Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Efficient solution of nonlinear elliptic problems using hierarchical matrices with Broyden updates

by

Mario Bebendorf and Ying Chen

Preprint no.: 51 2005

Efficient solution of nonlinear elliptic problems using hierarchical matrices with Broyden updates[*]

M. Bebendorf and Y. Chen

Fakultät für Mathematik und Informatik, University of Leipzig,
Augustusplatz 10/11, 04109 Leipzig, Germany

Abstract

The numerical solution of nonlinear problems is usually connected with Newton's method. Due to its computational cost, variants (so-called quasi-Newton methods) have been developed in which the arising inverse of the Jacobian is replaced by an approximation. In this article we present a new approach which is based on Broyden updates. Instead of updating the inverse we introduce a method which constructs updates of the *LU* decomposition. Since an approximate *LU* decomposition of finite element stiffness matrices can be efficiently computed in the set of hierarchical matrices, the complexity of the proposed method scales almost linearly. Numerical examples demonstrate the effectiveness of this new approach.

Keywords. Quasi-Newton methods, Broyden updates, nonlinear elliptic equations, hierarchical matrices.
Mathematics Subject Classification (2000). 34B15, 90C53, 65N30, 65F50, 65F05

1 Introduction

The main purpose of this paper is the efficient numerical solution of second-order nonlinear elliptic Dirichlet problems

$$-\nabla\cdot[\alpha(u)\nabla u-\beta(u)u]+\gamma(u)u=f \quad \text{in } \Omega, \tag{1a}$$
$$u=g \quad \text{on } \partial\Omega, \tag{1b}$$

where Ω is a bounded Lipschitz domain in $\mathbb{R}^n$, $n=2$ or 3, the coefficient functions α,β,γ are sufficiently smooth and f and g are given. For the coefficient α appearing in the principal part we assume that there is a real number $\alpha_0>0$ such that $\alpha(u)\geq\alpha_0$.

[*]This work was supported by the DFG priority program SPP 1146 "Modellierung inkrementeller Umformverfahren"

Nonlinear elliptic problems arise from many applications in physics and other fields, for instance in elastoplasticity, magnetic potential problems, viscous fluid flow problems, chemical reactions and pattern formation in biology. During the past three decades, many numerical approaches have been developed for solving nonlinear problems (see, e.g., [1, 9, 20]). Undoubtedly, Newton's method

$$u_{k+1} = u_k - (F'(u_k))^{-1} F(u_k), \quad k = 0, 1, 2, \ldots, \tag{2}$$

for the solution of the nonlinear problem $F(u) = 0$ is one of the most popular methods, which receives attention from applications due to its quadratic convergence. Newton's method can be used to treat nonlinear elliptic equations discretized not only by the finite element method, see [14], but also by the finite volume method, cf. [9]. Avoiding the computation of the Jacobian and its inverse in each Newton step, most of the attention in tackling the nonlinearity has been paid to quasi-Newton methods

$$u_{k+1} = u_k - B_k F(u_k), \quad k = 0, 1, 2, \ldots,$$

which are based on Newton's method while approximating the inverse $(F'(u_k))^{-1}$ of the Jacobian by a matrix B_k. A general description of quasi-Newton methods can be found in [2, 10, 13] and its applications to nonlinear elliptic problems in [1, 19].

In this article, we are concerned with a special quasi-Newton method. Broyden's method, cf. [8], is based on Broyden's update formula

$$B_{k+1} = B_k + \frac{(\delta_k - B_k y_k)\delta_k^T B_k}{\delta_k^T B_k y_k},$$

where $y_k = F(u_{k+1}) - F(u_k)$, $\delta_k = u_{k+1} - u_k$ and $B_0 = (F(u_0))^{-1}$. Hence, if B_k is known, then B_{k+1} can be computed by a rank-1 update. It is known, cf. [13], that in Broyden's method B_k is an approximation of the inverse $(F'(u_k))^{-1}$ of the Jacobian. The disadvantage of quasi-Newton methods is that the quadratic convergence is lost. Broyden's method can be shown to converge superlinearly, cf. [13]. Hence, usually more steps will be required to obtain the same accuracy as Newton's method. Despite this fact, Broyden's method is significantly cheaper than Newton's method, since the inverse of the Jacobian has to be computed only in the initial step of the iteration. Up to now, this interesting method could not be exploited for large-scale problems since the inverse in the initial step is required explicitly.

In this article we present a variant of Broyden's method for the solution of nonlinear elliptic problems (1). Instead of updating the inverse B_0 of the initial Jacobian $F'(u_0)$, we will propose a method which is based on the LU decomposition of $F'(u_0)$. Since in contrast to the inverse the LU decomposition of sparse matrices can be computed efficiently at least for moderate problem sizes, cf. [11], this new approach revitalizes Broyden's idea. Instead of keeping the triangular structure when updating the factors L and U of the LU decomposition, we generate and store appropriately transformed rank-1 updates of L and U from each rank-1 update applied to the coefficient matrix.

If large-scale problems are to be treated, the bandwidth of the factors L and U of the classical LU decomposition will lead to an uncompetitive number of operations. In this case hierarchical matrices ($\mathcal{H}$-matrices) originally introduced by Hackbusch et al., see [16, 17], can be used to accelerate the proposed methods. Hierarchical matrices can efficiently treat fully populated matrices arising from integral operators with asymptotically smooth kernels [3, 4]. In [5, 6] it has been proved that an $\mathcal{H}$-matrix approximant of the inverse FE matrix of general uniformly elliptic operators with L^∞-coefficients exists because the corresponding Green function can be approximated by a degenerate function. In addition to the approximate inversion of large sparse matrices also low-rank matrices can be added to $\mathcal{H}$-matrices with almost [1] linear complexity. Since the updated matrix B_{k+1} can be looked at as an approximation of $(F'(u_{k+1}))^{-1}$, the complexity of B_{k+1} can be expected to be of the same order as B_k. Hence, $\mathcal{H}$-matrices provide a setting in which Broyden's method even in its original form can be used to solve nonlinear problems with almost linear complexity.

Recently [7], it was proved that also the factors of the LU decomposition of FE stiffness matrices can be approximated by the $\mathcal{H}$-matrix structure with almost linear complexity. As for classical matrices, the variant of Broyden's method which will be based on the hierarchical LU decomposition instead of the hierarchical inverse will lead to a more efficient method.

This paper is organized as follows. In the next section, we review concepts and properties of $\mathcal{H}$-matrices. We quote the existence results for the $\mathcal{H}$-matrix approximation of both discrete inverses and LU decompositions of general elliptic linear second-order operators. These results lay ground to the efficiency of the proposed methods, since each Jacobian $F'(u_k)$ is an elliptic linear second-order operator. In Section 3 $\mathcal{H}$-matrix variants of Newton's and Broyden's method for solving nonlinear elliptic problems are proposed. We introduce a method which updates the factors of an initially computed LU decomposition of a discrete Jacobian each time a Broyden update is applied to it. Finally in Section 5, we provide numerical examples to demonstrate the effectiveness of the presented methods. All methods from this article will be seen to have almost linear complexity while the method which updates the factors of the LU decomposition is significantly faster than the others.

2 Hierarchical matrices

In this section we briefly describe the structure of $\mathcal{H}$-matrices which was introduced by Hackbusch et al. [16, 17]. We will present the main principles such as the admissibility condition, the hierarchical partitioning of the matrix into blocks and the blockwise restriction to low-rank matrices on which the efficient approximation of matrices is constructed.

Let $A \in \mathbb{R}^{N\times N}$ be a finite element stiffness matrix with entries

$$a_{ij} = a(\varphi_j, \varphi_i), \quad i, j = 1, \ldots, N,$$

[1] linear up to logarithmic factors

where $\varphi_i \in V$, $i \in I := \{1, \ldots, N\}$, are finite element basis functions with supports $X_i := \operatorname{supp} \varphi_i$, $i \in I$, and $a : V \times V \to \mathbb{R}$ is a bilinear form.

Assume that there is a partition P of the indices $I \times I$ of A such that each submatrix $A_{t \times s}$, $t, s \subset I$, can be approximated by a low-rank matrix, i.e.,

$$A_{t \times s} \approx UV^T, \quad U \in \mathbb{R}^{t \times k}, V \in \mathbb{R}^{s \times k}, \tag{3}$$

where k is a positive integer which is small compared with $|t|$ and $|s|$.

2.1 Cluster tree

Searching the set of possible partitions of $I \times I$ for a partition P which guarantees (3) seems practically impossible since this set is considerably large. By restricting ourselves to blocks $t \times s$ made up from rows t and columns s which are generated by recursive subdivision, P can be found with almost linear complexity. The structure which describes the way I is subdivided into smaller parts is the cluster tree. A tree T_I is called a *cluster tree* for an index set I if it satisfies the following conditions:

- (i) I is the root of T_I;
- (ii) If $t \in T_I$ is not a leaf, then it has two sons $t_1, t_2 \in T_I$ such that $t = t_1 \cup t_2$ and $t_1 \cap t_2 = \emptyset$.

For each $t \in T_I$, by $S(t) \subset T_I$ we denote the set of its sons and $\mathcal{L}(T_I)$ means the set of leaves of the tree T_I. Each level of T_I contains a partition of the index set I. We define the support of a cluster $t \in T_I$ as the union of the supports of the basis functions corresponding to its indices in t, i.e.,

$$X_t := \bigcup_{i \in t} X_i.$$

When generating a cluster tree by recursive subdivision of I, it is not advisable to subdivide the clusters until only one index is left. In practice, the recursion should be stopped if a certain cardinality $n_{\min}$ of the clusters is reached. In the case of quasi-uniform grids, the cost of building a cluster tree is $\mathcal{O}(N \log N)$, see [17].

2.2 Block-cluster tree

In order to be able to guarantee a sufficient approximation of each submatrix $A_{t \times s}$, $t \times s \in P$, of A by a matrix of low rank, the subblock $t \times s$ has to satisfy the so-called *admissibility condition*

$$\min\{\operatorname{diam} X_s, \operatorname{diam} X_t\} \leq \eta \operatorname{dist}(X_s, X_t), \tag{4}$$

where $\eta > 0$ is a given real number. Notice that in order to satisfy (4) the supports of t and s have to be far enough away from each other. This condition is caused by the fact that the Schwartz kernel $S(x,y)$ of elliptic operators possesses a singularity for $x=y$ only.

Based on a cluster tree T_I and the admissibility condition (4), we are ready to construct a block-cluster tree $T_{I\times I}$ by the following procedure.

```
procedure Build_BlockClusterTree($t \times s$)
    if $t \times s$ does not satisfy (4) and $|t| > n_{\min}$ and $|s| > n_{\min}$ then
        $S(t \times s) := \{t' \times s' : t' \in S(t),\ s' \in S(s)\}$;
        for $t' \times s' \in S(t \times s)$ do
            Build_BlockClusterTree($t' \times s'$);
    else $S(t \times s) := \emptyset$;
end.
```

Applying Build_BlockClusterTree to $I \times I$, we create a block-cluster tree with root $I \times I$. The set of leaves $\mathcal{L}(T_{I\times I})$ of the block-cluster tree $T_{I\times I}$ forms a partition P of $I \times I$ such that (4) is satisfied for large enough blocks. In the case of quasi-uniform grids, the cost of building a block-cluster tree is $\mathcal{O}(N)$, see [17].

For a given partition P of $I \times I$ and given rank r, we are now able to define the set of $\mathcal{H}$-matrices. A matrix A is called an $\mathcal{H}$-matrix of blockwise rank r if each admissible blocks $A_{t\times s}$, $t \times s \in P$ has at most rank r, i.e.,

$$\mathcal{H}(P,k) = \{A \in \mathbb{R}^{I\times I} : \text{rank } A_{t\times s} \leq r \text{ for each } t \times s \in P\}. \tag{5}$$

The storage requirement for $A \in \mathcal{H}(P,r)$ is of the order $rN\log N$. Multiplying A by a vector can be done with $\mathcal{O}(rN\log N)$ arithmetical operations. Two $\mathcal{H}$-matrices $A, B \in \mathcal{H}(P,r)$ can be added with complexity $\mathcal{O}(r^2 N \log N)$ if an arbitrarily small approximation error can be tolerated. Since low-rank matrices are a subset of $\mathcal{H}$-matrices, in particular rank-1 updates can be efficiently added to $\mathcal{H}$-matrices. The complexity of computing a rounded product of two $\mathcal{H}$-matrices is $\mathcal{O}(r^2 N \log^2 N)$, see [16, 17, 15].

2.3 Approximation of Galerkin FE inverses and of the factors of LU decompositions

Since by Newton's method (2) the nonlinear problem (1) or equivalently

$$F(u) := -\nabla \cdot [\alpha(u)\nabla u - \beta(u)u] + \gamma(u)u - f = 0 \tag{6}$$

is approximated by a sequence of linear problems

$$F'(u_k)(u_{k+1} - u_k) = -F(u_k), \quad k = 0, 1, 2, \ldots,$$

we first consider the linear case. Note that since

$$F'(u)v = -\nabla \cdot [\alpha(u)\nabla v - \{\beta(u) + \beta'(u)u - \alpha'(u)\nabla u\}v] + \{\gamma(u) + \gamma'(u)u\}v$$

the assumption $\alpha(u) \geq \alpha_0 > 0$ guarantees that each linear problem is in the class of second-order elliptic Dirichlet boundary value problems

$$Lu = f \quad \text{in } \Omega, \tag{7a}$$
$$u = g \quad \text{on } \partial\Omega \tag{7b}$$

with

$$Lu = -\nabla \cdot [A\nabla u + \boldsymbol{b}u] + \boldsymbol{c} \cdot \nabla u + du,$$

where $A(x) \in \mathbb{R}^{n\times n}$ is symmetric with entries $a_{ij} \in L^\infty(\Omega)$ and satisfies

$$0 < \lambda \leq \lambda(x) \leq \Lambda$$

for all eigenvalues $\lambda(x)$ of $A(x)$ and almost all $x \in \Omega$, $\boldsymbol{b}, \boldsymbol{c} \in (L^\infty(\Omega))^n$ and $d \in L^\infty(\Omega)$. For simplicity we assume that $g = 0$. Then the variational formulation of the linear boundary value problem (7) reads

$$\text{find } u \in V \text{ such that } a(u, v) = l(v) \quad \text{for all } v \in V,$$

where

$$a(u, v) = \int_\Omega \nabla v \cdot A\nabla u \,\mathrm{d}x + \int_\Omega \nabla v \cdot \boldsymbol{b}\, u \,\mathrm{d}x + \int_\Omega \boldsymbol{c} \cdot \nabla u\, v \,\mathrm{d}x + \int_\Omega duv \,\mathrm{d}x$$

and

$$l(v) = \int_{\partial\Omega} fv \,\mathrm{d}x.$$

Let ε_h denote the Galerkin finite element error. The following theorem, cf. [6], states that the inverse of the arising finite element stiffness matrix $A \in \mathbb{R}^{N\times N}$ can be approximated by an $\mathcal{H}$-matrix with almost linear complexity.

Theorem 2.1 *Let p be the depth of the cluster tree T_I defined in Section 2.1. Then there is a constant $c > 0$ defining $r := cp^2 \log^{n+1}(p/\varepsilon_h)$ and there is $C_\mathcal{H} \in \mathcal{H}(P, r)$ such that*

$$\|A^{-1} - C_\mathcal{H}\|_2 \leq c\,\varepsilon_h,$$

where $c = c(L, \Omega, \eta) > 0$ depends on the size of coefficients of L, the diameter of Ω and η. If $\varepsilon_h = \mathcal{O}(h^\beta)$ for some $\beta > 0$, then $r = \mathcal{O}(\log^{n+3} N)$ holds.

Although the constant c depends on the size of the coefficients of L, numerical experiments have shown that this dependence can hardly be observed.

Based on the previous result, in [7] a similar result for the approximation of the factors L and U of an LU decomposition of A has been proved. In the following theorem ρ_N denotes the *growth factor*, cf. [18], which plays a central role for the stability of the LU decomposition.

Theorem 2.2 *There are lower and upper triangular matrices* $L_{\mathcal{H}}, U_{\mathcal{H}} \in \mathcal{H}(P,r)$ *with*

$$r \sim (\log N)^2 \left[|\log \varepsilon| + (\log N)^2 + (\log N)(\log \rho_N \mathrm{cond}_2 A) \right]^{n+1}$$

such that

$$\|A - L_{\mathcal{H}} U_{\mathcal{H}}\|_2 \leq \varepsilon.$$

The asymptotic complexity of computing approximations of the inverse and of the factors of an LU decomposition is inherited from the approximate matrix multiplication. Hence, computing these approximations in $\mathcal{H}(P,r)$ can be done with $\mathcal{O}(r^2 N \log^2 N)$ arithmetical operations.

3 Quasi-Newton methods with Broyden updates

In each Newton step

$$F'(u_k)(u_{k+1} - u_k) = -F(u_k), \quad k = 0, 1, ... \tag{8}$$

we obtain a linear system

$$A(\underline{u}_k)\underline{\delta}_k = -F(\underline{u}_k) \tag{9}$$

arising from the Galerkin method applied to (8) for the unknown vector $\underline{\delta}_k := \underline{u}_{k+1} - \underline{u}_k$. Here, $\underline{u}_k \in \mathbb{R}^N$ denotes the Galerkin approximation of u_k. If (8) has been solved, then starting from a given vector $\underline{u}_0 \in \mathbb{R}^N$ update $\underline{u}_{k+1} = \underline{u}_k + \underline{\delta}_k$ until some tolerance is reached. For the sake of readability in the following we will not distinct between the vector $\underline{u}_k \in \mathbb{R}^N$ and u_k from (8).

Since the exact solution of (9) is expensive for large N, it is advisable to replace the coefficient matrix $A(u_k)$ by an approximation which allows a more efficient solution of (9). This class of variants of Newton's method is usually called quasi-Newton methods. Due to Theorem 2.1 it is possible to compute an approximate inverse of $A(u_k)$ with almost linear complexity. Hence, we can easily accelerate Newton's method by employing $\mathcal{H}$-matrices. In the following this obvious variant will be referred to as the $\mathcal{H}$-Newton method.

A special quasi-Newton method is Broyden's method, cf. [8], which starts from a given approximation $A_0 \in \mathbb{R}^{N \times N}$ of $A(u_0)$ and computes approximations A_k of $A(u_k)$ using rank-1 updates

$$A_{k+1} = A_k + \frac{(y_k - A_k \delta_k)\delta_k^T}{\delta_k^T \delta_k}, \tag{10}$$

where $y_k := F(u_{k+1}) - F(u_k)$. As for other quasi-Newton methods, the quadratic convergence of Newton's method is reduced to a superlinear one, cf. [13]. The matrix A_{k+1} can be seen to be the closest matrix B to A_k satisfying the secant equation $B\delta_k = y_k$, i.e., with $Q := \{B \in \mathbb{R}^{N \times N} : B\delta_k = y_k\}$ it holds

$$\|A_{k+1} - A_k\|_2 = \min_{B \in Q} \|A - B\|_2.$$

Since one is interested in the solution of (9), it is crucial that the inverse A_k^{-1} can also be updated in a similar way as A_k is updated in (10). With the Sherman-Morrison-Woodbury formula we have that

$$A_{k+1}^{-1} = A_k^{-1} + \frac{(\delta_k - A_k^{-1} y_k)\delta_k^T A_k^{-1}}{\delta_k^T A_k^{-1} \delta_k}. \tag{11}$$

Hence, if $B_0 = A_0^{-1}$ has been computed, B_{k+1} can be computed from B_k by

$$B_{k+1} = B_k + \frac{(\delta_k - B_k y_k)\delta_k^T B_k}{\delta_k^T B_k y_k},$$

which involves matrix-vector multiplications and a rank-1 update only. Since the last both operations can be performed in the set of $\mathcal{H}$-matrices with almost linear complexity, we obtain another obvious but efficient variant of Newton's method which will be referred to as the $\mathcal{H}$-Broyden method. Compared with the $\mathcal{H}$-Newton method only one $\mathcal{H}$-matrix inversion has to be computed when the initial matrix B_0 is generated. Matrix-vector multiplies and rank-1 updates can be expected to be done with much less time consumption than computing an approximate inverse in each step.

Recently [7], the existence of $\mathcal{H}$-matrix approximations of the factors L and U of the LU decomposition of finite element stiffness matrices was proved. The computation of the hierarchical LU decomposition was found to require much less time than the computation of the hierarchical inverse while the same robustness with respect to varying coefficients of the operator was observed. Hence, it seems desirable to replace the role of the inverse in (11) by the LU decomposition. However, no update formula like the Sherman-Morrison-Woodbury formula is known for the factors of the LU decomposition. In the following section we present a method which can be used for updating the factors L and U of an LU decomposition of A whenever A is updated by a rank-1 matrix.

4 An update method for the *LU* decomposition

Assume we want to update the matrix A with a given rank-1 matrix uv^T such that $A+uv^T$ is still non-singular. Let A be decomposed using the LU decomposition $A = LU$. Then the factors

$$\begin{aligned} \hat{L} &:= L + \alpha u w^T, \quad w := U^{-T} v, \quad \text{and} \\ \hat{U} &:= U + \beta z v^T, \quad z := L^{-1} u, \end{aligned}$$

with $\xi := w^T z$,

$$\alpha := \begin{cases} -\frac{1}{2}, & \xi < 0 \\ \frac{1}{2}, & \xi \geq 0 \end{cases} \quad \text{and} \quad \beta := \frac{1-\alpha}{1+\alpha\xi}$$

satisfy

$$\begin{aligned} \hat{L}\hat{U} &= LU + \alpha u w^T U + \beta L z v^T + \alpha\beta\xi u v^T \\ &= LU + (\alpha + \beta + \alpha\beta\xi) u v^T = A + u v^T \end{aligned}$$

since $\alpha + \beta + \alpha\beta\xi = 1$. Note that the matrices $\hat{L}$ and $\hat{U}$ are rank-1 updates of L and U, respectively, and they are not triangular in general. But they still can be used for an efficient computation. From $\hat{L} = L(I + \alpha z w^T)$ and $\hat{U} = (I + \beta z w^T)U$ it follows that $\hat{L}$ and $\hat{U}$ are non-singular since

$$\det \hat{L} = (1 + \alpha\xi) \det L \geq \det L \quad \text{and} \quad \det \hat{U} = (1 + \beta\xi) \det U = \frac{1 + \xi}{1 + \alpha\xi} \det U$$

and $1 + \xi \neq 0$ due to the assumption that $A + uv^T$ is non-singular.

Assume that A has been updated k times and let $w_{k+1} = U_k^{-T} v_{k+1}$ and $z_{k+1} = L_k^{-1} u_{k+1}$, $k = 0, 1, 2, \ldots$, where L_k and U_k denote the matrices that have been constructed from L and U as above, i.e.,

$$L_k U_k = A_k := A + \sum_{\ell=1}^{k} u_\ell v_\ell^T.$$

When computing the inverses of L_k and U_k, we make the following observation:

$$L_{k+1}^{-1} = (L_k(I + \alpha_k z_k w_k^T))^{-1} = (I + \alpha_k z_k w_k^T)^{-1} L_k^{-1} = \left(I + \alpha_k' z_k w_k^T\right) L_k^{-1},$$

where $\alpha_k' := -\alpha_k/(1 + \alpha_k \xi_k)$. Analogously, we obtain

$$U_{k+1}^{-1} = U_k^{-1} \left(I - \beta_k' z_k w_k^T\right),$$

where $\beta_k' := -\beta_k/(1 + \beta_k \xi_k) = (\alpha_k - 1)/(1 + \xi_k)$ and $\xi_k := w_k^T z_k$. As a consequence we have

$$L_k^{-1} = \prod_{\ell=k}^{1} \left(I + \alpha_\ell' z_\ell w_\ell^T\right) L^{-1} \quad \text{and} \quad U_k^{-1} = U^{-1} \prod_{\ell=1}^{k} \left(I + \beta_\ell' z_\ell w_\ell^T\right).$$

Hence the solution x of $L_k x = b$ can be computed by the following algorithm:

(1a) compute x from $Lx = b$,

(1b) for $\ell = 1, \ldots, k$ set $x := x + \alpha_\ell'(w_\ell^T x) z_\ell$.

Analogously, one can solve $U_k x = b$ for x by the following procedure

(2a) set $y := b$,

(2b) for $\ell = k, \ldots, 1$ set $y := y + \beta_\ell'(w_\ell^T y) z_\ell$,

(2c) compute x from $Ux = y$.

If $A_k x = b$ with $A_k = L_k U_k$ is to be solved for x, this can be done as usual by the two-step procedure

1. solve $L_k y = b$ for y,
2. solve $U_k x = y$ for x.

With these ideas, a variant of Broyden's method which is based on the LU decomposition can be constructed. In the following this method will be referred to as the $\mathcal{H}$-Broyden-LU method. In the kth step of the resulting quasi-Newton method $A_k\delta_k = -F(u_k)$ has to be solved and the vectors w_{k+1} and z_{k+1} have to be computed and stored. Both can be done by applying the modified forward/backward substitutions from above. Hence, the number of operations in the kth step is determined by the complexity of the forward/backward substitutions (1a) and (2c) in $\mathcal{H}(P,r)$ and the updates (1b) and (2b). The first requires $\mathcal{O}(rN\log N)$ operations while the latter can be done with $\mathcal{O}(kN)$ operations. Hence, the number of operations for k steps of this variant of Newton's method is of order $krN\log N + k^2N$. The initial hierarchical LU decomposition of A_0 requires $r^2N\log^2 N$ operations.

If a symmetric positive definite matrix A is to be updated with a rank-1 matrix uu^T, then the factor

$$\hat{L} := L + \alpha u w^T \quad \text{with} \quad w := L^{-1}u \quad \text{and} \quad \alpha := \frac{1}{1+\sqrt{1+\|w\|_2^2}}$$

satisfies

$$\begin{aligned}\hat{L}\hat{L}^T &= LL^T + \alpha u(Lw)^T + \alpha Lwu^T + \alpha^2\|w\|_2^2 uu^T\\ &= LL^T + \alpha(2+\alpha\|w\|_2^2)uu^T = A + uu^T\end{aligned}$$

since $\alpha(2+\alpha\|w\|_2^2) = 1$. Note that the ideas of this section are not restricted to the hierarchical LU decomposition. For small N recent exact decomposition algorithms such as SuperLU, cf. [11], can be expected to improve the efficiency of the presented method.

5 Numerical experiments

In this section, we report results of numerical experiments which confirm the efficiency of $\mathcal{H}$-Newton and $\mathcal{H}$-Broyden method when solving nonlinear elliptic problems. It will be seen that the complexity of both methods scales almost linearly with respect to the number of degrees of freedom N. Furthermore, the dominance of a variant of Broyden's method which is based on the LU decomposition will be seen. In the following we list four examples of the model problem (6).

Example 5.1 *Let $a(u) = 1$, $b(u) = 0$, $c(u) = \lambda e^u$ and*

$$f = \Big([9\pi^2 + \lambda e^{(x^2-x^3)}\sin(3\pi y)](x^2-x^3) + 6x - 2\Big)\sin(3\pi y)$$

such that (1) *has the exact solution*

$$u = (x^2-x^3)\sin(3\pi y).$$

Example 5.2 *Let $a(u) = 1$, $b(u) = \frac{1}{2}u$, $c(u) = 0$ and*

$$f = 200\cos(2\pi x + \frac{\pi}{2}) \Big((y^2 - y^3)[2\pi^2 + 25\cos(2\pi x + \frac{\pi}{2})(2y - 3y^2) \\ -50\pi\sin(2\pi x + \frac{\pi}{2})(y^2 - y^3)] + 3y - 1\Big)$$

such that the exact solution of (1) *is*

$$u = 50\cos(2\pi x + \frac{\pi}{2})(y^2 - y^3).$$

Example 5.3 *Let $a(u) = u + 1$, $b(u) = 0$, $c(u) = 1$ and*

$$f = [(x - x^2)\sin(3\pi y) + 1][2 + 9\pi^2(x - x^2)]\sin(3\pi y) \\ - (1 - 2x)^2\sin^2(3\pi y) - 9\pi^2(x - x^2)^2\cos^2(3\pi y) + (x - x^2)\sin(3\pi y)$$

such that (1) *has the solution*

$$u = (x - x^2)\sin(3\pi y).$$

Example 5.4 *Let $a(u) = u + 1$, $b(u) = 0$, $c(u) = u$ and*

$$f = 2[(x - x^2)(y - y^2) + 1](x - x^2 + y - y^2) - (1 - 2x)^2(y - y^2)^2 \\ - (x - x^2)^2(1 - 2y)^2 + (x - x^2)^2(y - y^2)^2$$

with the exact solution

$$u = (x - x^2)(y - y^2).$$

The purpose of these numerical experiments is to compare the performance of $\mathcal{H}$-Broyden and $\mathcal{H}$-Broyden-*LU* with Newton's method, the $\mathcal{H}$-Newton method and Broyden's method. For all of these methods we use piecewise linear basis functions $(\varphi_i)_{i=1}^N$ on the unit square $\Omega = (0, 1) \times (0, 1)$.

Numerical results are listed in the following tables. "♯it" denotes the number of iterations. "time(s.)" consists of two parts. For $\mathcal{H}$-Broyden-*LU*, $\mathcal{H}$-Broyden and Broyden's method the first value is the time required for computing the *LU* decomposition and the inverse, respectively, and the second one is for Broyden's iteration. The iteration stops if the stopping criterion $\|u_{k+1} - u_k\|_2 < 10^{-6}$ is satisfied.

In the case of Example 5.1, the numerical results are listed in Table 1–3 for $\lambda = 10, 100$, respectively. The number of iterations is slightly higher when λ becomes larger. The results show that the $\mathcal{H}$-Newton method converges faster than the $\mathcal{H}$-Broyden method but requires much more computational work. $\mathcal{H}$-Broyden-*LU* requires significantly less time than $\mathcal{H}$-Broyden since the computation of the hierarchical *LU* decomposition is approximately 10 times faster than the computation of the hierarchical inverse. Additionally, in each step of the iteration of $\mathcal{H}$-Broyden-*LU* the update is not added to the $\mathcal{H}$-matrices like it is done in the $\mathcal{H}$-Broyden method. From these results we also confirm that the complexity of the $\mathcal{H}$-matrix accelerated versions of Newton's and Broyden's method are almost linear. Moreover, the number of iterations in all these methods does only slightly depend on the number of degrees of freedom N.

Similar trends can be observed for Example 5.2, 5.3 and 5.4 in Table 4–9.

	$\lambda = 10$					$\lambda = 100$				
	Newton		Broyden			Newton		Broyden		
N	$\sharp$ it	time(s.)	$\sharp$ it	time(s.)		$\sharp$ it	time(s.)	$\sharp$ it	time(s.)	
3969	6	41.0	8	6.6	2.3	10	66.9	19	6.7	7.1
9025	6	385.7	8	69.9	14.8	11	721.8	19	65.4	35.6
16129	6	1828.3	8	292.6	39.7	11	3294.1	19	283.0	102.8
65025	-	-	-	-	-	-	-	-	-	-

Table 1: iterations and time for Example 5.1 with $\lambda = 10$ and $\lambda = 100$.

	$\mathcal{H}$-Newton		$\mathcal{H}$-Broyden			$\mathcal{H}$-Broyden-LU		
N	$\sharp$ it	time(s.)	$\sharp$ it	time(s.)		$\sharp$ it	time(s.)	
3969	6	8.5	9	1.3	1.4	8	0.2	0.4
9025	7	36.9	9	4.7	3.6	8	0.5	1.0
16129	7	82.8	9	10.7	7.1	9	1.0	2.0
65025	8	667.1	10	77.9	42.2	9	6.3	8.4
130321	8	1984.2	10	226.8	96.1	10	15.8	21.0
261121	9	5001.5	10	523.1	214.5	11	37.1	42.0
522729	-	-	-	-	-	14	96.9	113.2

Table 2: iterations and time for Example 5.1 with $\lambda = 10$.

References

[1] O. Axelsson, *On global convergence of iterative methods*, in Iterative Solution of Nonlinear Systems of Equations, Lecture Notes in Math. 953, Springer, Berlin, 1982, pp. 1–19.

[2] R. E. Bank and D. J. Rose, *Global approximate Newton methods*, Numer. Math., 37 (1981), pp. 279–295.

[3] M. Bebendorf, *Approximation of Boundary Element Matrices*, Numer. Math. 86 (2000), pp. 565–589.

[4] M. Bebendorf and S. Rjasanow, *Adaptive Low-Rank Approximation of Collocation Matrices*, Computing 70 (2003), pp. 1–24.

[5] M. Bebendorf and W. Hackbusch, *Existence of $\mathcal{H}$-Matrix approximants to the inverse FE-matrix of elliptic operators with L^∞-Coefficients*, Numer. Math., 95 (2003), pp. 1–28.

[6] M. Bebendorf, *Efficient inversion of the Galerkin matrix of general second order elliptic operators with non-smooth coefficients*, Math. Comp., 74 (2005), pp. 1179–1199.

	$\mathcal{H}$-Newton		$\mathcal{H}$-Broyden			$\mathcal{H}$-Broyden-LU		
N	♯ it	time(s.)	♯ it	time(s.)		♯ it	time(s.)	
3969	10	14.6	18	1.2	3.0	17	0.2	0.8
9025	11	60.6	19	4.8	9.1	18	0.5	2.0
16129	11	136.0	19	10.8	17.3	19	1.0	3.9
65025	11	952.3	20	77.9	97.5	22	4.7	18.4
130321	12	3123.3	22	227.3	243.2	23	15.7	40.1
261121	12	6921.5	25	525.6	615.0	24	38.4	94.2
522729	-	-	-	-	-	21	97.8	169.0

Table 3: iterations and time for Example 5.1 with $\lambda = 100$.

	Newton		Broyden		
N	♯ it	time(s.)	♯ it	time(s.)	
3969	6	40.8	22	6.7	6.1
9025	6	395.4	22	68.8	33.7
16129	6	1820.1	23	313.9	124.8
65025	-	-	-	-	-

Table 4: iterations and time for Example 5.2.

[7] M. Bebendorf, *Why approximate LU decompositions of finite element discretizations of elliptic operators can be computed with almost linear complexity*, Preprint 8/2005, Max-Planck-Institute MiS, Leipzig.

[8] C. G. Broyden, *A class of methods for solving nonlinear simultaneous equations*, Math. Comp., 19 (1965), pp. 577–593.

[9] P. Chatzipantelidis, V. Ginting and R. D. Lazarov, *A finite volume element method for a nonlinear elliptic problem*, to appear in Numer. Linear Algebra Appl., 2005.

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

[11] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, J. W. Liu, *A supernodal approach to sparse partial pivoting.* SIAM J. Matrix Anal. Appl. 20 (1999), pp. 720–755.

[12] J. E. Dennis and J. J. Moré, *Quasi-Newton methods, motivation and theory*, SIAM Rev., 19 (1977), pp. 46–89.

[13] J. E. Dennis and R. B. Schnabel, Jr., *Numerical methods for unconstrained optimization and nonlinear equations*, SIAM society for industrial and applied mathematics, Philadelphia, PA, 1996.

[14] J. Douglas, Jr. and T. Dupont, *A Galerkin method for a nonlinear Dirichlet problem*, Math. Comp., 29 (1975), pp. 689–696.

	$\mathcal{H}$-Newton		$\mathcal{H}$-Broyden			$\mathcal{H}$-Broyden-LU		
N	$\sharp$ it	time(s.)	$\sharp$ it	time(s.)		$\sharp$ it	time(s.)	
3969	6	8.3	21	1.3	3.4	20	0.2	0.8
9025	6	31.1	22	5.0	9.5	21	0.7	1.9
16129	6	70.3	23	11.5	19.5	22	1.4	3.5
65025	7	576.8	23	82.7	104.6	22	8.3	17.2
130321	7	1726.4	23	240.6	254.4	22	19.9	35.6
261121	7	3845.8	23	558.9	530.0	23	49.2	74.8
522729	-	-	-	-	-	23	124.0	157.3

Table 5: iterations and time for Example 5.2.

	Newton		Broyden		
N	$\sharp$ it	time(s.)	$\sharp$ it	time(s.)	
3969	10	69.1	10	6.6	2.8
9025	10	647.1	10	64.3	19.9
16129	11	3212.9	10	298.0	54.8
65025	-	-	-	-	-

Table 6: iterations and time for Example 5.3.

[15] L. Grasedyck and W. Hackbusch, *Construction and arithmetics of $\mathcal{H}$-matrices*, Computing 70 (2003), pp. 295–334.

[16] W. Hackbusch, *A sparse matrix arithmetic based on $\mathcal{H}$-matrices. I. Introduction to $\mathcal{H}$-matrices*, Computing, 62 (1999), pp. 89–108.

[17] W. Hackbusch and B. N. Khoromskij, *A sparse $\mathcal{H}$-matrix arithmetic. II. Application to multi-dimensional problems*, Computing, 64 (2000), pp. 21–47.

[18] N. J. Higham: *Accuracy and stability of numerical algorithms.*, Second Edition, SIAM, Philadelphia, PA, 2002.

[19] J. Karátson and I. Faragó, *Variable preconditioning via quasi-Newton methods for nonlinear problems in Hilbert space*, SIAM J. Numer. Anal., 41 (2003), pp. 1242–1262.

[20] J. Xu, *Two-grid discretization techniques for linear and nonlinear PDEs*, SIAM J. Numer. Anal., 33 (1996), pp. 1759–1777.

	$\mathcal{H}$-Newton		$\mathcal{H}$-Broyden			$\mathcal{H}$-Broyden-LU		
N	$\sharp$ it	time(s.)	$\sharp$ it	time(s.)		$\sharp$ it	time(s.)	
3969	10	13.8	10	1.3	1.4	9	0.2	0.4
9025	10	49.9	10	5.1	4.3	9	0.7	0.8
16129	11	130.6	10	11.4	7.7	9	1.4	1.5
65025	11	900.4	10	82.1	42.5	9	8.3	6.4
130321	12	2940.8	10	241.6	99.9	9	20.3	15.6
261121	12	6568.8	11	548.1	238.2	10	49.8	33.7
522729	-	-	-	-	-	13	125.6	89.9

Table 7: iterations and time for Example 5.3.

	Newton		Broyden		
N	$\sharp$ it	time(s.)	$\sharp$ it	time(s.)	
3969	5	35.3	5	7.0	1.5
9025	5	336.4	5	62.7	7.3
16129	5	1422.5	5	300.7	26.8
65025	-	-	-	-	-

Table 8: iterations and time for Example 5.4.

	$\mathcal{H}$-Newton		$\mathcal{H}$-Broyden			$\mathcal{H}$-Broyden-LU		
N	$\sharp$ it	time(s.)	$\sharp$ it	time(s.)		$\sharp$ it	time(s.)	
3969	5	6.9	5	1.3	0.6	4	0.2	0.1
9025	5	25.9	5	5.1	1.7	4	0.7	0.3
16129	5	58.6	5	11.5	3.4	4	1.4	0.6
65025	5	411.4	5	82.4	18.8	5	8.4	3.1
130321	5	1232.1	6	247.0	55.5	7	20.4	10.1
261121	6	3305.4	6	549.8	114.0	8	49.4	22.8
522729	-	-	-	-	-	13	123.6	71.8

Table 9: iterations and time for Example 5.4.