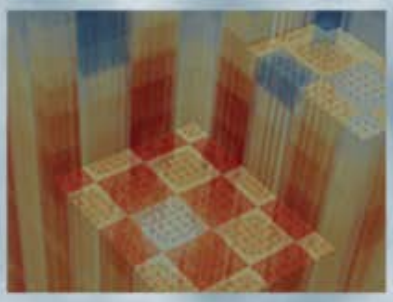
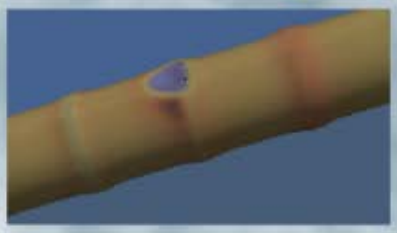
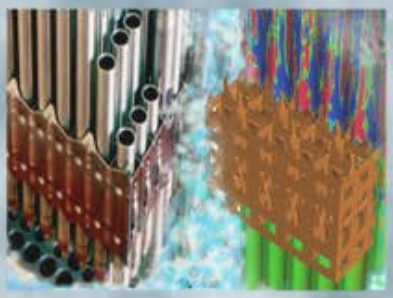
CASL-U-2015-0177-000

A Modified Moving Least Square Algorithm for Solution Transfer on a Spacer Grid Surface







Stuart R. Slattery, Steven P. Hamilton, and Thomas M. Evans
Oak Ridge National Laboratory

April 19, 2015

U.S. DEPARTMENT OF ENERGY | Nuclear Energy

CASL-U-2015-0177-000

ANS MC2015 - Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method · Nashville, Tennessee · April 19–23, 2015, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2015)

A MODIFIED MOVING LEAST SQUARE ALGORITHM FOR SOLUTION TRANSFER ON A SPACER GRID SURFACE

Stuart R. Slattery, Steven P. Hamilton, and Thomas M. Evans
Oak Ridge National Laboratory*
One Bethel Valley Road, Oak Ridge, TN, 37831
slatterysr@ornl.gov; hamiltonsp@ornl.gov; evanstm@ornl.gov

ABSTRACT

Solution transfer operators constructed from a moving least square basis have recently been researched for multiphysics simulations of pressurized water reactors. Useful for solution transfer between surfaces and volumes, moving least square schemes are attractive because they do not require a computational mesh but instead use a point cloud representation of the discretized domain. In addition, they are often able to achieve a very accurate and conservative solution reconstruction and are readily parallelized to scale on leadership class computing facilities. When studying the moving least square technique, we discovered that when the algorithm was applied to solution transfers on a spacer grid surface, numerical instabilities in the singular value decomposition algorithm used to generate the moving least square basis resulted in large errors after a few solution transfer iterations, rendering the method unusable in many situations. In this work we assess these instabilities and show that using a truncated singular value decomposition and simply augmenting the threshold in the algorithm does not alleviate the instabilities without introducing additional error into the solution transfer and adding an additional free parameter to the algorithm. We then modify the moving least square algorithm to ensure that the terms of the spatial polynomials used to construct the least square problem are linearly independent, resulting in a full rank linear system which can be used to calculate an optimal truncation threshold. We show that the new algorithm significantly improves the stability of repeated solution transfers on the spacer grid surface.

Key Words: solution transfer, multiphysics, moving least square, singular value decomposition, spacer grid

1 INTRODUCTION

Solution transfer algorithms are a fundamental requirement for multiphysics simulations where different computational grids and numerics are used for each physics involved. For solution transfer of nodal fields between physics modules, several methods are used in practice when a mesh is available including basic interpolation as well as weighted residual schemes which are available in both conservative and accurate formulations [1, 2]. As an alternative to methods that leverage the meshes discretizing a shared interface or domain, significant work has been performed in recent years primarily in the aerospace community on mesh-free techniques that instead apply the solution

*Notice: This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

CASL-U-2015-0177-000

transfer between sets of point clouds that represent the discrete interface [3–9]. In these methods, one removes the functional support provided by the grid and underlying physics discretization and replaces it with support of a higher order constructed from various basis functions. Often, compactly supported radial basis functions are used to provide this support and only require the nearest neighbors of a given node in its local spatial neighborhood.

Of the available mesh-free algorithms, the moving least square reconstruction algorithm first presented in the context of multiphysics solution transfer in [4] is particularly attractive because it is both accurate and fully decouples each node in the grid, offering a straight forward path to parallelization. In the literature, moving least square algorithms are used most often for solution transfers on grids with curved surfaces. However, many components of a light water reactor geometry do not have curved surfaces and, as indicated by [4], these flat surfaces may result in node configurations that lead to ill-conditioned or potentially singular least square problems. We found this to be the case when applying the algorithm to two different meshes discretizing the surface of a spacer grid with numerical instabilities in the solution scheme resulting in large errors after a few iterations. A spacer grid is an important and complex geometry in many reactor applications including coupling between fluid dynamics and heat transfer. In this application, solution transfer algorithms are used to reconstruct temperatures and heat fluxes on the surface of the spacer grid which serves as a boundary between the fluid domain and solid domain. In this paper, we assess the solution of the least square problem for these spacer grid discretizations and provide a modified solution scheme that is demonstrated to yield a more stable solution transfer operator than the originally published algorithm.

In this paper we first describe the moving least square reconstruction algorithm and define the solution transfer operator. Next, we use the operator to perform solution transfer on a spacer grid geometry where we show that computing a truncated singular value decomposition (SVD) to solve the least square problems that construct the operator basis can alleviate the instabilities at the cost of additional error in the solution transfer as well as giving an additional free parameter to the user. In particular, we show that large errors arise on the spacer grid surface when the singular values are not handled correctly. We then eliminate the need for selection of a truncation parameter and present an algorithm that poses the moving least square problem using a full rank but potentially lower order spatial polynomial matrix to build the moving least square basis. This not only improves the robustness of the algorithm but also automatically computes an optimal truncation parameter for each individual node in the system. Using the modified algorithm we repeat the calculations with our results indicating that the stability of the algorithm is improved. Finally, we conclude the paper with a summary of our results and indicate future work.

2 MOVING LEAST SQUARE RECONSTRUCTION ALGORITHM

In this section we present the original moving least square reconstruction technique. First, we define the solution transfer problem in terms of functions supported on a point cloud and define the solution transfer operator. Next, we briefly introduce the compactly supported radial basis functions used in the moving least square reconstruction. Finally, we define the moving least square solution transfer operator by building a basis from solutions to least square problems.

2.1 Solution Transfer Problem

The solution transfer problem is defined as the reconstruction of a function represented by a discretization on one grid to a representation on a potentially different grid with a different discretization. The grid on which the function is initially represented is defined as the *source grid* and the grid onto which the function will be transferred is defined as the *target grid*. In its representation on the source grid, the function will be defined as, f, the *source function*. The representation of the function on the target grid will be, g, the *target function*. We define the source function over M nodes or control points as:

$$f_i = f(\mathbf{s}_i) \ , \tag{1}$$

where f_i is the source function evaluated at the ith control point in the source grid, $\mathbf{s}_i$. Equivalently, we have the target function defined over N control points as:

$$g_i = g(\mathbf{t}_i) \ . \tag{2}$$

where g_i is the target function evaluated at the ith control point in the target grid. We then define a *solution transfer operator*, $\mathbf{H}$, such that $g = \mathbf{H}f$.

2.2 Compactly Supported Radial Basis Functions

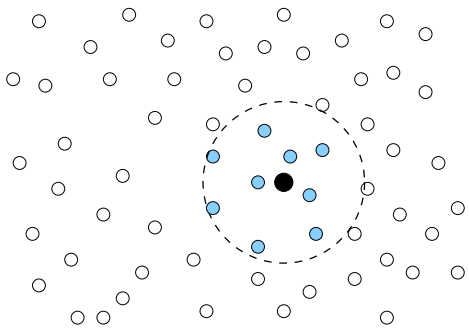
Mesh-free methods for solution transfer do not consider the discretization of fields when constructing the interpolant. Instead, the domain is represented by arbitrary point clouds with topology-independent functional support. Radial basis functions permit the construction of functional support in arbitrary point clouds by using the Euclidean distance, d, between a point, i, and any one of its supporting points, j:

$$d = ||\mathbf{x}_i - \mathbf{x}_j||_2 \ , \tag{3}$$

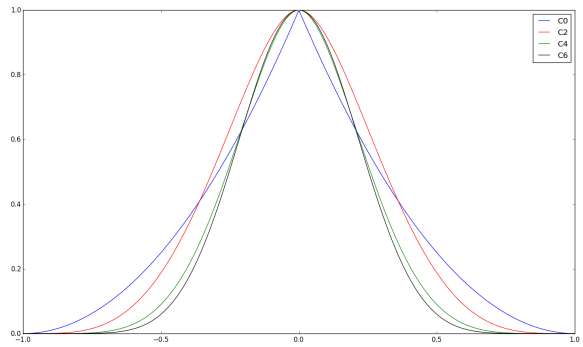
where $\mathbf{x}_i$ and $\mathbf{x}_j$ are the physical coordinates of the points i and j respectively. Compact radial basis functions are desirable for large interpolation problems because of the sparsity they introduce into the problem by requiring only a subset of the control points in the coupling domain to construct support as shown in Figure 1a. As an example, consider Wendland's C^2 function, part of a broader family of functions shown in Figure 1b:

$$\phi(d) = \left(1 - \frac{d}{\rho}\right)_+^4 \left(4\frac{d}{\rho} + 1\right) \tag{4}$$

where ρ is the physical support radius of the function and the $(\cdot)_+$ notation on the first term indicates that term evaluates to zero if $(1 - d/\rho) < 0$, thus truncating support at the boundary of the physical support radius. The choice of ρ will dictate how many control points are used for support with more points typically providing a smoother interpolation. We refer the reader to [10] for more details and examples of these functions.



(a) Radial basis functions are supported by the local neighborhood of nodes within a radius ρ.



(b) Wendland radial basis family. Functions of varying smoothness are available with C^0, C^2, C^4 and C^6 shown here.

Figure 1. Compactly supported radial basis functions build an interpolant with localized support.

2.3 Moving Least Square Solution Transfer Operator

Quaranta, Masarati, and Mantegazza presented a function reconstruction technique for arbitrary point clouds based on a moving least square discretization [4]. In this method, support and subsequently the solution transfer operator is constructed through solutions to local least square kernels defined by compactly supported radial basis functions. First, the solution at each target point is formulated as a polynomial sum:

$$g_i = \sum_{i=1}^{n} p_i(\mathbf{t}_i) a_i(\mathbf{t}_i) \ , \tag{5}$$

where the $a_i(\mathbf{t}_i)$ are polynomial coefficients and $p_i(\mathbf{t}_i)$ are quadratic polynomials in terms of spatial coordinates. In vector form for an arbitrary three-dimensional point, q, these polynomials are:

$$\mathbf{p}(q) = \begin{bmatrix} 1 & x_q & y_q & z_q & x_q y_q & x_q z_q & y_q z_q & x_q^2 & y_q^2 & z_q^2 \end{bmatrix}^T \ . \tag{6}$$

We can create the interpolant by minimizing the solution transfer residual through a weighted least square procedure at each individual target point, $\mathbf{t}_i$:

$$\underset{a_i(\mathbf{t}_i)}{\text{minimize}} \ \frac{\partial}{\partial a_i(\mathbf{t}_i)} \int_{\Omega} \phi(\mathbf{t}_i - \mathbf{s})(\mathbf{g} - \mathbf{f})^2 d\Omega(\mathbf{s}) \ , \tag{7}$$

where the $a_i(\mathbf{t}_i)$ serve as the free parameters in the minimization for which we are solving, $\Omega(\mathbf{s})$ is the domain of interest specifically defined by the supporting source control points, and compactly supported radial basis functions defined at the target control point and supported by the source control points, $\phi(\mathbf{t}_i - \mathbf{s})$, serve as the weights in the minimization. To close the problem, Eq (5) provides a natural constraint for the minimization procedure. Solving the minimization problem gives the following kernel to be evaluated at each target control point *:

$$\xi(t_i) = \mathbf{p}(t_i)^T \left[\mathbf{P}(s_{t_i})^T \mathbf{\Phi}(s_{t_i}) \mathbf{P}(s_{t_i}) \right]^{-1} \mathbf{P}(s_{t_i})^T \mathbf{\Phi}(s_{t_i}) \ , \tag{8}$$

*See [4] for a derivation of this kernel.

with s_{t_i} the set of n source points within the support radius of the ith target point, $\mathbf{t_i}$. The polynomial component is defined as a Vandermonde matrix:

$$\mathbf{P}(s_{t_i}) = \begin{bmatrix} \mathbf{p}(s_{t_1}) \\ \mathbf{p}(s_{t_2}) \\ \cdots \\ \mathbf{p}(s_{t_n}) \end{bmatrix} = \begin{bmatrix} 1 & x_{s_{t_1}} & y_{s_{t_1}} & z_{s_{t_1}} & x_{s_{t_1}}y_{s_{t_1}} & x_{s_{t_1}}z_{s_{t_1}} & y_{s_{t_1}}z_{s_{t_1}} & x^2_{s_{t_1}} & y^2_{s_{t_1}} & z^2_{s_{t_1}} \\ 1 & x_{s_{t_2}} & y_{s_{t_2}} & z_{s_{t_2}} & x_{s_{t_2}}y_{s_{t_2}} & x_{s_{t_2}}z_{s_{t_2}} & y_{s_{t_2}}z_{s_{t_2}} & x^2_{s_{t_2}} & y^2_{s_{t_2}} & z^2_{s_{t_2}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{s_{t_n}} & y_{s_{t_n}} & z_{s_{t_n}} & x_{s_{t_n}}y_{s_{t_n}} & x_{s_{t_n}}z_{s_{t_n}} & y_{s_{t_n}}z_{s_{t_n}} & x^2_{s_{t_n}} & y^2_{s_{t_n}} & z^2_{s_{t_n}} \end{bmatrix}, \tag{9}$$

and the basis component a diagonal matrix:

$$\mathbf{\Phi}(s_{t_i}) = \begin{bmatrix} \phi_{s_{t_1}t_i} & 0 & \cdots & 0 \\ 0 & \phi_{s_{t_2}t_i} & \cdots & 0 \\ \cdots & \cdots & \ddots & \cdots \\ 0 & 0 & \cdots & \phi_{s_{t_n}t_i} \end{bmatrix}. \tag{10}$$

We can then reconstruct the target function through independent evaluations of the moving least square kernel in Eq (8) at each target control point. Alternatively, we can store the solutions to the kernels as a matrix to form the solution transfer operator:

$$H_{ij} = \xi(\mathbf{t}_i)_j, \iff \mathbf{s}_j \in s_{t_i}\ , \tag{11}$$

where i is the index of the target control point and j the index of the source control point providing support within the radius of $\mathbf{t}_i$.

3 TRUNCATED SINGULAR VALUE DECOMPOSITION

The entries of the solution transfer operator $\mathbf{H}$ are explicitly constructed through the solution of local least square problems defined by Eq (8). To construct these problems, we use a radius search to build the set of source control points supporting each target control point, s_{t_i}. We formulate each kernel as a small, dense linear algebra problem and utilize LAPACK and BLAS routines for manipulating the resulting matrices [11]. To construct the kernel in Eq (8), we solve the linear system

$$\mathbf{AX} = \mathbf{B}\ , \tag{12}$$

with:

$$\mathbf{A} = \mathbf{P}(s_{t_i})^T\mathbf{\Phi}(s_{t_i})\mathbf{P}(s_{t_i})\ , \tag{13}$$

the operator,

$$\mathbf{B} = \mathbf{P}(s_{t_i})^T\mathbf{\Phi}(s_{t_i})\ , \tag{14}$$

a matrix of right hand side vectors and,

$$\mathbf{X} = \left[\mathbf{P}(s_{t_i})^T\mathbf{\Phi}(s_{t_i})\mathbf{P}(s_{t_i})\right]^{-1}\mathbf{P}(s_{t_i})^T\mathbf{\Phi}(s_{t_i})\ , \tag{15}$$

the matrix of solution vectors. The kernel for the ith target center, t_i, is then:

$$\xi(t_i) = \mathbf{p}(t_i)^T \mathbf{X} \,. \tag{16}$$

In the aerospace literature, solving Eq (12) is almost trivial for curved airfoil grids, requiring either an LU or QR decomposition scheme. However, as mentioned previously, flat surfaces and sharp edges on the spacer grid pose an issue with rank deficiency arising in the matrix of spatial polynomials in Eq (9). This occurs because one or two dimensions of space are effectively lost for constructing the coefficients of the polynomial. Because of this, to solve Eq (12) we use a singular value decomposition as a means of producing a factorization of the operator which reveals information about the rank of the system.

The singular value decomposition of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is defined as:

$$\mathbf{U}^T \mathbf{A} \mathbf{V} = \mathbf{\Sigma} \,, \tag{17}$$

where $\mathbf{U}^T \in \mathbb{R}^{m \times m}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$ and $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$ contains the singular values of $\mathbf{A}$. We can then find a minimum norm solution of Eq (12) by computing:

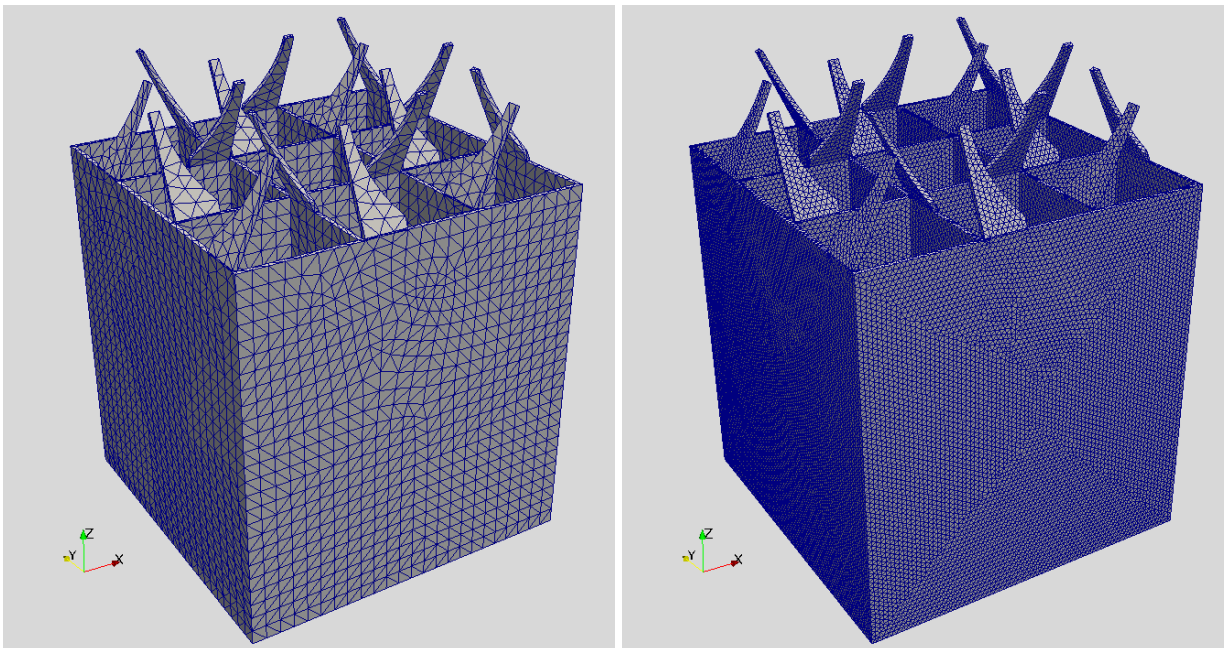
$$x_i = \sum_{i=1}^{rank(\mathbf{A})} \frac{u_i^T b_i}{\sigma_{ii}} v_i \,, \tag{18}$$

where $(\cdot)_i$ indicates the ith column of the given matrix, σ_{ii} the ith singular value, and the rank of $\mathbf{A}$ can usually be inferred from the singular values. However, it is not always trivial to determine this rank, especially when the operator in Eq (9) does not specify a fully orthogonal basis. In this case we choose to compute the pseudoinverse of $\mathbf{A}$:

$$\mathbf{X} = \mathbf{A}^+ \mathbf{B} \,. \tag{19}$$

To build the pseudoinverse, we use a truncated SVD procedure where a parameter ϵ is selected so that singular values that satisfy $\sigma_i < \epsilon \sigma_{max}$ are ignored in the computation of the minimum norm solution in Eq (18). Per [12], a good choice for the truncation is $\epsilon = \frac{1}{\kappa(\mathbf{A})}$ as the condition number of $\mathbf{A}$ provides a measure for the number of correct digits in our factorization. However, for rank deficient cases estimation of $\kappa(\mathbf{A})$ may not be possible if $\mathbf{A}$ is singular and an LU factorization is not available. Therefore, when using the original moving least square algorithm, we simply choose a value of ϵ to be used for the solution of Eq (12) at each target point. We refer the reader to [12] and [11] for more details on computing the truncated singular value decomposition and condition estimates of matrices.

To demonstrate the effectiveness of the truncated singular value decomposition, we apply the moving least square scheme to a 3×3 spacer grid geometry with mixing vanes. We used the triangular surface mesh from a coarse and fine linear tetrahedral discretization of the geometry as shown in Figure 2. Although these grids were selected arbitrarily, the numerical issues addressed in this work were found to occur when several different grid variations were used including those with linear hexahedral elements. To given an idea of the size of the solution transfer problem, grid A contains 10,973 nodes on the surface while grid B contains 94,621 nodes.



(a) Spacer grid mesh A. **(b) Spacer grid mesh B.**

Figure 2. Coarse mesh and fine mesh of 3×3 spacer grid geometry. Both meshes are composed of linear tetrahedrons.

In the absence of physics, it is common to study solution transfer operators by repeatedly applying them to a known function iteratively between two grids, in this case grids A and B, and observe how the operators perturb the function. For our tests we define the following simple linear function on the surface nodes of grid A that should be easily reconstructed by the algorithm in order to expose errors in the moving least square operator:

$$u(\mathbf{x}) = 1.0 + |x| + |y| + |z| \; . \tag{20}$$

We then apply the first solution transfer operator to move the function to the surface nodes of grid B and then apply the second operator to move the function back to the surface of grid A. After each iteration, accuracy of the solution transfer will be measured by $||\mathbf{e}||_2/||\mathbf{u}||_2$ where the error at a node, $\mathbf{x_i}$, is:

$$\mathbf{e}_i = f(\mathbf{x}_i) - u(\mathbf{x}_i) \; . \tag{21}$$

We applied the transfer operators in 100 iterations between grids A and B for $\epsilon = \epsilon_{machine} = 2.22045 \times 10^{-16}$, 1×10^{-14}, 1×10^{-12}, 1×10^{-8}, and 1×10^{-4} to capture the case of the original algorithm without truncation and then varying values of truncation using an implementation of the moving least square algorithm in the DataTransferKit library [13]. Figure 3 gives the error as a function of iteration for these calculations. For the case without truncation ($\epsilon = \epsilon_{mach}$), numerical instabilities cause an immediate and unbounded rise in error resulting from linear dependencies amongst nodes in the grid creating singular values near machine precision. Adding truncation stabilizes the iteration sequence at the cost of adding more error to the computation. As truncation

is increased, more information is thrown away from the least square problems yielding more error in the solution transfer. Based on these results, it is clear that the singular values play an important role in the accuracy and stability of the solution transfer. However, user selection of truncation is not ideal as it is not known a priori whether the parameter will provide the desired accuracy.

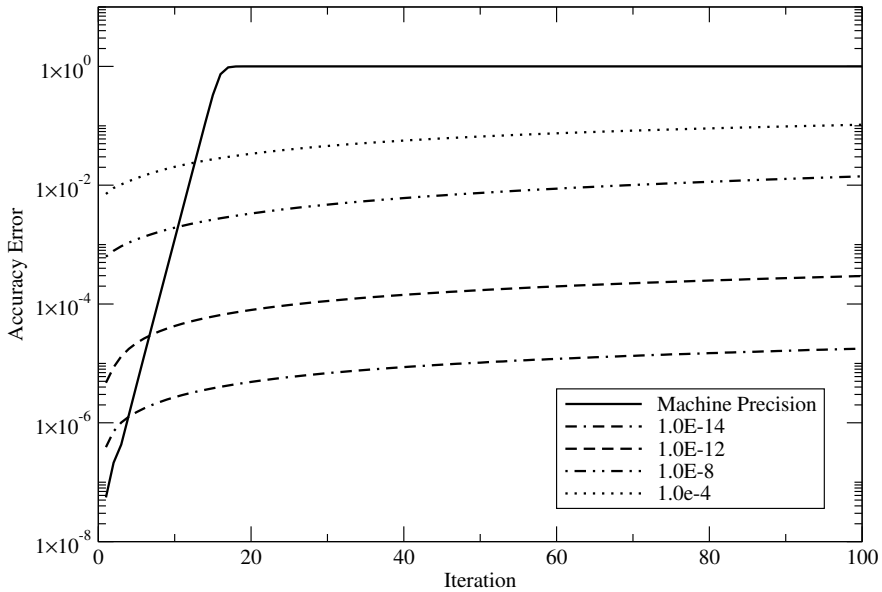


Figure 3. Effect of SVD truncation parameter on solution transfer accuracy. The accuracy is extremely sensitive to the chosen truncation parameter.

4 MODIFIED ALGORITHM

To improve the moving least square algorithm we focus on two deficiencies indicated by the analysis in the previous section. First, we were led to use the singular value decomposition as a solution scheme for building the moving least square kernel because we could not guarantee a full rank linear system for arbitrary discretizations of the spacer grid. We will first address rank deficiency by building a full rank approximation to the least square problem. Second, with a full rank system, it is viable to use an LU factorization to estimate a more optimal truncation parameter.

Full Rank Vandermonde Matrix Rank deficiency in the operator $\mathbf{A}$ is caused by lack of an orthogonal basis defined by $\mathbf{P}(s_{t_i})$ and not $\mathbf{\Phi}(s_{t_i})$. Considering that $\mathbf{\Phi}(s_{t_i})$ will always have non-zero elements on the diagonal due to all support points being within the radius of the target point of interest, it will always scale $\mathbf{A}$ and never introduce zero rows. To create a full rank variant of $\mathbf{A}$, we must then always ensure that $\mathbf{P}(s_{t_i})$ for a given target point is of full rank. We do this with the following procedure to ensure an orthogonal basis:

1. Add each coefficient in Eq (6) column-by-column starting with the linear terms to approximate $\mathbf{P}(s_{t_i}) \approx \hat{\mathbf{P}}(s_{t_i})$

2. For each column added, compute an SVD of the current approximation: $\mathbf{U}^T\hat{\mathbf{P}}(s_{t_i})\mathbf{V} = \mathbf{\Sigma}$
3. If any σ_{ii} in $\mathbf{\Sigma}$ is less than $\epsilon_{mach}\sigma_{max}$, discard the newest column of $\hat{\mathbf{P}}(s_{t_i})$ and move on to the next polynomial coefficient column

For this work we limit ourselves to the coefficients available in the quadratic spatial polynomial defined in Eq (6). However, there is no reason the order could not be increased to cubic or larger. This means that we have the potential to generate a reduced order polynomial for those points in which we have rank deficiency. For example, if all supporting points have spatial coordinates with $z = 1$, the resulting Vandermonde matrix approximation would be:

$$\hat{\mathbf{P}}(s_{t_i}) = \begin{bmatrix} 1 & x_{s_{t_1}} & y_{s_{t_1}} & x_{s_{t_1}}y_{s_{t_1}} & x^2_{s_{t_1}} & y^2_{s_{t_1}} \\ 1 & x_{s_{t_2}} & y_{s_{t_2}} & x_{s_{t_2}}y_{s_{t_2}} & x^2_{s_{t_2}} & y^2_{s_{t_2}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{s_{t_n}} & y_{s_{t_n}} & x_{s_{t_n}}y_{s_{t_n}} & x^2_{s_{t_n}} & y^2_{s_{t_n}} \end{bmatrix} . \tag{22}$$

Once a $\hat{\mathbf{P}}(s_{t_i})$ has been computed, it is used instead of $\mathbf{P}(s_{t_i})$ for the remainder of the least square algorithm with the new basis:

$$\hat{\xi}(t_i) = \hat{\mathbf{p}}(t_i)^T \left[\hat{\mathbf{P}}(s_{t_i})^T \mathbf{\Phi}(s_{t_i})\hat{\mathbf{P}}(s_{t_i})\right]^+ \hat{\mathbf{P}}(s_{t_i})^T \mathbf{\Phi}(s_{t_i}) \ , \tag{23}$$

using the pseudoinverse of $\hat{\mathbf{A}} = \hat{\mathbf{P}}(s_{t_i})^T \mathbf{\Phi}(s_{t_i})\hat{\mathbf{P}}(s_{t_i})$.

Optimal Truncation Parameter With $\hat{\mathbf{A}}$ now defined to within at least some tolerance to be a full rank system, we can additionally estimate the condition number of $\hat{\mathbf{A}}$ via an LU factorization and then use an implementation of the condition estimation algorithm in [12] as provided in LAPACK. This condition estimation is performed for the matrix $\hat{\mathbf{A}}$ constructed at each target point after the full rank system has been generated to give a new truncation parameter $\epsilon = \frac{1}{\kappa(\hat{\mathbf{A}})}$.

5 MODIFIED ALGORITHM RESULTS

Using the same iterative analysis to test the truncated SVD algorithm, we apply our new modified algorithm to the spacer grid problem using both the orthogonal polynomial basis as well as the orthogonal basis in tandem with an optimal truncation parameter computed via a condition estimate of the system. Iterative results for the new algorithm and the original algorithm without truncation are given in Figure 4. These results indicate that a combination of both a full rank polynomial matrix as well as an optimal truncation parameter are required in order to ensure long-term stability of the solution transfer. Because a small but arbitrary parameter of ϵ_{mach} was used to compute the rank of the approximate Vandermonde matrix, additional truncation is needed in the SVD solution to ensure accurate results for very ill-conditioned problems. This also indicates that a simpler solution scheme such as an LU or QR factorization could not in general be used to solve the most ill-conditioned of problems even with the orthogonal set of spatial polynomials. In

addition, we note that we achieve a similar accuracy error after 100 iterations with the new modified algorithm as we did with $\epsilon = 1 \times 10^{-14}$ shown in Figure 3 without an ad hoc choice of truncation parameter.

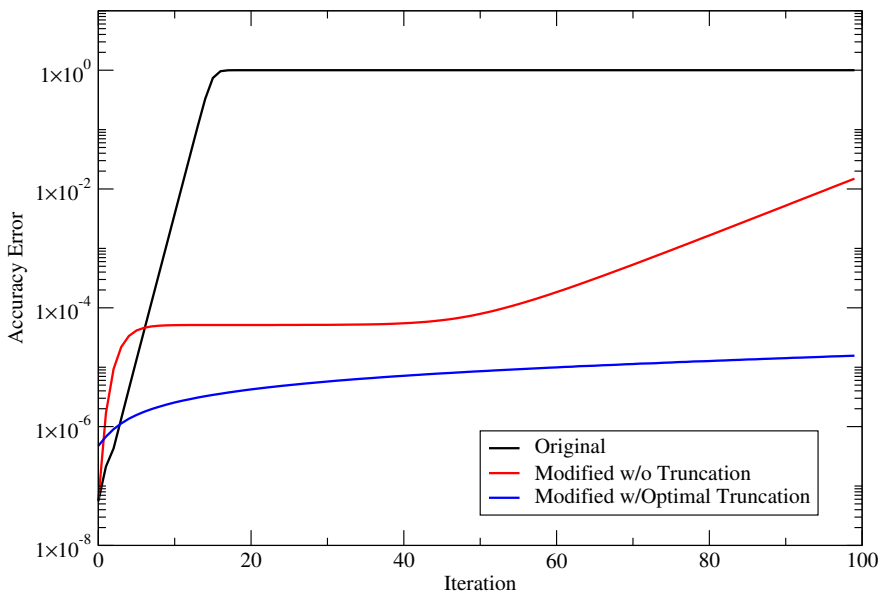


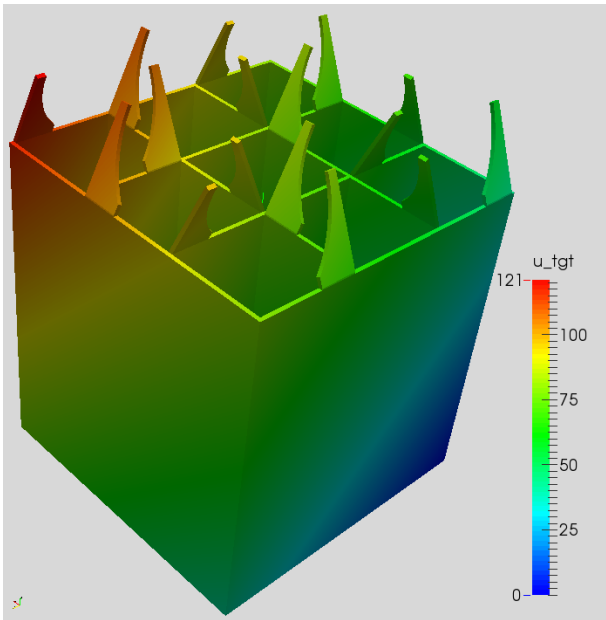
Figure 4. Solution transfer accuracy error vs. iteration. The modified algorithm without truncation improves the stability of the original algorithm but still exhibits a large error component. Adding an optimal truncation parameter eliminates this extra error component.

In Figure 5, we have additionally plotted the original function and then the original algorithm and both variants of the modified algorithm after 100 iterations. While the modified algorithm with optimal truncation respects the bounds of the original function, the modified algorithm without truncation shows errors on the surface of some mixing vanes, indicating that the least square problems at these points were particularly ill-conditioned because they were near sharp features of the grid. The original algorithm error grows unbounded if no truncation is applied resulting in loss of all accuracy in the solution transfer after $O(10)$ iterations.

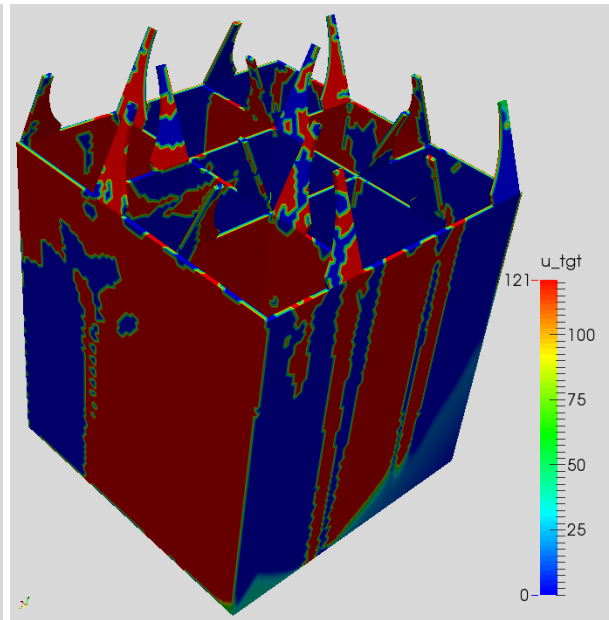
6 CONCLUSIONS

In this paper we have developed a modified moving least square reconstruction algorithm for solution transfer in multiphysics applications. For solution transfer on the surface of a spacer grid, certain mesh node configurations were found to result in ill-conditioned or possibly singular weighted least square problems. We discovered that when using a singular value decomposition to solve these problems, the selection of which singular values to truncate gave varying results for accuracy and stability. To eliminate this selection, we modified the algorithm to first build a fully orthogonal but potentially reduced order set of spatial polynomials and then used this new system to automatically compute an optimal truncation parameter for each node of the mesh, ultimately resulting in a more accurate and stable algorithm. In future work, we would like to expand the

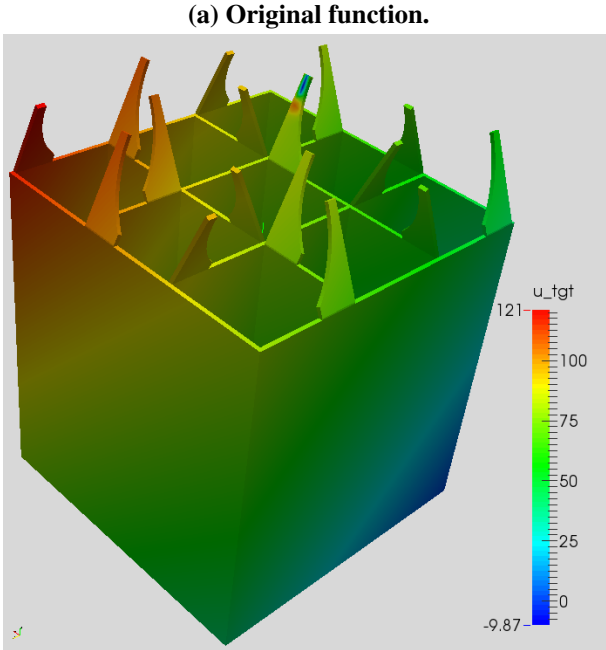
algorithm to use spatial polynomials of cubic order or higher and study its effects on the accuracy and performance of the solution transfer.



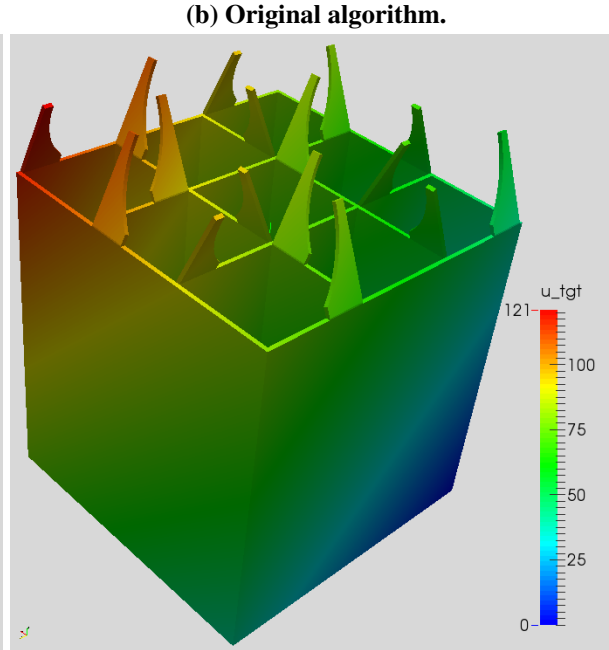
(a) Original function.



(b) Original algorithm.



(c) Modified algorithm without truncation.



(d) Modified algorithm with optimal truncation.

Figure 5. Solutions on the fine grid after 100 transfer iterations. The scale of (a), (c), and (d) are true to the solution while the scale of (b) is fixed to the scale of the original function to show the violation of the solution minima and maxima. Note the solution error in (c) on the mixing vane surface, an example of errors near sharp features.

7 ACKNOWLEDGMENTS

This research was supported by the Consortium for Advanced Simulation of Light Water Reactors (`www.casl.gov`), an Energy Innovation Hub (`http://www.energy.gov/hubs`) for Modeling and Simulation of Nuclear Reactors under U.S. Department of Energy Contract No. DE-AC05-00OR22725.

8 REFERENCES

[1] X. Jiao and M. T. Heath, "Common-refinement-based data transfer between non-matching mesh in multiphysics simulation," *International Journal for Numerical Methods in Engineering*, **61**, pp. 2402–2427 (2004).

[2] J. R. Cebral and R. Lohner, "Conservative load projection and tracking for fluid-structure problems," *AIAA Journal*, **35**, *4*, pp. 687–692 (1997).

[3] A. Beckert and H. Wendland, "Multivariate interpolation for fluid-structure-interaction problems using radial basis functions," *Aerosp. Sci. Technol.*, **5**, *2*, pp. 125–134 (2001).

[4] G. Quaranta, P. Masarati, and P. Mantegazza, "A Conservative Mesh-Free Approach for Fluid-Structure Interface Problems," *Int. Conf. on Computational Methods for Coupled Problems in Science and Engineering* (2005).

[5] T. Rendall and C. Allen, "Unified fluid-structure interpolation and mesh motion using radial basis functions," *International Journal for Numerical Methods in Engineering*, **74**, pp. 1519–1559 (2008).

[6] T. Rendall and C. Allen, "Improved radial basis function fluid-structure coupling via efficient localized implementation," *International Journal for Numerical Methods in Engineering*, **78**, pp. 1188–1208 (2009).

[7] W. Costin and C. Allen, "Radial Basis Function Interpolation For Data Transfer Across a Mesh Interface," *20th AIAA Computational Fluid Dynamics Conference* (2011).

[8] T. Rendall and C. Allen, "Evaluation of RBFs for volume data interpolation in CFD," *International Journal for Numerical Methods in Engineering*, **72**, pp. 748–769 (2013).

[9] M. Cordero-Gracia, M. Gomez, and E. Valero, "A radial basis function algorithm for simplified fluid-structure data transfer," *International Journal for Numerical Methods in Engineering* (2014).

[10] M. Buhmann, "Radial basis functions," *Acta Numerica* (2000).

[11] E. Anderson et al., *LAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, (1999).

[12] G. Golub and C. V. Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, fourth edition, (2013).

[13] S. Slattery, P. Wilson, and R. Pawlowski, "The Data Transfer Kit: A Geometric Rendezvous-Based Tool for Multiphysics Data Transfer," *International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering (M&C 2013) (CD-ROM)*, May, 2013.