AT&T
Bell Laboratories

Numerical Analysis Manuscript 90-11

Tradeoffs in Algorithms for Separable Nonlinear Least Squares

David M. Gay
Linda Kaufman

December 4, 1990

This appears on pp. 157–158 of *IMACS '91*, Proceedings of the 13th World Congress on Computational and Applied Mathematics, edited by R. Vichnevetsky and J. J. H. Miller, Criterion Press, Dublin, 1991.

TRADEOFFS IN ALGORITHMS FOR SEPARABLE NONLINEAR LEAST SQUARES

DAVID M. GAY
AT&T Bell Laboratories
600 Mountain Avenue, Room 2C–463
Murray Hill, NJ 07974-2070, U.S.A.

AND

LINDA KAUFMAN
AT&T Bell Laboratories
600 Mountain Avenue, Room 2C–461
Murray Hill, NJ 07974-2070, U.S.A.

***Abstract*—When nonlinear least squares problems are separable, i.e., some parameters appear linearly, it is often useful to project the linear parameters out of the problem, which leaves a nonlinear least-squares problem involving only the nonlinear parameters. For large-residual problems, it saves iterations but costs work and storage to use an ''augmented model'' in solving this problem. When there are multiple right-hand sides, the costs can be large, making a Levenberg-Marquardt or even a normal-equations algorithm more attractive; for some such problems, the associative law and extra orthogonal transformations can significantly reduce operation counts.**

Nonlinear least-squares problems often have the form

$$\text{minimize } \phi(c,x) := \tfrac{1}{2} \| \Phi(x)c - y(x) \|_2^2 ,$$

where Φ: $\mathbb{R}^p \longrightarrow \mathbb{R}^{n \times \ell}$ and y: $\mathbb{R}^p \longrightarrow \mathbb{R}^n$ are continuously differentiable. As Golub and Pereyra [5,6] have shown, minimizing $\phi(c,x)$ is equivalent to minimizing

$$\hat{\phi}(x) := \tfrac{1}{2} \| P_\Phi^\perp y(x) \|_2^2 , \tag{1}$$

where $P_\Phi^\perp = P_{\Phi(x)}^\perp$ projects orthogonally onto the null space of $\Phi(x)$. Indeed, if x^* minimizes $\hat{\phi}$, and if $c^* := \Phi(x^*)^- y(x^*)$, where Φ^- is a symmetric generalized inverse of Φ (i.e., $\Phi\Phi^-\Phi = \Phi$ and $(\Phi\Phi^-)^\mathrm{T} = \Phi\Phi^-$), then (c^*, x^*) minimizes ϕ.

Let $r(x) := P_{\Phi(x)}^\perp y(x)$ denote the residual in (1), and let $J = J(x)$ denote the Jacobian matrix of r, i.e., the matrix of first partials ($J_{i,j} = \partial r_i(x)/\partial x_j$). Since $\hat{\phi}$ has the form $\hat{\phi}(x) = \tfrac{1}{2} \| r(x) \|_2^2$, the gradient $\nabla\hat{\phi}$ of $\hat{\phi}$ is given by $\nabla\hat{\phi}(x) = J^\mathrm{T} r(x)$, and the Hessian (second derivative matrix) is

$$\nabla^2 \hat{\phi}(x) = J^\mathrm{T} J + \sum_{i=1}^{n} r_i(x) \nabla^2 r_i(x) . \tag{2}$$

The Gauss-Newton method for nonlinear least squares may be viewed as a variant of Newton's method in which the Hessian (2) is approximated by $J^\mathrm{T} J$; embedding the Gauss-Newton method in a trust-region method (for step-size control) gives Moré's variant [11] of the Levenberg-Marquardt method. Sometimes the term $\sum_i r_i \nabla^2 r_i$ in (2) is important, and approximating it by a secant update [2] can be worthwhile. In this case we speak of using an ''augmented model'' (of the objective function $\hat{\phi}$).

For separable nonlinear least squares, the jth column of the Jacobian matrix is given [5,6] by

$$J_{\cdot,j}(x) = -[(P_\Phi^\perp \frac{\partial \Phi}{\partial x_j} \Phi^-) + (P_\Phi^\perp \frac{\partial \Phi}{\partial x_j} \Phi^-)^\mathrm{T}] y(x) + P_\Phi^\perp \frac{\partial y}{\partial x_j}(x) . \tag{3}$$

Thus J has the form $J = K + L$, where $K_{\cdot,j} = P_\Phi^\perp [\frac{\partial y}{\partial x_j}(x) - \frac{\partial \Phi}{\partial x_j} \Phi^- y(x)]$; since the columns of K lie in the null space of Φ^T and those of L lie in the range of Φ, $K^\mathrm{T} L = \mathbf{0}$, $\nabla\hat{\phi} = K^\mathrm{T} r$, and $L^\mathrm{T} r = \mathbf{0}$. Kaufman [8] has suggested approximating J by K, i.e., using $K^\mathrm{T} K$ rather than $K^\mathrm{T} K + L^\mathrm{T} L$ as the ''Gauss-Newton'' Hessian approximation. Although this gives a less accurate Hessian approximation, the differences in function and gradient evaluations to solve a problem are often small and may be offset by savings in linear algebra time. Table 1 illustrates this. It shows function and gradient evaluations required to solve the separable test problems considered in [2], with both choices of Gauss-Newton model; the underlying least-squares solver is the variant of NL2SOL [3] that appears in the PORT3 subroutine library [1]. (Different local minimizers are found for ''KOW&OSB2'', the Golub-Pereyra one being smaller, but we attach little significance to this.)

Problem	n	ℓ	p	$K^\mathrm{T}K + L^\mathrm{T}L$ NF	$K^\mathrm{T}K + L^\mathrm{T}L$ NG	$K^\mathrm{T}K$ only NF	$K^\mathrm{T}K$ only NG
BARD	15	1	2	7	7	8	8
BEALE	3	1	1	5	5	5	5
BOX	10	1	2	7	7	7	7
FRDSTEIN	2	1	1	10	9	11	9
HELIX	3	1	2	16	13	16	13
KOW&OSB1	11	1	3	11	10	11	10
KOW&OSB2	11	1	3	14	11	16	15
MEYER	16	1	2	20	12	18	10
OSBORNE1	33	3	2	7	7	7	7
OSBORNE2	65	4	7	15	12	15	13
ROSNBROK	2	1	1	3	2	3	2
WOODS	7	2	2	47	31	43	27

Table 1. Golub-Pereyra versus Kaufman choices of Gauss-Newton Hessian

The algebra behind (3) deserves some discussion. If we factor $\Phi(x)$ as

$$\Phi = Q \begin{bmatrix} R & \bar{R} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} Z , \tag{4}$$

where Q is an orthogonal matrix (e.g., implicitly represented by the product of Householder matrices or Givens rotations — see [7]), $R \in \mathbb{R}^{\rho \times \rho}$ is an upper-triangular matrix of full rank $\rho \le \ell$, $\bar{R}$ is vacuous unless Φ is rank-deficient (i.e., unless $\rho < \ell$), and Z is a permutation matrix, then

$$P_\Phi^\perp = Q \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_{n-\rho} \end{bmatrix} Q^\mathrm{T} \tag{5}$$

(where $I_{n-\rho}$ is the $(n-\rho)\times(n-\rho)$ identity matrix), and a convenient choice of Φ^- is

$$\Phi^- = Z^\mathrm{T} \begin{bmatrix} R^{-1} \\ \mathbf{0} \end{bmatrix} (I_\rho, \mathbf{0}) Q^\mathrm{T} .$$

Kaufman's Gauss-Newton model permits cheaper linear algebra: rather than computing $r(x) := P_{\Phi(x)}^\perp y(x)$ and $K_{\cdot,j} = P_\Phi^\perp [\frac{\partial y}{\partial x_j}(x) - \frac{\partial \Phi}{\partial x_j} \Phi^- y(x)]$, it suffices to compute $\tilde{r}(x) := Q^\mathrm{T} r(x)$ and $\tilde{K}(x) := Q^\mathrm{T} K(x)$, i.e., to omit the multiplication by Q in (5), for we have $\nabla\hat{\phi} = K^\mathrm{T} r = \tilde{K}^\mathrm{T} \tilde{r}$ and $K^\mathrm{T} K = \tilde{K}^\mathrm{T} \tilde{K}$. This cheaper linear algebra makes the time to compute the final two columns in Table 1 roughly 25% less than that to compute the preceding two columns (on an SGI 4D/240S computer); such savings and the similar performances of the Golub-Pereyra and Kaufman Gauss-Newton models led us to use the latter in the separable nonlinear least-squares solvers that we put into the PORT3 subroutine library.

Separable nonlinear least-squares problems with multiple right-hand sides have further structure that is well worth exploiting. In such problems, Φ is a block-diagonal matrix with each diagonal block the same, say

$$\Phi(x) = \begin{bmatrix} G & & & \\ & G & & \\ & & \ddots & \\ & & & G \end{bmatrix}$$

with q diagonal blocks $G = G(x) \in \mathbb{R}^{\bar{n} \times \bar{\ell}}$, $n = q\bar{n}$, and $\ell = q\bar{\ell}$. Each successive block of $\bar{n}$ components of y is a different right-hand side and has its own linear parameters; the blocks are coupled by the nonlinear parameters, which all share. As Golub and LeVeque [4] have pointed out, a block QR factorization of Φ is appropriate; if we factor G as

$$G = \tilde{Q} \begin{bmatrix} \tilde{R} & \tilde{\tilde{R}} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tilde{Z}, \tag{6}$$

then we can use $Q = I_q \otimes \tilde{Q}$ and $R = I_q \otimes \tilde{R}$ in (4). This reduces the cost of computing (4) by a factor of about q^{-3}.

When $q = 1$, it makes most sense to compute the term $P_\Phi^\perp \frac{\partial \Phi}{\partial x_j} \Phi^- y(x)$ in (3) or Q^{T} times this term, i.e., the term $\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_{n-\rho} \end{bmatrix} Q^{\mathrm{T}} \frac{\partial \Phi}{\partial x_j} \Phi^- y(x)$ of $\tilde{K}_{\cdot,j}$, as $P_\Phi^\perp \left[\frac{\partial \Phi}{\partial x_j} [\Phi^- y(x)] \right]$ or $\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_{n-\rho} \end{bmatrix} \left[Q^{\mathrm{T}} \left[\frac{\partial \Phi}{\partial x_j} [\Phi^- y(x)] \right] \right]$, i.e., to apply Q^{T} after accumulating $\frac{\partial \Phi}{\partial x_j} c$, where $c = \Phi^- y(x)$ is the current estimate of the linear parameters. When there are multiple right-hand sides ($q > 1$) and y depends on sufficiently few components of x, it pays to parenthesize the expressions for $J_{\cdot,j}$ or $\tilde{K}_{\cdot,j}$ differently. Specifically, it is sometimes faster to apply $\tilde{Q}^{\mathrm{T}}$ to $\frac{\partial G}{\partial x_j}$ before multiplying by the relevant components of c, because this last multiplication then involves shorter vectors. If there are d nonzero partial-derivative columns of G, and if y does not depend on x, then this saves $\rho[2(pq - d)\tilde{n} + dq]$ multiplications, assuming $\tilde{Q}$ is a product of Householder matrices.

For problems in which y depends on at most a few components of x, further savings sometimes accrue if we adjoin transformations to $\tilde{Q}$ that shorten the nonzero portions of the $\tilde{Q}^{\mathrm{T}} \partial G / \partial x_j$ [9]. Let $M = M(x)$ be the $\tilde{n} \times d$ matrix of nonzero partial derivative columns of G, and assume $\tilde{n} > d$. Parenthesizing the computation of J or $\tilde{K}$ as in the last paragraph, we compute $\tilde{Q}^{\mathrm{T}} M$. It is straightforward to modify $\tilde{Q}$ by multiplying it on the right by orthogonal matrices so that (6) is preserved and the last $\tilde{n} - \rho - \mathrm{rank}(M)$ rows of $\tilde{Q}^{\mathrm{T}} M$ are zero. Kaufman and Sylvester [10] describe examples in which this reduces the overall operation count for computing $\tilde{K}$ and its QR factorization by factors of about 4 or 5. (In one of these examples, $d = 20$, but M has rank 10. These examples have structure such that adjoining further transformations to Q saves still more time — see [10] for details.)

The PORT3 nonlinear least-squares iteration drivers allow the Jacobian matrix to be passed in pieces. This is very convenient for problems with multiple right-hand sides, as it substantially reduces the storage that would otherwise be required. When some of the Jacobian pieces themselves have blocks of zeros, as in [10], passing appropriate sub-pieces also reduces the operation count for computing the Jacobian's QR factorization.

For dense nonlinear least-squares problems, using Householder matrices to compute a QR factorization of the Jacobian matrix takes about twice as long as using the normal equations (i.e., computing $J^{\mathrm{T}} J$ or $K^{\mathrm{T}} K$ explicitly, then computing its Cholesky factorization), but gives improved accuracy when the Jacobian is ill-conditioned. When there are multiple right-hand sides, it is sometimes possible to parenthesize the computations for the normal-equations so that computing a Cholesky factor of the Gauss-Newton Hessian approximation is substantially more than twice as fast as computing a QR factorization of the Jacobian matrix. For simplicity, assume that y does not depend on x. If we partition the current estimate $c = \Phi^- y$ of the linear parameters as $(c^{(1)\mathrm{T}}, c^{(2)\mathrm{T}}, \cdots, c^{(q)\mathrm{T}})^{\mathrm{T}}$ to conform with the block structure of Φ, then

$$K^{\mathrm{T}} K = \sum_{1 \le i \le j \le \bar{\ell}} \left[\sum_{k=1}^{q} c_{i,j}^{(k)} \right] W^{i,j},$$

where $W_{g,h}^{i,j} = (2 - \delta_{i,j}) \left[\frac{\partial G_{\cdot,i}}{\partial x_g} \right]^{\mathrm{T}} P_G^\perp \frac{\partial G_{\cdot,j}}{\partial x_h}$ is often zero in the examples where this is useful. This leads to a further several-fold savings in the large examples described in [10].

REFERENCES

[1] *The PORT Mathematical Subroutine Library,* AT&T Bell Laboratories, Murray Hill, NJ, 1984. Third edition.

[2] J. E. Dennis, Jr., D. M. Gay, and R. E. Welsch, ''An Adaptive Nonlinear Least-Squares Algorithm,'' *ACM Trans. Math. Software* **7** (1981), pp. 348–368.

[3] J. E. Dennis, Jr., D. M. Gay, and R. E. Welsch, ''Algorithm 573. NL2SOL—An Adaptive Nonlinear Least-Squares Algorithm,'' *ACM Trans. Math. Software* **7** (1981), pp. 369–383.

[4] G. H. Golub and R. LeVeque, ''Extensions and Uses of the Variable Projection Algorithm for Solving Nonlinear Least Squares Problems,'' Report SU 326 (1978), Dept. of Computer Science, Stanford University.

[5] G. H. Golub and V. Pereyra, ''The Differentiation of Pseudo-Inverses and Nonlinear Least-Squares Problems Whose Variables Separate,'' *SIAM J. Numer. Anal.* **10** (1973), pp. 413–432.

[6] G. H. Golub and V. Pereyra, ''Differentiation of Pseudo-Inverses, Separable Nonlinear Least-Squares Problems and Other Tales,'' pp. 303–324 in *Generalized Inverses and Applications*, ed. M. Z. Nashed, Academic Press (1976).

[7] G. H. Golub and C. F. Van Loan, *Matrix Computations,* Johns Hopkins University Press, Baltimore, MD, 1989. Second edition.

[8] L. Kaufman, ''A Variable Projection Method for Solving Separable Nonlinear Least Squares Problems,'' *BIT* **15** (1975), pp. 49–57.

[9] L. Kaufman, ''Separable Nonlinear Least Squares with Multiple Right Hand Sides,'' Numerical Manuscript 90–03 (1990), AT&T Bell Laboratories, Murray Hill, NJ.

[10] L. Kaufman and G. Sylvester, ''A Block Separable Nonlinear Least Squares Problem,'' Numerical Manuscript 90–04 (1990), AT&T Bell Laboratories, Murray Hill, NJ.

[11] J. J. Moré, ''The Levenberg-Marquardt Algorithm: Implementation and Theory,'' pp. 105–116 in *Numerical Analysis, Dundee 1977*, ed. G. A. Watson, Springer-Verlag, Berlin (1978).