

ON THE CONVERGENCE OF THE GENERALIZED
LINEAR LEAST SQUARES ALGORITHM *C. NEGOITA¹ and R. A. RENAUT² †

¹*Department of Mathematics, Oregon Institute of Technology, Klamath Falls, OR 97601, USA. email: negoita@oit.edu*

²*Department of Mathematics and Statistics, Arizona State University, Tempe, AZ 85287-1804, USA. email: renaud@asu.edu*

Abstract

This paper considers the issue of parameter estimation for biomedical applications using nonuniformly sampled data. The generalized linear least squares (GLLS) algorithm, first introduced by Feng and Ho (1993), is used in the medical imaging community for removal of bias when the data defining the model are correlated. GLLS provides an efficient iterative linear algorithm for the solution of the non linear parameter estimation problem. This paper presents a theoretical discussion of GLLS and introduces use of both Gauss Newton and an alternating Gauss Newton for solution of the parameter estimation problem in nonlinear form. Numerical examples are presented to contrast the algorithms and emphasize aspects of the theoretical discussion.

AMS subject classification: 65F10.

Key words: Iterative Methods, Parameter Estimation, Generalized Linear Least Squares.

1 Introduction.

The generalized linear least squares (GLLS) algorithm was briefly introduced in [5] for the solution of a biomedical inverse problem. A more-detailed presentation of the algorithm followed in [8]. GLLS is an extension of the generalized least squares (GLS) algorithm to non-uniformly sampled data [8]. A careful discussion of the motivation for the introduction of GLLS is also provided in [8].

Although GLLS has been successfully applied and adopted in biomedical studies [5, 8, 3] there is no analytic proof nor theoretical discussion of its convergence. The algorithm is not well known in the numerical analysis literature, and it should not be confused with generalized linear least squares problems [1, 11].

*Received December 2003. Revised November 2004. Communicated by Lars Eldén.

†This work was partially supported by the Arizona Center for Alzheimer's Disease Research, by NIH grant EB 2553301 and for the second author by NSF CMG-02223.

Despite its derivation as an iterative algorithm, GLLS has mostly been used as a one step iteration [8, 5, 6, 9].

1.1 The forward problem

Consider the general system input/system output (SISO) linear continuous dynamic system which is described by the n th order differential equation

$$(1) \quad \frac{d^n y}{dt^n} - \sum_{j=1}^n \alpha_j \frac{d^{n-j} y}{dt^{n-j}} = \sum_{j=1}^n \beta_j \frac{d^{n-j} u}{dt^{n-j}}.$$

Here $y = y(t)$ is the system output, $u = u(t)$ is the system input, coefficients $\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_n$ are the system transfer parameters, and $\frac{d^j}{dt^j}$ indicates the j th order time derivative operator. In the context of the derivation of the GLLS, it is convenient to derive $y(t)$ as a function of $u(t)$ using the Laplace transform approach. Denote the Laplace transform of a function $f(t)$ by

$$(\mathcal{L}f)(s) = F(s) = \int_0^{\infty} f(t)e^{-st} dt.$$

Then the inverse Laplace transform is $\mathcal{L}^{-1}(F(s))(t) = f(t)$, [17].

Let $Y(s)$ and $U(s)$ be Laplace transforms of $y(t)$ and $u(t)$, respectively, and introduce the polynomials

$$(2) \quad \begin{aligned} A(s) &= s^n - \sum_{j=1}^n \alpha_j s^{n-j}, \\ B(s) &= \sum_{j=1}^n \beta_j s^{n-j} \quad \text{and} \quad V(s) = \sum_{j=1}^n v_j s^{n-j}. \end{aligned}$$

Then, taking the Laplace transform of (1) yields in the Laplace domain

$$(3) \quad A(s)Y(s) = B(s)U(s) + V(s),$$

where $v_j, j = 1 \dots n$ are linear combinations of the initial conditions on the input and output and their higher order derivatives. Here, without loss of generality, these conditions are all assumed identically zero so that $V(s) = 0$ and

$$(4) \quad A(s)Y(s) = B(s)U(s) \quad \text{implies}$$

$$(5) \quad Y(s) = \frac{B(s)}{A(s)}U(s).$$

Consider the case in which $\lambda_j, j = 1 \dots n$ are distinct, real, nonzero roots of $A(s)$, and use the notation $A'(s)$ to denote the derivative of A with respect to s . Denote the inverse Laplace transform of a function $F(s)$ by $\mathcal{L}^{-1}[F(s)](t)$. Thus

$$(6) \quad \mathcal{L}^{-1} \left[\frac{s^q}{A(s)} \right] (t) = \sum_{j=1}^n \frac{\lambda_j^q}{A'(\lambda_j)} e^{\lambda_j t},$$

and, with the additional notation

$$(7) \quad h_j(t) = e^{\lambda_j t}, \quad \phi_j(t) = \frac{(h_j * u)(t)}{A'(\lambda_j)},$$

where here $*$ denotes the convolution,

$$(f * g)(t) = \int_0^t f(w)g(t-w)dw,$$

one also has

$$\begin{aligned} \mathcal{L}^{-1} \left[\frac{s^q}{A(s)} U(s) \right] (t) &= \mathcal{L}^{-1} \left[\frac{s^q}{A(s)} \right] (t) * u(t) \\ &= \sum_{j=1}^n \frac{\lambda_j^q}{A'(\lambda_j)} (e^{\lambda_j t} * u(t)). \end{aligned}$$

Hence, applying the inverse Laplace transform to (5) yields $y(t)$ analytically as a nonlinear function of $u(t)$

$$(8) \quad y(t) = \sum_{l=1}^n \beta_l \sum_{j=1}^n \lambda_j^{n-l} \phi_j(t),$$

dependent on the system transfer parameters.

1.2 Nonlinear inverse problem

The problem of interest in this paper is the inverse problem in which both $u(t)$ and $y(t)$ are sampled at some time points t_i , $i = 1 \dots m$, and from which estimates of the system parameters should be obtained. As given in (8) this defines the system parameters as the solution of the nonlinear minimization

$$(9) \quad \theta_{\text{nls}} = \operatorname{argmin}_{\theta \in \mathcal{R}^{2n}} \|\mathbf{s}_{\text{nls}}\|^2,$$

where \mathbf{s}_{nls} is the vector with entries

$$(10) \quad s_{\text{nls}}(t_i) = y(t_i) - \sum_{l=1}^n \beta_l \sum_{j=1}^n \lambda_j^{n-l} \phi_j(t_i), \quad i = 1 \dots m,$$

and

$$(11) \quad \theta = [\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_n]^T \in \mathcal{R}^{2n}.$$

Notice that while the dependency on β_l is linear, the dependency on α_l is nonlinear through λ_j and $\phi_j(t)$, and thus repeated solution of this nonlinear minimization, as required in certain biomedical applications [13], may be computationally prohibitive.

1.3 A linear inverse problem

Dividing (4) through by s^n and rearranging yields

$$(12) \quad Y(s) = \left[\sum_{j=1}^n \alpha_j s^{-j} \right] Y(s) + \left[\sum_{j=1}^n \beta_j s^{-j} \right] U(s).$$

Now, again assuming that all initial conditions are zero, inverse Laplace transform gives

$$(13) \quad y(t) = \sum_{j=1}^n \alpha_j I_j(y) + \beta_j I_j(u),$$

where for any integrable function $g(t)$ the notation used is

$$I_1(g) = \int_0^t g(\tau) d\tau, \quad I_j(g) = \int_0^t \int_0^{\tau_1} \dots \int_0^{\tau_{j-1}} g(\tau) d\tau d\tau_{j-1} \dots d\tau_1, \quad j \geq 2.$$

Sampling (13) in time at m points, $m > 2n$, provides the overdetermined linear system

$$(14) \quad X\theta = \mathbf{y}$$

where $\mathbf{y} = [y_1, y_2, \dots, y_m]^T \in \mathcal{R}^m$, with $y_i = y(t_i)$, and coefficient matrix $X \in \mathcal{R}^{m \times 2n}$ has entries which are integrals of the input and output functions

$$(15) \quad X_{ij} = \begin{cases} I_j(y(t_i)), & 1 \leq j \leq n, \\ I_{j-n}(u(t_i)), & n+1 \leq j \leq 2n. \end{cases}$$

Assuming that X is of full column rank,

$$(16) \quad \theta_{\text{ls}} = (X^T X)^{-1} X^T \mathbf{y}$$

is the unique least squares (LS) solution for

$$(17) \quad \min_{\theta \in \mathcal{R}^{2n}} \|X\theta - \mathbf{y}\|_2^2 = \min_{\theta \in \mathcal{R}^{2n}} \sum_{i=1}^m (X\theta - \mathbf{y})_i^2.$$

Consider the general system (14) in which \mathbf{y} is contaminated by white noise ϵ

$$(18) \quad \mathbf{y} = X\theta + \epsilon.$$

Then, by the Gauss–Markoff theorem [1], θ_{ls} is the best unbiased estimator. In the specific form for (14) considered here, however, assuming that $u(t)$ is known exactly but accounting for measurement noise in \mathbf{y} , the first n columns of X which are integrals of $y(t)$ are also contaminated by noise. The vector ϵ contains contributions from both sources of error

$$(19) \quad \epsilon = \epsilon_{\mathbf{y}} + \epsilon_X$$

where $\epsilon_{\mathbf{y}}$ is white noise in \mathbf{y} and $\epsilon_X = \varepsilon_X \alpha$ is a vector of correlated noise. Here matrix $\varepsilon_X \in \mathcal{R}^{m \times n}$ has entries $(\varepsilon_X)_{ij} = I_j(\epsilon_{\mathbf{y}}(t_i))$, $1 \leq j \leq n, 1 \leq i \leq m$. Thus, for this application, the correlation matrix is not the identity and θ_{ls} is not unbiased. The GLLS algorithm is designed to remove the bias for systems of this form derived from (1).

REMARK 1.1. *Another approach that can be used when matrix X is in error is the total least squares (TLS) solution [1, 11]. For TLS, however, it is assumed that all rows of the error in X are independently and identically distributed with zero mean and equal variance, which does not apply for this application.*

1.4 Derivation of GLLS

Including the measurement noise in $y(t)$, as in (18), yields the equation in the Laplace domain

$$(20) \quad A(s)Y(s) = B(s)U(s) + A(s)\Upsilon(s),$$

where $\Upsilon(s)$ is the Laplace transform of the white noise. The fundamental idea of the GLLS is to suppose that $A(s)$ can be improved iteratively such that in the limit the ratio $A^{(k)}$ to $A^{(k-1)}$ tends to the identity, where use of superscript k for any variable indicates the estimate of that variable at the k th iteration. Then, (20) becomes

$$(21) \quad \frac{A^{(k)}(s)}{A^{(k-1)}(s)}Y(s) \approx \frac{B^{(k)}(s)}{A^{(k-1)}(s)}U(s) + \Upsilon(s)$$

and the noise is whitened. Specifically, GLLS is an iterative approach to update $A(s)$ using (21) to define the update equation given the initial estimate of $A^{(0)}(s)$ from θ_{ls} .

The ratio $A^{(k)}(s)Y(s)/A^{(k-1)}(s)$ may be rewritten as

$$\begin{aligned} \frac{A^{(k)}(s)}{A^{(k-1)}(s)}Y(s) &= \frac{A^{(k-1)}(s) - [A^{(k-1)}(s) - A^{(k)}(s)]}{A^{(k-1)}(s)}Y(s) \\ &= Y(s) - \frac{s^n - \sum_{j=1}^n \alpha_j^{(k-1)} s^{n-j} - s^n + \sum_{j=1}^n \alpha_j^{(k)} s^{n-j}}{A^{(k-1)}(s)}Y(s) \\ (22) \quad &= Y(s) + \sum_{j=1}^n \alpha_j^{(k-1)} \frac{s^{n-j}}{A^{(k-1)}(s)}Y(s) - \sum_{j=1}^n \alpha_j^{(k)} \frac{s^{n-j}}{A^{(k-1)}(s)}Y(s). \end{aligned}$$

Hence, substituting in (21) and ignoring the noise term, yields

$$\begin{aligned} Y(s) &+ \sum_{j=1}^n \alpha_j^{(k-1)} \frac{s^{n-j}}{A^{(k-1)}(s)}Y(s) \\ (23) \quad &= \sum_{j=1}^n \alpha_j^{(k)} \frac{s^{n-j}}{A^{(k-1)}(s)}Y(s) + \sum_{j=1}^n \beta_j^{(k)} \frac{s^{n-j}}{A^{(k-1)}(s)}U(s). \end{aligned}$$

Then, using the same notation as for derivation of (8), with the λ_j the roots of $A^{(k-1)}$, and introducing additionally

$$(24) \quad \psi_j(t) = \frac{(h_j * y)(t)}{(A^{(k-1)})'(\lambda_j)},$$

inverse Laplace transform of (23) yields

$$(25) \quad \begin{aligned} y(t) &+ \sum_{l=1}^n \alpha_l^{(k-1)} \sum_{j=1}^n \lambda_j^{n-l} \psi_j(t) \\ &= \sum_{l=1}^n \left[\alpha_l^{(k)} \sum_{j=1}^n \lambda_j^{n-l} \psi_j(t) + \beta_l^{(k)} \sum_{j=1}^n \lambda_j^{n-l} \phi_j(t) \right]. \end{aligned}$$

Sampling in time provides the update equation

$$(26) \quad Z^{(k-1)}\theta^{(k)} = \mathbf{r}^{(k-1)},$$

for θ where $Z^{(k-1)}$ and $\mathbf{r}^{(k-1)}$ have row components, subscript i , in which all relevant variables are evaluated at iteration level $k-1$

$$(27) \quad \mathbf{z}_i = \left[\sum_{j=1}^n \lambda_j^{n-1} \psi_j(t_i), \dots, \sum_{j=1}^n \psi_j(t_i), \sum_{j=1}^n \lambda_j^{n-1} \phi_j(t_i), \dots, \sum_{j=1}^n \phi_j(t_i) \right]$$

$$(28) \quad r_i = \left[y_i + \sum_{l=1}^n \alpha_l^{(k-1)} \sum_{j=1}^n \lambda_j^{n-l} \psi_j(t_i) \right], \quad i = 1, \dots, m.$$

ALGORITHM 1. **GLLS**

Given $\theta^{(0)} = \theta_{\text{ls}}$, form $A(s) = s^n - \sum_{j=1}^n \alpha_j^{(0)} s^{n-j}$. Iterate to convergence.

1. **For** $k = 1, 2, \dots$ **Do**

- (a) Find $\lambda_j = \lambda_j^{(k-1)}$, the roots of $A(s)$, $j = 1, \dots, n$.
- (b) **If** there is any repeated root **Break**.
- (c) Calculate $A'(\lambda_j)$.
- (d) **For** $i = 1, \dots, m$ calculate $\mathbf{z}_i^{(k-1)}$ and $r_i^{(k-1)}$ using (27,28) **End**
- (e) Solve $Z^{(k-1)}\theta^{(k)} = \mathbf{r}^{(k-1)}$ for $\theta^{(k)}$.
- (f) **If** termination criterion satisfied **Break**.
- (g) Let $A(s) = s^n - \sum_{j=1}^n \alpha_j^{(k)} s^{n-j}$.

2. **End**

In the original algorithm [8] the termination criterion used was $\|\theta^{(k)} - \theta^{(k-1)}\|_2 < \text{tol}$ for some given tolerance tol.

REMARK 1.2. Note, if the initial conditions on input and output and their higher derivatives are unknown the algorithm is modified so that these conditions are parameters to be determined [8]. Additionally, if roots of the polynomial $A^{(k-1)}(s)$ at any step are no longer distinct the algorithm is ammended appropriately [16].

1.5 Outline.

In Section 2 of the paper consistency conditions for the GLLS algorithm are obtained. In Section 3 these are extended to consider the impact for the non-linear formulation of the problem. Numerical results, including a discussion of convergence conditions, are presented in Section 4. Conclusions follow in Section 5.

2 Theoretical Analysis.

Let

- $\Psi \in \mathcal{R}^{m \times n}$ be the matrix defined by (24) with entries $\Psi_{ij} = \psi_j(t_i)$, dependent on the output function y .
- $\Phi \in \mathcal{R}^{m \times n}$ be the matrix defined by (7) with entries $\Phi_{ij} = \phi_j(t_i)$, dependent on the input function u .
- $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_n]^T \in \mathcal{R}^n$ be the vector of nonzero, distinct, real roots of the polynomial $A(s)$, and V the Vandermonde matrix of powers of λ , $V = [\lambda^{n-1}, \lambda^{n-2}, \dots, \lambda^0] \in \mathcal{R}^{n \times n}$, where λ^q denotes the vector where each component is raised to power q .

With this notation, (28) and (27) are resp.

$$(29) \quad \mathbf{r}^{(k-1)} = \mathbf{y} + [\Psi\lambda^{n-1}, \Psi\lambda^{n-2}, \dots, \Psi\lambda^0]\alpha$$

$$(30) \quad Z^{(k-1)} = [\Psi, \Phi]V,$$

where all appropriate elements are evaluated at iteration level $k-1$. Hence, (26) is rewritten as

$$[\Psi^{(k-1)}, \Phi^{(k-1)}]V^{(k-1)} \begin{bmatrix} \alpha^{(k)} \\ \beta^{(k)} \end{bmatrix} = \mathbf{y} + [\Psi^{(k-1)}, 0]V^{(k-1)} \begin{bmatrix} \alpha^{(k-1)} \\ \beta^{(k-1)} \end{bmatrix},$$

where specifically $\Psi^{(k)} = \Psi(\alpha^{(k)})$, $\Phi^{(k)} = \Phi(\alpha^{(k)})$ and $V^{(k)} = V(\alpha^{(k)})$, and note that the right hand side of this equation is only dependent on the first n entries of the solution vector θ at step $(k-1)$.

Introducing the notation

$$(31) \quad C = \Psi V, \quad D = \Phi V,$$

yields the normal equations

$$(32) \quad \begin{bmatrix} C^T C & C^T D \\ D^T C & D^T D \end{bmatrix} \begin{bmatrix} \alpha^{(k)} \\ \beta^{(k)} \end{bmatrix} = \begin{bmatrix} C^T \\ D^T \end{bmatrix} (\mathbf{y} + C\alpha^{(k-1)}).$$

Equivalently

$$(33) \quad \begin{aligned} C^T C \alpha^{(k)} + C^T D \beta^{(k)} &= C^T \mathbf{y} + C^T C \alpha^{(k-1)}, \\ D^T C \alpha^{(k)} + D^T D \beta^{(k)} &= D^T \mathbf{y} + D^T C \alpha^{(k-1)}, \end{aligned}$$

where $C^T C = V^T \Psi^T \Psi V$ and $D^T D = V^T \Phi^T \Phi V$. Because the matrix V is Vandermonde the invertibility of these matrix products depends on the invertibility of $\Psi^T \Psi$ and $\Phi^T \Phi$, resp.

LEMMA 2.1. *Assume*

(i) $A(s)$ and h_j are defined by (2) and (7), and A has distinct, real, nonzero roots λ_j .

(ii) $u(t) > 0$ for any $t > 0$, $u(0) = 0$.

(iii) $y(t)$ satisfies (1).

Then matrices Φ and Ψ are of full column rank and $\Phi^T \Phi$, $\Psi^T \Psi$ are invertible.

PROOF. Consider first matrix Φ . If matrix Φ is of full column rank then its columns are linearly independent and $\Phi^T \Phi$ is invertible. It thus suffices to show that $\sum_{j=1}^n \gamma_j \phi_j = 0$ if and only if $\gamma_j = 0$ for all $j = 1, \dots, n$, where ϕ_j denotes the j^{th} column of Φ . Equivalently, $\sum_{j=1}^n \gamma_j (h_j * u)(t) = 0$ if and only if $\gamma_j = 0$ for each t . By linearity, $\sum_{j=1}^n (\gamma_j h_j * u)(t) = ((\sum_{j=1}^n \gamma_j h_j) * u)(t) = 0$. Hence $H(s)U(s) = 0$ in the Laplace domain, where $H(s)$ is the Laplace transform of the function $\sum_{j=1}^n \gamma_j h_j(t)$, and $U(s)$ is nonzero by assumption (ii). Note, each h_j is an independent exponential function and thus $H(s) = 0$ if and only if $\gamma_j = 0$ for each j . The result thus follows. The proof for Ψ follows similarly. \square

COROLLARY 2.2. *Assuming the conditions of Lemma 2.1, the iteration (33) can be written as*

$$(34) \quad \alpha^{(k)} = \alpha^{(k-1)} - C^\dagger (D\beta^{(k)} - \mathbf{y}),$$

$$(35) \quad \beta^{(k)} = D^\dagger \mathbf{y} - D^\dagger C (\alpha^{(k)} - \alpha^{(k-1)}),$$

where C^\dagger , D^\dagger are the pseudo-inverse matrices of C , respectively D , ie $C^\dagger = (C^T C)^{-1} C^T$.

Let $R(\Phi)$ denote the range of Φ , $N(\Phi^T)$ the null space of Φ^T , and $P_\Phi = \Phi(\Phi^T \Phi)^{-1} \Phi^T$ the orthogonal projector onto the range of Φ .

THEOREM 2.3. *Suppose the GLLS iteration converges to $\alpha^* = \lim_{k \rightarrow \infty} \alpha^{(k)}$, $\beta^* = \lim_{k \rightarrow \infty} \beta^{(k)}$. Then, by consistency*

$$1. \quad (36) \quad \beta^* = V^{-1} \Phi^\dagger \mathbf{y},$$

and

$$2. \quad (37) \quad (P_\Phi - I_m) \mathbf{y} \in N(\Phi^T) \cap N(\Psi^T),$$

where Φ , V depend on the converged α^* .

PROOF. At convergence

$$(38) \quad C^T D \beta = C^T \mathbf{y}, \text{ and}$$

$$(39) \quad D^T D \beta = D^T \mathbf{y},$$

where each of C and D depend on the converged α^* . By Lemma 2.1,

$$(40) \quad \beta^* = V^{-1}(\Phi^T \Phi)^{-1} \Phi^T \mathbf{y},$$

and the residual $D\beta^* - \mathbf{y} = (P_\Phi - I_m)\mathbf{y} \in N(\Phi^T)$. Now, using the expression for β^* in (38) yields $\Psi^T(I_m - P_\Phi)\mathbf{y} = 0$, as required. \square

REMARK 2.1. *If \mathbf{y} is sampled exactly and D is calculated analytically, then by (8) $\mathbf{y} = D\beta$, so that $\mathbf{y} \in R(\Phi)$ and $(P_\Phi - I_m)\mathbf{y} = 0$. Practically, \mathbf{y} is only known approximately, and $(P_\Phi - I_m)\mathbf{y}$ is never zero.*

2.1 Nonlinear Formulation

Consider the nonlinear formulation obtained from (8),

$$(41) \quad \theta_{\text{nls}} = \operatorname{argmin}_\theta S(\theta)$$

$$(42) \quad = \operatorname{argmin}_{[\alpha, \beta]} \frac{1}{2} \|D\beta - \mathbf{y}\|_2^2$$

$$(43) \quad = \operatorname{argmin}_\theta \frac{1}{2} \mathbf{s}(\theta)^T \mathbf{s}(\theta)$$

where $D(\alpha) = \Phi(\alpha)V(\alpha)$ depends only on α , and $\mathbf{s}(\theta)$ is appropriately defined by (10). Now

$$(44) \quad \begin{aligned} \frac{\partial}{\partial \alpha_j} \Phi &= [H - \Phi \operatorname{diag}(f_1, \dots, f_n)] \operatorname{diag} \left(\frac{\partial \lambda_1}{\partial \alpha_j}, \dots, \frac{\partial \lambda_n}{\partial \alpha_j} \right) \\ &= [H - \Phi F] \Lambda'_j \end{aligned}$$

where the matrix H has entries

$$H_{ij} = \frac{([t h_j(t)] * u(t))|_{t_i}}{A'(\lambda_j)},$$

now the prime in Λ'_j denotes differentiation with respect to α_j , and

$$(45) \quad f_j = \frac{A''(\lambda_j)}{A'(\lambda_j)} = \frac{n(n-1)\lambda_j^{n-2} + \sum_{l=1}^{n-2} (n-l)(n-l-1)\alpha_l \lambda_j^{n-l-2}}{n\lambda_j^{n-1} + \sum_{l=1}^{n-1} (n-l)\alpha_l \lambda_j^{n-l-1}}.$$

THEOREM 2.4. *Suppose that the GLLS algorithm converges, then at convergence*

1.

$$\frac{\partial S}{\partial \beta_j} = 0, \quad \text{and}$$

2.

$$\frac{\partial S}{\partial \alpha_j} = (\beta^*)^T V^T \Lambda'_j H^T (P_\Phi - I_m) \mathbf{y}.$$

Hence $\theta^* = (\alpha^*, \beta^*)$ is a critical point for S only if in addition to (36,37) $\mathbf{y}^T \Phi (\Phi^T \Phi)^{-1} \lambda' H^T (P_\Phi - I_m) \mathbf{y} = 0$.

PROOF. By (41)

$$(46) \quad \frac{\partial S}{\partial \alpha_j} = \beta^T \left(\frac{\partial}{\partial \alpha_j} D^T \right) \mathbf{s}(\theta), \text{ and}$$

$$(47) \quad \frac{\partial S}{\partial \beta_j} = \mathbf{d}_j^T \mathbf{s}(\theta),$$

where \mathbf{d}_j is the j^{th} column of D . The first statement is thus immediate. Suppose now that β^* solves (47) for the converged α^* . Then, using the prime to denote differentiation with respect to α , and substituting for β^* in (46), yields

$$\begin{aligned} \frac{\partial S}{\partial \alpha_j} &= \beta^{*T} \left(\frac{\partial}{\partial \alpha_j} D^T \right) \mathbf{s}(\theta^*) \\ &= \beta^{*T} [(V')^T \Phi^T + V^T (\Phi')^T] [\Phi (\Phi^T \Phi)^{-1} \Phi^T - I_m] \mathbf{y} \\ &= \beta^{*T} [(V')^T (I_n \Phi^T - \Phi^T I_m) \mathbf{y} + V^T (\Phi')^T (\Phi (\Phi^T \Phi)^{-1} \Phi^T - I_m) \mathbf{y}], \end{aligned}$$

and immediately the first term cancels. The second term can be rewritten as

$$\begin{aligned} \frac{\partial S}{\partial \alpha_j} &= \mathbf{y}^T \Phi (\Phi^T \Phi)^{-1} (\Phi')^T (P_\Phi - I_m) \mathbf{y}, \\ &= \mathbf{y}^T \Phi (\Phi^T \Phi)^{-1} \Lambda'_j [H^T - F \Phi^T] (P_\Phi - I_m) \mathbf{y}, \end{aligned}$$

noting that F and Λ'_j are diagonal. But now, by (37), the second term of this expression also cancels and the remaining result follows. \square

REMARK 2.2. *Theorem 2.4 demonstrates that a converged solution of GLLS yields a local minimum with respect to β but not necessarily with respect to α . Thus GLLS does not always solve the nonlinear problem.*

3 Nonlinear GLLS Iteration.

Consider now a nonlinear iteration for solving (43) based on the Gauss Newton method [1]

$$(48) \quad \theta^{(k)} = \theta^{(k-1)} + \mathbf{p}^{(k-1)}$$

where $\mathbf{p}^{(k-1)}$ solves the linear system

$$(49) \quad \left(\mathbf{J}^T(\theta^{(k-1)}) \mathbf{J}(\theta^{(k-1)}) \right) \mathbf{p}^{(k-1)} = -\mathbf{J}^T(\theta^{(k-1)}) \mathbf{s}(\theta^{(k-1)}),$$

and $\mathbf{J}(\theta) = \nabla_\theta \mathbf{s}(\theta)$ is the Jacobian matrix of $\mathbf{s}(\theta)$. This system can be recognized as the normal equations for the linear system $\mathbf{J}(\theta^{(k-1)}) \mathbf{p}^{(k-1)} \approx -\mathbf{s}(\theta^{(k-1)})$, so that in effect the Gauss Newton method replaces a nonlinear least squares problem by a sequence of linear least squares problems, which arise from the Newton method where second order terms are dropped.

The Jacobian matrix is obtained from (44) and

$$V' = \frac{\partial}{\partial \alpha_j} V = \Lambda_j' V E,$$

where E is the matrix with column entries $E_j = (n - j)\mathbf{e}_{j+1}$, and \mathbf{e}_j is the unit vector of length n . Let $G \in \mathcal{R}^{m \times n}$ be the matrix whose columns have the form

$$(50) \quad \mathbf{g}_j = [(H - \Phi F)\Lambda_j' V + \Phi \Lambda_j' V E] \beta.$$

Then $\nabla_{\theta} S = [G, D]^T \mathbf{s}$, and the Jacobian matrix is given by $\mathbf{J} = [G, D]$ which is independent of the matrix Ψ .

Rewriting (49) yields

$$(51) \quad G^T G(\alpha^{(k)} - \alpha^{(k-1)}) + G^T D(\beta^{(k)} - \beta^{(k-1)}) = -G^T (D\beta^{(k-1)} - \mathbf{y}),$$

$$(52) \quad D^T G(\alpha^{(k)} - \alpha^{(k-1)}) + D^T D(\beta^{(k)} - \beta^{(k-1)}) = -D^T (D\beta^{(k-1)} - \mathbf{y}),$$

and, assuming G and D are full rank,

$$(53) \quad \alpha^{(k)} = \alpha^{(k-1)} - G^\dagger (D\beta^{(k)} - \mathbf{y}),$$

$$(54) \quad \beta^{(k)} = D^\dagger \mathbf{y} - D^\dagger G(\alpha^{(k)} - \alpha^{(k-1)}).$$

Equivalently,

$$\begin{aligned} G^T (I_m - P_D) G(\alpha^{(k)} - \alpha^{(k-1)}) &= G^T (I_m - P_D) \mathbf{y}, \\ D^T (I_m - P_G) D \beta^{(k)} &= D^T (I_m - P_G) \mathbf{y}, \end{aligned}$$

and it is clear that at convergence β^* is again given by (36) while α^* is such that $(I_m - P_D)\mathbf{y} \in N(G^T)$. Here D is nonlinear in α and G is nonlinear in both α and β .

REMARK 3.1. Does C approximate G ?

Comparing the nonlinear iteration (53)-(54) with the linear iteration (34)-(35) it is clear that the GLLS iteration is the nonlinear least squares Gauss Newton iteration where the matrix G is approximated by C . The local convergence of the undamped Gauss Newton iteration is well understood [1, 12]. Thus, if C is a good approximation to G at each iteration, with the same spectral properties, GLLS will converge under the same convergence conditions as Gauss Newton. In addition, it is well known that if G is rank deficient, the Gauss Newton iteration may not proceed. In this case either second derivatives need to be considered, or regularization methods need to be employed. The GLLS algorithm may still converge as long as C is a full rank approximation to G . If, however, C is far from G , even if C is full rank and the iteration is convergent, the GLLS solution will not be a true minimum for (43).

Note also that by the requirement for convergence with respect to β , then (47) is solved for any α by β^* as given in (36). This suggests an alternative development of Gauss Newton (GN) to find the solution of (37) for fixed β , from

which β is then updated to satisfy (47). Specifically, decoupling the steps yields the alternating iterative scheme

$$(55) \quad \beta^{(k)} = D^\dagger \mathbf{y}$$

$$(56) \quad \alpha^{(k)} = \alpha^{(k-1)} - G^\dagger (D\beta^{(k)} - \mathbf{y}),$$

in which at both steps matrix D depends on $\alpha^{(k-1)}$ but at the second step (56), G also depends on $\alpha^{(k-1)}$ and the updated value $\beta^{(k)}$. This iteration is similar to the variable projection algorithms used for separable problems [1] p. 351.

ALGORITHM 2. GN

Initialize $\theta^{(0)} = \theta_{\text{ls}}$ and $A(s) = s^n - \sum_{j=1}^n \alpha_j^{(0)} s^{n-j}$. Iterate to convergence.

1. **For** $k = 1, 2, \dots$ **Do**
 - (a) Find $\lambda_j^{(k-1)}$ the roots of $A(s)$, $j = 1, \dots, n$
 - (b) **If** there is any repeated root **Break**
 - (c) Calculate the matrices D , G as functions of $\lambda_j^{(k-1)}$.
 - (d) Form residual $\mathbf{s}^{(k-1)} = D\beta^{(k-1)} - \mathbf{y}$.
 - (e) Solve (49) and form update (48).
 - (f) **If** termination criterion satisfied **Break**
 - (g) Let $A(s) = s^n - \sum_{j=1}^n \alpha_j^{(k)} s^{n-j}$.

2. **End**

ALGORITHM 3. Alternating Gauss Newton (AGN)

Initialize $\theta^{(0)} = \theta_{\text{ls}}$ and $A(s) = s^n - \sum_{j=1}^n \alpha_j^{(0)} s^{n-j}$. Iterate to convergence.

1. **For** $k = 1, 2, \dots$ **Do**
 - (a) Find $\lambda_j^{(k-1)}$ the roots of $A(s)$, $j = 1, \dots, n$
 - (b) **If** there is any repeated root **Break**
 - (c) Calculate the matrix D as a function of $\lambda_j^{(k-1)}$.
 - (d) Form update $\beta^{(k)} = (D^T D)^{-1} D^T \mathbf{y}$.
 - (e) Calculate the matrix G as function of $\beta^{(k)}$ and $\alpha^{(k-1)}$.
 - (f) Update $\alpha^{(k)} = \alpha^{(k-1)} - (G^T G)^{-1} G^T (D\beta^{(k)} - \mathbf{y})$
 - (g) **If** termination criterion satisfied **Break**
 - (h) Let $A(s) = s^n - \sum_{j=1}^n \alpha_j^{(k)} s^{n-j}$.

2. **End**

For the updates in the above algorithms a QR decomposition, facilitated by *Matlab's* **qr** command, is used. Note, matrices $D^T D$ and $G^T G$ are not explicitly formed.

4 Numerical investigations.

4.1 Experiments

Numerical experiments, motivated by clinical Positron Emission Tomography (PET) studies [2], are reported for (1) with $n = 2$. Data are non-uniformly sampled according to the clinical protocol $t = [0, .1, .2167, .25, .2833, .3167, .35, .3833, .4167, .45, .55, .7167, .9, 1.25, 2, 3, 4.25, 5.75, 8.25, 12.5, 17.5, 25, 45]$ minutes. The input $u(t)$ is given analytically by a form suggested in [16] $u(t) = 28.0975(e^{-.857642t} - e^{-1.21986t})$. For specific choices of the transfer parameters, listed in Table 1, in (1), $y(t)$ is explicitly available as $y(t) = (f*u)(t)$, where

$$(57) \quad f(t) = \frac{\beta_1}{\lambda_+ - \lambda_-} \left[\left(\frac{\beta_2}{\beta_1} + \lambda_+ \right) e^{\lambda_+ t} - \left(\frac{\beta_2}{\beta_1} + \lambda_- \right) e^{\lambda_- t} \right],$$

and λ_+, λ_- are the roots of $A(s) = 0$. The performance of the algorithms is illustrated for three cases which are chosen to contrast their characteristics for noisy data.

Table 1: System parameters used for each of the numerical tests.

Test	α_1	α_2	β_1	β_2
1	-.1988	-.0009	.1020	.0070
2	-1.0053	-.1225	.1389	.1115
3	-1.2614	-.0037	.1630	.2049

Case 1 For clinical studies the integrals are estimated numerically. Here the composite trapezoidal rule is used for both $I_1(y)$ and $I_2(y)$ with cubic spline interpolation used to provide the needed intermediate values. For Z , the convolution $h_j * u$ is calculated analytically, while $h_j * y$ again requires cubic spline interpolation for $y(t)$. The resulting expression is integrated analytically. Hence this case investigates effects of the noise introduced through truncation errors.

Case 2 Following other studies, [12], and motivated by typical noise seen in clinical data [13], $y(t)$ is replaced by

$$(58) \quad \tilde{\mathbf{y}} = \mathbf{y} + \frac{\cos(60\mathbf{y})}{200\sqrt{\Delta t}}, \quad \tilde{\mathbf{y}}_1 = y(0) = 0,$$

where Δt is the vector of time differences, $\{t_{i+1} - t_i\}_{i=1}^{m-1}$.

Case 3 In contrast to case 2 noise is added in which the peak of the output data is overestimated,

$$(59) \quad \tilde{\mathbf{y}} = \mathbf{y} + \mathbf{y} \frac{\cos^2(3\mathbf{y})}{8\sqrt{\Delta t}}, \quad \tilde{\mathbf{y}}_1 = y(0) = 0.$$

The profiles of the output data on a log scale for cases 2 and 3, and each parameter choice are illustrated in Figure 1.

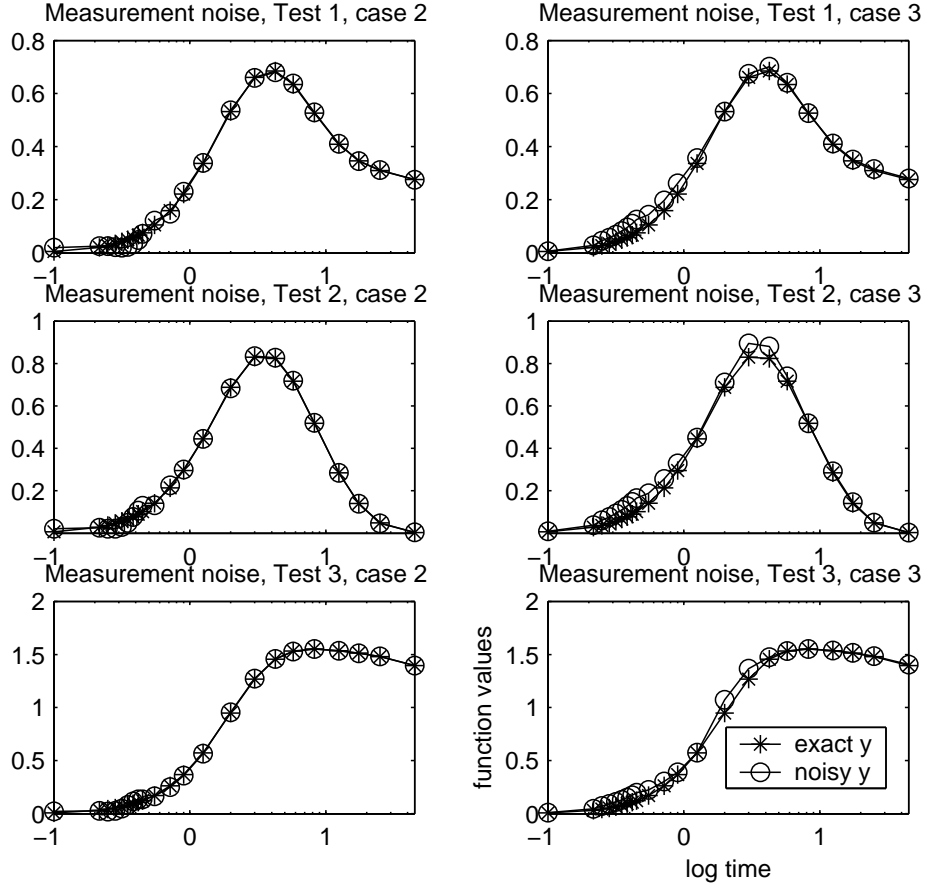


Figure 1: Time-log plots for Tests 1, 2 and 3, cases 2 and 3.

4.2 Convergence criteria

Convergence of each algorithm is determined based on the scale independent variable

$$(60) \quad c^{(k)} := \frac{\|\mathbf{J}^T(\theta^{(k)})\mathbf{s}(\theta^{(k)})\|_2}{\|\mathbf{J}^T(\theta^{(k)})\|_2 \max(\|\mathbf{s}(\theta^{(k)})\|_2, \|\mathbf{s}(\theta_{\text{typical}})\|_2)},$$

as suggested in [4], where a typical value of θ is given by θ_{ls} . The iterative scheme is terminated either when the maximum number of iterations, set to 10,000 and denoted by Max, is reached or when

$$(61) \quad c^{(k)} < \text{tol} = 10^{-p}, \quad \text{where}$$

$$\text{for GLLS, } c_{\text{GLLS}}^{(k)} \approx \frac{\|\mathbf{Z}^T(\theta^{(k)})\mathbf{s}(\theta^{(k)})\|_2}{\|\mathbf{Z}^T(\theta^{(k)})\|_2 \max(\|\mathbf{s}(\theta^{(k)})\|_2, \|\mathbf{s}(\theta_{\text{ls}})\|_2)}.$$

AGN uses (60) with $\mathbf{J} = [G, D]$ and G and D as given in (55) and (56).

4.3 Results

The solution θ_{ls} is used to initialize the iteration in each case. At each iteration the relative errors of $\theta^{(k)}$ as approximation to the exact θ , and of $\mathbf{y}^{(k)}$ as an estimate of \mathbf{y} are recorded. In each case the number of iterations to convergence, K , as measured by (60), and the values of θ at convergence are tabulated for all three algorithms. The convergence rate is estimated by the appropriate approximation for each algorithm of the ratio $\|\mathbf{J}(\theta^{(k)})\mathbf{p}^{(k)}\|_2/\|\mathbf{J}(\theta^{(k-1)})\mathbf{p}^{(k-1)}\|_2$, see [1]. In the presented results the notation $x.y(z)$ is used to indicate $x.y \times 10^z$ and the tolerance is indicated by p . Moreover, the process is terminated if convergence to required tolerance has not been reached, or if a rank deficient matrix is encountered, denoted RD.

REMARK 4.1. *Comparing C and G*

Additional results, in particular relating to Remark 3.1, are presented in [14].

4.3.1 Test 1

Case 1 All algorithms perform well, converging to the exact values for θ as shown in Table 2. Both GLLS and GN converge to the given values already at the lower tolerances with $c_{\text{GLLS}}^{(K)} = 2.9(-2)$ and $c_{\text{GN}}^{(K)} = 1.4(-2)$, $3(-2)$, for tolerances -1 and -2 , resp. Moreover, the convergence of GLLS and GN is far superior to that of AGN. Behavior of GLLS supports the comments in Remark 2.1.

Cases 2 and 3 Results are tabulated in Tables 3 and 4, resp. The performance of all algorithms is similar in terms of the accuracy of the estimates, with case 2 more accurate than case 3, while computationally AGN is the most expensive.

GLLS estimates improve only slightly with iteration. Figure 2 demonstrates the behavior of GLLS as compared to LS, and after just one iteration of GLLS, for case 3. The difference between \mathbf{y}_{ls} and \mathbf{y}_{GLLS} is large, however, iteration with GLLS has little impact. Convergence rate for AGN is linear, as is illustrated for case 3 in Figure 3.

4.3.2 Test 2

Case 1 Results are tabulated in Table 5. GLLS and GN converge rapidly but AGN does not converge to 10^{-3} .

Cases 2 and 3 Results are tabulated in Tables 6, 7. Convergence properties are poor, reflecting the fact that the initial guesses are apparently too far from the exact solution. GLLS and AGN converge to inaccurate solutions and the one step GLLS estimate is bad. In each case, GN encounters a rank deficient matrix, which does not correspond to a case in which there are double roots of $A(s)$, and diverges.

Table 2: Test 1, case 1.

	p	α_1	α_2	β_1	β_2	K	$c^{(K)}$	$\ \theta^{(K)} - \theta\ _2$
EX		-.1988	-.0009	.1020	.0070	n/a	n/a	n/a
LS		-.1915	-.0005	.1018	.0064	0	n/a	7.3(-3)
GLLS		-.1985	-.0009	.1020	.0070	1	6.4(-1)	3.0(-4)
	-3	-.1988	-.0009	.1020	.0070	4	6.3(-5)	0
GN	-3	-.1988	-.0009	.1020	.0070	4	8.4(-6)	0
AGN	-1	-.1906	-.0006	.1017	.0064	3	9.1(-2)	8.2(-3)
	-2	-.1975	-.0008	.1019	.0069	148	1.0(-2)	1.3(-3)
	-3	-.1987	-.0009	.1020	.0070	318	9.9(-4)	1.0(-4)

Table 3: Test 1, case 2.

	p	α_1	α_2	β_1	β_2	K	$c^{(K)}$	$\ \theta^{(K)} - \theta\ _2$
EX		-.1988	-.0009	.1020	.0070	n/a	n/a	n/a
LS		-.1807	-.0020	.1011	.0058	0	n/a	1.8(-2)
GLLS		-.1920	-.0008	.1012	.0067	1	2.1(-1)	6.9(-3)
	-3	-.1927	-.0008	.1012	.0067	4	7.8(-5)	6.2(-3)
GN	-1	-.1925	-.0008	.1013	.0067	2	7.8(-2)	6.3(-3)
	-2	-.1925	-.0008	.1012	.0067	3	3.2(-3)	6.4(-3)
	-3	-.1925	-.0008	.1012	.0067	4	1.5(-5)	6.4(-3)
AGN	-1	-.1794	-.0003	.1010	.0057	2	3.8(-2)	1.9(-2)
	-2	-.1839	-.0006	.1007	.0062	52	9.9(-3)	1.5(-2)
	-3	-.1917	-.0008	.1012	.0067	245	9.9(-4)	7.2(-3)
	-4	-.1924	-.0008	.1012	.0067	422	9.9(-5)	6.5(-3)

4.3.3 Test 3

Case 1 GLLS converges slowly to a correct solution and results are tabulated in Table 8. GN encounters a rank deficient matrix in its iteration, while the AGN algorithm stagnates at the LS solution.

Cases 2 and 3 GLLS converges quickly to inaccurate solutions in both cases, whereas AGN stagnates at the LS solution for case 2 but converges linearly for case 3. GN again encounters a rank deficient matrix in the former case, but converges quickly for case 3, to a solution close to the one obtained by AGN.

5 Conclusions.

Iterative GLLS has been studied theoretically and numerically. Analysis suggests that it should be implemented as a nonlinear algorithm. Two basic nonlinear algorithms GN and AGN were considered.

The theoretical analysis suggests that GLLS will not always solve the equivalent nonlinear problem (43). Explicitly, numerical results show that it behaves

Table 4: Test 1, case 3.

	p	α_1	α_2	β_1	β_2	K	$c^{(K)}$	$\ \theta^{(K)} - \theta\ _2$
EX		-.1988	-.0009	.1020	.0070	n/a	n/a	n/a
LS		-.2416	-.0014	.1101	.0093	0	n/a	4.3(-2)
GLLS		-.2546	-.0018	.1113	.0101	1	3.4(-2)	5.6(-2)
	-2	-.2546	-.0018	.1113	.0101	2	2.2(-3)	5.7(-2)
	-3	-.2557	-.0018	.1114	.0102	3	1.7(-4)	5.8(-2)
GN	-2	-.2663	-.0021	.1121	.0109	4	4.5(-3)	5.9(-2)
	-3	-.2663	-.0021	.1121	.0109	5	9.0(-4)	5.9(-2)
AGN	-1	-.2419	-.0014	.1106	.0092	1	3.5(-2)	4.4(-2)
	-2	-.2541	-.0017	.1113	.0100	43	9.8(-3)	5.6(-2)
	-3	-.2652	-.0020	.1120	.0108	171	9.9(-4)	6.7(-2)

Table 5: Test 2, case 1.

	p	α_1	α_2	β_1	β_2	K	$c^{(K)}$	$\ \theta^{(K)} - \theta\ _2$
EX		-1.0053	-.1225	.1389	.1115	n/a	n/a	n/a
LS		-.1694	-.0022	.1344	.0022	0	n/a	8.5(-1)
GLLS		-.2241	-.0100	.1348	.0093	1	2.8(-1)	7.9(-1)
	-1	-.2677	-.0162	.1351	.0150	2	7.2(-2)	7.5(-1)
	-2	-.4580	-.0434	.1363	.0397	6	8.5(-3)	5.6(-1)
	-3	-.9894	-.1202	.1388	.1094	14	5.5(-4)	1.6(-2)
	-4	-1.0039	-.1223	.1389	.1113	17	5.8(-5)	1.4(-3)
GN	-1	-1.0054	-.1225	.1391	.1117	20	2.4(-2)	3.0(-4)
	-2	-1.0053	-.1225	.1389	.1115	21	2.2(-3)	0
	-3	-1.0053	-.1225	.1389	.1115	22	5.6(-6)	0
AGN	-1	-.1693	-.0022	.1345	.0022	2	5.7(-2)	8.5(-1)
	-2	-.1688	-.0024	.1342	.0023	17	9.8(-3)	8.5(-1)
	-3	-.1923	-.0057	.1343	.0053	Max	1.0(-3)	8.3(-1)

like an undamped Gauss Newton iteration, where the Jacobian matrix \mathbf{J} is approximated by Z . This approximation is only valid, however, for certain parameter ranges. When the iteration matrix C in GLLS does approximate that of the GN method, it will converge like a GN method. In some cases, GLLS acts as a regularization of GN and thus may converge in cases where GN does not, as shown in Test 2. Relative to the use of GLLS as an unconstrained iterative algorithm for solution of the specific medical application (PET), it is not reliable as an iterative procedure, see Test 1, Table 4. On the other hand, for parameter ranges observed in PET studies [15, 9], as used for Test 1, a one step iteration obtained from GLLS is always close to the converged values, but not as close as desired to the exact system parameter values.

With respect to the nonlinear algorithms proposed here, GN and AGN, their performance is consistent with the solution (43). It is clear from the results that the convergence of GN does depend, as expected, on a good initial guess. On the

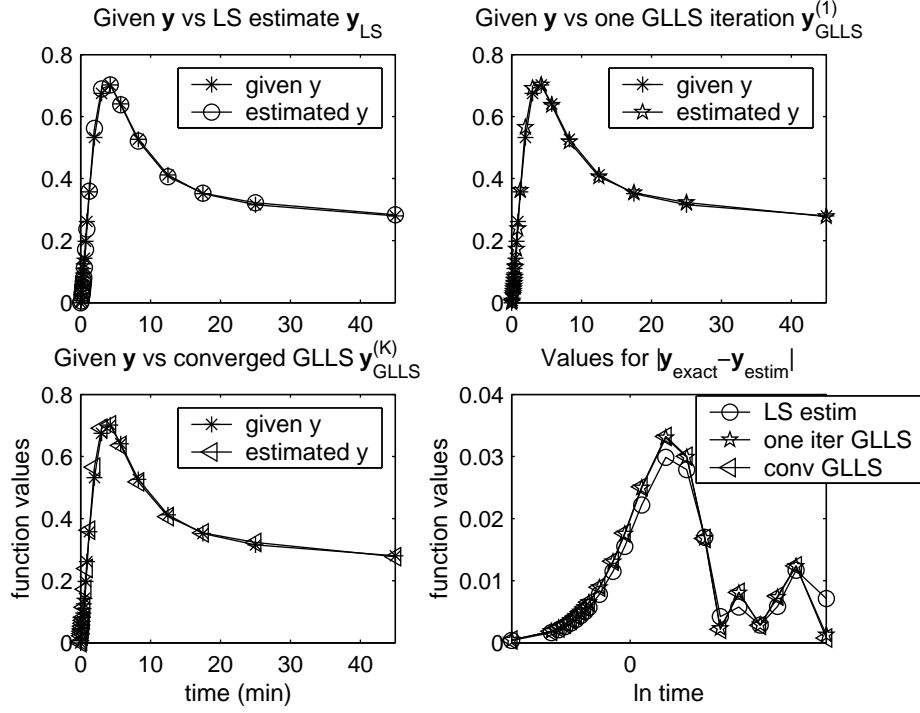


Figure 2: Test 1, case 3: comparison of estimates obtained for output data \mathbf{y} using the system parameter values θ_{ls} , θ_{GLLS} and parameter values obtained after one iteration of the GLLS algorithm.

other hand, decoupling using AGN does not exhibit the rank deficiency encountered with GN, and when it converges, it is at a linear rate. Typical approaches to improve convergence rates for GN type methods are given for example in [1, 12], such as use of a damped version or inclusion of second derivative information by quasi-Newton. From this observation, and the contrasting performance of GN and AGN, it can be presumed that their successful application will depend on incorporation of regularization, such as in the Levenberg-Marquardt algorithm. Future work will consider damping methods, regularization, the incorporation of parameter constraints for these algorithms to improve accuracy and computational efficiency, and a study of the sensitivity of these algorithms to noise in input as well as in output data.

REFERENCES

1. Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, PA, 1996.
2. K. Chen, D. Bandy, E. Reiman, S. C. Huang, M. Lawson, D. Feng, L. Yun, and A.

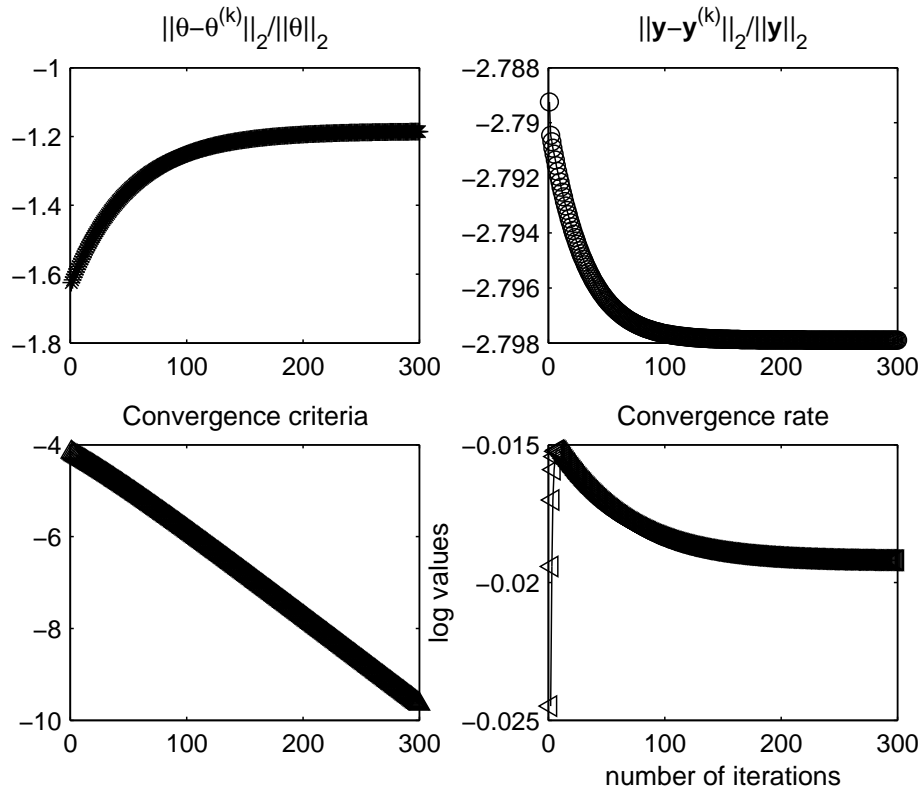


Figure 3: Test 1, case 3: convergence behavior for AGN.

Palant, *Noninvasive Quantification of the Cerebral Metabolic Rate for Glucose Using Positron Emission Tomography, F-Fluoro-2-Deoxyglucose, the Patlak Method, and an Image Derived Input Function*, J. Cereb. Blood Flow Metab., 18 (1998), pp. 716–723.

3. K. Chen, M. Lawson, E. Reiman, A. Cooper, D. Feng, S. C. Huang, D. Bandy, D. Ho, L. Yen, and A. Palant, *Generalized Linear Least Squares Method for Fast Generation of Myocardial Blood Flow Parametric Images with N-13 Ammonia PET*, IEEE Trans. Med. Imag., 17 (1998), no. 2, pp. 236–243.
4. A. R. Conn, N. I. M. Gould and P. L. Toint, *Trust-Region Methods*, MPS-SIAM, Philadelphia, PA, 2000.
5. D. Feng and D. Ho, *Parametric imaging algorithms for multicompartmental models dynamic studies with positron emission tomography*, in Quantification of Brain Function: Tracer Kinetics and Image Analysis in Brain PET, K. Uemura, N. A. Lassen, T. Jones, and I. Kanno, eds., Elsevier Science Publishers, Amsterdam, The Netherlands, 1993, pp. 127–137.
6. D. Feng, D. Ho, K. Chen, L. Wu, J. Wang, R. Liu, and S. Yeh, *An Evaluation of the Algorithms for Determining Local Cerebral Metabolic Rates of Glucose Using*

Table 6: Test 2, case 2.

	p	α_1	α_2	β_1	β_2	K	$c^{(K)}$	$\ \theta^{(K)} - \theta\ _2$
EX		-1.0053	-.1225	.1389	.1115	n/a	n/a	n/a
LS		-.1574	-.0006	.1345	.0008	0	n/a	8.6(-1)
GLLS		-.2030	-.0072	.1348	.0067	1	1.1(-1)	8.1(-1)
	-1	-.2365	-.0120	.1350	.0111	2	2.8(-2)	7.8(-1)
	-2	-.2912	-.0198	.1353	.0183	4	5.2(-3)	7.3(-1)
	-3	-1.3104	-.1676	.1397	.1524	16	9.1(-4)	3.1(-1)
GN		-46.91	-7.06	-0.66	5.73	14	RD	
AGN	-1	-.1573	-.0007	.1346	.0007	2	1.5(-2)	8.6(-1)
	-2	-.1571	-.0007	.1345	.0008	6	9.4(-3)	8.6(-1)
	-3	-.1568	-.0008	.1343	.0008	29	9.7(-4)	8.6(-1)

Table 7: Test 2, case 3.

	p	α_1	α_2	β_1	β_2	K	$c^{(K)}$	$\ \theta^{(K)} - \theta\ _2$
EX		-1.0053	-.1225	.1389	.1115	n/a	n/a	n/a
LS		-.9011	-.0024	.1943	.1457	0	n/a	1.7(-1)
GLLS	-1	-.3180	-.0210	.1496	.0204	1	2.1(-2)	7.0(-1)
	-2	-.3290	-.0224	.1500	.0218	2	4.0(-3)	6.9(-1)
	-3	-.3295	-.0224	.1501	.0219	3	8.4(-4)	6.9(-1)
GN		-10^{23}				RD		
AGN	-1	-.2651	-.0135	.1486	.0134	1	2.2(-2)	7.5(-1)
	-2	-.2652	-.0135	.1486	.0134	2	4.3(-3)	7.5(-1)
	-3	-.2655	-.0135	.1487	.0133	7	9.3(-4)	7.5(-1)

Positron Emission Tomography Dynamic Data, IEEE Trans. Med. Imag., 14 (1994), no. 4, pp. 697–710.

7. D. Feng, S. C. Huang, and Z. Wang, *Models for Computer Simulation Studies of Input Functions for Tracer Kinetic Modeling with Positron Emission Tomography*, Intern. J. Bio-Med. Comp., 32 (1993), pp. 95–110.
8. D. Feng, S. C. Huang, Z. Wang, and D. Ho, *An Unbiased Parametric Imaging Algorithm for Nonuniformly Sampled Biomedical System Parameter Estimation*, IEEE Trans. Med. Imag., 15 (1996), no. 4, pp. 512–518.
9. D. Feng, Z. Wang and S. C. Huang, *A Study on Statistically Reliable and Computationally Efficient Algorithms for Generating Local Cerebral Blood Flow Parametric Images with Positron Emission Tomography*, IEEE Trans. Med. Imag., 12 (1993), no. 2, pp. 182–188.
10. N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, PA, 1996.
11. G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.
12. C. T. Kelley, *Iterative Methods for Optimization*, SIAM, Philadelphia, PA, 1999.
13. C. Negoita, *Global Kinetic Imaging Using Dynamic Positron Emission Tomography Data*, PhD. thesis, Department of Mathematics and Statistics, Arizona State

Table 8: Test 3, case 1.

	p	α_1	α_2	β_1	β_2	K	$c^{(K)}$	$\ \theta^{(K)} - \theta\ _2$
EX		-1.2614	-.0037	.1630	.2049	n/a	n/a	n/a
LS		-.2898	-.0008	.1629	.0467	0	n/a	9.8(-1)
GLLS		-.4556	-.0013	.1629	.0737	1	6.4(-1)	8.1(-1)
	-1	-.5683	-.0017	.1629	.0921	2	2.9(-2)	7.0(-1)
	-2	-.7200	-.0021	.1629	.1168	4	8.8(-3)	5.5(-1)
	-3	-1.1060	-.0032	.1630	.1796	18	9.9(-4)	1.6(-1)
	-4	-1.2440	-.0036	.1630	.2021	46	9.3(-5)	1.8(-2)
GN		1.3767	.0082	-.0375	.0519	RD	2.6(-1)	2.6
AGN		-.2898	-.0008	.1629	.0467	Max	7.6(-5)	9.8(-1)

Table 9: Test 3, case 2.

	p	α_1	α_2	β_1	β_2	K	$c^{(K)}$	$\ \theta^{(K)} - \theta\ _2$
EX		-1.2614	-.0037	.1630	.2049	n/a	n/a	n/a
LS		.0286	.0001	.1628	-.0051	0	n/a	1.3
GLLS	-1	.0239	.0001	.1628	-.0044	1	2.3(-2)	1.3
	-2	.0221	.0001	.1628	-.0041	2	4.7(-3)	1.3
	-3	.0204	.0001	.1628	-.0038	5	4.7(-6)	1.3
GN		1.0102	.0030	.0184	-.0187	RD	3.5(-1)	2.2
AGN		.0286	.0001	.1628	-.0051	Max	9.9(-4)	1.3

University, AZ., 2003.

14. C. Negoita and R. A. Renaut, *On the Convergence of the Generalized Linear Least Squares Algorithm*, <http://snoopy.oit.edu/~negoitac> or <http://math.asu.edu/~rosie>.
15. M. Piert, R. Koeppe, B. Giordani, S. Berent, and D. Kuhl, *Diminished Glucose Transport and Phosphorylation in Alzheimer's Disease Determined by Dynamic FDG-PET*, *J. Nuclear Med.*, 37 (1996), pp. 201-208.
16. K. Wong and D. Feng, *Generalized linear least squares algorithm for non-uniformly sampled biomedical system identification with possible repeated eigenvalues*, *Comp. Methods Prog. Biomed.*, 57 (1998), pp. 167-177.
17. D. Zwillinger, *Standard Mathematical Tables and Formulae*, CRC Press LLC, Boca Raton, FL, 1996.

Table 10: Test 3, case 3.

	p	α_1	α_2	β_1	β_2	K	$c^{(K)}$	$\ \theta^{(K)} - \theta\ _2$
EX		-1.2614	-.0037	.1630	.2049	n/a	n/a	n/a
LS		.0225	.0001	.1643	-.0043	0	n/a	1.3
GLLS	-1	-.5846	-.0012	.1916	.0931	1	1.3(-2)	6.8(-1)
	-2	-.6847	-.0016	.1924	.1098	2	5.4(-3)	5.9(-1)
	-3	-.6617	-.0015	.1922	.1059	4	5.9(-4)	6.1(-1)
	-4	-.6592	-.0015	.1922	.1055	6	6.1(-5)	6.1(-1)
GN	-1	-1.1163	-.0029	.1704	.1457	1	7.8(-2)	1.6(-1)
	-2	-1.1416	-.0030	.2005	.1806	2	5.8(-3)	1.3(-1)
	-3	-1.1489	-.0030	.2025	.1857	4	4.8(-4)	1.2(-1)
	-4	-1.1494	-.0030	.2027	.1860	6	3.2(-5)	1.2(-1)
AGN	-1	-.9034	-.0023	.1973	.1454	1	1.3(-2)	3.6(-1)
	-2	-1.1046	-.0029	.2018	.1786	175	9.9(-3)	1.6(-1)
	-3	-1.1450	-.0030	.2026	.1853	383	9.9(-4)	1.2(-1)
	-4	-1.1490	-.0030	.2027	.1860	586	9.9(-5)	1.2(-1)