



# A Flow Perspective on Nonlinear Least-Squares Problems

Hans Georg Bock<sup>1</sup> · Jürgen Gutekunst<sup>1</sup>  · Andreas Potschka<sup>1</sup> · María Elena Suárez Garcés<sup>1</sup>

Received: 25 November 2019 / Accepted: 8 July 2020 / Published online: 3 October 2020  
© The Author(s) 2020

## Abstract

Just as the damped Newton method for the numerical solution of nonlinear algebraic problems can be interpreted as a forward Euler timestepping on the Newton flow equations, the damped Gauß–Newton method for nonlinear least squares problems is equivalent to forward Euler timestepping on the corresponding Gauß–Newton flow equations. We highlight the advantages of the Gauß–Newton flow and the Gauß–Newton method from a statistical and a numerical perspective in comparison with the Newton method, steepest descent, and the Levenberg–Marquardt method, which are respectively equivalent to Newton flow forward Euler, gradient flow forward Euler, and gradient flow backward Euler. We finally show an unconditional descent property for a generalized Gauß–Newton flow, which is linked to Krylov–Gauß–Newton methods for large-scale nonlinear least squares problems. We provide numerical results for large-scale problems: An academic generalized Rosenbrock function and a real-world bundle adjustment problem from 3D reconstruction based on 2D images.

**Keywords** Nonlinear least squares · Gauß–Newton · Globalization · Continuous flows

**Mathematics Subject Classification (2010)** 58C15 · 65H20 · 65K05 · 65L07

---

Dedicated to Volker Mehrmann on the occasion of his 65th birthday.

✉ Jürgen Gutekunst  
juergen.gutekunst@iwr.uni-heidelberg.de

Hans Georg Bock  
bock@iwr.uni-heidelberg.de

Andreas Potschka  
potschka@iwr.uni-heidelberg.de

María Elena Suárez Garcés  
mariaelenasuarezgarces@gmail.com

<sup>1</sup> Interdisciplinary Center for Scientific Computing (IWR), Heidelberg University, Im Neuenheimer Feld 205, 69120, Heidelberg, Germany

# 1 Introduction

We consider nonlinear least-squares problems of the form

$$\min \frac{1}{2} \|f(x)\|_2^2 \quad \text{over } x \in \mathbb{R}^n, \tag{1}$$

where  $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$  is twice continuously differentiable. The necessary optimality conditions of (1) require that in a minimum  $x^* \in \mathbb{R}^n$  the gradient of the objective function vanishes

$$F(x) := f'(x)^T f(x) = 0, \tag{2}$$

where  $f'$  denotes the Jacobian of  $f$ . Equation (2) is called the normal equation. It is important to realize that the normal equation is only a necessary condition for a minimum, which is also satisfied in other spurious stationary points (maxima, saddle points) of the objective. Stationary points can often be characterized by their Hessian

$$F'(x) = f'(x)^T f'(x) + \sum_{i=1}^m f_i(x) \nabla^2 f_i(x) =: H(x) + Q(x),$$

which splits up into a first order term  $H$  based solely on  $f'$  and a second-order term  $Q$ , in which the current residual  $f$  and its Hessian tensor enter (but no first order derivatives).

*Example 1* (Parameter estimation) The main area of application for this type of problem is maximum likelihood estimation for parameter estimation problems with normally distributed measurement errors (see, e.g., [4]): In this case, we would like to determine an unknown but deterministic model parameter  $\bar{x} \in \mathbb{R}^n$ , which parametrizes a deterministic model response function  $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ . We assume we can obtain measurements from the real-world system modeled by  $h$  and  $\bar{x}$  with normally distributed measurements  $\eta \in \mathbb{R}^m$  with expected value  $h(\bar{x})$  and known covariance  $\Sigma$ , which is an  $m$ -by- $m$  symmetric positive semi-definite matrix. With  $\Gamma = \Sigma^{-\frac{1}{2}}$ , the maximum likelihood estimation problem becomes a nonlinear least squares problem of the form (1) by setting

$$f(x) = \Gamma[h(x) - \eta].$$

Obviously, the measurements  $\eta$  enter linearly into  $f$  and thus all derivatives of  $f$  are independent of  $\eta$ . In particular, they only enter in the Hessian  $F'(x)$  through the second order term  $Q$  in a multiplicative fashion.

**Related previous work** The Gauß–Newton method (see, e.g., [4, 5, 17, 25]), which solves a sequence of linear least-squares problems (linearized within the norm), is the method of choice to solve problem (1) in many cases. Like all Newton-type methods, it is in general only locally convergent. Because its search direction is a descent direction for the objective, line-search methods such as [3] are applicable to enforce convergence from arbitrary starting points. Alternatively, the Levenberg–Marquardt method [15, 16] uses an adaptive Tikhonov-type regularization to enforce convergence, which is equivalent to a trust-region globalization [17, 27]. Starting with the seminal paper by Davidenko [7], a large body of literature is dedicated to the construction of globalization methods by applying time-stepping methods for ordinary differential equations to the Newton flow equations (see, e.g., [6, 8–10, 14]) or generalized Newton flow equations [22, 23, 28]. Similarly, gradient descent and proximal point algorithms can be interpreted as forward or backward Euler time-stepping on the gradient flow equations (we refer the reader to the discussion in [20, Sec. 4]). A detailed study of the Gauß–Newton flow equations appears to be missing up to now.

**Contributions and structure of the paper** We review the local convergence of the Gauß–Newton method and the notion of statistically stable solutions, which are defined by a spectral radius condition (Section 2). We introduce the Gauß–Newton flow and compare it with the Newton flow and gradient flow (Section 3). The Levenberg–Marquardt method is recovered as a backward Euler timestepping on the gradient flow, which means that it is a proximal point method. The damped Gauß–Newton method is a forward Euler timestepping scheme on the Gauß–Newton flow equations. We illustrate certain salient features of the different flows on the well-known Rosenbrock function. We also learn from this example that the local spectral radius condition has no meaningful global counterpart along the Gauß–Newton flow, which in turn implies that full steps have to be taken locally to guarantee convergence to only statistically stable solutions (Section 4). For the case of inexact Gauß–Newton methods based on approximate solutions of the linearized least-squares systems with the aid of Krylov subspace methods, we show that the LSQR search direction [19] is always a descent direction (Section 5). Finally, we provide in Section 6 numerical results for Krylov–Gauß–Newton methods on large-scale academic and real-world problems: A generalized Rosenbrock function and a bundle adjustment problem from 3D reconstruction based on 2D images.

## 2 The Gauß–Newton Method

The (local) Gauß–Newton method generates a sequence  $(x_k)_k \subset \mathbb{R}^n$  from an initial solution guess  $x_0 \in \mathbb{R}^n$  via the iteration

$$x_{k+1} = x_k + s_k,$$

where the increments  $s_k \in \mathbb{R}^n$  are the solutions of the linear least-squares problems (linearization under the norm)

$$\min \frac{1}{2} \|f'(x_k)s + f(x_k)\|_2^2 \quad \text{over } s \in \mathbb{R}^n. \tag{3}$$

This problem is convex and always has a solution. If  $f'(x_k)$  has full column rank, the solution is uniquely given by the Moore–Penrose pseudoinverse  $f'(x_k)^+ = [f'(x_k)^T f'(x_k)]^{-1} f'(x_k)^T$  via

$$s_k = -f'(x_k)^+ f(x_k).$$

There is also a different viewpoint: The Gauß–Newton method is a Newton-type method on the normal equation (2), where we discard the second-order term  $Q$  in the approximation of the inverse of the Hessian  $F'$ , i.e.,

$$x_{k+1} = x_k - f'(x_k)^+ f(x_k) = x_k - H(x_k)^{-1} F(x_k).$$

At the center of the analysis of the Gauß–Newton method and the Gauß–Newton flow (to be described in Section 4) is the matrix

$$G(x) := H(x)^{-1} Q(x).$$

**Lemma 1** *If  $f'(x)$  has full column rank, then the spectrum of  $G(x)$  is real and its eigenvectors are real and orthogonal with respect to the scalar product endowed by  $f'(x)^T f'(x)$ .*

*Proof* Any eigenpair  $(\lambda, v) \in \mathbb{C} \times \mathbb{C}^n$  of  $G(x)$  is a solution of the generalized eigenvalue problem

$$Q(x)v = \lambda H(x)v.$$

Because  $Q(x)$  is symmetric and  $f'(x)$  has full rank,  $H(x) = f'(x)^T f'(x)$  is positive definite, which implies that  $(\lambda, v)$  are real and the orthogonality of the eigenvectors with respect to  $H(x)$  (see, e.g., [21]). □

It is well-known that the local convergence of the Gauß–Newton method to a solution  $x^* \in \mathbb{R}^n$  of the normal equations (2) is only guaranteed by Ostrowski’s theorem [13, 18] if the spectral radius condition

$$\kappa_{GN} := \rho(G(x^*)) < 1 \tag{4}$$

holds. The convergence is linear with rate equal to  $\kappa_{GN}$ . Conversely, if  $\kappa_{GN} > 1$ , it is easy to see that there exists an arbitrarily small perturbation of  $x^*$  in the direction of principal eigenvector of the matrix in (4), which leads the iterates away from  $x^*$ .

Because the residuals  $f(x)$  enter multiplicatively in  $Q(x)$ , solutions with small residual will satisfy the condition (4). It might appear as a disadvantage of the Gauß–Newton method that it will not be attracted by solutions of the normal equation (2) that violate (4) in comparison with, e.g., a Newton method, which exhibits local convergence to any solution to the normal equation (2). For parameter estimation problems, however, this behavior is actually an advantage:

*Example 2* (Mirror trick, see [5]) We continue with Example 1. Assume we have obtained a minimizer  $x^* \in \mathbb{R}^n$  of (1) for which  $\kappa_{GN} > 1$ . By Lemma 1, there exists an eigenvalue  $\lambda \in \mathbb{R}^n$  of  $G(x^*)$  with modulus greater than one. Let  $v \in \mathbb{R}^n \setminus \{0\}$  denote the eigenvector of  $G(x^*)$  corresponding to  $\lambda$ , which implies that

$$Q(x^*)v = \lambda H(x^*)v. \tag{5}$$

Because  $x^*$  is a minimizer, the Hessian must be positive semi-definite, which together with (5) leads to

$$0 \leq v^T F'(x^*)v = v^T H(x^*)v + v^T Q(x^*)v = (1 + \lambda)v^T H(x^*)v = (1 + \lambda)\|f'(x^*)v\|_2^2.$$

If  $f'(x^*)$  has full column rank, then  $\lambda > -1$  and because  $|\lambda| = \kappa_{GN} > 1$  it follows that  $\lambda > 1$ .

By our statistical assumptions, the measurement errors are symmetrically distributed around the true model response. Let us, in a thought experiment, mirror the measurements  $\eta_i$  around  $h_i(x^*)$ , i.e., we generate statistically plausible measurements

$$\tilde{\eta} := 2h(x^*) - \eta.$$

The corresponding least-squares residuals are

$$\tilde{f}(x) = \Gamma [h(x) - \tilde{\eta}] = \Gamma [h(x) - (2h(x^*) - \eta)],$$

which indeed implies mirrored residuals  $\tilde{f}(x^*) = -f(x^*)$ . Because we changed  $f$  only by a constant, the derivatives of  $\tilde{f}$  and  $f$  coincide. Hence,  $x^*$  also solves the normal equations (2) for  $\tilde{f}$  due to

$$\tilde{F}(x^*) = \tilde{f}'(x^*)\tilde{f}(x^*) = -f'(x^*)f(x^*) = -F(x^*) = 0.$$

For the Hessian, the sign of the  $Q$  term swaps because

$$\tilde{F}'(x^*) = f'(x^*)^T f'(x^*) - \sum_{i=1}^m f_i(x^*)\nabla^2 f_i(x^*) = H(x^*) - Q(x^*). \tag{6}$$

From (6) and (5), we obtain

$$v^T \tilde{F}'(x^*)v = (1 - \lambda)v^T H(x^*)v = (1 - \lambda)\|f'(x^*)v\|_2^2.$$

Because  $\lambda > 1$ , this shows that the Hessian  $\tilde{F}'(x^*)$  has negative curvature in the direction of  $v$  if  $f'(x^*)$  has full column rank. It is, thus, not a minimizer of (1) with the mirrored measurements. In this sense,  $x^*$  is not a *statistically stable* solution of (1) for the original measurements  $\eta$ . It is indeed advantageous that the Gauß–Newton method will not be attracted to such spurious solutions of the normal equation (2), in contrast to, e.g., the Newton method.

### 3 Continuous Flows and Timestepping Methods for Least-Squares Problems

In connection with (1), we give three examples of ordinary differential initial value problems. The corresponding flows are the Newton flow, the gradient flow, and the Gauß–Newton flow. Together with appropriate timestepping in the form of forward and backward Euler methods, we can derive the most popular globalization techniques for solving nonlinear least-squares problems.

#### 3.1 The Newton Flow

We can derive the Newton flow by embedding the normal equation (2) into a family of root-finding problems that depend on a parameter  $t \in [0, \infty)$  according to

$$F(x) - e^{-t} F(x_0) = 0.$$

Obviously,  $x_0$  is a solution for  $t = 0$  and we obtain a solution of the normal equations for  $t = \infty$ . If  $F'(x_0)$  is invertible, the Implicit Function Theorem guarantees a local solution  $x(t)$  satisfying

$$\dot{x}(t) = -F'(x(t))^{-1} F(x(t)), \quad x(0) = x_0. \tag{7}$$

A forward Euler step on the Newton flow equation (7) delivers one Newton step

$$x_{k+1} = x_k - F'(x_k)^{-1} F(x_k).$$

We strongly discourage the use of the Newton method for solving (1), because the iterates are attracted to spurious solutions of the normal equation (2) such as maxima or saddle points of (1). Furthermore, the use of the full Hessian  $F'(x)$  is problematic if  $F'(x_0)$  is indefinite, because any continuous path  $x(t)$  from  $x_0$  to a minimum  $x^*$  of (1), where  $F'(x^*)$  is necessarily positive semi-definite, must entail a point  $\bar{t}$  at which  $F'(x(\bar{t}))$  is singular by the intermediate value theorem applied to the (real) eigenvalues of  $F'(x(t))$ . In this case, the Newton flow and the Newton step computation break down.

#### 3.2 The Gradient Flow

A flow that is only attracted to minima is the gradient flow

$$\dot{x}(t) = -\nabla \left( \frac{1}{2} \|f(x(t))\|_2^2 \right) = -f'(x(t))^T f(x) = -F(x), \quad x(0) = x_0. \tag{8}$$

Its equilibria are exactly the solutions of the normal equation (2) and equilibria  $x^*$  with indefinite or negative definite Hessian  $F'(x^*)$  are not asymptotically stable. It is interesting to note that the objective decreases along the gradient flow

$$\frac{d}{dt} \left( \frac{1}{2} \|f(x(t))\|_2^2 \right) = f(x(t))^T f'(x(t)) \dot{x}(t) = -\|F(x)\|_2^2.$$

A forward Euler discretization leads to the method of steepest descent (also called gradient descent)

$$x_{k+1} = x_k - t_k F(x_k),$$

which can be extremely slow for medium to badly conditioned problems. A backward Euler discretization leads to the step equations

$$x_k = x_{k+1} + t_k F(x_{k+1}) \Leftrightarrow F(x_{k+1}) + \frac{1}{t_k} (x_{k+1} - x_k) = 0.$$

These equations are the necessary optimality conditions of

$$\min \frac{1}{2} \|f(x)\|_2^2 + \frac{1}{2t_k} \|x - x_k\|_2^2 \quad \text{over } x \in \mathbb{R}^n.$$

Thus, backward Euler on the gradient flow delivers the Levenberg–Marquardt method, which adds a Tikhonov regularization term. This method has a strong link to trust-region methods, there the inverse steplength  $1/t_k$  plays the role of an optimal Lagrange multiplier for a quadratic trust-region constraint.

### 3.3 The Gauß–Newton Flow

The Gauß–Newton flow is given by

$$\dot{x}(t) = -f'(x(t))^+ f(x(t)), \quad x(0) = x_0. \tag{9}$$

As for the gradient flow (8), the objective decreases along the Gauß–Newton flow because

$$\frac{d}{dt} \left( \frac{1}{2} \|f(x(t))\|_2^2 \right) = f(x(t))^T f'(x(t)) \dot{x}(t) = -F(x(t))^T H(x(t))^{-1} F(x(t)) \leq 0.$$

Forward Euler timestepping delivers the damped Gauß–Newton method

$$x_{k+1} = x_k - t_k f'(x_k)^+ f(x_k).$$

Backward Euler timestepping yields the step equations

$$x_k = x_{k+1} + t_k f'(x_{k+1})^+ f(x_{k+1}) \Leftrightarrow F(x_{k+1}) + t_k H(x_{k+1}) [x_{k+1} - x_k] = 0,$$

which could be interpreted as a variable metric Levenberg–Marquardt method. Unfortunately, the step equation cannot be readily interpreted as an optimality condition. If we approximate  $H(x_{k+1})$  by  $H(x_k)$ , we essentially arrive again at a damped Gauß–Newton method with damping factor  $1/(1 + t_k)$ .

An interesting alternative interpretation of the Gauß–Newton method is to embed the original problem (1) in a family of problems parameterized by  $t \in [0, \infty)$ , whose solution is  $x_0$  for  $t = 0$  and a solution  $x^*$  of (1) for  $t = \infty$ :

$$\min \frac{1}{2} \|f(x) - e^{-t} f(x_0)\|_2^2 \quad \text{over } x \in \mathbb{R}^n. \tag{10}$$

This gives rise to a residual in the normal equations of the form

$$F(x) = e^{-t} f'(x)^T f(x_0).$$

Using the implicit function theorem analogously to the Newton flow case, we arrive at the initial value problem

$$\left( H(x(t)) + \sum_{i=1}^m [f_i(x(t)) - e^{-t} f_i(x_0)] \nabla^2 f_i(x(t)) \right) \dot{x}(t) = -F(x(t)), \quad x(0) = x_0. \tag{11}$$

If  $f'(x_0)$  has full column rank, then a local solution  $x(t)$  exists and the initial flow direction equals the Gauß–Newton flow

$$\dot{x}(0) = -f'(x(0))^+ f(x(0)).$$

There is no guarantee, however, that the matrix on the left-hand side of (11) stays invertible for all  $t \geq 0$ . Nonetheless, we can embed the solution of (10) with a finite  $t$  in an outer loop, in which we sequentially update  $x_0$  by an approximate solution of (10). This leads to a sequential homotopy method for Gauß–Newton methods similar to the one proposed in [24] for inexact Sequential Quadratic Programming methods.

### 3.4 Comparison of the Newton, Gradient, and Gauß–Newton Flows

We compare the different flows for the classical Rosenbrock function, which fits into the class of parameter estimation problems (Example 1) with  $n = m = 2$  and

$$h(x) = \begin{pmatrix} x_2 - x_1^2 \\ 1 - x_1 \end{pmatrix}, \quad \eta = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \Gamma = \begin{pmatrix} 10 & 0 \\ 0 & 1 \end{pmatrix},$$

and which yields the well-known optimization problem

$$\min \frac{1}{2} [100(x_2 - x_1^2)^2 + (1 - x_1)^2] \quad \text{over } x \in \mathbb{R}^2.$$

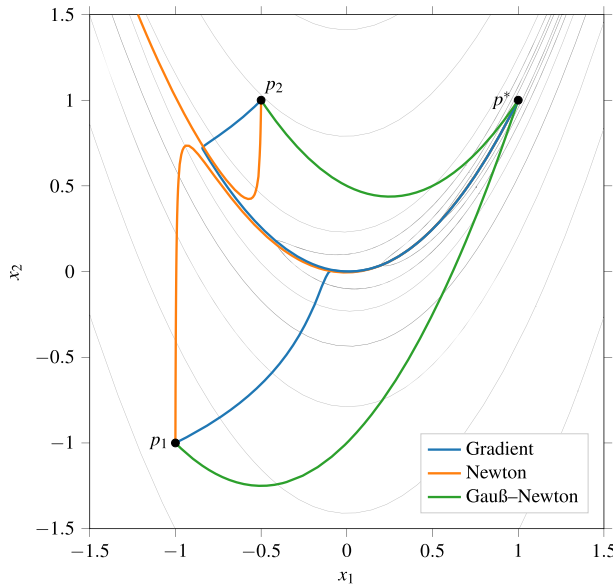
The point  $x^* = (1, 1)^T$  is a unique global minimum, because it is the only point which attains the lower bound 0 of the objective. We can easily compute

$$\begin{aligned} f'(x) &= \begin{pmatrix} -20x_1 & 10 \\ 1 & 0 \end{pmatrix}, & F(x) &= \begin{pmatrix} -200x_1(x_2 - x_1^2) + x_1 - 1 \\ 100(x_2 - x_1^2) \end{pmatrix}, \\ F'(x) &= \begin{pmatrix} 600x_1^2 - 200x_2 + 1 & -200x_1 \\ -200x_1 & 100 \end{pmatrix}. \end{aligned}$$

Solving for  $\det F'(x) = 0$ , we obtain that the Hessian  $F'(x)$  is singular along the parabola  $x_2 = x_1^2 + \frac{1}{200}$ , which is just slightly above the trough of the banana valley  $x_2 = x_1^2$ . The right-hand sides of the gradient, Newton, and Gauß–Newton flows can be verified to be

$$\begin{aligned} -F(x) &= \begin{pmatrix} 200x_1(x_2 - x_1^2) + 1 - x_1 \\ -100(x_2 - x_1^2) \end{pmatrix}, \\ -F'(x)^{-1}F(x) &= \frac{1}{200x_1^2 - 200x_2 + 1} \begin{pmatrix} 1 - x_1 \\ 200x_1^4 - 400x_1^2x_2 - x_1^2 + 2x_1 + 200x_2^2 - x_2 \end{pmatrix}, \\ -f'(x)^+ f(x) &= \begin{pmatrix} 1 - x_1 \\ 2x_1 - x_1^2 - x_2 \end{pmatrix}. \end{aligned}$$

Obviously, the Newton flow has a singularity along the shifted parabola  $x_2 = x_1^2 + \frac{1}{200}$ . Thus, the Newton flow cannot cross this barrier if started from above it (cf. Fig. 1).



**Fig. 1** A comparison of gradient, Newton, and Gauß–Newton flows emanating from the two different starting points  $p_1 = (-1, -1)$  and  $p_2 = (-\frac{1}{2}, 1)$  for the Rosenbrock example. Level sets of the objective function are indicated in black. The unique global optimum lies at  $p^* = (1, 1)$ . The gradient flow is perpendicular to the level sets. The Newton flow emanating from the upper point cannot cross the manifold  $x_2 = x_1^2 + \frac{1}{200}$ , where the Hessian  $F'(x)$  is singular, and diverges to infinity. The Gauß–Newton flows do not exhibit sharp turns

Moreover, we see that in the Rosenbrock example, the Gauß–Newton flow does not suffer from stiffness, which is induced in the gradient and Newton flow by fast transients towards the trough of the banana valley (cf. Fig. 1), where they take a sharp turn.

Two remarks are appropriate here: First, fast attraction to the banana-shaped bottom of the valley is detrimental for line-search and trust-region based globalization if no second-order correction is employed, because stepsizes and trust-region radii are required to be rather tiny to ensure descent in the objective. Second, from a stability point of view, forward Euler timestepping requires tiny stepsizes for stiff equations, which thus applies for the gradient and Newton method but not for the Gauß–Newton method. These two properties speak clearly in favor of using the Gauß–Newton method.

The obtuse angle of very nearly  $135^\circ$  between the Newton and the Gauß–Newton flows at  $\hat{x} = (-1, -1)$  in Fig. 1 hints at another important point: The convergence theory of inexact Newton methods based on generalized Newton flows (see [22, 23]) is not helpful in the analysis of the Gauß–Newton method. More rigorously, we can algebraically check that

$$\begin{aligned} \left\| F(\hat{x}) - F'(\hat{x})H(\hat{x})^{-1}F(\hat{x}) \right\|_2 &= \left\| 200 \begin{pmatrix} -\hat{x}_1^3 + \hat{x}_1^2 + \hat{x}_1\hat{x}_2 - \hat{x}_2 \\ 0 \end{pmatrix} \right\|_2 = 800 \\ &> \left\| \begin{pmatrix} -402 \\ -200 \end{pmatrix} \right\|_2 = \|F(\hat{x})\|_2. \end{aligned}$$

This means that the Gauß–Newton direction does not satisfy the contravariant  $\kappa$ -condition A2 in [23]. We also learn that its norm-free linearized counterpart, the  $\kappa_{GN}$ -condition (4), is



meaningful only in the vicinity of a solution  $x^*$  and that  $\rho(G(x))$  has no relevance for the globalization of Gauß–Newton methods for  $x$  far away from a solution  $x^*$ .

### 4 Asymptotic Stability of the Gauß–Newton Flow

The interpretation of the Gauß–Newton method as a Newton-type method provided in Section 2 allows for the study of generalized Newton flows [22, 23], which determine the behavior of the damped iteration

$$x_{k+1} = x_k + t_k s_k$$

for small stepsizes  $t_k \in (0, 1]$ . We can then interpret the Gauß–Newton iteration as a forward Euler timestepping method with stepsize  $t_k$  on the continuous Gauß–Newton flow  $x(t)$  determined by the initial value problem (9).

**Lemma 2** (Asymptotic stability of critical points) *Assume  $x^* \in \mathbb{R}^n$  is an equilibrium point of the Gauß–Newton flow  $x(t)$  in (9) and  $f'(x^*)$  has full column rank. With  $\lambda \in \mathbb{R}$  we denote the smallest eigenvalue of the matrix  $G(x^*)$ . Then,  $x^*$  is asymptotically stable if  $\lambda > -1$  and unstable if  $\lambda < -1$ .*

*Proof* We study the spectrum of the linearization of the flow right-hand side about  $x^*$ . To simplify notation, we abbreviate  $A(x) = f'(x)$ . The Moore–Penrose pseudoinverse can be differentiated with the formula [12, Theorem 4.3]

$$\frac{d}{dx} A^+ = -A^+ \frac{dA}{dx} A^+ + A^+ (A^+)^T \frac{dA^T}{dx} (I_m - AA^+) + (I_n - A^+A) \frac{dA^T}{dx} (A^+)^T A^+.$$

At  $x^*$ , we can exploit  $A(x^*)^+ f(x^*) = 0$  and in addition the general identities  $\frac{dA}{dx}(x) f(x) = Q(x) = Q(x)^T$  and  $A^+(A^+)^T = [A^T A]^{-1}$  to obtain

$$\left( \frac{d}{dx} A(x)^+ \right) \Big|_{x=x^*} f(x^*) = [A(x^*)^T A(x^*)]^{-1} Q(x^*).$$

Hence, we obtain for the linearization of the flow right-hand side around the critical point  $x^*$  that

$$\begin{aligned} \frac{d}{dx} (-f'(x)^+ f(x)) \Big|_{x=x^*} &= - \left( \frac{d}{dx} A(x^*)^+ \right) f(x^*) - f'(x^*)^+ f'(x^*) \\ &= -I_n - H(x^*)^{-1} Q(x^*) = -I_n - G(x^*), \end{aligned}$$

which has purely real eigenvalues by Lemma 1. This matrix has only negative eigenvalues for  $\lambda > -1$ , which implies that  $x^*$  is asymptotically stable, and at least one positive eigenvalue for  $\lambda < -1$ , in which case  $x^*$  is unstable. □

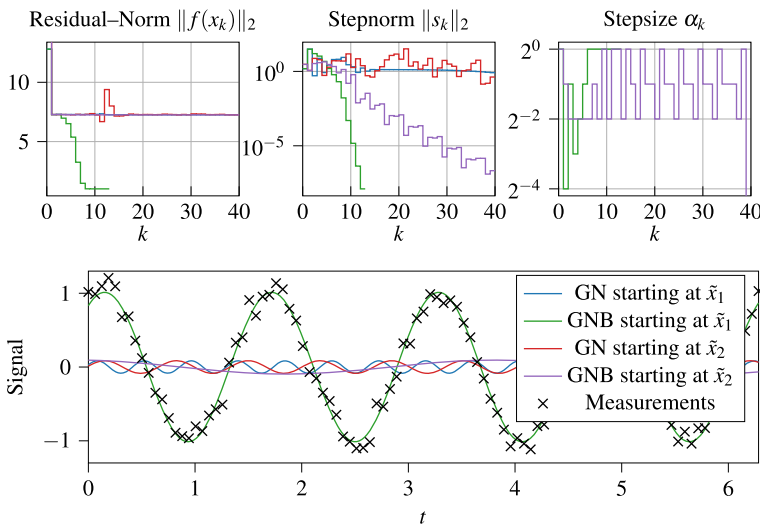
We believe a warning is appropriate at this point: In contrast to the spectral radius condition (4) on  $G(x^*)$  for the discrete Gauß–Newton iteration, the condition for asymptotic stability of the Gauß–Newton flow concerns only the negative part of the spectrum of  $G(x^*)$ . In the light of the mirror trick of Example 2, the flow might be attracted to statistically unstable solutions. We illustrate this behavior in the following example by the means of a simple parameter estimation problem that we want to solve with a damped Gauß–Newton method using a simple Armijo backtracking line-search.

*Example 3* (Frequency reconstruction) Given a sequence of  $n + 1$  measurements  $(\eta_i)_{0 \leq i \leq n}$  taken at time  $t_i = \frac{2\pi i}{n}$  from the signal  $h_i(a^*, \omega^*, \Phi^*) = a^* \sin(\omega^* t_i + \Phi^*)$  we aim to reconstruct the amplitude, frequency and phase  $x^* = (a^*, \omega^*, \Phi^*) \in \mathbb{R}^3$ . Under the assumption that the measurement noise  $\eta_i - h_i(x^*)$  is normally distributed around 0 with covariance  $\sigma = 0.1$ , this leads to the maximum likelihood parameter estimation problem

$$\min \frac{1}{2\sigma} \|f(x)\|_2^2 \quad \text{over } x = (a, \omega, \Phi) \in \mathbb{R}^3$$

with  $f(x) = [\eta_0 - h_0(x), \dots, \eta_n - h_n(x)]^\top$ . For  $n = 100$  and measurements generated with the true signal parameters  $x^* = (1, 4, 1)$  we apply a Gauß–Newton method with simple Armijo backtracking line-search using  $\alpha = \frac{1}{2}$  as backtracking factor and  $\beta = 0.1$  for the descent condition and compare the behavior with the full-step Gauß–Newton method for two different initial values (see Fig. 2).

Example 3 suggests how to interpret the convergence behavior of a damped Gauß–Newton method with line-search. The line-search condition will always guarantee descent and thus convergence to a stationary point within the current level-set. However, only if the convergence finally happens with full-steps, the solution will be statistically stable – a property that can be used in algorithms to distinguish statistically stable from statistically unstable solutions. To summarize it, a GN method with stepsize damping can help to globalize the convergence but in general it can not always prevent the iterates from getting trapped at a statistically unstable critical point, in particular when the initial guess is far away from



**Fig. 2** A comparison of the full-step Gauß–Newton method (denoted GN) with a damped Gauß–Newton method using a simple Armijo backtracking line-search (denoted GNB), each starting from two different points  $\tilde{x}_1 = (1.5, 6, 1.5)^\top$  and  $\tilde{x}_2 = (3, 2, 2)^\top$ . The upper plots represent the residual-norm of all iterates, the stepnorm of all iterates, and the stepsize of the iterates using the line-search method until the termination criterion  $\|s_k\|_2 \leq 5 \cdot 10^{-8}$  or  $k \geq 40$  is reached. The lower plot represents the measurements and the model response of the final iterates. It can be seen that for both starting points the full-step GN method failed to converge and just diverges. The backtracking GN method converges for both points but the stepsize of the Armijo backtracking line-search only converges with full steps for the first point (green line)

the solution. In such a case domain knowledge has to be used to find a good initial guess for the iterates.

### 5 Krylov–Gauß–Newton Methods

For large-scale problems, we are forced to solve the linearized least-squares problems (3) only approximately, e.g., by Krylov subspace methods, which leads to the name Krylov–Gauß–Newton or Truncated Gauß–Newton method (cf. [13]).

Two particular Krylov subspace methods for the approximate solution of (3) are LSQR and LSMR [11, 19]. In their  $i$ -th subiteration, they construct an orthonormal basis  $V_i(x_k) \in \mathbb{R}^{n \times i}$  of the  $i$ -dimensional Krylov subspace

$$K_i(H(x_k), F(x_k)) := \{p(H(x_k))F(x_k) \mid p \text{ is a polynomial of degree at most } i - 1\}$$

by a Golub–Kahan bidiagonalization process. (If the dimension of  $K_i(H(x_k), F(x_k))$  is smaller than  $i$ , then the solution is already contained in  $K_{i-1}(H(x_k), F(x_k))$ .) We then solve the reduced space linear least-squares problems

$$\min \frac{1}{2} \|f'(x_k)V_i(x_k)y + f(x_k)\|_2^2 \quad \text{over } y \in \mathbb{R}^i, \tag{LSQR}$$

$$\min \frac{1}{2} \|f'(x_k)^T [f'(x_k)V_i(x_k)y + f(x_k)]\|_2^2 \quad \text{over } y \in \mathbb{R}^i. \tag{LSMR}$$

In that sense, LSQR strives for the best linearized fit, while LSMR strives for the lowest violation of the linearized normal equation.

For LSQR, the normal equation

$$V_i(x_k)^T H(x_k)V_i(x_k)y + V_i(x_k)^T F(x_k) = 0$$

delivers the solution

$$y_i^*(x_k) = - [V_i(x_k)^T H(x_k)V_i(x_k)]^{-1} V_i(x_k)^T F(x_k),$$

provided that  $f'(x_k)$  has full column rank. For fixed  $i$ , we can consider the LSQR–Gauß–Newton flow equations

$$\dot{x}(t) = V_i(x(t))y_i^*(x(t)), \quad x(0) = x_0.$$

For  $i \geq 1$ , the objective value is guaranteed to decrease along the flow due to

$$\frac{d}{dt} \left( \frac{1}{2} \|f(x)\|_2^2 \right) = F(x)^T \dot{x} = -F(x)^T V_i(x) [V_i(x)^T H(x)V_i(x)]^{-1} V_i(x)^T F(x) < 0$$

whenever  $F(x) \neq 0$  and  $f'(x)$  has full column rank (suppressing the arguments of  $x = x(t)$ ). We see that there is no further condition on the required accuracy to solve (3) other than  $i \geq 1$ : We always obtain a descent direction. This observation suggests the following numerical strategy: Far away from a solution, we approximate solutions to (3) with rather low accuracy. When we are close to a solution, indicated for instance by stagnation in the nonlinear objective value, we tighten the accuracy more and more to finally benefit from the statistical stability properties of the exact Gauß–Newton direction. A simple implementation of this approach is illustrated in Algorithm 1.

---

**Algorithm 1** LSQR–Gauß–Newton method.

---

```

1: function LSQRGAUSSNEWTON( $x_0$ ,  $x_{tol}$ ,  $o_{tol}$ ,  $k_{max}$ ,  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\sigma$ ,  $\tau$ ,  $\tau_{min}$ )
2:   Evaluate  $f(x_0)$ 
3:   for  $k = 0, \dots, k_{max}$  do
4:     Compute  $s_k$  as an approximate solution of (3) by LSQR [11] with BTOL = 0
       and ATOL =  $\tau$ 
5:     Set  $t \leftarrow 1$ , evaluate  $f(x_k)^T f'(x_k)s_k$ 
6:     loop ▷ Armijo backtracking line-search
7:       Set  $x_{k+1} \leftarrow x_k + ts_k$ , evaluate  $f(x_{k+1})$ 
8:       if  $\|f(x_{k+1})\|_2^2 \leq \|f(x_k)\|_2^2 + 2t\beta f(x_k)^T f'(x_k)s_k$  then break
9:       Backtrack  $t \leftarrow \alpha t$ 
10:    if  $\|f(x_k)\|_2 - \|f(x_{k+1})\|_2 \leq \sigma \max\{\|f(x_{k+1})\|_2, 1\}$  then reduce  $\tau \leftarrow \max\{\gamma\tau, \tau_{min}\}$ 
11:    if  $\|s_k\|_2 \leq x_{tol}$  or  $\|f(x_k)\|_2 - \|f(x_{k+1})\|_2 \leq o_{tol}\|f(x_0)\|_2$  then terminate

```

---

For LSMR, the normal equation reads

$$V_i(x_k)^T H(x_k)^2 V_i(x_k)y + V_i(x_k)^T H(x_k)F(x_k) = 0.$$

Repetition of the steps above leads to an objective value variation along the LSMR–Gauß–Newton flow of

$$\frac{d}{dt} \left( \frac{1}{2} \|f(x)\|_2^2 \right) = -F(x)^T V_i(x) \left[ V_i(x)^T H(x)^2 V_i(x) \right]^{-1} V_i(x)^T H(x)F(x).$$

From here, it is not clear whether we always get a descent direction. We have found LSMR to be working well in practice without giving rise to directions of ascent. This is in line with observation 1 in [11, Sec. 7.1], where the authors report that the residual  $r(x) = f'(x)V_i(x)y_i^*(x) + f(x)$  for LSMR “seems to be monotonic (no counterexamples were found)”.

## 6 Numerical Results

We report on our numerical experience with a line-search LSQR–Gauß–Newton method applied to two test cases: The first is an extended Rosenbrock function in variable dimensions with and without synthetic random measurements. The second is a challenging real-world problem called Bundle Adjustment for the construction of a 3D environment from a large number of markers in 2D pictures.

The results were obtained with a Python implementation of Algorithm 1. We used CasADi [2] for the computation of function derivatives. The default parameters were  $\alpha = \frac{1}{2}$  for the backtracking factor,  $\beta = \frac{1}{10}$  for the Armijo descent condition,  $\sigma = 10^{-4}$  for the objective stagnation test to trigger a reduction by a factor of  $\gamma = \frac{1}{10}$  of the LSQR tolerance  $\tau$ , initially set to  $10^{-3}$  and bounded below by  $\tau_{min} = 10^{-12}$ .

All computations were performed on a 64bit Ubuntu 16.04 Linux machine with 128 GB of RAM and Intel Core i7-5820K computational cores at 3.3 GHz. No parallelization was used.

### 6.1 An Extended Rosenbrock Function

We consider the following  $n$ -dimensional extension of the Rosenbrock function (cf. [26]) in the form of Example 1 with

$$\left. \begin{aligned} h_{2i-1}(x) &= x_i - 1, \\ h_{2i}(x) &= x_i^2 - x_{i+1}, \end{aligned} \right\} i = 1, \dots, n-1, \quad \Gamma_{ij} = \text{diag}(1, 10, \dots, 1, 10) \in \mathbb{R}^{(2n-2) \times (2n-2)}.$$

with measurements  $\eta \in \mathbb{R}^{2n-2}$  that are normally distributed with expectation 0 and covariance  $\Gamma^{-2}$ . For  $\eta = 0$ , the unique global optimum is  $x_i^* = 1$  for  $i = 1, \dots, n$ .

We run Algorithm 1 on this example with  $\text{xtol} = 10^{-5}$  and  $\text{otol} = 10^{-12}$  for 20 runs with different random realizations of  $\eta$  each for  $n = 10^1, \dots, 10^7$  with an initial guess  $(x_0)_i = 1$  for  $i = 1, \dots, n$ , which is not a solution because  $\eta \neq 0$ . The resulting statistics are displayed in Table 1. We see that we can find solutions of the large instances with a relatively small total number of LSQR iterations in comparison to the problem dimension. The required CPU time appears to grow linearly in  $n$ .

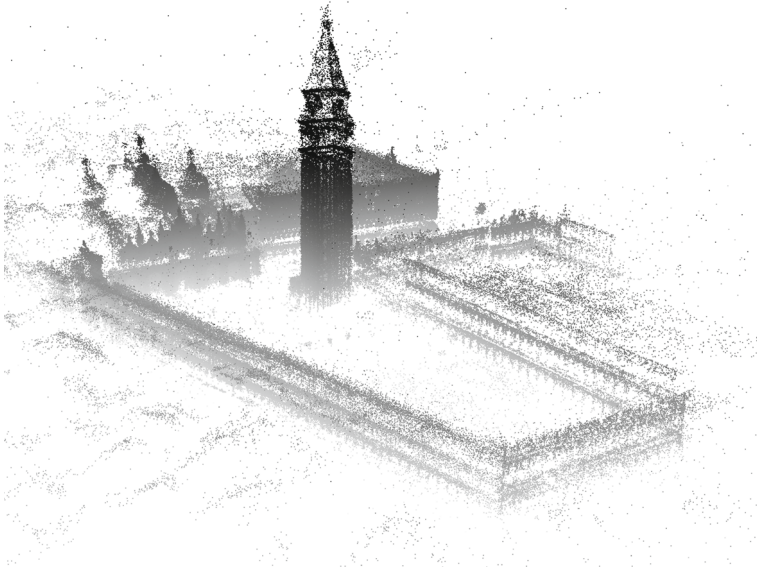
### 6.2 Large Scale Bundle Adjustment

Bundle Adjustment is a method of Structure from Motion (SfM) for estimating geometric 3D data from big sequences of 2D images. The core ingredient that couples the real world 3D position of an object and the resulting 2D position of the corresponding marker in the image is the camera model that depends on camera position, orientation, and some additional parameters such as focal lengths and lense distortion parameters. The camera model gives a prediction of the 2D positions of the markers depending on the real world 3D coordinates of an object and the camera parameters. By defining the residual function  $f$  as the difference of predicted 2D positions and the measurements, a large scale nonlinear least squares problem can be set up to estimate the most likely 3D positions and camera parameters. For a detailed description of the camera model and how the least-squares problem is defined we refer the reader to [1]. We note that, since 3D translation of the real-world coordinates and the camera positions by an arbitrary vector results in the same 2D image measurements, the bundle adjustment problems exhibit an intrinsic ambiguity, which results in a structural violation of the full-rank condition on  $f'(x)$ . As in [1], we rely on the regularizing properties of Krylov-space methods for our computations.

We use the bundle adjustment datasets *Ladybug*, *Trafalgar Square*, *Dubrovnik*, and *Venice*, which are freely available at <http://grail.cs.washington.edu/projects/bal> and consist

**Table 1** Statistics (minimal, median, and maximal numbers) of the results for the extended Rosenbrock example over 20 runs each

$n$	Iterations			Total LSQR iterations			CPU time		
	min	med	max	min	med	max	min	med	max
10	7	15	34	45	131	315	0.006	0.017	0.036
100	9	12	31	79	174	392	0.011	0.020	0.048
1000	8	11	29	65	225	646	0.019	0.040	0.111
10000	9	14	38	123	326	1201	0.127	0.246	0.792
100000	8	13	38	102	325	1285	1.110	2.362	7.723
1000000	9	12	24	107	254	668	13.764	23.041	54.875



**Fig. 3** 3D visualization of the largest *Venice* problem instance results

of several problems with varying numbers of 2D pictures and 3D objects (see Fig. 3 for a visualization resulting from the largest *Venice* dataset). The evaluation of the function  $f$  is partially based on the SciPy Cookbook Python Notebook, which is available at [https://scipy-cookbook.readthedocs.io/items/bundle\\_adjustment.html](https://scipy-cookbook.readthedocs.io/items/bundle_adjustment.html). We use the same Python implementation of Algorithm 1 as in the previous example. The derivatives are obtained using CasADi [2].

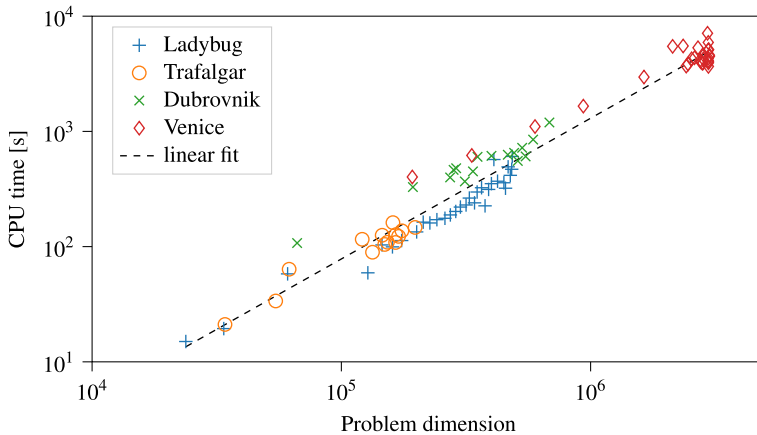
### 6.2.1 Results

We run Algorithm 1 with the parameters  $\alpha = \frac{1}{2}$  for the back-tracking factor,  $\beta = 10^{-3}$  for the Armijo descent condition,  $\sigma = 10^{-2}$  for the objective stagnation test to trigger a reduction by a factor of  $\tau = \frac{1}{10}$  of the LSQR tolerance  $\tau$ , initially set to  $\frac{1}{10}$  and bounded from below by  $\tau_{\min} = 10^{-4}$ . We chose  $\text{xtol} = 10^{-10}$  and  $\text{otol} = 10^{-7}$ . As in the synthetic extended Rosenbrock example, variable and residual dimensions are very nearly proportional for all problems.

In all cases, Algorithm 1 terminates within a maximum of 56 Krylov–Gauß–Newton iterations, always taking full steps eventually. As can be seen in Table 2, even the

**Table 2** Statistics (minimal, median, and maximal numbers) of the results for the Bundle Adjustment problem datasets

Dataset	Iterations			Total LSQR iterations		
	min	med	max	min	med	max
Ladybug	18	24	43	629	1933	4806
Trafalgar	22	28	42	2730	3290	4822
Dubrovnik	24	27	56	2944	4005	8561
Venice	22	30	44	3712	4831	9480



**Fig. 4** The CPU time for the approximate solution of different bundle adjustment problems with an LSQR–Gauß–Newton method grows linearly with the problem dimension

problems with large problem dimension are solved within relatively few Krylov–Gauß–Newton steps. We observe that the total CPU time appears to grow linearly with the problem dimension, see Fig. 4. When it comes to the total number of inner LSQR iterations, we note that again relatively few are needed compared to the problem dimensions, see Table 2.

## 7 Conclusion

In the full-rank case, the full-step Gauß–Newton method has the favorable property of not being attracted to statistically unstable minima. The nonlinear least-squares objective decreases along the the Gauß–Newton flow and the Krylov–Gauß–Newton flow if LSQR is used. The damped (Krylov–)Gauß–Newton method is equivalent to forward Euler timestepping on the (Krylov–)Gauß–Newton flow, while the Levenberg–Marquardt method is equivalent to backward Euler timestepping on the gradient flow. We showed for the popular example of the Rosenbrock function that the Gauß–Newton flow equations constitute a non-stiff differential equation, in contrast to the Newton and gradient flow equations, which give rise to fast transients towards the trough of the notorious banana-shaped valley. From this vantage point, line-search globalization for the Newton and gradient methods suffer from much stricter stepsize restrictions than the Gauß–Newton method. Krylov–Gauß–Newton methods have an intrinsic regularizing property to make them appropriate also for ill-conditioned large-scale nonlinear least-squares problems.

**Acknowledgements** The authors gratefully acknowledge support by the German Federal Ministry for Education and Research, grants no. 05M17MBA-MOPhaPro and 05M18VHA-MORENet.

**Funding** Open Access funding provided by Projekt DEAL.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Agarwal, S., Snavely, N., Seitz, S.M., Szeliski, R.: Bundle adjustment in the large. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) Proceedings of the 11th European Conference on Computer Vision: Part II, ECCV'10, pp. 29–42. Springer, Berlin (2010)
2. Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., Diehl, M.: CasADi: a software framework for nonlinear optimization and optimal control. *Math. Program. Comput.* **11**, 1–36 (2019)
3. Armijo, L.: Minimization of functions having Lipschitz continuous first partial derivatives. *Pac. J. Math.* **16**, 1–3 (1966)
4. Bates, D.M., Watts, D.G.: Nonlinear Regression Analysis and its Applications. In: Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. Wiley, New York (1988)
5. Bock, H.G.: Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen. *Bonner Mathematische Schriften*, vol. 183. Universität Bonn, Bonn (1987)
6. Bock, H.G., Kostina, E., Schlöder, J.P.: On the role of natural levelfunctions to achieve global convergence for damped newton methods. In: Powell, M.J.D., Scholtes, S. (eds.) *System Modelling and Optimization* (Cambridge, 1999), pp. 51–74. Kluwer Acad. Publ, Boston, MA (2000)
7. Davidenko, D.F.: On a new method of numerical solution of systems of nonlinear equations. *Dokl. Akad. Nauk SSSR (N.S.)* **88**, 601–602 (1953)
8. Deuffhard, P.: A modified newton method for the solution of ill-conditioned systems of nonlinear equations with application to multiple shooting. *Numer. Math.* **22**, 289–315 (1974)
9. Deuffhard, P.: Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms. *Springer Series in Computational Mathematics*, vol. 35. Springer, Berlin (2004)
10. Deuffhard, P.: The grand four: affine invariant globalizations of newton's method. *Vietnam J. Math.* **46**, 761–777 (2018)
11. Fong, D.C.-L., Saunders, M.: LSMR: An iterative algorithm for sparse least-squares problems. *SIAM J. Sci. Comput.* **33**, 2950–2971 (2011)
12. Golub, G.H., Pereyra, V.: The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM J. Numer. Anal.* **10**, 413–432 (1973)
13. Gratton, S., Lawless, A.S., Nichols, N.K.: Approximate Gauss–Newton methods for nonlinear least squares problems. *SIAM J. Optim.* **18**, 106–132 (2007)
14. Hohmann, A.: Inexact Gauss Newton methods for parameter dependent nonlinear problems. Ph.D. thesis, Freie Universität Berlin (1994)
15. Levenberg, K.: A method for the solution of certain non-linear problems in least squares. *Q. Appl. Math.* **2**, 164–168 (1944)
16. Marquardt, D.W.: An algorithm for least-squares estimation of nonlinear parameters. *J. Soc. Indust. Appl. Math.* **11**, 431–441 (1963)
17. Nocedal, J., Wright, S.J.: *Numerical Optimization*. Springer Series in Operations Research. Springer-Verlag, New York (1999)
18. Ortega, J.M., Rheinboldt, W.C.: *Iterative Solution of Nonlinear Equations in Several Variables*. Classics in Applied Mathematics, vol. 30. SIAM, Philadelphia, PA (2000). Reprint of the 1970 original
19. Paige, C.C., Saunders, M.A.: LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Softw.* **8**, 43–71 (1982)
20. Parikh, N., Boyd, S.: Proximal algorithms. *Found. Trends Optim.* **1**, 127–239 (2014)
21. Parlett, B.N.: The Symmetric Eigenvalue Problem. *Classics in Applied Mathematics*, vol. 20. SIAM, Philadelphia, PA (1998). Corrected reprint of the 1980 original
22. Potschka, A.: Backward step control for global newton-type methods. *SIAM J. Numer. Anal.* **54**, 361–387 (2016)
23. Potschka, A.: Backward step control for Hilbert space problems. *Numer. Algor.* **81**, 151–180 (2019)



24. Potschka, A., Bock, H.G.: A sequential homotopy method for mathematical programming problems. arXiv:1902.06984 (2019)
25. Schlöder, J.: Numerische Methoden Zur Behandlung Hochdimensionaler Aufgaben Der Parameteridentifizierung. Bonner Mathematische Schriften, 187 Universität Bonn, Bonn (1988)
26. Shang, Y.-W., Qiu, Y.-H.: A note on the extended rosenbrock function. *Evol. Comput.* **14**, 119–126 (2006)
27. Sorensen, D.C.: Trust region methods for unconstrained minimization. In: *Nonlinear Optimization, 1981* (Cambridge, 1981), NATO Conf. Ser. II: Systems Sci., pp. 29–38. Academic Press, London (1982)
28. Suárez Garcés, M.E.: Iterative linear algebra for parameter estimation. Ph.D. Thesis Heidelberg University (2018)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.