mathfrak{n}}}^s] \in \mathbb{R}^{n\times \min\{s,n\}}, \\
\Delta G_i^s &= G(w_s) - G(w_i) \ \ (i = \tilde{\mathfrak{n}}, \dots, s-1), \\
W_s^{wG} &= [\Delta G_{s-1}^s \ \Delta G_{s-2}^s \ \dots \ \Delta G_{\tilde{\mathfrak{n}}}^s] \in \mathbb{R}^{n\times \min\{s,n\}},
\end{aligned}$$

where $\tilde{\mathfrak{n}} = \max\{0, s - n\}$.

$$\begin{aligned}
\hat{G}'_s &= W_s^{wG} \left((V_s^{wG})^T V_s^{wG} \right)^{-1} (V_s^{wG})^T, \\
\widehat{(K^{-1})}'_s &= \hat{G}'_s - I = W_s^{wG} \left((V_s^{wG})^T V_s^{wG} \right)^{-1} (V_s^{wG})^T - I.
\end{aligned}$$

Setting $K^{-1}(w_i) = p_i$ for $i = 0, 1, \dots$ (i.e., $w_i = K(p_i)$ and $G(w_i) = H(K^{-1}(w_i)) = H(p_i)$) and modifying the notation accordingly, we get

$$\widehat{(K^{-1})}'_s = \hat{M}'_s = W_s^{KH} \left((V_s^{wG})^T V_s^{wG} \right)^{-1} (V_s^{wG})^T - I, \tag{4.1}$$

where

$$\Delta K_i^s = K(p_s) - K(p_i) \ \ (i = \tilde{\mathfrak{n}}, \dots, s-1), \tag{4.2a}$$
$$V_s^{KH} = [\Delta K_{s-1}^s \ \Delta K_{s-2}^s \ \dots \ \Delta K_{\tilde{\mathfrak{n}}}^s] \in \mathbb{R}^{n\times \min\{s,n\}}, \tag{4.2b}$$
$$\Delta H_i^s = H(p_s) - H(p_i) \ \ (i = \tilde{\mathfrak{n}}, \dots, s-1), \tag{4.2c}$$
$$W_s^{KH} = [\Delta H_{s-1}^s \ \Delta H_{s-2}^s \ \dots \ \Delta H_{\tilde{\mathfrak{n}}}^s] \in \mathbb{R}^{n\times \min\{s,n\}}. \tag{4.2d}$$

For more details on the background of this method see [9, 17, 18].

Note that (4.1) can be used only for $s \geq 1$, because $\hat{M}'_s$ cannot be constructed earlier. We will therefore use $p_1 = H(p_o)$. As such, we could define $\hat{M}'_o$ as $\hat{M}'_o = -I$ to make (4.1) conform for $s = 0$. Alternatively, this could be seen as setting the initial approximation of $G'(w)$ equal to zero, which is as good as any guess, as we don't know anything about this Jacobian a priori.

The complete algorithm can be described as follows.

Algorithm 4.1. *QN-ILS.*

```
1. Startup.
   Take a starting value $p_o$;
   compute $p_1 = H(p_o)$.
   Set $s = 1$.
2. Loop until sufficiently converged:
   a. Compute $K(p_s)$ and $H(p_s) = K(p_s) + p_s$.
   b. Construct the approximation of the inverse of the Jacobian $\hat{M}'_s$
      using (4.1) and (4.2).
   c. Quasi-Newton step: $p_{s+1} = p_s - \hat{M}'_s K(p_s)$.
   d. Set $s = s + 1$.
```

4.2. General theorems regarding QN-ILS. The construction in section 4.1 corresponds to the formulation used in Lemma 2.7, as long as the columns of V_s^{KH} remain linearly independent. (See section 4.6 for more details about this requirement.)

As $W_s^{KH} = A_H A_K^{-1} V_s^{KH} = (A_K^{-1} + I)V_s^{KH}$, this means we can write

$$\hat{M}'_s = A_H A_K^{-1} \mathcal{L}_s^{KH} (\mathcal{L}_s^{KH})^T - I,$$

where the columns of $\mathcal{L}_s^{KH}$ form an orthonormal basis for the range of V_s^{KH}.

For the theoretical analysis we will assume that the orthonormal bases are constructed such that $\mathcal{L}_{s+1}^{KH} = [\mathcal{L}_s^{KH} | \bar{L}_{s+1}]$ where $\bar{L}_{s+1}$ is the newly added basis vector for the range of V_{s+1}^{KH}.

We also have that $\|\hat{M}'_{s+1} - A_K^{-1}\| \leq \|\hat{M}'_s - A_K^{-1}\|$ by applying Theorem 2.10 with $T_1 = A_H A_K^{-1}$ and $T_2 = I$, which means that $\hat{M}'_s$ converges in a monotone manner toward A_K^{-1} for $s \to n$.

Theorem 4.1. *Consider the method described in Algorithm* 4.1. *Assume $\hat{M}'_s$ and A_K are nonsingular. Then the following relation for the residual holds:*

$$\forall j \in \{0, 1, \ldots, s\} : \ (I - A_K \hat{M}'_s)(r_s - r_j) = 0. \tag{4.3}$$

Proof. As $\forall j \in \{0, 1, \ldots, s\} : \mathcal{R}(V_j^{KH}) \subset \mathcal{R}(V_s^{KH})$, we can write $\forall x \in \mathcal{R}(V_j^{KH}),\ j = 0, 1, \ldots, s$:

$$\begin{aligned} \hat{M}'_s x &= \left(A_H A_K^{-1} \mathcal{L}_s^{KH} (\mathcal{L}_s^{KH})^T - I\right) x \\ &= A_H A_K^{-1} x - x \\ &= (A_K + I) A_K^{-1} x - x \\ &= A_K^{-1} x \\ (I - A_K \hat{M}'_s) x &= 0. \end{aligned}$$

We can conclude that $\mathcal{R}(V_s^{KH})$ is part of the null space of $I - A_K \hat{M}'_s$.

From the definition of V_s^{KH} we have that $(r_s - r_j) \in \mathcal{R}(V_s^{KH})$ and (4.3) follows. ∎

Corollary 4.2. $\forall x \perp \mathcal{R}(V_j^{KH}),\ j = 0, 1, \ldots, s : \hat{M}'_s x = -x$.

Proof. $\forall x \perp \mathcal{R}(V_j^{KH}),\ j = 0, 1, \ldots, s$:

$$\begin{aligned} \hat{M}'_s x &= \left(A_H A_K^{-1} \mathcal{L}_s^{KH} (\mathcal{L}_s^{KH})^T - I\right) x, \\ \hat{M}'_s x &= A_H A_K^{-1} \mathcal{L}_s^{KH} (\mathcal{L}_s^{KH})^T x - x = 0 - x = -x. \qquad ∎ \end{aligned}$$

COROLLARY 4.3. *Consider the method described in Algorithm* 4.1 *applied to the linear problem given in* (1.1). *Assume* $\hat{M}'_s$ *is nonsingular. Then* $e_1 = A_H e_o$, $r_1 = A_H r_o$, *and, for* $s \geq 1$, $\exists \{\gamma_{i,s+1}\}_{i=1}^{i=s}$, *such that*

$$r_{s+1} = A_H r_o + A_H \sum_{i=1}^{s} \gamma_{i,s+1}(r_i - r_o), \tag{4.4}$$

$$e_{s+1} = A_H e_o + A_H \sum_{i=1}^{s} \gamma_{i,s+1}(e_i - e_o). \tag{4.5}$$

Proof. From Theorem 4.1 it follows that

$$\begin{aligned} r_{s+1} &= r_o - A_K \hat{M}'_s r_o \\ &= r_o - A_K((A_K + I)A_K^{-1}\mathcal{L}_s^{KH}(\mathcal{L}_s^{KH})^T - I)r_o \\ &= r_o - (\mathcal{L}_s^{KH}(\mathcal{L}_s^{KH})^T + A_K(I - \mathcal{L}_s^{KH}(\mathcal{L}_s^{KH})^T))r_o \\ &= (A_K + I)(I - \mathcal{L}_s^{KH}(\mathcal{L}_s^{KH})^T)r_o \\ &= A_H r_o - A_H \mathcal{L}_s^{KH}(\mathcal{L}_s^{KH})^T r_o. \end{aligned}$$

As $\mathcal{L}_s^{KH}(\mathcal{L}_s^{KH})^T$ is a projection operator on span$\{r_1 - r_o, r_2 - r_o, \dots, r_s - r_o\}$, the latter expression can be written as

$$r_{s+1} = A_H r_o + A_H \sum_{i=1}^{s} \gamma_{i,s+1}(r_i - r_o). \tag{4.6}$$

To prove (4.5) it suffices to multiply both sides of (4.6) by the inverse of A_K from the left and rearrange the terms as follows:

$$r_{s+1} = (A_K + I)r_o + (A_K + I)\sum_{i=1}^{s} \gamma_{i,s+1}(r_i - r_o),$$

$$\begin{aligned} A_K^{-1} r_{s+1} = e_{s+1} &= A_K^{-1}(A_K + I)r_o + A_K^{-1}(A_K + I)\sum_{i=1}^{s} \gamma_{i,s+1}(r_i - r_o) \\ &= (I + A_K^{-1})r_o + (I + A_K^{-1})\sum_{i=1}^{s} \gamma_{i,s+1}(r_i - r_o) \\ &= (I + A_K^{-1})A_K e_o + (I + A_K^{-1})A_K \sum_{i=1}^{s} \gamma_{i,s+1}(e_i - e_o) \\ &= A_H e_o + A_H \sum_{i=1}^{s} \gamma_{i,s+1}(e_i - e_o), \end{aligned}$$

which completes our proof. □

Consequence. According to Theorem 4.1, we have that $\forall j \in \{0, 1, \dots, s\}$

$$\begin{aligned} r_s - A_K \hat{M}'_s r_s &= r_j - A_K \hat{M}'_s r_j, \\ A_K e_s - A_K \hat{M}'_s r_s &= A_K e_j - A_K \hat{M}'_s r_j, \end{aligned}$$

and thus, for A_K nonsingular (and since $p_s - p_j = e_s - e_j$)

$$\begin{aligned} e_s - \hat{M}'_s r_s &= e_j - \hat{M}'_s r_j, \\ p_{s+1} = p_s - \hat{M}'_s K(p_s) &= p_j - \hat{M}'_s K(p_j). \end{aligned}$$

This means that the previous iterates only contribute to the solution process by creating a better approximate Jacobian $\hat{M}_s'$.

4.3. The secant property. The class of quasi-Newton methods encompasses every scheme where the inverse of the exact Jacobian (or, in the linear case, the system matrix) is replaced with an approximation $\hat{M}_s'$. Sometimes the name *quasi-Newton* has been used to describe the subclass of methods that respect the secant equation [3][1]

$$p_s - p_{s-1} = \hat{M}_s' \cdot (K(p_s) - K(p_{s-1})) \tag{4.7}$$

for all $s = 1, 2, \ldots$. Following the nomenclature of [7] we call the methods respecting (4.7) *secant methods* and prove that the QN-ILS algorithm is a secant method.

THEOREM 4.4. *Consider the method described in Algorithm* 4.1 *and assume* $\hat{M}_s'$ *is nonsingular; then the quasi-Newton iteration is a secant method; i.e., it satisfies the secant equation (4.7).*

Proof. We recall that $\hat{M}_s' = A_H A_K^{-1} \mathcal{L}_s^{KH} (\mathcal{L}_s^{KH})^T - I$. It follows that

$$\hat{M}_s' \cdot (K(p_s) - K(p_{s-1})) = (A_H A_K^{-1} \mathcal{L}_s^{KH} (\mathcal{L}_s^{KH})^T - I)(r_s - r_{s-1}).$$

As $(r_s - r_{s-1}) \in \mathcal{R}(V_s^{KH})$ and hence $\mathcal{L}_s^{KH} (\mathcal{L}_s^{KH})^T (r_s - r_{s-1}) = r_s - r_{s-1}$, this becomes

$$\begin{aligned} \hat{M}_s' \cdot (K(p_s) - K(p_{s-1})) &= (A_H A_K^{-1} - I)(r_s - r_{s-1}) \\ &= A_K^{-1}(r_s - r_{s-1}) \\ &= p_s - p_{s-1}. \qquad \square \end{aligned}$$

COROLLARY 4.5. *If K is an affine mapping, then for the QN-ILS method (Algorithm* 4.1*) the following expression holds:*

$$\forall x, y \in \mathbb{R}^{n \times 1} : x - y \in \mathcal{R}(V_s^{KH}) : x - y = \hat{M}_s'(K(x) - K(y)). \tag{4.8}$$

The proof is an immediate consequence of Theorem 4.4 and Lemma 2.7.

Remark. Similarly one can prove that GMRes, written as a quasi-Newton method (see (3.4)), is a secant method.

This means that $\hat{M}_s'$ is correct for every direction in $\mathcal{R}(V_s^{KH})$; for every direction $v \perp \mathcal{R}(V_s^{KH})$ we have $\hat{M}_s' v = -v$. The latter can be interpreted as setting the approximation of $G'(w)$ equal to zero in every direction of which no information is available.

We would like to point out that one can construct other methods than QN-ILS for which (4.8) holds, as this equation provides $s \times n$ equations only for the n^2 unknowns of the approximate Jacobian and is thus underdetermined.

Also note that, according to Theorem 2.10, ensuring that (4.8) holds for all previous iterates is beneficial when solving linear systems. However, this matter should be investigated in more detail for nonlinear systems, as secant properties obtained with points that are far from the actual solution might not be representative of the actual tangent hyperplane and thus hamper convergence.

COROLLARY 4.6. *Consider the method described in Algorithm* 4.1. *Then the following expression holds:*

$$r_{s+1} = (I - (\hat{M}_{s+1}')^{-1} \hat{M}_s') r_s. \tag{4.9}$$

Proof. We have $p_{s+1} = p_s - \hat{M}_s' r_s$ and (from Theorem 4.4) $\hat{M}_{s+1}'(r_{s+1} - r_s) = p_{s+1} - p_s$. Equation (4.9) follows. $\square$

[1] The secant equation is sometimes called *the fundamental equation of quasi-Newton methods.*

4.4. Obtaining $\hat{M}'_{s+1}$ by a rank-one update. We will now show that the Jacobian $\hat{M}'_{s+1}$ can be obtained by applying a rank-one update to $\hat{M}'_s$.

THEOREM 4.7. *Suppose that $\hat{M}'_s$ is constructed according to* (4.1)*; then $\hat{M}'_{s+1}$ is linked to $\hat{M}'_s$ by the following expression:*

$$\hat{M}'_{s+1} = \hat{M}'_s + A_H A_K^{-1} \bar{L}_{s+1} \bar{L}^T_{s+1}, \tag{4.10}$$

where $\bar{L}_{s+1}$ is the $(s+1)$th column of $\mathcal{L}^{KH}_{s+1}$, with $s < n$.

Proof. $\forall x \in \mathbb{R}^{n\times 1}$ we have, according to Lemma 2.7, that

$$\begin{aligned}(\hat{M}'_{s+1} - \hat{M}'_s)x &= (A_H A_K^{-1} \mathcal{L}^{KH}_{s+1} (\mathcal{L}^{KH}_{s+1})^T - I)x - (A_H A_K^{-1} \mathcal{L}^{KH}_{s} (\mathcal{L}^{KH}_{s})^T - I)x \\ &= A_H A_K^{-1} (\mathcal{L}^{KH}_{s+1} (\mathcal{L}^{KH}_{s+1})^T - \mathcal{L}^{KH}_{s} (\mathcal{L}^{KH}_{s})^T)x.\end{aligned}$$

As $\mathcal{R}(V^{KH}_s) \subset \mathcal{R}(V^{KH}_{s+1})$ we have

$$(\hat{M}'_{s+1} - \hat{M}'_s)x = A_H A_K^{-1} \bar{L}_{s+1} \bar{L}^T_{s+1} x.$$

As this is true $\forall x \in \mathbb{R}^{n\times 1}$, we obtain

$$\hat{M}'_{s+1} - \hat{M}'_s = A_H A_K^{-1} \bar{L}_{s+1} \bar{L}^T_{s+1},$$

which completes our proof. □

COROLLARY 4.8. *Consider the method described in Algorithm* 4.1. *Then the following expression holds:*

$$\delta p_{s+1} = -A_H A_K^{-1} \bar{L}_{s+1} \bar{L}^T_{s+1} r_s, \tag{4.11}$$

with $\delta p_s = p_{s+1} - p_s$, for $s = 0, 1, 2, \ldots$.

Proof. From Theorem 4.7 we have

$$I = (\hat{M}'_{s+1})^{-1} \hat{M}'_s + (\hat{M}'_{s+1})^{-1} A_H A_K^{-1} \bar{L}_{s+1} \bar{L}^T_{s+1}.$$

Inserting this in (4.9) (Corollary 4.6) gives

$$\begin{aligned} r_{s+1} &= (\hat{M}'_{s+1})^{-1} A_H A_K^{-1} \bar{L}_{s+1} \bar{L}^T_{s+1} r_s, \\ \hat{M}'_{s+1} r_{s+1} &= A_H A_K^{-1} \bar{L}_{s+1} \bar{L}^T_{s+1} r_s, \\ \delta p_{s+1} &= -A_H A_K^{-1} \bar{L}_{s+1} \bar{L}^T_{s+1} r_s, \end{aligned}$$

which completes our proof. □

Note that, by simple algebraic manipulation of (4.10), the rank-one update can also be written as

$$\hat{M}'_{s+1} = \hat{M}'_s + \frac{(\delta p_s - \hat{M}'_s \delta K_s)\left((I - \mathcal{L}^{KH}_s (\mathcal{L}^{KH}_s)^T)\delta K_s\right)^T}{\langle \delta K_s, (I - \mathcal{L}^{KH}_s (\mathcal{L}^{KH}_s)^T)\delta K_s \rangle}.$$

4.5. Comparing QN-ILS with Krylov subspace methods. We will now show that QN-ILS in the form of Algorithm 4.1 shows some similarities with Krylov subspace methods (Definition 2.4) as at the sth iterate we have $p_s \in \mathcal{Y}_s = \mathcal{K}_s\{A_K; r_o\}$, which is the same search subspace as GMRes; on the other hand, we have $r_s \perp \mathcal{Z}_{s-1} = (A_H^T)^{-1} A_K \mathcal{K}_{s-1}\{A_K; r_o\}$. As r_s is orthogonal only to an $(s-1)$-dimensional subspace, it is not a Krylov subspace method in the classical sense.

THEOREM 4.9. *Consider the method described in Algorithm* 4.1 *applied to the linear problem given in* (1.1). *Assume* $\hat{M}'_{s-1}$ *is nonsingular. Then we have that*

$$e_s \in e_o + \mathcal{K}_s\{A_K; r_o\}, \tag{4.12a}$$

$$p_s \in p_o + \mathcal{K}_s\{A_K; r_o\}, \tag{4.12b}$$

$$r_s \in r_o + A_K\mathcal{K}_s\{A_K; r_o\}. \tag{4.12c}$$

Proof. Let $\mathcal{P}_k = \{q(x) \in \mathbb{R}[x] : q(x) = \sum_{i=1}^{k} \kappa_i x^i\}$, i.e., the space of real polynomials of degree k, or lower, with zero constant term, where $q(A_K)$ represents a polynomial in A_K.

We first note that $\mathcal{P}_k$ over $\mathbb{R}$ is a vector-space of dimension k, and that as such

$$\forall t_1(x), t_2(x) \in \mathcal{P}_k, \forall \alpha_1, \alpha_2 \in \mathbb{R} : \alpha_1 t_1(x) + \alpha_2 t_2(x) \in \mathcal{P}_k$$

and that $\forall l \leq k = \mathcal{P}_l \subset \mathcal{P}_k$.

We will now give our proof by induction.

We know that

$$e_1 = A_H e_o = e_o + (A_H - I)e_o = e_o + q_1(A_K)e_o,$$

where $q_1 \in \mathcal{P}_1$ and $A_K = A_H - I$. We know (from Corollary 4.3) that

$$e_2 = A_H \sum_{i=1}^{1} \gamma_{i,2}(e_i - e_o) + A_H e_o = A_H \gamma_{1,2}(e_1 - e_o) + A_H e_o,$$
$$e_2 = \gamma_{1,2}(A_H - I)^2 e_o + (1 + \gamma_{1,2})(A_H - I)e_o + e_o,$$
$$e_2 = e_o + q_2(A_K)e_o,$$

where $q_2 \in \mathcal{P}_2$. We now prove that, if we have

$$e_k = e_o + q_k(A_K)e_o$$

for $k = 1, 2, \ldots, s-1$, where $q_k \in \mathcal{P}_k$, it follows that

$$e_s = e_o + q_s(A_K)e_o,$$

where $q_s \in \mathcal{P}_s$.

We have (from Corollary 4.3) that

$$e_s = A_H e_o + A_H \sum_{k=1}^{s-1} \gamma_{k,s}(e_k - e_o) = A_H e_o + A_H \sum_{k=1}^{s-1} \gamma_{k,s}(q_k(A_K)e_o).$$

Knowing that $\forall k \leq s-1 : q_k \in \mathcal{P}_k \Rightarrow q_k \in \mathcal{P}_{s-1}$ and that $\mathcal{P}_{s-1}$ is a vector-space over $\mathbb{R}$, we can write

$$e_s = A_H \tilde{q}_{s-1}(A_K)e_o + A_H e_o,$$

where $\tilde{q}_{s-1} \in \mathcal{P}_{s-1}$. We also have that

$$e_s = (A_H - I)\tilde{q}_{s-1}(A_K)e_o + \tilde{q}_{s-1}(A_K)e_o + (A_K)e_o + e_o.$$

As $\forall q(x) \in \mathcal{P}_k : xq(x) \in \mathcal{P}_{k+1}$ and as $x \in \mathcal{P}_{k+1}$ we can finally write

$$e_s = e_o + q_s(A_K)e_o,$$

where $q_s \in \mathcal{P}_s$.

Thus $e_{s+1} \in e_o + \mathrm{span}\{A_K e_o, A_K^2 e_o, A_K^3 e_o, \dots, A_K^s e_o\}$.

Noting that $\mathrm{span}\{A_K e_o, A_K^2 e_o, A_K^3 e_o, \dots, A_K^s e_o\} = \mathrm{span}\{r_o, A_K r_o, A_K^2 r_o, \dots, A_K^{s-1} r_o\}$ we have proven (4.12a). (4.12b) and (4.12c) follow immediately. □

THEOREM 4.10. *Consider the method described in Algorithm* 4.1 *applied to the linear problem given in* (1.1). *Assume* $\hat{M}'_{s-1}$ *is nonsingular. Then we have that* $r_s \perp (A_H^T)^{-1} A_K \mathcal{K}_{s-1}\{A_K; r_o\}$.

Proof. From (1.3b) we know that

$$r_s = r_{s-1} - A_K \hat{M}'_{s-1} r_{s-1}$$

and hence

$$\begin{aligned} r_s &= r_{s-1} - A_K (A_H A_K^{-1} \mathcal{L}_{s-1}^{KH} (\mathcal{L}_{s-1}^{KH})^T - I) r_{s-1} \\ &= r_{s-1} - ((I + A_K) \mathcal{L}_{s-1}^{KH} (\mathcal{L}_{s-1}^{KH})^T - A_K) r_{s-1} \\ &= (I + A_K)(I - \mathcal{L}_{s-1}^{KH} (\mathcal{L}_{s-1}^{KH})^T) r_{s-1} \\ &= A_H (I - \mathcal{L}_{s-1}^{KH} (\mathcal{L}_{s-1}^{KH})^T) r_{s-1}. \end{aligned}$$

It follows that $(I - \mathcal{L}_{s-1}^{KH} (\mathcal{L}_{s-1}^{KH})^T) r_{s-1} \in (\mathcal{R}(V_{s-1}^{KH}))^\perp$ and hence that

$$\exists z \in (\mathcal{R}(V_{s-1}^{KH}))^\perp : r_s = A_H z.$$

As $\forall x \in (\mathcal{R}(V_{s-1}^{KH}))^\perp, \forall y \in \mathcal{R}(V_{s-1}^{KH}) : \langle x, y \rangle = 0$, it follows that

$$\forall y \in \mathcal{R}(V_{s-1}^{KH}) : \langle r_s, (A_H^T)^{-1} y \rangle = 0.$$

From the definition of V_s^{KH} and (4.12c), we see that $\mathcal{R}(V_{s-1}^{KH}) = \mathcal{A}_K K_{s-1}\{A_K; r_o\}$ if V_{s-1}^{KH} is of rank $s-1$. r_s is thus orthogonal to $(A_H^T)^{-1} A_K \mathcal{K}_{s-1}\{A_K; r_o\}$. □

Note that $(A_H^T)^{-1} A_K \mathcal{K}_{s-1}\{A_K; r_o\}$ is only a subspace of dimension $s-1$; hence the method does not comply with the definition of a Krylov subspace method (Definition 2.4).

We would like to point out some further subtle differences between GMRes and QN-ILS. In the original framework in which QN-ILS was developed [6], the performance was measured by the number of "calls" $H(p)$, which in this context correspond to matrix-vector products with A_H (or A_K). If we start from p_o, then at the sth iteration GMRes has used $s+1$ matrix-vector products and p_n (or an earlier iterate) will be the solution [16]; for every iterate its residual is readily available.[2] In QN-ILS we have used only s matrix vector products to create p_s, and p_{n+1} (or an earlier iterate) will be the solution [9]; of course, in that case we will not have knowledge of r_{n+1}, which might be advisable when working in finite precision, for which an extra matrix-vector product would be needed. The fact that r_{n+1} is zero is reflected in Theorem 4.10.

Furthermore GMRes assumes that for every $x \in \mathbb{R}^{n \times 1}$ we can form $A_K x$. In the original problem setting of QN-ILS this was not available as we could only form $A_K x - b$; it was found that the most stable way to adapt GMRes to the latter setting was to explicitly compute b, which required an extra matrix-vector product [9].

[2] In practical implementations it is even so that only the residual, not the iterate, is available at every iteration.

4.6. Nonsingularity of the approximate Jacobian for QN-ILS.

We show in the next theorem that, in exact arithmetic, the approximate Jacobian for the QN-ILS method never becomes singular before the solution has been reached.

THEOREM 4.11. *Consider the QN-ILS method described in Algorithm* 4.1. *Let $\breve{s}$ be the first value for which $\delta K_{\breve{s}+1}$ $(= K_{\breve{s}+2} - K_{\breve{s}+1})$ is linearly dependent on $\delta K_o, \delta K_1, \ldots, \delta K_{\breve{s}}$, and then $p_{\breve{s}+2}$ is the solution of $K(p) = 0$.*

Proof. If $\delta K_{\breve{s}+1}$ is linearly dependent on $\delta K_o, \delta K_1, \ldots, \delta K_{\breve{s}}$, then

$$\delta K_{\breve{s}+1} = \sum_{j=0}^{\breve{s}} k_j \delta K_j,$$

$$\hat{M}'_{\breve{s}+1} \delta K_{\breve{s}+1} = \sum_{j=0}^{\breve{s}} k_j \hat{M}'_{\breve{s}+1} \delta K_j.$$

Then we have, according to Corollary 4.5, that

$$\hat{M}'_{\breve{s}+1} \delta K_{\breve{s}+1} = \sum_{j=0}^{\breve{s}} k_j A_K^{-1} \delta K_j = A_K^{-1} \delta K_{\breve{s}+1}. \tag{4.13}$$

Also

$$\begin{aligned} p_{\breve{s}+2} &= p_{\breve{s}+1} - \hat{M}'_{\breve{s}+1} K(p_{\breve{s}+1}), \\ K(p_{\breve{s}+2}) &= K(p_{\breve{s}+1}) - A_K \hat{M}'_{\breve{s}+1} K(p_{\breve{s}+1}). \qquad (4.14) \end{aligned}$$

We also have

$$K(p_{\breve{s}+2}) = K(p_{\breve{s}+1}) + \delta K_{\breve{s}+1}. \tag{4.15}$$

Using (4.13) in (4.15) we obtain

$$K(p_{\breve{s}+2}) = K(p_{\breve{s}+1}) + A_K \hat{M}'_{\breve{s}+1} \delta K_{\breve{s}+1}. \tag{4.16}$$

Combining (4.16) and (4.14) we get

$$\begin{aligned} (A_H - I) \hat{M}'_{\breve{s}+1} \delta K_{\breve{s}+1} &= -A_K \hat{M}'_{\breve{s}+1} K(p_{\breve{s}+1}), \\ A_K \hat{M}'_{\breve{s}+1} K(p_{\breve{s}+2}) &= 0, \end{aligned}$$

and as $\hat{M}'_{\breve{s}+1}$ and A_K are nonsingular it follows that $K(p_{\breve{s}+2}) = 0$ and that $p_{\breve{s}+2}$ is the solution of $K(p) = 0$. □

5. Adding a step-length parameter $\boldsymbol{\theta_s}$ to QN-ILS.

5.1. Long-term effect of step-length parameters.

We could consider modifying the basic quasi-Newton iteration (1.2) with a step-length parameter $\theta_s \in \mathbb{R}$:

$$p_{s+1} = p_s - \theta_s \hat{M}'_s r_s. \tag{5.1}$$

The resulting error, resp. residual, equation is

$$\begin{aligned} e_{s+1} &= e_s - \theta_s \hat{M}'_s A_K e_s, \\ r_{s+1} &= r_s - \theta_s A_K \hat{M}'_s r_s. \end{aligned}$$

We now show that the effect of the step-length parameter is limited to the current iterate.

Theorem 5.1. *Consider the method described in section* 4.1*, but using a relaxation factor as in* (5.1)*, then the choice of $\theta_s \in \mathbb{R}_o$ is irrelevant for the value of* p_{s+2}[3].

Proof. We have, by posing $r_s = r_{s,\|} + r_{s,\perp}$, where $r_{s,\|} \in \mathcal{R}(V_s^{KH})$, $r_{s,\perp} \in (\mathcal{R}(V_s^{KH}))^{\perp}$ and using Theorem 4.1, that

$$\begin{aligned} r_{s+1} &= r_s - \theta_s A_K \hat{M}_s' r_s \\ &= r_{s,\|} + r_{s,\perp} - \theta_s A_K \hat{M}_s' r_{s,\perp} - \theta_s r_{s,\|} \\ &= (1-\theta_s) r_{s,\|} + r_{s,\perp} - \theta_s A_K \hat{M}_s' r_{s,\perp} \\ &= (1-\theta_s) r_{s,\|} + r_{s,\perp} - \theta_s r_{s,\perp} + \theta_s r_{s,\perp} - \theta_s A_K \hat{M}_s' r_{s,\perp}. \end{aligned}$$

Using Corollary 4.2 we obtain

$$\begin{aligned} r_{s+1} &= (1-\theta_s) r_{s,\|} + r_{s,\perp} - \theta_s r_{s,\perp} - \hat{M}_s' \theta_s r_{s,\perp} - \theta_s A_K \hat{M}_s' r_{s,\perp} \\ &= (1-\theta_s) r_{s,\|} + (1-\theta_s) r_{s,\perp} - \theta_s (A_K + I) \hat{M}_s' r_{s,\perp} \\ &= (1-\theta_s) r_{s,\|} + (1-\theta_s) r_{s,\perp} - \theta_s A_H \hat{M}_s' r_{s,\perp} \\ &= \underbrace{(1-\theta_s)\left(r_{s,\|} + r_{s,\perp} + A_H \hat{M}_s' r_{s,\perp}\right)}_{(*)} - \underbrace{A_H \hat{M}_s' r_{s,\perp}}_{(**)}. \end{aligned} \tag{5.2}$$

The new column vector added to update V_s^{KH} to V_{s+1}^{KH} will depend on $\delta r_s = r_{s+1} - r_s$:

$$\delta r_s = -\theta_s \left(r_{s,\|} + r_{s,\perp} + A_H \hat{M}_s' r_{s,\perp}\right).$$

We can thus conclude that the direction of δr_s is independent of θ_s. We see that r_{s+1} has a part that is parallel to δr_s (part $(*)$ in (5.2)) and a remaining part that is independent of θ_s (part $(**)$). $(*)$ will be eliminated completely at the next iteration, according to Theorem 4.1, as it lies in $\mathcal{R}(V_{s+1}^{KH})$. We can thus conclude that θ_s will have no effect at the next iteration. □

Corollary 5.2. *Consider the method described in section* 4.1*. If we use for the first iteration*

$$p_1 = (1-\theta) p_o + \theta H(p_o),$$

then the value of $\theta \in \mathbb{R} \setminus \{0\}$ has no influence on the following iterations.

Proof. The proof is analogous to that of Theorem 5.1. □

Theorem 5.1 and Corollary 5.2 show that line searches, which are often part of a (quasi-) Newton method, do not improve the long-term convergence of our algorithm, but can improve the instantaneous convergence. While classical line searches [1, 15] require supplemental function evaluations and would be very expensive if counted by the number of matrix-vector products, we will discuss a cheap and easy alternative in section 5.2.

5.2. Variants of QN-ILS. In this section we assume that, contrary to the previous sections, we are able to form $A_K x$ for every $x \in \mathbb{R}^{n \times 1}$. This allows us to

[3] The value of p_{s+1} will be affected, however.

rewrite Algorithm 4.1 as in the form of Algorithm 5.1, in which we also include a step-length parameter θ_s. This form is based on ideas formulated in [8].

Algorithm 5.1. *Variant 1 of QN-ILS.*

```
1. Startup.
   Take a starting value $p_o$;
   compute $r_o = A_K p_o - b$;
   set $\hat{M}'_o = -I$.
   Set $s = 0$.
2. Loop until sufficiently converged
   a. $\Delta_s = \hat{M}'_s r_s$
   b. $q_s = A_K \Delta_s$
   c. $p_{s+1} = p_s - \theta_s \Delta_s$
   d. $r_{s+1} = r_s - \theta_s q_s$
   e. If $s = 0$: $\bar{L}_{s+1} = \frac{q_s}{\|q_s\|}$
      else: $\bar{L}_{s+1} = \frac{(I-\mathcal{L}_s^{KH}(\mathcal{L}_s^{KH})^T)q_s}{\|(I-\mathcal{L}_s^{KH}(\mathcal{L}_s^{KH})^T)q_s\|}$
   f. If $s = 0$: $\mathcal{L}_{s+1}^{KH} = \bar{L}_{s+1}$
      else: $\mathcal{L}_{s+1}^{KH} = [\mathcal{L}_s^{KH} | \bar{L}_{s+1}]$
   g. $\hat{M}'_{s+1} = \hat{M}'_s - \dfrac{\hat{M}'_s (q_s - r_s) \bar{L}_{s+1}^T}{\langle q_s, \bar{L}_{s+1} \rangle}$
   h. Set $s = s + 1$.
```

It is easy to show that Algorithm 5.1 is identical to Algorithm 4.1 when using the rank-one update formula of section 4.4, if we set $\theta_s = 1$. We also note that adding the parameter θ_s does not change the search subspace of the original method. From Theorem 5.1 we know that adding this parameter changes only the current iterate.

If we use the formulation of QN-ILS given in Algorithm 5.1, we can find the value of θ_s that minimizes r_s in the Euclidean norm. To do so, we impose

$$r_{s+1} \perp q_s,$$

with $r_{s+1} = r_s - \theta_s q_s$, which leads to

$$\langle r_{s+1}, q_s \rangle = \langle r_s - \theta_s q_s, q_s \rangle = 0$$

$$\theta_s = \frac{\langle r_s, q_s \rangle}{\langle q_s, q_s \rangle}. \tag{5.3}$$

If we compare Algorithm 5.1 with GMRes, we see that r_s lies in the same Krylov subspace $(r_o + A_K \mathcal{K}_s\{A_K; r_o\})$ for both methods. When using the optimal step-length parameter in (5.3) Algorithm 5.1 searches for the smallest residual r_s in that subspace, but only along the direction q_{s-1}. Even though this will improve convergence (in exact arithmetic), it will in general result in a larger residual than for GMRes as the latter searches for the smallest residual $r_s \in r_o + A_K \mathcal{K}_s\{A_K; r_o\}$, but in all directions contained in that subspace.

If we want to modify Algorithm 5.1 to make it analytically identical to GMRes, we should alter it as in Algorithm 5.2.

Algorithm 5.2. *Variant* 2 *of QN-ILS.*

```
As Algorithm 5.1, but with
2. c. $p_{s+1} = p_s - \sum_{i=o}^{s} \theta_{s,i} \Delta_i$
2. d. $r_{s+1} = r_s - \sum_{i=o}^{s} \theta_{s,i} q_i$.
```

To find the optimal parameters we define $\Theta_s = [\theta_{s,o}\ \theta_{s,1}\ \dots\ \theta_{s,s}]^T$ and impose

$$r_{s+1} \perp q_i$$

$(i = 0, 1, \dots s)$; then this leads to, by analogy with (5.3),

$$\Theta_s = (Q^T Q)^{-1} Q^T r_s, \tag{5.4}$$

where $Q = [q_o\,|q_1\,|\dots|q_s]$.

Because of the nonsingularity of $\hat{M}'_s$ (cf. section 4.6) we know that $\{\Delta_i\}_{i=o}^s$ forms a basis for the Krylov subspace $\mathcal{K}_s\{A_K; r_o\}$ and $\{q_i\}_{i=o}^s$ forms a basis for the Krylov subspace $A_K\mathcal{K}_s\{A_K; r_o\}$ to which r_{s+1} is now orthogonal. It follows that Algorithm 5.2 is analytically identical to GMRes.

Note that the modifications to obtain Algorithms 5.1 and 5.2 do not demand extra matrix-vector products and only conceptually differ from the original formulation of Algorithm 4.1 in their ability to form $A_K x$ for all $x \in \mathbb{R}^{n\times 1}$ as opposed to $A_K x - b$.

6. Some illustrative examples.

In this section we compare

- GMRes in the numerically stable form given by Barrett and co-workers [2];
- QN-ILS as given in Algorithm 5.1 using $\theta_s = 1$;
- QN-ILS as given in Algorithm 5.1 using θ_s given by (5.3);
- QN-ILS as given in Algorithm 5.2.

As the QN-ILS method was originally developed for problems in which a matrix-vector product can be considered as extremely expensive,[4] we will use the necessary number of matrix-vector products to reach convergence as the performance measure.

6.1. Test matrix from the Matrix Market Repository.

In this test we take for A_K a square non-semidefinite positive matrix $\in \mathbb{R}^{32\times 32}$ from the Matrix Market Repository [12] called IBM32.

For the vector b we choose $[b]_i = 1$ $(i = 1, \dots, n)$.

All iterations will start from $p_o = [0\ 0\ \dots\ 0]^T$. As a convergence requirement we take a relative reduction of the residual $\frac{r_s}{r_o} \leq 10^{-5}$ and measure the number of matrix-vector products necessary to obtain convergence. This is shown in Figure 6.1.

We see that Algorithm 5.1 with the optimal value of θ_s (see (5.3)) gives an initial improvement over the same algorithm with $\theta_s = 1$ (called *baseline algorithm* hereafter). After about 10 iterations this improvement starts to wane, and after about 15 iterations the residual coincides with that of the baseline algorithm. At the end, the convergence drops slightly behind. The second variant of QN-ILS (Algorithm 5.2) is identical to GMRes apart from a small difference in the last iteration.

6.2. One-dimensional advection-diffusion equation.

In [8] the following ODE boundary value problem was proposedas a test case:

$$-\frac{d^2u}{dx^2} + \beta\frac{du}{dx} = 0 \quad \text{on }]0,1[, \tag{6.1a}$$

$$u(0) = 1, \tag{6.1b}$$

$$u(1) = 0. \tag{6.1c}$$

[4] In the original method the actual matrix-vector product was not computed, but represented an implicit procedure. Alternatively the simple matrix-vector product could be the concatenation of a series of matrix-vector products with intermediate dimensions surpassing the concatenated dimensions by several orders. For more details see [11].

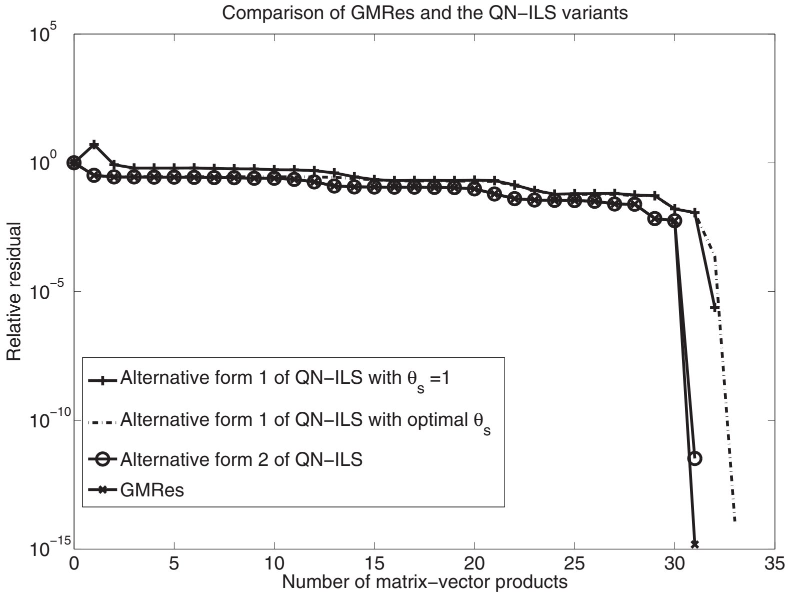

FIG. 6.1. *Convergence history of the different solvers for the IBM32 matrix test case.*

Equation (6.1) describes a one-dimensional advection-diffusion problem and is discretized on a uniform grid with step-size $h = \frac{1}{n+1}$ using first-order finite difference upwind discretization for the advection term and second-order central discretization for the diffusion term. This leads to a linear system that can be written as $K(p) = A_K p - b = 0$.

For β we take the value 10^{-1}; as in [8] we take $n = 50$. p_o is chosen as $p_o = [1\ 1\ \ldots\ 1]^T$.

The convergence criterion is a relative reduction of the residual of 10^{-5}, and as a performance measure the number of matrix-vector products is used. This is shown in Figure 6.2.

We see that Algorithm 5.1 with the optimal value of θ_s (see (5.3)) initially gives a clear improvement over the baseline algorithm but finally stagnates. (It was found that this stagnation was due to limit-cycle behavior.) We also see from the convergence history that Algorithm 5.2 is identical to GMRes apart from a very small difference in the last iteration.

While the exact details of the numerical instabilities for $\theta_s = 1$ are still unclear, the experiments pointed toward numerical instabilities due to bad conditioning of the linear systems that had to be solved in the routine.

6.3. Two-dimensional advection-diffusion equation. We propose the following PDE problem as a test case:

$$\vec{\lambda} \cdot \nabla u(x,y) - \nu \nabla^2 u(x,y) = f(x,y) \quad \text{on } \Omega = \,]-1,1[\times]-1,1[, \tag{6.2a}$$

$$u(x,y) = 0 \quad \text{on } \partial\Omega. \tag{6.2b}$$

Equation (6.2) describes a two-dimensional advection-diffusion problem and is discretized using a residual distribution scheme on an unstructured triangular mesh with 441 nodes (361 interior nodes) [5, 10]. This leads to a linear system that can be written as $K(p) = A_K p - b = 0$.

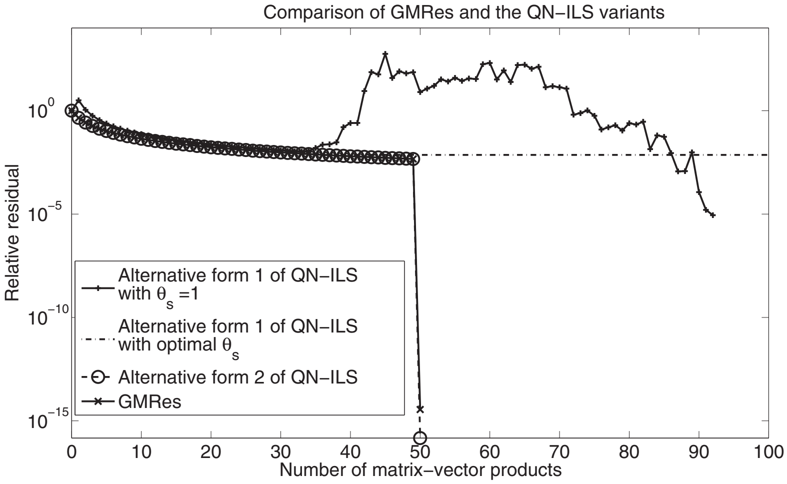

FIG. 6.2. *Convergence history of the different solvers for the one-dimensional advection-diffusion problem.*

We take $\vec{\lambda} = (1, 1)$, $\nu = 0.1$, and $f(x, y) = (x^2 - 1)(y^2 - 1)^2$. We start from $p_o = [0\ 0\ \ldots\ 0]^T$.

The convergence criterion is a relative reduction of the residual of 10^{-5}, and as a performance measure the number of matrix-vector products is used. This is shown in Figure 6.3.

We see that Algorithm 5.1 with the optimal value of θ_s (see (5.3)) largely coincides with the baseline algorithm apart from some values in the middle of the convergence

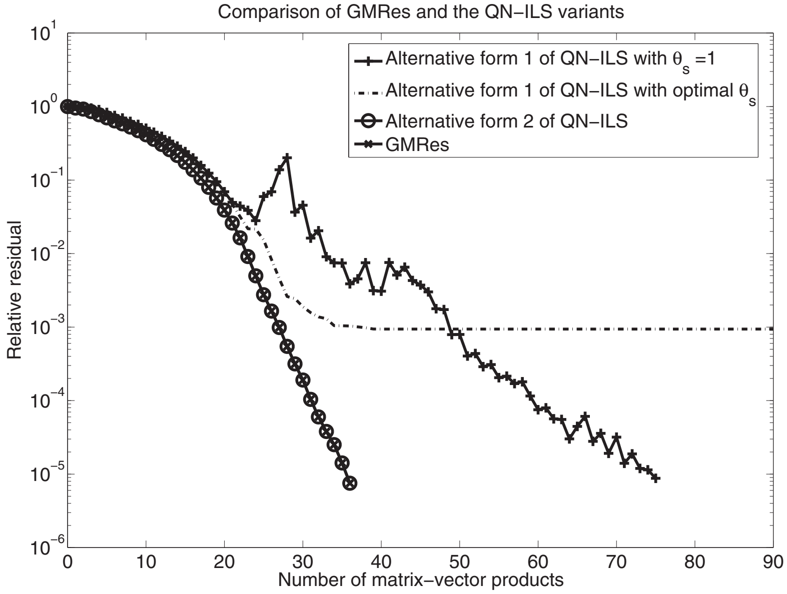

FIG. 6.3. *Convergence history of the different solvers for the two-dimensional advection-diffusion problem.*

history. We also see from the convergence history that Algorithm 5.2 is identical to GMRes for all iterations.

7. Conclusions. We have shown that GMRes can be considered as a quasi-Newton method. We have analyzed a particular quasi-Newton method, QN-ILS, originally developed for nonlinear systems. It was shown that the iterates for this method share the same Krylov search subspace as those of GMRes. It is also shown that adding suitable step-length parameters to a particular implementation of QN-ILS can result in an improved method. Rewriting the algorithm further, using multiple parameters, makes this method analytically equivalent to GMRes for systems of linear equations without the need for extra matrix-vector products.

REFERENCES

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

[2] R. Barrett et al., *Templates for the solution of linear systems: Building blocks for iterative methods,* Miscellaneous Titles in Applied Mathematics Series 43, SIAM, Philadelphia, 1993.

[3] C. G. Broyden, *Quasi-Newton methods and their applications to function minimization*, Math. Comp., 21 (1967), pp. 368–381.

[4] E. Cătinaş, *The inexact, inexact perturbed and quasi-Newton methods are equivalent models*, Math. Comp., 74/249 (2005), pp. 291–301.

[5] H. Deconinck, K. Sermeus, and R. Abgrall, *Status of Multidimensional Upwind Residual Distribution Schemes and Applications in Aeronautics*, AIAA paper 2000-2328, 2000.

[6] J. Degroote, K.-J. Bathe, and J. Vierendeels, *Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction*, Comput. & Structures, 87 (2009), pp. 798–801.

[7] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[8] P. Deuflhard, R. Freund, and A. Walter, *Fast secant methods for the iterative solution of large nonsymmetric linear systems*, IMPACT Comput. Sci. Engrg., 2 (1990), pp. 244–276.

[9] R. Haelterman, J. Degroote, D. Van Heule, and J. Vierendeels, *The quasi-Newton least squares method: A new and fast secant method analyzed for linear systems*, SIAM J. Numer. Anal., 47 (2009), pp. 2347–2368.

[10] R. Haelterman, *Aggregation Multi-Grid for Residual Distribution Schemes*, Project report 2005-14, Von Karman Institute for Fluid Dynamics, Rhode-St.-Gènese, Belgium, 2005.

[11] R. Haelterman, *Analytical Study of the Least Squares Quasi-Newton Method for Interaction Problems*, Ph.D. thesis, Ghent University, Gent, Belgium, 2009.

[12] Matrix Market Repository. http://math.nist.gov/MatrixMarket/.

[13] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.

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

[15] Z.-J. Shi, *Convergence of quasi-Newton method with new inexact line search*, J. Math. Anal. Appl., 315 (2006), pp. 120–131.

[16] H. A. van der Vorst, *Iterative Krylov methods for large linear systems*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, New York, 2003.

[17] J. Vierendeels, L. Lanoye, J. Degroote, and P. Verdonck, *Implicit coupling of partitioned fluid-structure interaction problems with reduced order models*, Comput. & Structures, 85 (2007), pp. 970–976.

[18] J. Vierendeels, *Implicit coupling of partitioned fluid-structure interaction solvers using reduced-order models*, in Fluid-Structure Interaction, Modelling, Simulation, Optimisation, Lecture Notes in Computational Science and Engineering 53, H.-J. Bungartz, and M. Schäfer, eds., Springer, New York, 2006, pp. 1–18.

[19] C. Vuik and H. A. van der Vorst, *A comparison of some GMRes-like methods*, Linear Algebra Appl., 160 (1992), pp. 131–162.