

Constrained Hermite TLS for Mesh-free Derivative Estimation Near and On Boundaries

Robert A. McDonald*
and Alejandro Ramos†

California Polytechnic State University, San Luis Obispo, CA, 93407

A Taylor Series Least Squares approach to estimating derivatives on scattered data is extended to include derivative observation or specification. This approach allows improved estimation of derivatives near boundaries as compared to the standard ghost node approach. It also enables a unique method of estimating derivatives in a surface using only surface data.

Nomenclature

$(\cdot)_0$	Quantity at point of interest
$(\cdot)_i$	Quantity at point i
$(\cdot)_{con}$	Elements of (\cdot) which correspond to constraint equation
$(\cdot)_{lsq}$	Elements of (\cdot) which correspond to least squares minimization
$\Delta(\cdot)_i$	Change in quantity (\cdot) from point of interest at point i
Δn	Spacing between points used in ghost node extrapolation
ϵ_1, ϵ_2	Combined measure of error for first and second derivative predictions
ϵ_{C_p}	Error in pressure coefficient
ϵ_{h_0}	Measure of error for directional derivative prediction
A	Taylor series coefficient matrix for function observations: $A_{i,k}$
B	Taylor series coefficient matrix for derivative observations: $B_{i,k}$
C	TLS Solution coefficient matrix
D, E	HTLS Solution coefficient matrices
F, G	CHTLS solution coefficient matrices (with $\{H\}$)
Q	Orthogonal matrix from QR decomposition
q	Submatrix of Q
R	Upper-triangular matrix from QR decomposition
v	Direction unit vector: $v_j; j \in [1, N]$
V_{inf}	Freestream velocity
W	Combined weight matrix
W_b	Derivative observation weight matrix: $W_{b,i,j} \equiv w_i \delta_{i,j}$
W_f	Function observation weight matrix: $W_{f,i,j} \equiv w_i \delta_{i,j}$
x	Vector of input quantities to f : $x_j; j \in [1, N]$
θ, ϕ	Spherical polar coordinates
$\{d\}$	Vector of Taylor series derivatives: $d_k; k \in [1, t]$
$\{f\}$	Vector of function difference observations: $\Delta f_i; i \in [1, n_f]$
$\{H\}$	CHTLS solution coefficient vector (with F and G)
$\{h\}$	Vector of directional derivative observations: $h_i; i \in [1, n_b]$
$\{V_0\}$	Direction derivative \mathbf{v}_0 rearranged to order corresponding to $\{d\}$
a	Index of differentiation variable
a, b, c	Ellipsoid axis lengths

*Associate Professor, Aerospace Engineering, One Grand Avenue, AIAA Senior Member.

†Graduate Research Assistant, Aerospace Engineering, One Grand Avenue, AIAA Student Member.

f	Function to be approximated by Taylor series
f_d	Function value in domain used in ghost node extrapolation
f_g	Extrapolated ghost node function value
f_s	Function value on surface used in ghost node extrapolation
k	Derivative vector index
l_{nom}	Nominal length
m_1	Derivative exponent and summation index series first instance: $m_1 \equiv q - \sum_{j=2}^N m_j$
m_j	Derivative exponents and summation index series: $m_j; j \in [2, N]$
$m_{j,k}$	Derivative exponent for vector of derivatives
N	Number of input variables to f
n_b	Number of directional derivative observations
n_f	Number of function observations
o	Number of constraints in constrained least squares problem: $o = 1$
q	Derivative exponent and summation index and limit
q_k	Derivative exponent for vector of derivatives
R	Non-zero element of \mathbf{R}
r	Order of derivatives included in truncated Taylor series
S	Sum of weighted squared residuals
s	Number of terms in truncated Taylor series: $s = \binom{r+N}{N}$
t	Number of derivatives in truncated Taylor series: $t = s - 1$
w_i	Residual weight: $w_i \equiv \left(\sum_{j=1}^N \Delta x_{j,i}^2 \right)^{-p/2}$
$\delta_{i,j}$	Kronecker delta

I. Introduction

THE Taylor series least squares (TLS) method has been used recently to estimate derivatives when no regular mesh is available.¹⁻³ This includes unstructured meshes and meshfree applications. Although this derivative estimation is useful in its own right, the TLS method has also been applied to the solution of partial differential equations (PDE's) in the same way that finite difference estimations of derivatives are used to solve PDE's in the finite difference method.

When the TLS method has previously been applied to aerospace problems, the derivative estimates near the boundaries of the domain have presented challenges. The usual solution to this problem has been to introduce a set of "ghost nodes" outside of the domain. The "ghost nodes" are assigned a function value which will imply a desired boundary condition at the wall. For example, to enforce a zero velocity condition at the wall, one would assign each "ghost node" a velocity equal and opposite its in-domain counterpart.⁴⁻⁹

In this research, the TLS method has been extended to allow the use in a least squares sense of a Neumann boundary condition in the derivative estimate. We call this the Hermite Taylor Least Squares method or HTLS. The HTLS method has been further extended to allow the exact specification of a Neumann condition at the point of interest. We call this the Constrained Hermite Taylor Least Squares method or CHTLS. The CHTLS method has already improved the ability of the generalized finite difference method to model compressible potential flows.¹⁰ The CHTLS also enables the calculation of derivatives on the surface of a body immersed in a domain – using only information on the surface. It is envisioned that this technique will be used to calculate the surface velocities and pressure coefficient for a low-order unstructured panel code.

II. Taylor Series Least Squares

The Taylor series of a function, f , of N variables, truncated to include derivatives up to order r , may be written as in Equation 1, where \mathbf{x} is the vector of input quantities $x_j; j \in [1, N]$. There are $s = \binom{r+N}{N}$ terms in this truncated series.

$$f(\mathbf{x} + \Delta\mathbf{x}) = \sum_{q=0}^r \sum_{m_2=0}^q \sum_{m_3=0}^{(q-m_2)} \cdots \sum_{m_N=0}^{(q-\sum_{j=2}^{N-1} m_j)} \left(\frac{\partial^q f}{\partial x_1^{m_1} \cdots \partial x_N^{m_N}} \prod_{j=1}^N \frac{\Delta x_j^{m_j}}{m_j!} \right) \quad (1)$$

In this form, the quantity m_1 is defined $m_1 \equiv q - \sum_{j=2}^N m_j$; this definition in conjunction with the limits

in the N nested sums ensure that the derivatives are of consistent order. In practice, a general purpose implementation for N -dimensional problems would generate the terms in the series through a recursive algorithm rather than the N nested for-loops implied by the repeated sums in Equation 1.

Note that the j index is used twice in Equation 1 and its use is implied in the $N - 4$ omitted sums implied by the ellipsis. These indices are entirely independent and their scope is limited by the sum or product operator which introduces them.

The $q = 0$ case of $\partial^q f$ is interpreted as f , thus providing the central function value in the series; this also relies on the facts that $\partial x_j^0 = 1$, $\Delta x_j^0 = 1$, and $0! = 1$. When the $q = 0$ term is removed from the series and replaced with f_0 , there remain $t = s - 1$ terms in the series.

The t Taylor series derivatives are collected into an $t \times 1$ column vector, $\{d\}$, with entries d_k ; $k \in [1, t]$. The sequence of the derivatives in $\{d\}$ is determined by the set of exponents $q_k, m_{j,k}$; while any sequence may be selected, once selected care must be taken to maintain consistency.

$$d_k = \frac{\partial^{(q_k)} f}{\partial x_1^{(m_{1,k})} \dots \partial x_N^{(m_{N,k})}} \quad (2)$$

The most straightforward approach to sequencing the series derivatives for a problem of fixed dimensionality involves a series of N nested for-loops corresponding to the nested sums which appear in Equation 1. The k index is initialized before the loops execute and is incremented inside the innermost loop, where $m_{1,k} = q_k - \sum_{j=2}^N m_{j,k}$ is also calculated. The recursive approach mentioned earlier may be used to generate the coefficients for the general N -dimensional problem. Using the sequenced vector of derivatives, the truncated Taylor series may be rewritten as a single summation.

$$f(\mathbf{x} + \Delta \mathbf{x}) = f_0 + \sum_{k=1}^t d_k \prod_{j=1}^N \frac{\Delta x_j^{(m_{j,k})}}{m_{j,k}!} \quad (3)$$

The TLS approach to estimating derivatives uses a set of n_f function observations f_i ; $i \in [1, n_f]$ in the neighborhood of a point of interest f_0 . By subtracting the point of interest function observation, and applying Equation 3 to the set of observations, we attain the following equation, where $\Delta(\cdot)_i \equiv (\cdot)_i - (\cdot)_0$.

$$\Delta f_i = \sum_{k=1}^t d_k \underbrace{\prod_{j=1}^N \frac{\Delta x_{j,i}^{(m_{j,k})}}{m_{j,k}!}}_{A_{i,k}} \quad (4)$$

By defining vector or matrix equivalents to terms from Equation 4, it may be written concisely in the matrix form given by Equation 5. The $n_f \times 1$ column vector $\{f\}$ of function difference observations Δf_i must be defined as well as the $n_f \times t$ matrix \mathbf{A} with components $A_{i,k}$ as indicated in Equation 4.

$$\mathbf{A}\{d\} = \{f\} \quad (5)$$

When $n_f > t$, Equation 5 corresponds to an overdetermined system which will be solved in the least squares sense. The least squares problem minimizes S , the sum of the weighted and squared residuals, where $S \equiv \sum_{i=1}^{n_f} w_i (A_{i,k} d_k - \Delta f_i)^2$. S is minimized by choosing values for a set of variables; this is done by differentiating R with respect to each of the variables and then solving for the values which simultaneously zero those derivatives. In this case, the variables are actually d_k , the derivatives which appear in the truncated Taylor series. Rather than follow these defining steps to the solution, this exposition follows the equivalent process frequently presented in linear algebra texts.

The weights matrix \mathbf{W}_f is defined as a matrix with the weights, w_i , calculated from the locations, $\Delta \mathbf{x}_i$, of the n_f function observations along the diagonal i.e. $W_{f,i,j} \equiv w_i \delta_{i,j}$, where $\delta_{i,j}$ is the Kronecker delta. The weight, $w_i \equiv \prod_{j=1}^N \Delta x_{j,i}^{-2p/2}$, is an inverse distance weighting function used to improve problem conditioning. The parameter p controls the decay of the weighting with distance; p must satisfy $p \geq 0$ and was chosen to be 1 for this study. The weights are applied by pre-multiplying both sides of Equation 5 by the weight matrix.

$$\mathbf{W}_f \mathbf{A}\{d\} = \mathbf{W}_f \{f\}$$

Both sides of the equation are then pre-multiplied by \mathbf{A}^T . The matrix $\mathbf{A}^T \mathbf{W}_f \mathbf{A}$ is of size $t \times t$ and can be inverted through conventional techniques, allowing the solution of the system as given in Equation 6.

$$\{d\} = \underbrace{[\mathbf{A}^T \mathbf{W}_f \mathbf{A}]^{-1} \mathbf{A}^T \mathbf{W}_f \{f\}}_{\mathbf{C}} \quad (6)$$

In an iterative solution problem, it is useful to define the $t \times n_f$ matrix \mathbf{C} as indicated in Equation 6; this allows the equation to be written $\{d\} = \mathbf{C}\{f\}$. The matrix \mathbf{C} is only a function of the spacing of the n_f points in the neighborhood of the point of interest and the number of terms retained in the truncated Taylor series. Consequently, it can be calculated once and stored at the beginning of a solution process. At each subsequent iteration, \mathbf{C} may be recalled, multiplied by the current function difference observations $\{f\}$, resulting in a new estimate of the local partial derivatives $\{d\}$.

III. Hermite Taylor Series Least Squares

When the point of interest is near a region where information about the derivative of the function is known, that information can be used to help improve the estimate of the derivatives. In fluid dynamic applications, this situation frequently arises when the point of interest is near a domain boundary and the physics of the problem determine the behavior of the derivative at the boundary.

The truncated Taylor series of the first derivative of f with respect to x_a , may be written as follows. There are t terms in this series. As with Equation 1, m_1 is defined $m_1 \equiv q - \sum_{j=2}^N m_j$.

$$\frac{\partial f}{\partial x_a} = \sum_{q=1}^r \sum_{m_2=0}^q \sum_{m_3=0}^{(q-m_2)} \cdots \sum_{m_N=0}^{(q-\sum_{j=2}^{N-1} m_j)} \frac{\partial^q f}{\partial x_1^{m_1} \cdots \partial x_N^{m_N}} m_a \prod_{j=1}^N \frac{\Delta x_j^{(m_j - \delta_{ja})}}{m_j!}$$

This series may also be written in terms of d_k , the sequenced set of derivatives defined in Equation 2.

$$\frac{\partial f}{\partial x_a} = \sum_{k=1}^t d_k m_{a,k} \prod_{j=1}^N \frac{\Delta x_j^{(m_{j,k} - \delta_{ja})}}{m_{j,k}!} \quad (7)$$

The Hermite Taylor series least squares (HTLS) approach to estimating derivatives augments the TLS approach with a set of n_b directional derivative observations h_i ; $i \in [1, n_b]$ also in the neighborhood of a point of interest. Matching the directional derivative allows imposition of a single scalar condition rather than the N conditions required to match each derivative component, but also requires specification of the direction unit vector for each derivative observation. The direction unit vectors are specified as \mathbf{v}_i ; $i \in [1, n_b]$, where each \mathbf{v}_i is a vector of components $v_{i,j}$; $j \in [1, N]$.

As expected, the directional derivative is the dot product of the direction unit vector with the gradient $h = \mathbf{v} \cdot \nabla f$. In many fluid dynamic problems, \mathbf{v}_i will be a surface normal vector. Furthermore, when used to enforce Neumann boundary conditions in fluid dynamic problems, the directional derivative observation h_i will often be zero. In Equation 8 below for h_i , the sum over k has been brought outside the dot product represented by the sum over a .

$$h_i = \sum_{k=1}^t \left[\underbrace{d_k}_{B_{i,k}} \sum_{a=1}^N v_{i,a} m_{a,k} \prod_{j=1}^N \frac{\Delta x_{j,i}^{(m_{j,k} - \delta_{ja})}}{m_{j,k}!} \right] \quad (8)$$

By defining vector or matrix equivalents to terms from Equation 8, it may be written concisely in the matrix form given by Equation 9. The $n_b \times 1$ column vector $\{h\}$ to hold the directional derivative observations h_i must be defined as well as the $n_b \times t$ matrix \mathbf{B} with components $B_{i,k}$ as indicated in Equation 8.

$$\mathbf{B}\{d\} = \{h\} \quad (9)$$

Equations 5 and 9 may be combined by vertically concatenating matrices \mathbf{A} and \mathbf{B} and vectors $\{f\}$ and $\{h\}$. The combined system given as Equation 10 will be solved in the least squares sense to find the set of Taylor series derivatives $\{d\}$ which best matches both the function and directional derivative observations contained in $\{f\}$ and $\{h\}$.

$$\begin{matrix} (n_f+n_b) \times t & & (n_f+n_b) \times 1 \\ \left[\begin{array}{c} \mathbf{A} \\ \mathbf{B} \end{array} \right] & \{d\} = & \left\{ \begin{array}{c} f \\ h \end{array} \right\} \end{matrix} \quad (10)$$

The same process for developing the least squares solution which lead to Equation 6 is applied to Equation 10. The weights matrix \mathbf{W} is constructed by placing the weight values for both the function and directional derivative observations along the diagonal. This may also be interpreted as placing the individual weights matrices \mathbf{W}_f and \mathbf{W}_b along the diagonal as shown below; by construction, the combined weight matrix \mathbf{W} is $(n_f + n_b) \times (n_f + n_b)$.

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_f & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_b \end{bmatrix}$$

Each side of Equation 10 is then pre-multiplied by the combined weights matrix \mathbf{W} and then the transpose of the combined coefficient matrix $\mathbf{A}^T \mathbf{B}^T$. These operations result in a system which can be solved for $\{d\}$ while satisfying both the function and derivative observations in a least squares sense.

$$\{d\} = \left[\mathbf{A}^T \mathbf{B}^T \mathbf{W} \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \right]^{-1} \mathbf{A}^T \mathbf{B}^T \mathbf{W} \left\{ \begin{array}{c} f \\ h \end{array} \right\}$$

Alternatively, considerable simplification may be achieved by exploiting the sparse structure of \mathbf{W} to re-arrange this equation into separate terms for the function and derivative observations. The combined least squares solution is presented as Equation 11.

$$\{d\} = \underbrace{\mathbf{A}^T \mathbf{W}_f \mathbf{A} + \mathbf{B}^T \mathbf{W}_b \mathbf{B}^{-1} \mathbf{A}^T \mathbf{W}_f}_{\mathbf{D}} \{f\} + \underbrace{\mathbf{A}^T \mathbf{W}_f \mathbf{A} + \mathbf{B}^T \mathbf{W}_b \mathbf{B}^{-1} \mathbf{B}^T \mathbf{W}_b}_{\mathbf{E}} \{h\} \quad (11)$$

When using HTLS to estimate derivatives in an iterative solution procedure, the $t \times n_f$ matrix \mathbf{D} and the $t \times n_b$ matrix \mathbf{E} defined in Equation 11 may be calculated once and stored for later use. These matrices are a function only of the point spacing for the function and directional derivative observations, the order of the truncated series, and the direction vectors specified for the derivative observations.

As discussed at the beginning of this section, in many fluid dynamic problems, the directional derivative will be specified as zero at a boundary and the $\{h\}$ vector will vanish, allowing the second term to be dropped. Note that the zero derivative specification still has an influence on the solution because of the \mathbf{B} and \mathbf{W}_b matrices present in the calculation of \mathbf{D} .

IV. Constrained Hermite Taylor Series Least Squares

The constrained Hermite Taylor series least squares (CHTLS) approach to estimating derivatives augments the HTLS approach with a constraint to exactly match the specified directional derivative at the point of interest. In fluid dynamic applications, this situation frequently arises when the point of interest is on a domain boundary and the physics of the problem determine the behavior of the derivative at the boundary.

Matching the directional derivative only at the point of interest provides a single, scalar constraint on the least squares problem. The directional derivative observation at the point of interest is designated h_0 . As before, specification of a directional derivative allows the treatment of a scalar quantity instead of a vector of N quantities. However, this requires the specification of the direction for the directional derivative. The direction unit vector at the point of interest is specified as \mathbf{v}_0 , a vector of components $v_{0,j}$; $j \in [1, N]$.

Recall that the directional derivative at a point is the dot product of the direction unit vector with the gradient $h = \mathbf{v} \cdot \nabla f$. At the point of interest, the first derivatives from the Taylor series may be used directly as ∇f . This leads to a straightforward equation for the directional derivative at the point of interest.

$$h_0 = \sum_{a=1}^N v_{0,a} \frac{\partial f}{\partial x_a}$$

The components of \mathbf{v}_0 are collected into the $t \times 1$ column vector, $\{V_0\}$ in the positions corresponding to the sequence of the derivatives in $\{d\}$ defined in Equation 2. All of the components of $\{V_0\}$ which correspond

to a higher derivative are zero. Definition of $\{V_0\}$ allows the equation for h_0 to be re-written in vector form as Equation 12.

$$\{V_0\}^T \{d\} = h_0 \quad (12)$$

Equation 12 is the single linear constraint to be enforced on the least squares minimization problem from the previous section. In a general constrained least squares problem with o constraints, the QR decomposition of the transpose of the constraint equation coefficient matrix is found. From the decomposition, \mathbf{Q} is a $t \times t$ orthogonal matrix and \mathbf{R} is a $t \times o$ upper-triangular matrix. In this problem, the decomposition results from $\{V_0\} = \mathbf{Q}\mathbf{R}$, where there is only a single constraint, i.e. $o = 1$. Consequently, there is only a single non-zero element of \mathbf{R} ; this element is identified as R .

The orthogonal \mathbf{Q} matrix is used to transform the unknown vector $\{d\}$ into an alternate set of unknowns. These unknowns are then partitioned as shown by Equation 13 according to whether they are determined by the constraint equation or the least squares minimization. In general, $\{d_{con}\}$ will be $o \times 1$ and $\{d_{lsq}\}$ will be $(t - o) \times 1$; in this problem, d_{con} is a scalar.

$$\{d\} = \mathbf{Q} \begin{matrix} d_{con} \\ d_{lsq} \end{matrix} \quad (13)$$

The QR decomposition is substituted for $\{V_0\}$ into Equation 12 and the variable transformation is applied to replace $\mathbf{Q}^T \{d\}$ with d_{con} ; because of the sparsity of \mathbf{R} , the $\{d_{lsq}\}$ rows are eliminated. In this case of a single constraint, a scalar equation results which will be solved for d_{con} .

$$R d_{con} = h_0 \quad (14)$$

The overdetermined system given by Equation 10 is transformed into the new set of variables to be solved in the least squares sense. The orthogonality of \mathbf{Q} leads to the fact that $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}$; this is inserted into Equation 10 and $\mathbf{Q}^T \{d\}$ is recognized as the transformed variables.

$$\begin{matrix} \mathbf{A} \\ \mathbf{B} \end{matrix} \mathbf{Q} \begin{matrix} d_{con} \\ d_{lsq} \end{matrix} = \begin{matrix} f \\ h \end{matrix}$$

The coefficient matrix from this equation may be partitioned further into $\{A_{con}\}$ and $\{B_{con}\}$, the column which corresponds to the d_{con} equation and, \mathbf{A}_{lsq} and \mathbf{B}_{lsq} the columns which correspond to the $\{d_{lsq}\}$ equations as shown below. Note that in this problem, $\{A_{con}\}$ is a $n_f \times 1$ column vector, \mathbf{A}_{lsq} is a $n_f \times (t - 1)$ matrix, $\{B_{con}\}$ is a $n_b \times 1$ column vector, and \mathbf{B}_{lsq} is a $n_b \times (t - 1)$ matrix.

$$\begin{matrix} \mathbf{A} \\ \mathbf{B} \end{matrix} \mathbf{Q} = \begin{matrix} \{A_{con}\} & \mathbf{A}_{lsq} \\ \{B_{con}\} & \mathbf{B}_{lsq} \end{matrix}$$

Partitioning the coefficient matrix allows the system to be re-arranged in a way which separates the d_{con} equations from the $\{d_{lsq}\}$ equations.

$$\begin{matrix} A_{con} \\ B_{con} \end{matrix} d_{con} + \begin{matrix} \mathbf{A}_{lsq} \\ \mathbf{B}_{lsq} \end{matrix} \{d_{lsq}\} = \begin{matrix} f \\ h \end{matrix}$$

Equation 14 is used to determine d_{con} , which allows the first term of this equation to be brought to the right hand side as a known quantity. This manipulation provides the overdetermined system to be solved in a least squares sense for $\{d_{lsq}\}$.

$$\begin{matrix} \mathbf{A}_{lsq} \\ \mathbf{B}_{lsq} \end{matrix} \{d_{lsq}\} = \begin{matrix} f - A_{con} d_{con} \\ h - B_{con} d_{con} \end{matrix}$$

The same process which was used to develop Equation 11 is applied to this system. Both sides of the system are pre-multiplied by $[\mathbf{A}_{lsq}^T \mathbf{B}_{lsq}^T] \mathbf{W}$. Furthermore, the sparse structure of \mathbf{W} was exploited to separate terms for the function and derivative observations. This leads to the following equation for the least squares solution of $\{d_{lsq}\}$; this equation also provides definitions for matrices \mathbf{F}' and \mathbf{G}' .

$$\{d_{lsq}\} = \frac{\left[\mathbf{A}_{lsq}^T \mathbf{W}_f \mathbf{A}_{lsq} + \mathbf{B}_{lsq}^T \mathbf{W}_b \mathbf{B}_{lsq} \right]^{-1} \mathbf{A}_{lsq}^T \mathbf{W}_f \{f - A_{con} d_{con}\} +}{\mathbf{F}'} \frac{\left[\mathbf{A}_{lsq}^T \mathbf{W}_f \mathbf{A}_{lsq} + \mathbf{B}_{lsq}^T \mathbf{W}_b \mathbf{B}_{lsq} \right]^{-1} \mathbf{B}_{lsq}^T \mathbf{W}_b \{h - B_{con} d_{con}\}}{\mathbf{G}'}$$

Matrices \mathbf{F}' and \mathbf{G}' can be pulled through the vectors at the right, which results in Equation 15 and leads to the definition of $\{H'\}$. Note that the negation of the third term becomes part of the definition of $\{H'\}$.

$$\{d_{lsq}\} = \mathbf{F}' \{f\} + \mathbf{G}' \{h\} - \underbrace{\{\mathbf{F}' A_{con} + \mathbf{G}' B_{con}\}}_{\{H'\}} d_{con} \quad (15)$$

Equation 15 for $\{d_{lsq}\}$ and Equation 14 for d_{con} represent the complete solution to the CHTLS problem. However, this solution is still in terms of the transformed variables and must be transformed back to the original variables through Equation 13. Straightforward substitution of Equation 15 into Equation 13 results in the following equation for the solution in the original variables.

$$\{d\} = \mathbf{Q} \left(\mathbf{F}' \{f\} + \mathbf{G}' \{h\} + \{H'\} d_{con} \right)$$

This equation can also be re-arranged into a form appropriate for an iterative solution procedure; this is done by breaking the substituted expression for $\{d_{lsq}\}$ into separate terms. The \mathbf{F}' matrix is augmented by placing a $1 \times n_f$ row vector of zeros above its first row. Similarly, the \mathbf{G}' matrix is augmented with a $1 \times n_b$ row vector of zeros. The $\{H'\}$ column vector is augmented with a single 1 entry before the first entry. Equation 14 is solved for d_{con} allowing h_0/R to be substituted in its place. Finally, the \mathbf{Q} matrix is multiplied through each of the terms.

$$\{d\} = \mathbf{Q} \underbrace{\begin{matrix} \mathbf{0} \\ \mathbf{F}' \end{matrix}}_{\mathbf{F}} \{f\} + \mathbf{Q} \underbrace{\begin{matrix} \mathbf{0} \\ \mathbf{G}' \end{matrix}}_{\mathbf{G}} \{h\} + \frac{1}{R} \mathbf{Q} \underbrace{\begin{matrix} 1 \\ H' \end{matrix}}_{\{H\}} h_0 \quad (16)$$

Equation 16 presents a convenient form for the repeated estimation of derivatives through the CHTLS method. The indicated definitions for \mathbf{F} , \mathbf{G} , and $\{H\}$ allow the solution to be written compactly as $\{d\} = \mathbf{F}\{f\} + \mathbf{G}\{h\} + \{H\}h_0$. In the situation where the directional derivative observations surrounding the point of interest and the directional derivative at the point of interest are zero, there is no need to construct or store \mathbf{G} and $\{H\}$. However, the directional derivatives maintain significant impact on the derivative estimates through their role in the construction of \mathbf{F} .

V. Interpretation of QR Decomposition

The given problem of a single constraint which only affects a handful of the variables allows a simplified treatment of the QR decomposition as well as an enhanced physical interpretation of the variable transformation. Recall that the $\{V_0\}$ matrix is constructed by placing the N elements of the direction unit vector at the point of interest into a vector according to the positions of the corresponding first derivatives in the $\{d\}$ vector. For simplicity of discussion, it is safe to assume that the N first derivatives occupy the first N elements of $\{d\}$ and consequently, the first N elements of $\{V_0\}$ correspond to the direction unit vector at the point of interest.

The QR decomposition must satisfy the following equation.

$$\{V_0\} = \mathbf{Q}\mathbf{R} \quad (17)$$

Because the LHS is a vector, we know that \mathbf{R} has only one non-zero element. We can choose $R = 1$ for that entry. Because only the first N entries in $\{V_0\}$ are non-zero, we only need to concern ourselves with the

upper-left $N \times N$ submatrix of \mathbf{Q} . The remaining entries are zero, except along the diagonal, where they are equal to one. This $N \times N$ submatrix is orthogonal and will be referred to as \mathbf{q} .

$$\mathbf{Q} = \begin{pmatrix} \mathbf{q} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \quad (18)$$

We are left with the following equation to determine \mathbf{q} .

$$\mathbf{v}_0 = \mathbf{q} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad (19)$$

From this, we observe that the first column of \mathbf{q} must equal \mathbf{v}_0 , and that the remaining columns may be chosen in any way that preserves orthogonality. In the common case of a problem where the input quantities correspond to spatial dimensions, this is a straightforward process.

Recall Equation 14 where the transformed variable is solved for in terms of the applied constraint value. With R chosen to be one, the transformed variable is the applied constraint value. We are lead to interpret the transformation of variables as a transformation of the first derivatives in the N problem dimensions into directional derivatives. The first unknown, d_{con} is the directional derivative in the specified direction, while the first $N - 1$ entries of d_{lsq} are directional derivatives which complete the orthogonal basis set.

VI. Results

The TLS, HTLS, and CHTLS methods have been applied to a number of test problems used to demonstrate the effectiveness of the formulation and to verify the correctness of this implementation. These methods have also been used by the authors to solve the full potential equation for subcritical compressible flows.¹⁰ The authors have also conducted numerous other tests varying point distribution, order of accuracy, test function, etc. These tests have revealed that these methods are generally robust and well behaved.

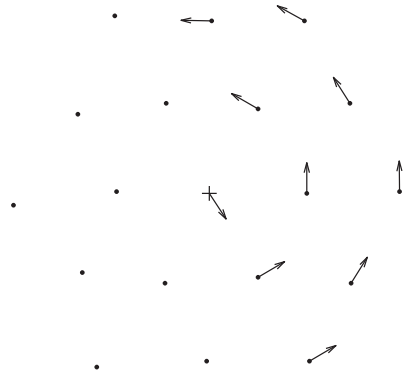
In the first set of tests, the methods were used to estimate the derivatives of a two dimensional test function, $f = \sin x_1^2 + x_2^3$, centered at $\mathbf{x}_0 = [0.24, 1.23]$. This function was chosen because it is bounded, non-symmetric, and it is infinitely differentiable; further, its derivatives are bounded and generally nontrivial.

The cloud of 18 points surrounding \mathbf{x}_0 used to provide the supporting data is depicted in Figure 1(a). These points were extracted from a two dimensional unstructured mesh; although these points are well distributed, they are not perfectly spaced. The TLS test case utilized the function value at all 18 surrounding points. The HTLS test case discarded the function value at the nine right-hand points and instead specified the directional derivative at those points; the direction for the derivative observations was chosen to be radial about the point of interest. The CHTLS test case augmented the supporting data used by the HTLS case with specification of the directional derivative at the point of interest; the direction for the derivative specification was chosen to be down and to the right.

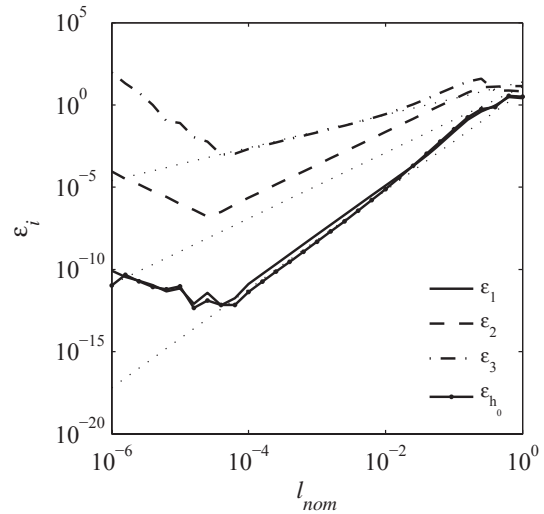
For these tests, the Taylor series were truncated to include first, second, and third derivatives, i.e. $r = 3$, resulting in $t = 9$ derivatives for a two dimensional function. The Matlab symbolic toolbox was used to programmatically obtain all analytical derivatives for comparison. Rather than report the error in all nine quantities separately, error in derivatives of common order were combined by calculating their RMS; consequently, ϵ_1 is a combined measure of error in the first derivative predictions, ϵ_2 for second derivatives, and so on. The error in the prediction of the directional derivative at the point of interest is also reported as ϵ_{h_0} ; because it is a quantity based on first derivatives, its behavior will generally parallel ϵ_1 .

Each method was used to estimate the derivatives of the test function at the point of interest using a progressively refined point cloud. The point cloud depicted in Figure 1(a) was scaled for each case such that the nominal distance between neighboring points is l_{nom} . The error metrics were calculated for each case and plotted against l_{nom} on a log-log chart. The results of this test of the TLS method were included as Figure 1(b). In addition to the error metrics, sloped guide lines representing first, second, and third order behavior were included on each of these charts.

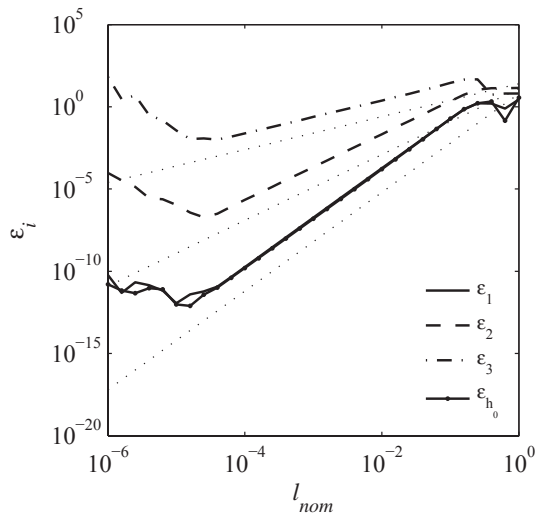
As revealed by Figure 1(b), the TLS method behaves largely as expected. Because third order derivatives were included in the truncated Taylor series, the prediction of first derivatives is third order accurate. Rounding error begins to dominate truncation error for $l_{nom} < 10^{-4}$. Each higher derivative prediction is



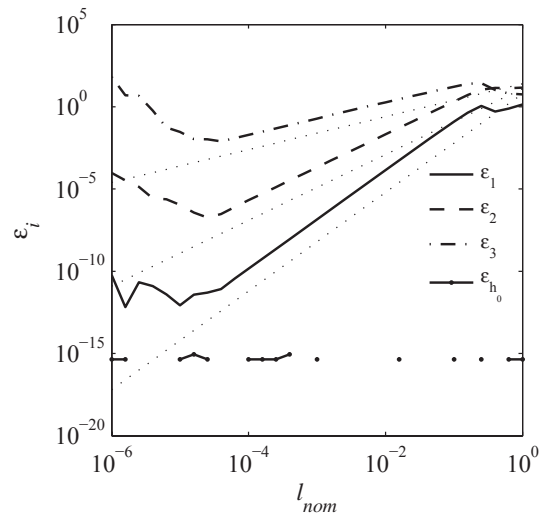
(a) Cloud of points which provided supporting data.



(b) Test of TLS method.



(c) Test of HTLS method.



(d) Test of CHTLS method..

Figure 1. Two dimensional tests using Taylor series including first, second, and third derivatives.

one order less accurate. The prediction of the directional derivative at the point of interest behaves similarly to the prediction of the first derivatives.

Figure 1(c) depicts the results of the HTLS method on this test problem. The HTLS method behaves like the TLS method in most ways, however it is somewhat less accurate in this case. This inaccuracy is not surprising because specifying values for the directional derivative provides less information about the behavior of the function than specifying the function directly. In usual practice, the directional derivative information would augment rather than replace the function value information. In this case, the behavior of the directional derivative at the point of interest is indistinguishable from the first derivative prediction.

Figure 1(d) depicts the behavior of the CHTLS method for the two dimensional test case. The CHTLS method behaves very much like the HTLS method with the exception of the error in the prediction of the directional derivative at the point of interest. The effect of the constraint can be plainly seen as the error in the directional derivative prediction is less than 10^{-15} for all values of l_{nom} . Note that the omission of some points in the ϵ_{h_0} curve is due to the fact that the plotting program can not represent a value exactly equal to zero on a log scale. In subsequent plots for the CHTLS method, the error in the directional derivative is too small in magnitude to appear on the selected scale.

VI.A. On-Boundary Results

A three dimensional test was performed to study the behavior of these methods when used to estimate derivatives when the point of interest lies on the boundary of a domain. The derivatives of a three dimensional function $f = \sin x_1^2 + x_2^3 + x_3^4$ were investigated at a point located on the surface of a unit radius sphere centered at the origin. The point of interest was located at $\theta = 25^\circ$, $\phi = 80^\circ$ in spherical polar coordinates. This test function shares the desirable properties of the two dimensional test function used earlier.

An unstructured grid generator was used to discretize a limited domain surrounding the point of interest on the surface of the sphere. The domain was limited to prevent the grids from growing extremely large as the nominal edge length, l_{nom} , was made extremely small. Figure 2(a) depicts a coarse ($l_{nom} = 0.2$), but otherwise typical mesh. Although a volume mesh is shown, the TLS approach only utilizes point location information; the point connectivity implied by the mesh is used only for visualization.

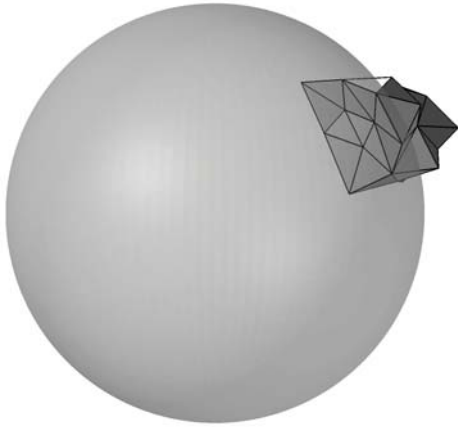
The grid generator was constrained to incorporate a seed triangle on the surface; the equilateral seed triangle was located with the point of interest at its centroid. Although the discretized domain was extended to a radius of eight edge lengths from the point of interest, most of those points were discarded. The tests were performed using only points which were a part of a tetrahedron which shared at least one point with the tetrahedron formed from the seed triangle. The grid generation process naturally lead to some variation in the point distribution and number of points used for each test. Consequently, these results are somewhat less regular than those resulting from the scaled point distributions of the other test cases.

For these tests, the Taylor series were once again truncated after the third derivatives, i.e. $r = 3$, resulting in $t = 19$ derivatives for a three dimensional function. As with the two dimensional tests, errors in derivatives of common order were combined by calculating their RMS.

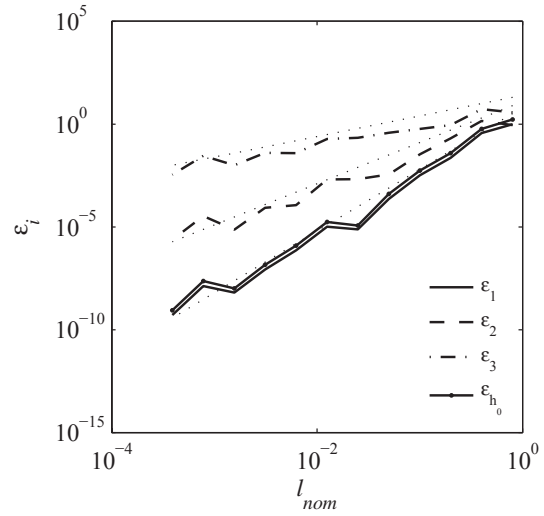
The behavior of the TLS method at predicting derivatives for this three dimensional test function is depicted in Figure 2(b). As expected, the combined error of the first derivative predictions generally follows the third order guide line and each higher order derivative is predicted with reduced accuracy. The results are somewhat less regular than the two dimensional results because of the variation introduced by the grid generation process. The results do not extend into the region dominated by rounding error because the grid generator could not produce grids when rounding error became dominant. The error in predicting the directional (surface normal) derivative at the point of interest is slightly greater than the combined first derivative error measure, but very closely parallels its behavior.

The directional derivative normal to the sphere at the surrounding surface points was included in the supporting data for the test of the HTLS method. The results reported in Figure 2(c) are generally similar to the results for the TLS method. However, the slight downward shift of each curve indicates that there is a general reduction in error. Furthermore, the results are somewhat less erratic indicating that the method is somewhat less sensitive to the grid. The error in predicting the directional derivative is now consistently less than the combined first derivative error measure, and does not follow its behavior as consistently.

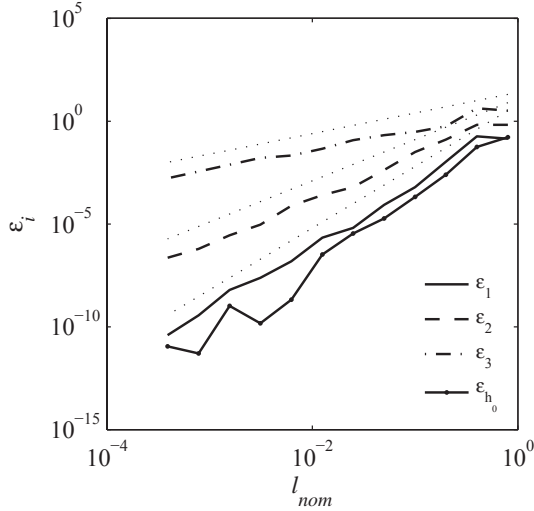
In Figure 2(d), the results of applying the CHTLS method to the three dimensional test function are reported. Constraint of the directional derivative at the point of interest was effective – error in its prediction was too small to be plotted on the selected scale. In all other regards, the results are nearly identical to those for the HTLS method.



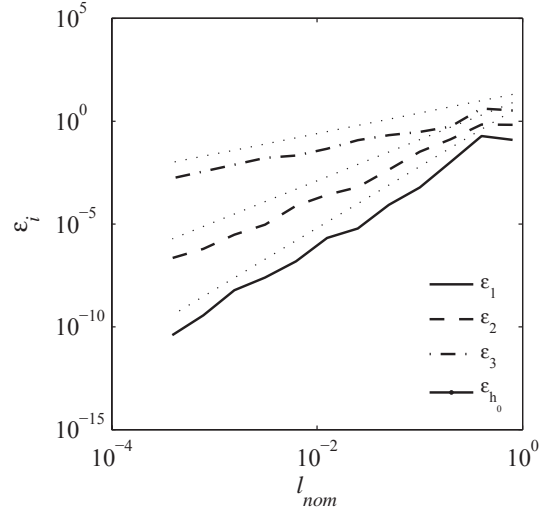
(a) Representative surface and volume mesh used for the three dimensional test cases ($l_{nom} = 0.2$).



(b) Test of TLS method.



(c) Test of HTLS method.



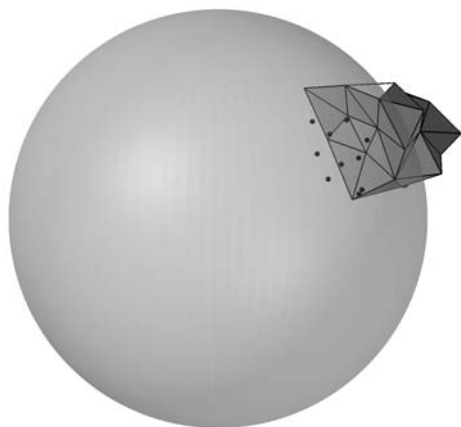
(d) Test of CHTLS method.

Figure 2. Three dimensional tests using Taylor series including first, second, and third derivatives.

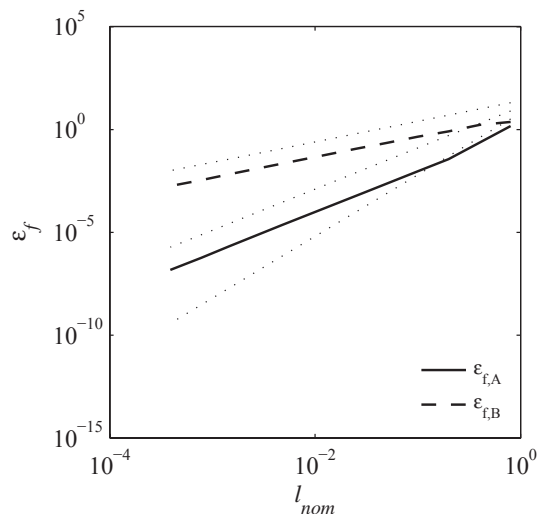
VI.B. Ghost Point On-Boundary Results

Other authors have used ghost points to improve the derivative prediction near the boundary of the domain when using the TLS method. This involves extrapolating function values from the domain to the ghost points – and then using those points in the TLS supporting data.

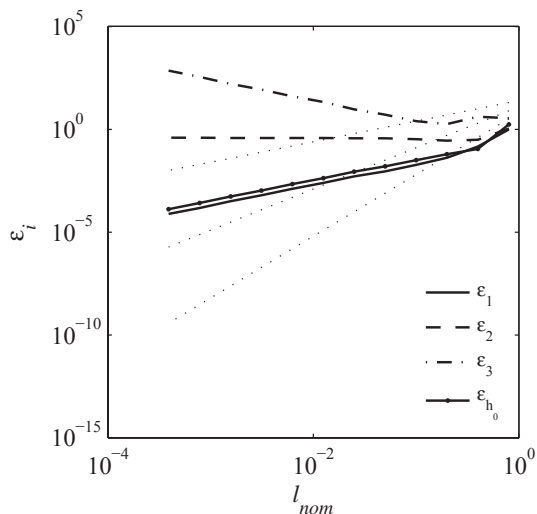
Ghost points were constructed by mirroring selected points through the surface of the sphere to its interior. All non-surface points connected to a surface point by a single edge were selected to construct ghost nodes. The reflected points were placed an equal distance inside the surface of the sphere as the original points were outside. An analytical representation of the surface (rather than the discretized approximate surface) was used to generate the ghost nodes. The ghost nodes constructed for the representative mesh are depicted in Figure 3(a).



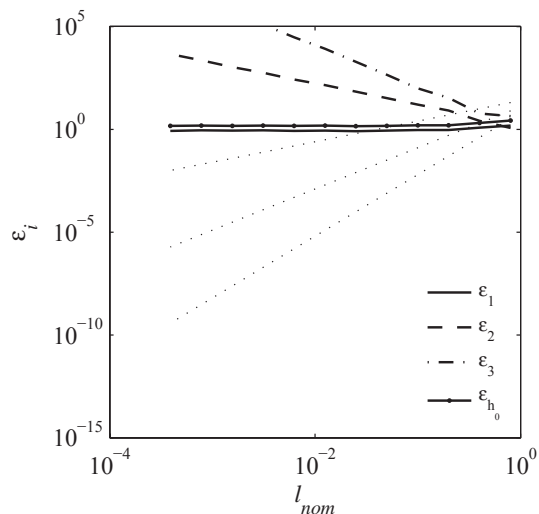
(a) Ghost points created for the representative mesh.



(b) Error in the extrapolated function value.



(c) Test of TLS method using ghost points extrapolated with technique 'A'.



(d) Test of TLS method using ghost points extrapolated with technique 'B'.

Figure 3. Three dimensional tests using the ghost point technique incorporating Taylor series including first, second, and third derivatives.

For these tests, two extrapolation techniques were compared. The first, referred to as technique 'A' is given as Equation 20. This extrapolation equation requires the function value in the domain, f_d , and on

the surface, f_s , to estimate the function value at the ghost node, f_g . This extrapolation equation assumes the three points are in-line and equally spaced; this required the creation of a surface point not depicted in Figure 3(a) and the evaluation of the function at that point. In problems with a physical Dirichlet boundary condition, the corresponding function value would be imposed.

$$f_g = 2f_s - f_d \quad (20)$$

The second extrapolation technique, referred to as technique ‘B’ is given as Equation 21. This extrapolation equation requires the function value in the domain, the normal derivative of the function at the surface, $\partial f/\partial n$, and the spacing between the points, Δn , to estimate the function value at the ghost node. Once again, an auxiliary surface point was used to calculate the normal derivative of the function; in a problem with a physical Neumann boundary condition, the corresponding derivative value would be imposed.

$$f_g = f_d + 2\frac{\partial f}{\partial n}\Delta n \quad (21)$$

The accuracy of the two extrapolation techniques was compared for the test case. In contrast to a problem with an actual boundary to its domain, the test function remains well defined on the interior of the sphere and the analytical function value can be used to evaluate the accuracy of the extrapolation procedure. For each mesh the extrapolated function value was calculated for each of the ghost points; a combined error measure was created by taking the RMS of all point errors. These results are reported for techniques ‘A’ and ‘B’ in Figure 3(b). From these results, it is clear that technique ‘A’ is second order accurate and technique ‘B’ is first order accurate at predicting the ghost node function value. Unfortunately, the practitioner does not typically have freedom to choose which technique to use on a particular problem, the choice is dictated by the physics of the problem.

The TLS method was used to estimate the derivatives at the point of interest using the ghost points and the extrapolated function values in the same manner as was used to produce Figure 2(b). The results from the TLS ghost node test are reported in Figures 3(c) and 3(d) for extrapolation techniques ‘A’ and ‘B’, respectively. Although these calculations were performed using truncated Taylor series which included derivatives up to third order, the results are considerably less accurate than those reported in Figure 2(b) which were obtained with the TLS method without ghost points. In each of these cases, the accuracy of the extrapolation of the ghost points served to limit the accuracy of the derivative estimate; the first derivative predictions were one order less accurate than the corresponding extrapolation accuracy. Furthermore, each higher derivative was predicted with one order less accuracy.

VI.C. In-Boundary Results

Section VI.A reported on the behavior of the TLS, HTLS, and CHTLS methods when used to predict the derivatives of a function when the point of interest was on the boundary of a three dimensional domain; in those tests, information from points both on the boundary and in the domain were used to support the derivative estimates. In this section, estimates of the derivatives of a function at a point of interest on the boundary will be made using only information from points on the boundary. Knowledge of the function derivative normal to the boundary is required for this in-boundary technique; consequently, only application of the CHTLS method is reported here.

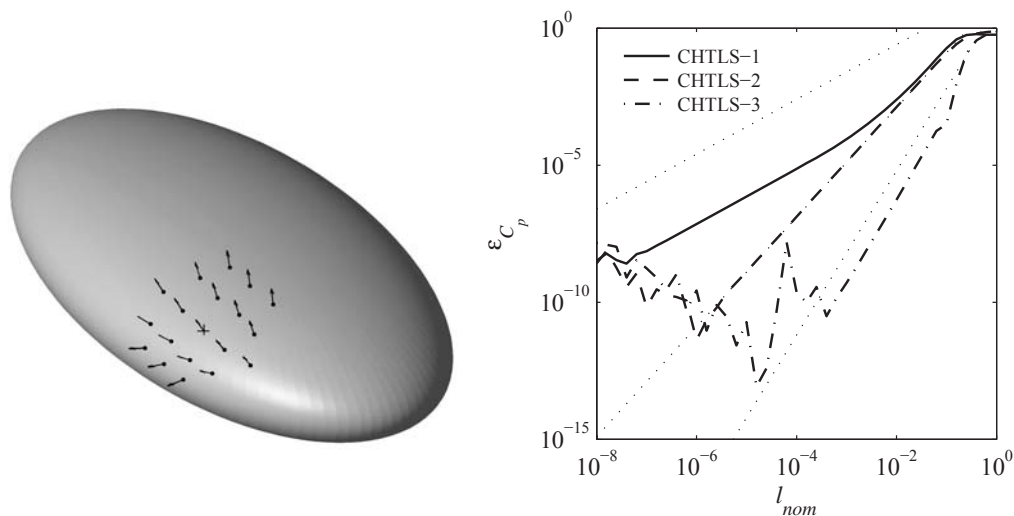
In-boundary derivative estimation can find application when derivatives are needed, but only boundary information is available. One such situation is the calculation of surface velocities and pressure when performing aerodynamic calculations with a panel code. In most panel codes, the surface is described with a structured mesh, and these derivatives are calculated in the parametric surface coordinates implied by the mesh structure.^{11,12} In existing unstructured panel codes, higher order singularity distributions are frequently used which allow surface derivatives to be taken across a single panel.^{13,14}

A representative in-boundary problem was studied by calculating the surface pressure coefficient about an ellipsoid with three unequal axes in potential flow. The analytical solution for the potential, velocities, and pressure coefficient are developed at length by Lamb¹⁵ and somewhat more clearly by Munk.¹⁶ However the greatly simplified treatment by Munk,¹⁷ which provides only surface data, was used. The surface of the ellipsoid was described by the following equation with parameters $a = 1.0$, $b = 2.0$, and $c = 0.5$.

$$\frac{x_1^2}{a^2} + \frac{x_2^2}{b^2} + \frac{x_3^2}{c^2} = 1$$

The point of interest investigated was located on the surface of the ellipsoid at $\theta = 53^\circ$, $\phi = 343^\circ$ in spherical polar coordinates. A non-axis-aligned freestream velocity of $\mathbf{V}_{\text{inf}} = [1.0, 0.5233, 0.3253]$ was selected.

For this test, the two dimensional point set depicted in Figure 1(a) was again put to use. For each case, the points were rotated into a plane tangent to the ellipsoid at the point of interest. The points were then scaled to the required spacing and then centered about the point of interest. Finally, the points were projected towards the center of the ellipsoid onto the ellipsoid surface. Figure 4(a) depicts the surface points and normal vectors used to predict surface pressure coefficient for a typical case in this test.



(a) Representative surface points and normal vectors used for the in-boundary test cases $r = 2$, and $r = 3$. ($l_{\text{nom}} = 0.2$).

Figure 4. CHTLS prediction of pressure coefficient on ellipsoid.

Rather than report the error in the prediction of numerous higher derivatives of potential, in this fluid dynamic problem, only the error in the prediction of pressure coefficient, ϵ_{C_p} , was reported. Calculating pressure coefficient requires the calculation of all three components of velocity which are in turn calculated as the three first derivatives of potential. Consequently, error in pressure coefficient serves to combine the first derivative errors in a similar way as the RMS calculations in the previous tests.

Figure 4(b) depicts the behavior of the error in pressure coefficient prediction using the CHTLS method to calculate velocities. Three cases were examined, each with a different number of derivatives included in the truncated Taylor series. Because the pressure coefficient calculation is based on first derivatives, its error behaves with the same order as the highest derivatives included in the truncated Taylor series.

VII. Conclusions

The TLS technique for estimating derivatives on scattered data has been extended to include derivative information in a least squares sense (HTLS) and also in a constrained least squares sense (CHTLS). A series of formal tests have verified that these techniques are an accurate means of estimating derivatives. In other work, the authors have used the CHTLS technique to solve the full potential equation for subcritical compressible flow.¹⁰

These techniques have been compared with a ghost node technique where fictitious nodes are added to the data set for derivative estimation; the ghost nodes outside the domain are assigned values extrapolated from inside the domain. The ghost node approach is shown to limit the accuracy of first derivative approximation to one order of accuracy less than the order of the extrapolation used for the function value. As the extrapolation techniques usually used are limited to first or second order accuracy, the ghost node approach may be limiting for some problems.

VIII. Reference Implementation

These techniques have been implemented as a Matlab function which implements the TLS, HTLS, or CHTLS methods for N-dimensional problems including as many terms from the Taylor series as desired. This function calculates the matrices required for iterative calculation of derivatives.

These results, further unpublished tests, and the full potential equation solutions conducted by the authors¹⁰ were calculated with this implementation of these techniques. This reference implementation is available for others to use freely; it is available for download from the Matlab File Exchange www.mathworks.com/matlabcentral/fileexchange/29122 or by contacting the first author.

IX. Acknowledgements

This work has been supported by NASA Research Announcement Grant NNX07AO14A under the supervision of technical monitor Craig Nickol.

References

- ¹MacNeal, R. H., "An Asymmetrical Finite Difference Network," *Quarterly of Applied Mathematics*, Vol. 11, 1953, pp. 295–310.
- ²Liszka, T. and Orkisz, J., "The Finite Difference Method at Arbitrary Irregular Grids and its Application in Applied Mechanics," *Computers and Structures*, Vol. 11, 1980, pp. 83–95.
- ³Liszka, T., "An Interpolation Method for an Irregular Net of Nodes," *International Journal for Numerical Methods in Engineering*, Vol. 20, 1984, pp. 1599–1612.
- ⁴Batina, J. T., "A Gridless Euler/Navier-Stokes Solution Algorithm for Complex Two-Dimensional Applications," TM-107631, NASA, 1992.
- ⁵Batina, J. T., "A Gridless Euler/Navier-Stokes Solution Algorithm for Complex-Aircraft Applications," Tech. rep., NASA TM-107727, Langley Research Center, Hampton, VA, 1993.
- ⁶Sridar, D. and Balakrishnan, N., "An Upwind Finite Difference Scheme for Meshless Solvers," *Journal of Computational Physics*, Vol. 189, 2003, pp. 1–29.
- ⁷Sridar, D. and Balakrishnan, N., "Convergence Acceleration of an Upwind Least Squares Finite Difference Based Meshless Solver," *AIAA Journal*, Vol. 44, 2006, pp. 2189–2196.
- ⁸Katz, A. and Jameson, A., "A Comparison of Various Meshless Schemes Within a Unified Algorithm," *47th AIAA Aerospace Sciences Meeting*, 2009.
- ⁹Katz, A. J., *Meshless Methods for Computational Fluid Dynamics*, Ph.D. thesis, Stanford University, 2009.
- ¹⁰Ramos, A., *Development of a Meshless Method to Solve Compressible Potential Flows*, M.S. Thesis, California Polytechnic State University, June 2010.
- ¹¹Ashby, D. L., Dudley, M. R., Iguchi, S. K., Browne, L., and Katz, J., "Potential Flow Theory and Operation Guide for the Panel Code PMARC_12," Tech. rep., NASA Ames Research Center, Moffett Field, CA, December 1992.
- ¹²Katz, J. and Plotkin, A., *Low-Speed Aerodynamics*, Cambridge University Press, Cambridge, MA, 2nd ed., 2001.
- ¹³Eller, D., *An efficient method for time-domain low-speed aerodynamics: Theory and examples*, Department of Aeronautics and Vehicle Engineering, Royal Institute of Technology, SE - 100 44 Stockholm, Sweden, December 2005.
- ¹⁴Willis, D. J., *An Unsteady, Accelerated, High Order Panel Method with Vortex Particle Wakes*, Ph.D. Thesis, Massachusetts Institute of Technology, 2006, pp. 48-49.
- ¹⁵Lamb, H., *Hydrodynamics*, Cambridge University Press, 6th ed., 1932, Also published by Dover Publications in 1945.
- ¹⁶Munk, M. M., *Aerodynamic Theory*, Vol. I, 1934.
- ¹⁷Munk, M. M., "Remarks on the Pressure Distribution over the Surface of an Ellipsoid, Moving Translationally Through a Perfect Fluid," , No. NACA-TN-196, June 1924.