Hazard Concentration with Three Sources and Four Observation Sites

COMPUTATIONAL SCIENCE | RESEARCH ARTICLE

Nonlinear least squares algorithm for identification of hazards

Robert E. White

Cogent Mathematics (2015), 2: 1118219
Nonlinear least squares algorithm for identification of hazards

Robert E. White

Abstract: Given observations of selected concentrations one wishes to determine unknown intensities and locations of the sources for a hazard. The concentration of the hazard is governed by a steady-state nonlinear diffusion–advection partial differential equation and the best fit of the data. The discretized version leads to a coupled nonlinear algebraic system and a nonlinear least squares problem. The coefficient matrix is a nonsingular $M$-matrix and is not symmetric. Iterative methods are compositions of nonnegative least squares and Picard/Newton methods, and a convergence proof is given. The singular values of the associated least squares matrix are important in the convergence proof, the sensitivity to the parameters of the model, and the location of the observation sites.

Subjects: Applied Mathematics; Inverse Problems; Mathematics & Statistics; Science

Keywords: advection–diffusion; impulsive sources; Schur complement; nonnegative least squares; nonlinear; singular values

AMS subject classifications: 35K60; 65H10; 93B40; 93E24

1. Introduction

The paper is a continuation of the work in White (2011). The first new result is a convergence proof of the algorithm to approximate a solution to a nonlinear least square problem. A second new result is a sensitivity analysis of the solution as a function of the model parameters, measured data, and observation sites. Here, the singular value decomposition of the least squares matrix plays an important role.

A convergence proof is given for the algorithm to approximate the solution of a coupled nonlinear algebraic system and a nonnegative least squares problem, see lines (9–11) in this paper and in White (2011). These discrete models evolve from the continuous models using the

ABOUT THE AUTHOR

Professor Robert E. White has published numerous articles on numerical analysis including approximate solutions of partial differential equations, numerical linear algebra, and parallel algorithms. The most recent work is on hazard identification given observation data. He has authored three textbooks. The second edition of Computational Mathematics: Models, Methods and Analysis with MATLAB and MPI will be published in the fall of 2015 by CRC Press/Taylor and Francis.

PUBLIC INTEREST STATEMENT

Chemical or biological hazards can move by diffusion or fluid flow. The goal of the research is to make predictions about the location and intensities of the hazards based on the downstream location of observations sites. Mathematical models of the diffusion and flow are used to extrapolate from the observations data sites to unknown upstream locations and intensities of the hazards. The convergence of the iterative process for these extrapolations is studied. The sensitivities of the calculations, because of variations of the model parameters, observation data, and location of the observation sites, are also studied.
advection–diffusion PDE in Equation (1) (see Pao, 1992). In contrast to the continuous models in Andrle and El Badia (2015), Hamdi (2007), and Mahar and Datta (1997), this analysis focuses on the linear and semilinear algebraic systems. In Hamdi (2007), the steady-state continuous linear model requires observations at all of the down stream portions of the boundary. In Andrle and El Badia (2015), the continuous linear velocity free model where the intensities of the source term are time dependent requires observations at the boundary over a given time interval. In both these papers, the Kohn–Vogelius objective function is used for the symmetric models in the identification problems.

Here and in White (2011) we assume the coefficient matrix associated with the steady-state discretized partial differential equation in Equation (1) is an $M$-matrix and may not be symmetric, which includes coefficient matrices from upwind finite difference schemes for the advection–diffusion PDE. The least squares matrix $C(\cdot, ssites)$, see Equations (6–8), is an $k \times l$ matrix where $k > l$ and $k$ and $l$ are the number of observation and source sites, respectively. The rank and singular values (see Meyer, 2000, section 5.12), of $C(\cdot, ssites)$ determines whether of not the observation and source sites are located so that the intensities can be accurately computed.

Four examples illustrate these methods. The first two examples in Section 3 are for $n = 2$ and $n = 3$ unknowns, and they show some of the difficulties associated with the nonlinear problem. Sections 4 and 5 contain the convergence proof. Examples three and four in Section 6 are applications of the 1D and 2D versions of the steady-state advection–diffusion PDE. Sensitivity of the computed solutions due to variation of the model parameters, the data and the relative location of the source and observation sites is illustrated. The connection with the sensitivity and the singular values of the least squares matrix is established.

2. Problem formulation

Consider a hazardous substance with sources from a finite number of locations. The sources are typically located at points in a plane and can be modeled by delta functionals. The hazard is described as concentration or density by a function of space and time, $u(x, y, t)$. This is governed by the advection–diffusion PDE

$$u_t - \nabla \cdot D \nabla u + v \cdot \nabla u + d_r u = f(u) + S$$

(1)

where $S$ is a source term, $D$ is the diffusion, $v$ is the velocity of the medium, $d_r$ is the decay rate, and $f(u)$ is the nonlinear term such as in logistic growth models. The source term will be a finite linear combination of delta functionals

$$S = \sum a_i \delta(x - \hat{x}_i)$$

(2)

where the coefficients $a_i$ are the intensities and the locations $\hat{x}_i$ are possibly unknown points in the line or plane. The source coefficients should be nonnegative, which is in contrast with impulsive culling where the coefficient of the delta functionals are negative and have the form $-c_i u(x_i)$ (see White, 2009).

We will assume the source sites and the observation sites do not intersect. The reason for this is the ability to measure concentrations most likely allows one to measure the intensities at the same site.

Identification problem: Given data for $u(osites)$ at a number of given observation sites, osites, determine the intensities ($a_i$) and locations ($\hat{x}_i$) at source sites, ssites, so that

$$\|data - u(osites)\|$$

is suitably small.
2.1. Linear least squares
Since the governing PDE is to be discretized, we first consider the corresponding linear discrete problem

\[ Au = d. \]

The coefficient matrix \( A \) is derived using finite differences with upwind finite differences on the velocity term. Therefore, we assume \( A \) is a nonsingular \( M \)-matrix (see Berman & Plemmons, 1994, chapter 6 or Meyer, 2000, section 7.10). The following approach finds the solution in one least squares step. The coefficient matrix is \( n \times n \) and nodes are partitioned into three ordered sets, whose order represents a reordering of the nodes:

- \( \text{osites} \) observe sites
- \( \text{ssites} \) source sites and
- \( \text{rsites} \) remaining sites.

The discrete identification problem is given data \( \text{data} \) at \( \text{osites} \), find \( d(\text{ssites}) \) such that

\[ Au = d \text{ such that } u(\text{osites}) = \text{data}. \]

In general, the reordered matrix has the following block structure with \( \text{other} = \begin{bmatrix} \text{ssites} & \text{rsites} \end{bmatrix} \)

\[ A = \begin{bmatrix} a & e \\ f & b \end{bmatrix} \]

where \( a = A(\text{osites}, \text{osites}), e = A(\text{osites}, \text{other}), f = A(\text{other}, \text{osites}) \) and \( b = A(\text{other}, \text{other}) \).

Assumptions of this paper:

1. \( A \) is an \( M \)-matrix so that \( A \) and \( a \) have inverses.
2. \( \#(\text{osites}) = k, \) and \( \#(\text{ssites}) = l < k \) so that \( n = k + l + \#(\text{rsites}) \).
3. \( d = \begin{bmatrix} 0 & d_1^T \end{bmatrix} \) and the only nonzero terms in \( d_1 \) are at the nodes in \( \text{ssites} \), that is, \( d_1 = \begin{bmatrix} 0 & d_1(\text{ssites})^T & 0 \end{bmatrix} = \begin{bmatrix} 0 & z^T & 0 \end{bmatrix} \).

Multiply Equation (3) by a block elementary matrix (as in assumptions 1 and 2)

\[ \begin{bmatrix} I_k & 0 \\ -fa^{-1} & I_{n-k} \end{bmatrix} \begin{bmatrix} a & e \\ f & b \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \end{bmatrix} = \begin{bmatrix} I_k & 0 \\ -fa^{-1} & I_{n-k} \end{bmatrix} \begin{bmatrix} 0 \\ d_1 \end{bmatrix} \]

(4)

\[ \begin{bmatrix} a & e \\ 0 & b \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \end{bmatrix} = \begin{bmatrix} 0 \\ d_1 \end{bmatrix} \]

(5)

where \( \hat{b} \equiv b - fa^{-1}e \). Solve Equation (5) for \( u_1 \) and then \( u_0 = -a^{-1}e(\hat{b})^{-1}d_1 \). Define

\[ d_1 = \begin{bmatrix} d_1(\text{ssites})^T & 0 \end{bmatrix} \) and \( C = -a^{-1}e(\hat{b})^{-1} \).

The computed approximations of the observed data are in \( u_0 \) and the source data are in \( z = d_1(\text{ssites}) \). This gives a least squares problem for \( z \)

\[ \text{data} = C(\text{ssites})z. \]

(7)
Since $a$ is $k \times k$, $e$ is $k \times (n - k)$, and $\hat{b}$ is $(n - k) \times (n - k)$, the matrix $C$ is $k \times (n - k)$ and the least squares matrix $C(:, ssites)$ is $k \times l$ where $k > l$. The least squares problem in (7) has a unique solution if the columns of $C(:, ssites)$ are linearly independent $(C(:, ssites)z = 0$ implies $z = 0)$.

Impose the nonnegative condition on the components of the least squares solution. That is, consider finding $z \geq 0$ such that

$$r(z)^T r(z) = \min_{y \geq 0} r(y)^T r(y) \text{ wherer}(z) \equiv data - C(:, ssites)z$$

(8)

If the matrix $C(:, ssites)$ has full column rank, then the nonnegative least squares problem (8) has a unique solution (see Bjorck, 1996, section 5.2). The nonnegative least squares solution can be computed using the MATLAB command lsqnonneg.m (see MathWorks, 2015).

2.2. Least squares and nonlinear term

Consider a nonlinear variation of $Au = d$

$$Au = d(z) + \hat{d}(u)$$

(9)

where $d(z) = [0 \ z^T \ 0]^T$ (as in assumption 3) and $\hat{d}(u) = [\hat{d}_0(u_i)].$

Use the reordering in (3) and repeat the block elementary matrix transformation in (4) and (5) applied to the above. Using the notation $d(u) = [\hat{d}_0(u)]^T \ hat{d}_1(u^T)^T$ one gets

$$\begin{bmatrix} a & e \\ 0 & \hat{b} \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \end{bmatrix} = \begin{bmatrix} I_k & 0 \\ -fa^{-1} & I_{n-k} \end{bmatrix} \left( \begin{bmatrix} 0 \\ d_1 \end{bmatrix} + \begin{bmatrix} \hat{d}_0(u) \\ \hat{d}_1(u) \end{bmatrix} \right)$$

$$= \begin{bmatrix} 0 \\ d_1 \end{bmatrix} + \begin{bmatrix} \hat{d}_0(u) \\ -fa^{-1}\hat{d}_0(u) + \hat{d}_1(u) \end{bmatrix}.$$ 

Solve for $u_1$ and then for $u_0$

$$u_0 = a^{-1}\hat{d}_0(u) - a^{-1}e(\hat{b})^{-1} \left( d_1 + (-fa^{-1}\hat{d}_0(u) + \hat{d}_1(u)) \right)$$

$$= g(u) + Cd_1$$

where $C \equiv -a^{-1}e(\hat{b})^{-1}$ and $g(u) \equiv a^{-1}\hat{d}_0(u) + C(-fa^{-1}\hat{d}_0(u) + \hat{d}_1(u))$.

This leads to a nonlinear least squares problem for nonnegative $z$

$$data - g(u) = C(:, ssites)z.$$ 

(10)

The solution of the nonnegative least squares problem is input to the nonzero components in $d$, that is, $d(ssites) = z$. The coupled problem from (9) and (10) is to find $z \geq 0$ and $u$ so that

$$data - g(u) = C(:, ssites)z \text{ and } Au = d(z) + \hat{d}(u).$$

(11)

3. Two algorithms for the coupled problem

If the nonnegative least squares problem has a unique solution $z = H(u)$ given a suitable $u$, then we can write (11) as

$$Au = d(H(u)) + \hat{d}(u),$$

or as a fixed point
$u = A^{-1}(d(H(u)) + \widehat{d}(u)) \equiv G(u)$.

The following algorithm is a composition of nonnegative least squares and a Picard update to form one iteration step. Using additional assumptions we will show this is contractive and illustrate linear convergence. Example 3.2 is a simple implementation with $n = 3$, and in Section 6 Example 6.1 uses the 1D steady-state version of the model in (1) and Example 6.2 uses the 2D steady-state model in (1).

**Algorithm 1.** NLS-Picard Method for (11)

- choose nonnegative $u^0$
- for $m = 0$, $\text{mmax}$
  - solve the nonnegative least squares problem
    $data - g(u^m) = C(:,:, ssites)z^{m+1}$
  - solve the linear problem
    $Au^{m+1} = d(z^{m+1}) + \widehat{d}(u^m)$
  - test for convergence
- end for-loop

The following algorithm uses a combination of nonnegative least squares and Newton like updates. Define $F(z, u)$ and $F_u(z, u)$ as

$F(z, u) \equiv Au - d(z) - \widehat{d}(u)$ and $F_u(z, u) \equiv A - \widehat{d}_u(u)$

where $\widehat{d}_u(u) \equiv [\widehat{d}_j'(u_j)]$ is the diagonal matrix with components $\widehat{d}_j'(u_j)$.

The matrix $F_u(z, u)$ is not the full derivative matrix of $F(z, u)$ because it ignores the dependence of $z$ on $u$. Choose $\lambda > 0$ so that $F_u + \lambda I$ has an inverse.

**Algorithm 2.** NLS-Newton Method for (11)

- choose nonnegative $u^0$ and $\lambda > 0$
- for $m = 0$, $\text{mmax}$
  - solve the nonnegative least squares problem
    $data - g(u^m) = C(:,:, ssites)z^{m+1}$
  - solve the linear problem
    $Au^{m+1/2} = d(z^{m+1}) + \widehat{d}(u^m)$
    compute $F(z^{m+1/2}, u^{m+1/2}) = Au^{m+1/2} - d(z^{m+1}) - \widehat{d}(u^{m+1/2})$
    $= \widehat{d}(u^m) - \widehat{d}(u^{m+1/2})$
    compute $F_u(z^{m+1/2}, u^{m+1/2}) \equiv A - \widehat{d}_u(u^{m+1/2})$
  - solve the linear problem
    $(F_u(z^{m+1/2}, u^{m+1/2}) + \lambda I) \Delta u = F(z^{m+1/2}, u^{m+1/2})$
    $u^{m+1} = u^{m+1/2} - \Delta u$
  - test for convergence
- end for-loop

Variations of the NLS-Newton algorithm include using over relaxation, $\omega$, at the first linear solve and using damping, $\alpha$, on the second linear solve in the Newton update:

$u^{m+1/2} = (1 - \omega)u^m + \omega u^{m+1/2}$ and

$u^{m+1} = u^{m+1/2} - \alpha \Delta u$.

Appropriate choices of $\omega$, $\alpha$ and $\lambda$ can accelerate convergence.

The calculations recorded in Table 4 and Figure 4 in White (2011), illustrate the NLS-Newton algorithm for the 1D steady-state problem with logistic growth term $\widehat{d}(u) = f(u) = c(100 - u)u$ and
variable growth rate \(c\). The larger the \(c\) parameter resulted in larger concentrations. The intensities at the three source sites were identified. However, the number of iterations of the NLS-Newton algorithm increased as the \(c\) parameter increased. The algorithm did start to fail for \(c\) larger than 0.0100.

The lack of convergence is either because the associated fixed point problem does not have a contractive mapping or because the solutions become negative. The following example with \(n = 2\), nonzero source term and nonsymmetric \(A\) illustrates the difficulty of solving a semilinear algebraic problem where the solution is required to be positive.

**Example 3.1** Let \(n = 2\) and use \(f_i(u_j) = c(100 - u_i)u_j\). Equation (9) has the form

\[
\begin{bmatrix}
1 & 0 \\
-\epsilon & 1
\end{bmatrix}
\begin{bmatrix}
u_0 \\
u_1
\end{bmatrix}
= \begin{bmatrix}
0 \\
z
\end{bmatrix}
+ \begin{bmatrix}
f_0(u_0) \\
f_1(u_1)
\end{bmatrix}.
\]

The first equation is \(u_0 = c(100 - u_0)u_0\) whose solution is either \(u_0 = 0\) or \(u_0 = 100 - 1/c > 0\). Put the nonzero solution into the second equation \(-\epsilon(100 - 1/c) + u_1 = z + c(100 - u_1)u_1\) and solve for \(u_1\). If \(\epsilon(100 - 1/c) + z > 0\), then the second equation is \(u_1 = c(100 - u_1)u_1 + \epsilon(100 - 1/c) + z\) and will have a positive solution.

The next example with \(n = 3\), given data at the first and second nodes and a source term at the third node illustrates the coupled problem in (11). This simple example can be solved both by-hand calculations and the NLS-Picard algorithm, and more details are given in Appendix 1.

**Example 3.2** Let the data be given at the first and second nodes

\[
data = \begin{bmatrix}
2.9711 \\
5.4971
\end{bmatrix}.
\]

Let the matrix \(A\) and nonlinear terms be given by system

\[
\begin{bmatrix}
3 & 0 & -1 \\
0 & 3 & -2 \\
-2 & -2 & 4
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
u_3
\end{bmatrix}
= \begin{bmatrix}
0 \\
z
\end{bmatrix}
+ \begin{bmatrix}
cu_1(10 - u_1) \\
cu_2(10 - u_2) \\
cu_3(10 - u_3)
\end{bmatrix}.
\]

The submatrices of \(A\) are

\[
a = \begin{bmatrix}
3 & 0 \\
0 & 3 \\
-1 & -2
\end{bmatrix}, e = \begin{bmatrix}
-1 \\
-2
\end{bmatrix}, f = \begin{bmatrix}
-2 & -2
\end{bmatrix} \text{ and } b = [4].
\]

Then one easily computes

\[
\hat{b} = b - fa^{-1}e = [2] \text{ and } c = -a^{-1}e\hat{b}^{-1} = (1/6)\begin{bmatrix}
1 \\
2
\end{bmatrix}.
\]

The \(g(u)\) in the nonlinear least square Equation (11) is

\[
g(u) = c\begin{bmatrix}
(4/9)u_1(10 - u_1) + (1/6)u_2(10 - u_2) + (1/9)u_3(10 - u_3) \\
(5/9)u_1(10 - u_1) + (2/6)u_2(10 - u_2) + (2/9)u_3(10 - u_3)
\end{bmatrix}.
\]

The calculation with the NLS-Picard algorithm for this example with \(n = 3\) in White (2015, testlsnonlc.m) confirms both the exact solution for \(c = 8/100 = 0.0800\) and convergence for \(c < 0.1667\), which are given in Appendix 1. The number of iterations needed for convergence increases as \(c\) increases, and convergence fails for \(c = 0.3000\). Both the exact and NLS-Picard calculations show \(z > 0\) for \(c \leq 0.1975\) and \(z = 0\) for \(c \geq 0.1976\). The outputs for five values of \(c = 0.08:0.04:0.24\)
are given in Figure 1. The left graphs indicate the five solutions of (10) for fixed \( z = z = 10 \) and increasing \( c \). The two “*” are the given data, which are from the solution of (10) with \( c = 0.0800 \) and 5% error. The right graphs are the converged solutions of (11) for \( c = 0.08:0.04:0.20 \).

4. Nonnegative least squares problem

The nonnegative least squares problem in (10) for \( z \geq 0 \) has several forms

\[
R(z)^TR(z) = \min_{y \geq 0} R(y)^TR(y) \quad \text{where} \quad R(z) \equiv data - g(u) - C(:, sites)z.
\]

This may be written as

\[
R(z)^TR(z) = (data - g(u))^T(data - g(u)) - 2J(z)
\]  

(12)

where

\[
J(z) \equiv \frac{1}{2}z^TC(:, sites)^TC(:, sites)z - z^TC(:, sites)^T(data - g(u)).
\]

If \( C(:, sites) \) has full column rank, then the matrix \( C(:, sites)^TC(:, sites) \) is symmetric positive definite (SPD). \( J(z) \) is the quadratic functional associated with the linear system (normal equations)

\[
0 = r(z) \equiv C(:, sites)^T(data - g(u)) - C(:, sites)^TC(:, sites)z
\]

When the nonnegative condition is imposed on \( z \geq 0 \) and the matrix is SPD, then the following are equivalent (see Cryer, 1971):

\[
J(z) = \min_{y \geq 0} J(y) \quad \text{(minimum quadratic functional)},
\]

\[
0 \geq r(z)^T(y - z) \quad \text{for all} \ y \geq 0 \quad \text{(variational inequality)} \quad \text{and}
\]

\[
0 = r(z)^Tz \quad \text{and} \ 0 \geq r(z) \quad \text{(linear complementarity problem)}.
\]

Any solution of a variational inequality is unique and depends continuously on the right-hand side. The next theorem is a special case of this.

**Theorem 4.1** Consider the nonnegative least squares problem in (10). If \( C(:, sites) \) has full column rank, then there is only one nonnegative solution for each \( u \). Let \( z = H(u) \) be the solution for each \( u \)
S where $S$ is a bounded set of nonnegative $n$-vectors. Moreover, if $\| \hat{a}(u) - \hat{a}(v) \| \leq \tilde{K} \| u - v \|$ then there is a constant $C_1$ such that

$$\| C(:, ssites)(H(u) - H(v)) \| \leq C_1 \| u - v \|$$

**Proof** Let $z$ and $w$ be solutions of (10). The variational inequalities for $z$ with $y = w$, and for $w$ with $y = z$ are:

$$0 \geq r(z)^T (w - z) \text{ and } 0 \geq r(w)^T (z - w).$$

Add these to get a contradiction for the case $z - w$ is not a zero vector:

$$0 \geq (z - w)^T C(:, ssites)^T C(:, ssites) (z - w) > 0.$$ 

In order to show the continuity, use the above with $z = H(u)$ and $w = H(v)$:

$$0 \geq |C(:, ssites)^T (data - g(u)) - C(:, ssites)^T C(:, ssites) H(u)| (H(v) - H(u))$$

$$0 \geq |C(:, ssites)^T (data - g(v)) - C(:, ssites)^T C(:, ssites) H(v)| (H(u) - H(u)).$$

Add these to obtain for $B \equiv C(:, ssites)^T C(:, ssites)$

$$(H(v) - Hu)^T B (H(v) - H(u)) \leq -(g(v) - g(u))^T C(:, ssites) (H(v) - H(u)).$$

Use the Cauchy inequality

$$\| C(:, ssites)(H(v) - H(u)) \|^2 \leq \| g(v) - g(u) \| \| C(:, ssites)(H(v) - H(u)) \|.$$ 

By the definition of $g$ and the assumption on $d$ we have for some $C_1$

$$\| C(:, ssites)(H(v) - H(u)) \| \leq C_1 \| v - u \|.$$ 

The solution of the variational problem for fixed $u$ has the form $z = H(u)$. The assumption that $C(:, ssites)$ has full column rank implies there is a positive constant $c_0$ such that $c_0 (w^T \hat{w}) \leq w^T C(:, ssites)^T C(:, ssites) w$. This and the above theorem give the following

$$c_0 \| H(v) - H(u) \| \leq \| C(:, ssites)(H(v) - H(u)) \| \leq C_1 \| v - u \|.$$ (13)

Although in Example 3.2 with $n = 3$ one can estimate $c_0$ and $C_1$, generally these constants are not easily approximated. However, $c_0$ can be estimated by the smallest singular value of the least squares matrix.

5. Convergence proof of the NLS-Picard algorithm

Let the assumptions in the previous theorem hold so that (12) is true for all $u, v \in S$. The mapping $G(u) \equiv A^{-1} (d(H(u)) + \hat{a}(u))$ may not give $G(u) \in S$. This appears to be dependent on the particular problem and properties of $\hat{a}(u)$. If the bound is chosen so that $\hat{a}(u)$ is also nonnegative, then assumption (1) implies $A^{-1} \geq 0$ and hence $G(u) \geq 0$. The least squares condition requires the solution to be “close” to the given data at the osites, which suggests the $G(u)$ should remain bounded. The following theorem gives general conditions on the problem.

**Theorem 5.1** Let the assumptions of Theorem 4.1 hold for $u, v \in S$. Assume there is a solution to the coupled problem (11), $u^* \in S$. Let $c_0$ and $C_1$ be from (12). If

$$\tilde{\gamma} \equiv \| A^{-1} (C_1/c_0 + \tilde{K}) \| < 1.$$
then the NLS-Picard algorithm will converge to this solution of (11).

**Proof** Let $u^* = G(u^*)$ and $u^{m+1} = G(u^m)$.

$$
\|u^* - u^{m+1}\| = \|G(u^*) - G(u^m)\|
= \left\|A^{-1}(d(H(u^*)) - d(H(u^m))) + A^{-1}(\tilde{d}(u^*) - \tilde{d}(u^m))\right\|
\leq \left\|A^{-1}(C_1/C_0)\|u^* - u^m\| + \|A^{-1}\|\tilde{R}\|u^* - u^m\|ight\|
= \hat{r}\|u^* - u^m\|
\vdots
\leq \hat{r}^m\|u^* - u^0\|.
$$

Since $\hat{r} < 1$, this implies convergence.

---

6. Applications and sensitivity analysis

Both the NLS-Picard and NLS-Newton methods converge linearly, that is, the errors at the next iteration are bounded by a contractive constant, $\hat{r} < 1$, times the error from the current iteration. However, the contractive constant for the NLS-Newton algorithm is significantly smaller than the contractive constant for the NLS-Picard algorithm, and this results in fewer iterations steps for the NLS-Newton algorithm. The calculations in the next two examples can be done by either algorithm.

The relevance of the singular values of the least squares matrix to the sensitivity of the calculations with variations in the data and the source and observation sites is demonstrated.

**Example 6.1** Consider the steady-state 1D model in (1). The following calculations were done by the MATLAB code in White (2015, poll1dlsnonla2.m) and with seven observation sites, three source sites, and $n = 160$. The differential Equation (1) was discretized by the upwind finite difference approximations, and the sources terms were the intensities times the approximation of the delta functionals.
where initially $d \equiv \text{zeros}(n,1)$. In order to check the accuracy of the finite difference model, calculations were done with $n = 160, 320$ and $480$, which gave similar results as reported below for $n = 160$.

The condition number (ratio of the largest and smallest singular values) of the least squares matrix controls the relative errors in the computed least squares problem. The condition numbers for the $n = 160, 320$, and $480$ are about the same at $17.0170, 16.5666$, and $16.4146$, respectively.

For $c = 0.010$ the NLS-Picard algorithm did not converge, but the NLS-Newton algorithm converged in 92 steps. The solid lines in Figure 2 are with no random error in the data, and the dotted lines are from data with 5% random error. For $c = 0.005$ the NLS-Picard algorithm converged in 100 steps with $\hat{r} \approx .83$, and the NLS-Newton algorithm converged in 19 steps with $\hat{r} \approx .41$. If $w$ is changed from 1.0 to 1.1, then the NLS-Newton algorithm converges in 16 steps with $\hat{r} \approx .28$. As $c$ decreases, the iterations required for convergence and $\hat{r}$ decrease.

The dotted lines indicate some uncertainty in the numerical solutions, which can be a function of the physical parameters and the location of the observation sites relative to the source sites. If there is no error in the data, the intensities of the sources are in the vector

$$z = \begin{bmatrix} 4.0000 & 1.0000 & 2.0000 \end{bmatrix}.$$ 

Use 100 computations with the $c = 0.001$ and 5% random errors. The means and standard deviations of the computed intensities at the three source sites are

$$\text{mean}(z) = \begin{bmatrix} 4.0357 & 0.9437 & 2.0187 \end{bmatrix}$$ and

$$\text{std}(z) = \begin{bmatrix} 0.2570 & 0.3631 & 0.1375 \end{bmatrix}.$$ 

The standard deviation of the intensities at the center source in Figure 2 is large relative to its mean value. The large computed least squares errors are a result of some relatively small singular values in the singular value decomposition of the least squares matrix

$$C(:, \text{ssites}) = U\Sigma V^T$$

where the singular values $\sigma$ are the diagonal components of $\Sigma$; $U$ is $7 \times 7$, $\Sigma$ is $7 \times 3$ and $V$ is $3 \times 3$.

In this example, the third singular value is small

$$\sigma = \begin{bmatrix} 0.2882 & 0.0916 & 0.0169 \end{bmatrix}.$$ 

The least square solution of (10) is given by the pseudo inverse of the least squares matrix times

$$V\Sigma^+U^T \hat{d} = v_1 u_1^T / \sigma_1 + v_2 u_2^T / \sigma_2 + v_3 u_3^T / \sigma_3.$$ 

Because $\sigma_3$ is relatively small, any errors in $\hat{d}$ will be amplified as well as the components of

$$v_3 = \begin{bmatrix} 0.5778 & -0.7882 & 0.2121 \end{bmatrix}^T.$$ 

This suggests the second component, the center source, of least squares solution will have significant errors. The errors can be decreased if the percentage random error is smaller or by alternative location of the observation sites. In the 1D model this is not too difficult to remedy. However, in the 2D model this is more challenging.
Example 6.2 Consider the steady-state 2D model in (1) with boundary conditions equal to zero upstream and zero normal derivative downstream. The calculations in Figure 3 were done by the MATLAB code in White (2015, testlsnonlf2da3.m) using NLS-Picard and with four variable observation sites and three fixed source sites. The three fixed sources are to the left near \( x = 0 \), and the four observation sites are indicated by the symbol “*” above the site. The intensities times the approximation of the delta functionals is

\[
    d(\text{sites}) = [a_1, a_2, a_3]^T / (dx \times dy)
\]

where initially \( d \equiv \text{zeros}(nn,1) \) and \( nn = (nx + 1) \times (ny + 1) \). If there is no random error in the data, then the computed source will be

\[
    \begin{bmatrix}
        8.0000 & 16.0000 & 4.0000 \\
    \end{bmatrix}.
\]

The numerical experiments involve varying the location of the observation site at \((x, 16)\) where \( x = L - \Delta L \) and varying the \( y \)-component of the velocity, \( \text{vely} \). The velocity term in (1) is \((\text{velx}, \text{vely}) = (1.00, 0.10)\). The base calculation in Figure 3 uses \( \Delta L = 30 \) and \( \text{vely} = 0.10 \) along with 5% random error in the data and \( c = 0.0004 \) growth coefficient in the nonlinear term. The singular values of the least squares matrix and the mean and standard deviations of the intensities of the three computed sources are recorded for 100 computations.

\( \Delta L = 30 \) and \( \text{vely} = 0.10 \):

\[
    \sigma = \begin{bmatrix}
        0.0472 & 0.0300 & 0.0046 \\
    \end{bmatrix}
\]

\[
    \text{mean}(z) = \begin{bmatrix}
        7.9947 & 16.0341 & 4.0040 \\
    \end{bmatrix} \quad \text{and}
\]

\[
    \text{std}(z) = \begin{bmatrix}
        0.2523 & 0.4966 & 0.1451 \\
    \end{bmatrix}.
\]

Changing the \( y \)-component of the velocity can cause the concentration to be moved away from the observation sites and cause more uncertainty in the calculations, which is indicated by larger standard deviations and smaller singular values of the least squares matrix. Change \( \text{vely} = 0.10 \) to \( \text{vely} = 0.15 \) and \( \text{vely} = 0.05 \).
ΔL = 30 and vely = 0.15:

\[\sigma = \begin{bmatrix} 0.0424 & 0.0302 & 0.0002 \\ \end{bmatrix}\]

\[\text{mean}(z) = \begin{bmatrix} 7.9562 & 16.034 & 23.6939 \end{bmatrix}\]

\[\text{std}(z) = \begin{bmatrix} 0.2399 & 0.5479 & 30.7650 \end{bmatrix}\].

Here, the third singular value has significantly decreased and can cause failure of the NLS-Picard algorithm or increased computation errors because of the variation in the data.

ΔL = 30 and vely = 0.05:

\[\sigma = \begin{bmatrix} 0.0853 & 0.0004 & 0.0001 \\ \end{bmatrix}\]

\[\text{mean}(z) = \begin{bmatrix} 7.7883 & 16.6030 & 3.9778 \end{bmatrix}\]

\[\text{std}(z) = \begin{bmatrix} 0.5744 & 2.0884 & 0.1167 \end{bmatrix}\].

In this case the second and third singular values of the least squares matrix decreased, and the standard deviation of the second source increased.

Changing the location of the third observation site can also cause more uncertainty in the calculations, which is indicated by larger standard deviations and smaller singular values of the least squares matrix. Change ΔL = 30 to ΔL = 20 and ΔL = 10 and keep vely = 0.10.

ΔL = 20 and vely = 0.10:

\[\sigma = \begin{bmatrix} 0.0472 & 0.0300 & 0.0005 \\ \end{bmatrix}\]

\[\text{mean}(z) = \begin{bmatrix} 8.0012 & 16.0350 & 4.5229 \end{bmatrix}\]

\[\text{std}(z) = \begin{bmatrix} 0.2350 & 0.5321 & 4.5265 \end{bmatrix}\].

Here, the third singular value has significantly decreased and can cause failure of the NLS-Picard algorithm or increased computation errors because of the variation in the data. The standard deviation of the third source has significantly increased.

Figure 4. Six sources and nine observations.
\( \Delta L = 10 \) and \( \text{vely} = 0.10: \)

\[
\sigma = \begin{bmatrix}
0.0472 & 0.0320 & 0.000396 \\
0.0782 & 0.0612 & 0.0459 & 0.0304 & 0.0178 & 0.0047 \\
\end{bmatrix}
\]

\[
\text{mean}(z) = \begin{bmatrix}
8.0005 & 15.9137 & 238.9531 \\
23.9404 & 16.0215 & 8.0135 & 3.9869 & 16.1539 & 7.9334 \\
\end{bmatrix}
\] and 

\[
\text{std}(z) = \begin{bmatrix}
0.2379 & 0.5007 & 396.3124 \\
0.8512 & 0.4408 & 0.2064 & 0.0549 & 0.9046 & 0.2879 \\
\end{bmatrix}.
\]

In this case the third singular values of the least squares matrix have dramatically decreased, and the standard deviation of the third source is very large relative to the mean.

The calculation in Figure 4 has six source sites (near the axes) and nine observation sites (downstream and below the “*”) (White, 2015, testlsnonlf2da6.m) uses NLS-Picard. Here the \( \text{vely} \) is larger with velocity the equal to (1.00, 0.30), and the random error in the data is smaller and equal to 1%:

\[
\sigma = \begin{bmatrix}
0.0782 & 0.0612 & 0.0459 & 0.0304 & 0.0178 & 0.0047 \\
0.0782 & 0.0612 & 0.0459 & 0.0304 & 0.0178 & 0.0047 \\
\end{bmatrix}
\]

\[
\text{mean}(z) = \begin{bmatrix}
23.9404 & 16.0215 & 8.0135 & 3.9869 & 16.1539 & 7.9334 \\
23.9404 & 16.0215 & 8.0135 & 3.9869 & 16.1539 & 7.9334 \\
\end{bmatrix}
\] and 

\[
\text{std}(z) = \begin{bmatrix}
0.8512 & 0.4408 & 0.2064 & 0.0549 & 0.9046 & 0.2879 \\
0.8512 & 0.4408 & 0.2064 & 0.0549 & 0.9046 & 0.2879 \\
\end{bmatrix}.
\]

The source closest to the origin is the most difficult to identify; with one percent error in the data the standard deviation is 0.8512 and with five percent error this goes up to 4.2348. This happens despite the relatively close distribution of the singular values from 0.0782 to 0.0047 for a condition number equal to 16.7658.

7. Summary

In order to determine locations and intensities of the source sites from data at the observation sites, there must more observations sites than source sites and the observation sites must be more chosen so that the least squares matrix \( C(\cdot, \text{ssites}) \) has full column rank. Furthermore, the singular values in the singular value decomposition of the least squares matrix are important in the convergence proof and the location of the observation sites.

Measurement errors in the observations as well as uncertain physical parameters can contribute to significant variation in the computed intensities of the sources. Multiple computations with varying these should be done, and the means and standard deviations should be noted. Large standard deviations from the mean indicate a lack of confidence in the computations. In this case, one must adjust the observation sites, which can be suggested by inspection of the smaller singular values for the least squares matrix and the corresponding columns in \( V \) of the singular value decomposition.

Funding

No external funding. Some of this research was done before retirement at North Carolina State University.

Author details

Robert E. White

E-mail: white@math.ncsu.edu

1 Department of Mathematics, Box 8205, North Carolina State University, Raleigh, NC 27695-8205, USA.

Citation information

Cite this article as: Nonlinear least squares algorithm for identification of hazards, Robert E. White, Cogent Mathematics (2015), 2: 1118219.

Cover image

Source: Author

References

Andrie, M., & El Badia, A. (2015). On an inverse problem. Application to a pollution detection problem, II. Inverse Problems in Science and Engineering, 23, 389–412.

Berman, A., & Plemmons, R. J. (1994). Nonnegative matrices in the mathematical sciences. Philadelphia, PA: SIAM.

Bjorck, A. (1996). Numerical methods for least squares problems. Philadelphia, PA: SIAM.

Cryer, C. W. (1971). The solution of a quadratic programming problem using systematic over ralaxation. SIAM Journal on Control and Optimization, 9, 385–392.

Hamdi, A. (2007). Identification of point sources in two-dimensional advection-diffusion-reaction equations: Application to pollution sources in a river. Stationary case, Inverse Problems in Science and Engineering, 15, 855–870.

Mahar, P. S., & Datto, B. (1997). Optimal monitoring network and ground-water-pollution source identification. Journal of Water Resources Planning and Management, 199, 199–207.

MathWorks. (2015). (123, pp. 199–207) Retrieved from http://www.mathworks.com

Meyer, C. D. (2000). Matrix analysis and applied linear algebra. Philadelphia, PA: SIAM.

Pao, C. V. (1992). Nonlinear parabolic and elliptic equations. New York, NY: Plenum Press.

White, R. E. (2009). Populations with impulsive culling: Identification and control. International Journal of Computer Mathematics, 86, 2143–2164.
Appendix 1

Details for Example 3.2 with \( n = 3 \)

The least square problem in (11) can be solved for nonnegative \( z = H(u) \)

\[
z = \left( \frac{6}{5} \right) \left( \text{data}(1) + 2 \text{data}(2) \right) - (16/15) c u_1 (10 - u_1) - (22/15) c u_2 (10 - u_2) - c u_3 (10 - u_3).
\]

Next, consider the system in (11). The first two equations give the first and second unknowns in terms of the third unknown

\[
\begin{align*}
    u_1 &= \frac{-(3 - c 10) + \sqrt{(3 - c 10)^2 + 4 cu_3}}{2c} \quad \text{and} \\
    u_2 &= \frac{-(3 - c 10) + \sqrt{(3 - c 10)^2 + 8 cu_3}}{2c}.
\end{align*}
\]

Insert the formulas for \( z, u_1, \) and \( u_2 \) into the third equation in the system to get a nonlinear problem for a single unknown \( u_3 \) and the single equation

\[
\sqrt{(3 - c 10)^2 + 4 cu_3} + 2 \sqrt{(3 - c 10)^2 + 8 cu_3} = 3(3 - 10) + 2c(\text{data}(1) + 2 \text{data}(2)).
\]

When \( c = 8/100 \), this can be solved by Newton’s method to give \( u_3 = 7.2526 \). Then solve for the first two unknowns and finally solve for \( z \)

\[
u_1 = 2.9748, u_2 = 5.4952 \quad \text{and} \quad z = 10.4764.
\]

In the above example, we were able to explicitly solve for nonnegative \( z \) in terms of the components of \( u \). The problem \( Au = d(H(u)) + \hat{d}(u) \) is

\[
\begin{bmatrix}
    3 & 0 & -1 \\
    0 & 3 & -2 \\
    -2 & -2 & 4
\end{bmatrix}
\begin{bmatrix}
    u_1 \\
    u_2 \\
    u_3
\end{bmatrix} =
\begin{bmatrix}
    cu_1 (10 - u_1) \\
    cu_2 (10 - u_2) \\
    D(u_1, u_2)
\end{bmatrix}
\]

where

\[
D(u_1, u_2) \equiv \left( \frac{6}{5} \right) \left( \text{data}(1) + 2 \text{data}(2) \right) - (16/15) c u_1 (10 - u_1) - (22/15) c u_2 (10 - u_2).
\]

The equivalent fixed point problem is \( u = A^{-1}(d(H(u)) + \hat{d}(u)) \)

\[
\begin{bmatrix}
    u_1 \\
    u_2 \\
    u_3
\end{bmatrix} = \frac{1}{18}
\begin{bmatrix}
    8 & 2 & 3 \\
    4 & 10 & 6 \\
    6 & 6 & 9
\end{bmatrix}
\begin{bmatrix}
    cu_1 (10 - u_1) \\
    cu_2 (10 - u_2) \\
    D(u_1, u_2)
\end{bmatrix} \equiv G(u).
\]
Expand the matrix-vector product to get
\[
G(u) \approx \begin{bmatrix}
  f_1(.26) + f_2(-.13) + 2.8 \\
  f_1(-.13) + f_2(.07) + 5.6 \\
  f_1(-.20) + f_2(-.40) + 8.4
\end{bmatrix}
\]
where \( f_i \equiv cu_i(10 - u_i). \)

Assume \( u, v \) are nonnegative and bounded and in
\[
S \equiv \{ u | u = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^T, 0 \leq u_i \leq 10, i = 1, 2, 3 \}.
\]
Then \( 0 \leq f_i \leq 25c \) and since \( z \geq 0, G(u) \geq 0. \) If \( c \leq 1 \), then \( G(u) < 10 \) and \( G \) maps \( S \) into \( S \).

The contraction mapping theorem requires \( G(u) \) to be contractive, that is,
\[
\|G(u) - G(v)\|_\infty \leq r\|u - v\|_\infty \text{ where } r < 1.
\]

One can approximate the differences
\[
\Delta f_i \equiv cu_i(10 - u_i) - cv_i(10 - v_i) \\
\|\Delta f\| \leq c10\|u - v\|.
\]

Then it is possible to estimate \( r \) as a function of \( c \)
\[
\|G(u) - G(v)\|_\infty \leq \begin{begin{pmatrix} 8 & 2 & 3 \\
  4 & 10 & 6 \\
  6 & 6 & 9 \\
\end{pmatrix} \begin{pmatrix} \Delta f_1 \\
  \Delta f_2 \\
  \Delta f_3
\end{pmatrix} \begin{pmatrix} 1/18 \end{pmatrix} = \begin{pmatrix} \Delta f_1(8 + 3(-16/15)) + \Delta f_2(2 + 3(-22/15)) \\
  \Delta f_1(4 + 6(-16/15)) + \Delta f_2(10 + 6(-22/15)) \\
  \Delta f_1(6 + 9(-16/15)) + \Delta f_2(6 + 9(-22/15))
\end{pmatrix}
\]
\[
\leq c10(10.8/18)\|u - v\|_\infty.
\]

Let \( r = c(108/18) < 1 \) and require \( c < 1/6 = 0.1667 \).