# APPROXIMATE INVERSE AND IMPLICIT FUNCTIONS USING AN M/NSET VARIATION*

David W. Dreisigmeyer†
Los Alamos National Laboratory
Los Alamos, NM 87544

February 27, 2007

**Abstract**

We employ a variation of multivariate or nonlinear state estimation techniques (M/NSET) in order to approximate a nonlinear mapping from one Euclidean space into another. Potential applications are approximating the implicit and inverse function theorems, approximating a chart of a manifold and, designing surrogate models.

**Keywords:** multivariate state estimation technique, nonlinear state estimation technique, manifold charting, implicit function theorem, inverse function theorem, surrogate models

**AMS subject classifications:** 26B10, 65D05

## 1 Introduction

The multivariate and nonlinear state estimation techniques (M/NSET) [1, 4] are methods for determining if a given observation is in a target set $\Omega \subset \mathbb{R}^n$. What they attempt to do is model the identity mapping on $\Omega$ using samples from $\Omega$. So, for each $\mathbf{x} \in \Omega$, we want to construct a function such that $\mathbf{f}(\mathbf{x}) \approx \mathbf{x}$, that is, $\mathbf{f} \approx \mathbf{id}_\Omega$. Then we can use the residual $\|\mathbf{f}(\mathbf{x}) - \mathbf{x}\|$ to determine if $\mathbf{x} \in \Omega$ or not.

Here we look at extending M/NSET to the following case: instead of trying to approximate $\mathbf{id} : \Omega \to \Omega$, we want to approximate more general nonlinear mappings $\mathbf{f} : \Omega \to \Psi$, where $\Psi \subset \mathbb{R}^m$. Specific examples that we have in mind for this are to approximate the functions arising out of an implicit or inverse function. So, we could try to approximate the $\varphi(\mathbf{y})$ such that

$$\mathbf{f}(\mathbf{y}, \varphi(\mathbf{y})) = \mathbf{0} \tag{1.1}$$

*LA-UR-07-1214
†email: dreisigm@lanl.gov

or, the $\mathbf{g}(\mathbf{y})$ such that

$$(\mathbf{g} \circ \mathbf{f})(\mathbf{x}) \quad = \quad \mathbf{x}. \tag{1.2}$$

We can use (1.1) to, e.g., put (approximate) local coordinate charts on a manifold $\mathcal{M}$ where $\mathbf{y}$ is an element of the tangent space $\mathcal{T}_{\mathbf{x}}\mathcal{M}$ at $\mathbf{x} \in \mathcal{M}$.

Our paper is organized as follows. In Section 2 we give an overview of the standard M/NSET and state our variation. Section 3 is devoted to an investigation of the geometry of the algorithm. We give some examples in Section 4. A discussion follows in Section 5.

## 2 The M/NSET variation

Standard M/NSET [1, 4] attempts to approximate the identity mapping $\mathbf{id}_\Omega$ on some set $\Omega$. In order to do this, one starts with a sampling of points $\mathbf{x}_i \in \Omega$ and forms the matrix $X = [\, \mathbf{x}_1, \ldots, \mathbf{x}_n \,]$. Then, with this $X$ we define our approximation to $\mathbf{id}_\Omega$ as

$$\begin{aligned} \widehat{\mathbf{x}} &= X(X \otimes X)^{-1}(X \otimes \mathbf{x}) && (2.1a) \\ &= \widehat{\mathbf{id}}_\Omega(\mathbf{x}), && (2.1b) \end{aligned}$$

where the operation $\otimes$ is some symmetric function acting on the columns of $X$. So, for instance, we could take

$$(X \otimes X)_{ij} \quad = \quad 1 - \frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2}{\|\mathbf{x}_i\|_2 + \|\mathbf{x}_j\|_2}\,, \tag{2.2}$$

which is the standard MSET operation. Note in particular that $\mathbf{x}_i = \widehat{\mathbf{id}}_\Omega(\mathbf{x}_i)$ for $\mathbf{x}_i \in X$ regardless of the definition of $(X \otimes X)$.

A simple generalization of this method is to have the image of our mapping be some other set $\Psi$. So now we are trying to approximate some nonlinear mapping $\mathbf{f} : \Omega \to \Psi$. For each $\mathbf{x}_i \in X \subset \Omega$ we will have a corresponding image point $\mathbf{y}_i = \mathbf{f}(\mathbf{x}_i) \in \Psi$. We can collect all of these image points into the matrix $Y = [\, \mathbf{y}_1, \ldots, \mathbf{y}_n \,]$. Then, instead of (2.1), we will have the mapping

$$\begin{aligned} \widehat{\mathbf{x}} &= Y(X \otimes X)^{-1}(X \otimes \mathbf{x}) && (2.3a) \\ &= \widehat{\mathbf{f}}(\mathbf{x}), && (2.3b) \end{aligned}$$

where we have that $\mathbf{y}_i = \widehat{\mathbf{f}}(\mathbf{x}_i)$.

Note that it is fairly easy to update the algorithm by appending/deleting columns to/from $X$ and $Y$. This will be equivalent to updating a QR factorization [3]. Now we'll look at the geometry of the algorithm in a little more detail.

# 3 The geometry of the algorithm

To begin exploring the geometry of our M/NSET variation, let us begin by using the QR decomposition of $(X \otimes X)$ to have

$$(X \otimes X) \quad = \quad QR. \tag{3.1}$$

Then, $(X \otimes \mathbf{x}) = QR\mathbf{y}$ and, in particular, $(X \otimes \mathbf{x}_i) = QR\mathbf{e}_i$, where $\mathbf{x}_i \in X$ and $\mathbf{e}_i$ is a standard basis element of $\mathbb{R}^n$. So we have that, see (2.3),

$$Y(X \otimes X)^{-1}(X \otimes \mathbf{x}) \quad = \quad Y\mathbf{y}. \tag{3.2}$$

The output of our mapping is simply a linear combination of the columns of $Y$ where the particular combination depends on how we define the $\otimes$ operation.

The $\otimes$ operation should ideally emphasis those variations in $\mathbf{x}$ that lead to variations in $\mathbf{f}(\mathbf{x})$. An extreme case is given by the function

$$f(x, y) \quad = \quad x^2. \tag{3.3}$$

We want to design a $\otimes$ that only depends on variations in $x$. For example, we could use

$$\|\mathbf{x}\| \quad = \quad |x| \tag{3.4}$$

in (2.2) instead of $\|\mathbf{x}\|_2$.

Now, let us specialize our procedure to the case where $(\mathbf{x} \otimes \mathbf{y}) = d(\|\mathbf{x} - \mathbf{y}\|_A)$, where

$$\|\mathbf{x} - \mathbf{y}\|_A^2 \quad = \quad (\mathbf{x} - \mathbf{y})^T A(\mathbf{x} - \mathbf{y}) \tag{3.5}$$

and, $d(x)$ is some appropriately chosen function, e.g., $d(x) = \exp(-x)$. We are splitting the determination of $\otimes$ into two steps. First, the matrix $A$ is picking our those variations of $\mathbf{x}$ that are important for changes in our function $\mathbf{f}(\mathbf{x})$. The function $d(x)$ is determining what length scale is important. So, if $d(x)$ is highly peeked around $x = 0$, only nearby neighbors are important for determining the output of $\widehat{\mathbf{f}}(\mathbf{x})$, see (2.3). A flatter $d(x)$ allows further neighbors to have a greater influence in determining the value of $\widehat{\mathbf{f}}(\mathbf{x})$. We could even have the situation where, e.g.,

$$d(x) \quad = \quad \begin{cases} d_1(x) & , 0 \leq x \leq 1 \\ d_2(x) & , x > 1 \end{cases} \tag{3.6}$$

and/or, $A$ is dependent on the separation of $\mathbf{x}$ and $\mathbf{y}$ i.e., $A = A(\|\mathbf{x} - \mathbf{y}\|_2)$.

# 4 Some examples

Our first example is using NSET to approximate the absolute value function $f(x) = |x|$. Here we let

$$(X \otimes X)_{ij} \quad = \quad \exp(-|x_i - x_j|). \tag{4.1}$$

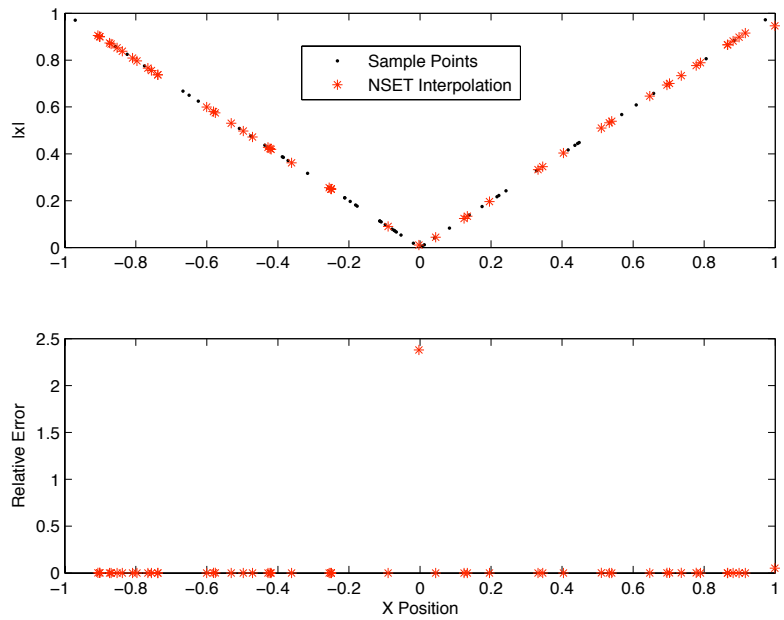


Figure 1: *The $|x|$ NSET interpolation example.*

We randomly choose $50$ points uniformly distributed in $[-1, 1]$ as our sample points. Then $20$ new points were also randomly chosen in $[-1, 1]$. We ran these points through our NSET operator to find the interpolation. The results are shown in Figure 1. The average relative error was $4.94\%$. The average absolute error was $.0017$.

Our next example will use NSET to approximate the $\sin^{-1}(x)$ function. We still use (4.1) as our definition of $\otimes$. Here we took $50$ randomly chosen points uniformly distributed on $[-\pi/2, \pi/2]$. These provided the $Y$ for (2.3a). The $X$ was given by the sine of these points. To test the NSET operator, $20$ points where randomly chosen from $[-\pi/2, \pi/2]$. The sine of these points were the input to the NSET operator. The average relative error was $.51\%$ and, the average absolute error was $.0071$. The results are shown in Figure 2.

Our final example will be placing an approximate coordinate chart on a manifold. As above, we took $50$ randomly chosen points uniformly distributed on $[-\pi/2, \pi/2]$. These will be the $X$ matrix for (2.3a) using (4.1) and will place local coordinates on the unit circle. Our $Y$ matrix will consist of the points in $\mathbb{R}^2$ given by $[\ \sin(x)\ \cos(x)\ ]^T$. We then choose $20$ random points from $[-\pi/2, \pi/2]$ and, used NSET to approximate the mapping to the unit circle. The results are shown in Figure 3. The average relative error of the mapping was $.39\%$.

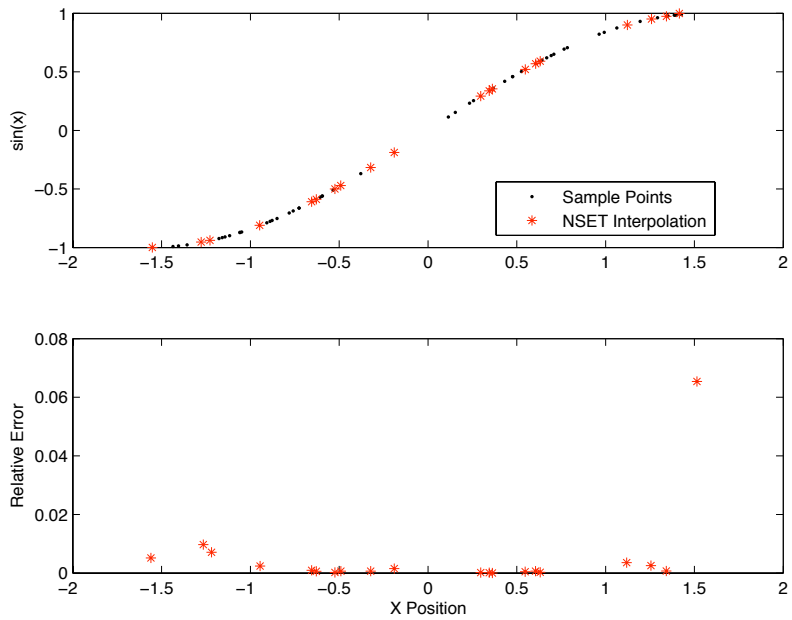


Figure 2: *The* $\sin^{-1}(x)$ *NSET interpolation example.*

# 5 Discussion

We've demonstrated that M/NSET can be usefully employed for approximating implicit and inverse functions. This extension of the standard M/NSET is obtained by simply allowing the image of our mapping to differ from the range, compare (2.3a) with (2.1a). In this way we are no longer concerned with trying to approximate the identity mapping on our set of interest. We have also seen that the extended M/NSET can be used to place local coordinate charts on manifolds.

We imagine that the charting procedure outlined here will be useful for implementing the Lipschitz implicit and inverse function theorems as needed by, e.g., the direct search methods in [2]. Here the M/NSET variation will provide a surrogate model for the implicit mapping from the tangent space of a manifold to the manifold. These initial guesses can then be refined by numerically implementing a Lipschitz implicit function theorem. As our sampling of the manifold becomes better, we can update the $X$ and $Y$ matrices to have a better surrogate for the implicit function.

The M/NSET variation could also be useful for more general surrogate models [5]. We would be able to easily update our surrogate when more information about the true function becomes available. This update could be done by adding/deleting columns to/from $X$ and $Y$ and/or, modifying the definition of $\otimes$.

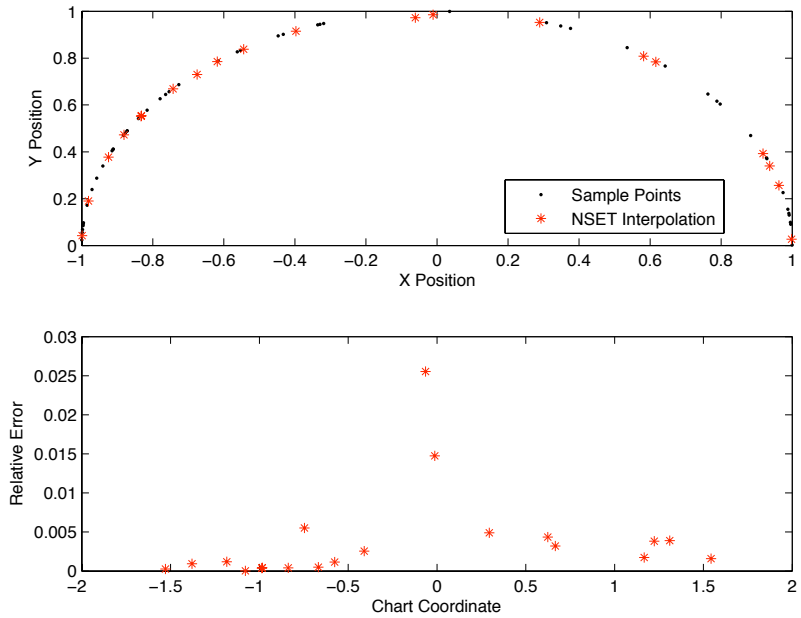


Figure 3: *The unit circle NSET charting example.*

# References


[1] C. L. Black, R. E. Uhrig, and J. W. Hines. System modeling and instrument calibration verification with a nonlinear state estimation technique. In *Proceedings of the Maintenance and Reliability Conference*, 1998.

[2] D. W. Dreisigmeyer. Direct search methods over Lipschitz manifolds. *Submitted to SIOPT*.

[3] G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins, 3rd edition, 1996.

[4] K. C. Gross, R. M. Singer, S. W. Wegericj, J. P. Herzog, R. van Alstine, and F. Bockhorst. Application of a model-based fault detection system to nuclear plant signals. In *Proceedings of the 9th International Conference on Intelligent Systems Application to Power Systems*, 1997.

[5] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker. Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, 41:1–28, 2005.