# The Large Residual Problem in Nonlinear Least Squares

Bastian Haase
Final Project Proposal

Nonlinear least squares problems form an important subcategory of numerical optimization. One of their most important applications lies in data fitting, which is, for instance, an important tool in statistics. A typical nonlinear least squares problem has the form $f(x) = \frac{1}{2}\sum_{i=1}^{m} r_j^2(x)$ for smooth functions $r_j : \mathbb{R}^n \to \mathbb{R}$ which are called *residuals*. Let $r = (r_j)_{j=1,..,m}^T$ denote the *residual vector* and $J$ its Jacobian. A short computation shows

$$\nabla f(x) = J(x)^T r(x) \;\text{ and }\; \nabla f(x) = J(x)^T J(x) + \sum_{i=1}^{m} r_j(x)\nabla^2 r_j(x)$$

which reveals an interesting feature of these problems: By computing the Jacobian $J$, one already gets the first term of the Hessian of $f$. Often, the residuals are small or nearly linear, in these cases the term $J^T J$ is a good approximation of the Hessian that can be used in Newton's method. This method is called Gauss-Newton and performs very well in a wide range of problems. However, one frequently occurs problems with large residuals where the described approximation does not yield satisfying results. In this work, I want to describe algorithms that also approximate the second term and analyze how well these algorithms performs in the large residual case.

## 1 Aim of the Project

The aim is to implement and compare 3 different strategies to approximate $S(x) = \sum_{i=1}^{m} r_i(x)\nabla^2 f(x)$ in a Newton method. We will test them against standard implementations of the Newton method, Gauss-Newton and BFGS. The first strategy relies on approximating $\nabla^2 f(x)$ by finite differences and to use this to approximate $S(x)$. Furthermore, we will implement two strategies proposed by Brown-Dennis ([BD71]) and Dennis-Gay-Welsh ([DGW81]) which approximate and update the term $S(X)$. It should be noted that Dennis-Gay-Welsh propose their update strategy in combination with a very sophisticated trust-region algorithm. This algorithm does not just adjust the trust region but also decides between their new update strategy or a standard Gauss-Newton model. As this work focuses on the approximation of $S(x)$, I will not implement the trust-region algorithm. Note that such an algorithm could also be paired with any other update strategy tested here. Finally, we will implement an algorithm proposed by Fletcher-Xu ([Fle86]). It is a hybrid algorithm of Gauss-Newton and BFGS using line search. This method chooses between Gauss-Newton and BFGS steps between each iteration depending on a parameter which we will tweak for every test problem.

### 1.1 Measurements

We will test all algorithms on four problems, all of which are classical examples to test nonlinear least squares algorithms. For each problem, we will pick a starting point with a large residual. Their dimensions range from $2 \leq n \leq 4$ and $2 \leq m \leq 20$. They differ in complexity of $\nabla^2 r_i(x)$ and one problem has a large residual even at the solution ($\approx 43000$). These problems seem to cover a wide range of scenarios and will give us indications on when the special update strategies are beneficial and which one performs best in different circumstances.

# References


[BD71] Kenneth M. Brown and John E. Dennis. New Computational Algorithms For Minimizing A Sum Of Squares Of Nonlinear Functions. *Research Report Number 71 - 6, Yale University*, 1971.

[DGW81] John E. Dennis, David M. Gay, and Roy E. Welsh. An Adaptive Nonlinear Least-Squares Algorithm. *ACM Transactions on Mathematical Software*, 7(3):348–368, 1981.

[Fle86] R Fletcher. *Practical methods of optimization.* Wiley, 1986.