AIAA SciTech Forum
9 - 13 January 2017, Grapevine, Texas
58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference

10.2514/6.2017-0131

Comparison of Unified and Sequential-Approximate Approaches to Multifidelity Optimization

Dean E. Bryson* and Markus P. Rumpfkeil†
University of Dayton, Dayton, Ohio, 45469, USA

Multifidelity approaches are frequently used in engineering design when high-fidelity models are too expensive to use directly and lower fidelity models of reasonable accuracy exist. In optimization, corrected low-fidelity data is typically used in a series of sequential optimizations bounded by trust regions around the approximate model. Each sub-optimization is independent of the previous one, except for the starting design and trust region size. A new unified multifidelity quasi-Newton approach is presented that preserves an estimate of the inverse Hessian between iterations, determining search directions from high-fidelity data and using approximate models for single line searches. The proposed algorithm produces better search directions, maintains larger step sizes, and requires significantly fewer low-fidelity function evaluations than Trust Region Model Management. The multifidelity quasi-Newton method also provides an expected optimal point which is forward looking and is useful in building superior low-fidelity corrections. As part of the multifidelity framework, a warm starting technique is demonstrated to initialize high-fidelity optimization when transition away from approximate models is deemed fruitful. The new approach is compared to Trust Region Model Management on a variety of analytic test problems using both polynomial and kriging correction functions. In summary, the unified multifidelity quasi-Newton approach required fewer high-fidelity function evaluations than Trust Region Model Management in about two-thirds of the test cases.

I. Introduction

Fidelity is the degree to which engineering prediction (whether by analysis or simulation) matches physical reality. Typically, high-fidelity methods (those capturing the greatest number of physical phenomena) are resource intensive which may preclude their use in much of the engineering design process. Fortunately, in many cases, methods of lesser fidelity are often sufficient to model the physics driving the design. Thus, we resort to using cheaper, lower fidelity methods when available. An underlying assumption of using multiple fidelities (when treated in a hierarchical sense), is that a lower fidelity model produces less accurate or less reliable information than a higher-fidelity model but at a lower cost. However, an underlying difficulty may be knowing what sufficient fidelity is, particularly as we depart from well explored regions of the design space.

The use of multiple fidelities may also make design optimization more efficient. Similar to early design, optimizers make repeated calls to simulations with variations in the design configuration. The likelihood that a design point will be rejected after being analyzed is greater in early iterations than in later iterations. That is, as the optimizer builds a better picture of the design space, it becomes more effective in selecting designs that satisfy the acceptance criteria (e.g., sufficient decrease and curvature). It is often times sufficient to only know the trend of a response function with respect to design variable changes. For example, we do not need to know the exact value of induced drag, but only that it decreases as aspect ratio increases. The use of multifidelity tools lends itself to this relaxed requirement. The low-fidelity model may be sampled frequently and cheaply to obtain design trends, and high-fidelity model evaluations are reserved for occasions when a more accurate prediction of the actual response value is required.

Several approaches may be taken when combining fidelities in design optimization. Perhaps the least sophisticated methods for combining analyses of varying fidelity are sequential, uncoupled methods. These

*Ph.D. Student, Dept. of Mechanical and Aerospace Engineering, brysond1@udayton.edu, Member AIAA
†Associate Professor, Dept. of Mech. and Aerospace Engineering, Markus.Rumpfkeil@udayton.edu, Senior Member AIAA

methods rank analysis fidelities, with the assumption that higher fidelity analyses better represent truth. However, there is no coupling of information across fidelities. The classic example of this is a series of design optimizations, with the initial design for optimization with a higher fidelity being the optimal design of the previous lower fidelity. An example of this strategy is Rethore et al.[1] in the optimization of offshore wind farms. The advantage of this strategy is that no calibration or correlation between fidelities is required. Ideally, the lower fidelity optimizations would find a design near the high-fidelity optimum, reducing the number of high-fidelity optimization iterations required. However, this behavior is not guaranteed and is a matter of how well the low-fidelity analysis matches the trends of the high-fidelity analysis globally. An example of this mismatch is the case when fidelity means the tightness of multidisciplinary coupling, and loosely or uncoupled analyses may yield very different results than tightly coupled analyses.

In some instances, data of different fidelities may be combined as an ensemble in which all sources are combined to create the best estimate of a response rather than a single fidelity being trusted above all others.[2–4] The advantage of this approach is that it is based on the premise that all models have some error or inadequacy. For example, Allaire, Willcox & Toupet[2] present calculating the maximum likelihood estimator as a multi-fidelity technique. The first iteration begins with the optimal design found using the lowest fidelities for all disciplines. The variances for each model are then estimated, and the disciplines contributing most to the variance of the optimal design are selected for incrementing the level of fidelity. The data for different fidelities within a single discipline are combined to produce a maximum likelihood estimator. The process continues, with each iteration having increased fidelity for the discipline contributing most to the variance in the optimal design, until the design variance falls below a specified tolerance. An advantage of this approach is that it provides an indication of which disciplines would benefit from increased fidelity. Disadvantages, though, are that the approach assumes approximate levels of error for each model are historically known or subjectively assigned, and there is no guarantee that the low-fidelity optimum lies near the high-fidelity optimum.

More commonly, some hierarchy of data is presumed to exist in which one fidelity is more reliable than another. Thus, a method is sought to leverage intensely-sampled low-fidelity data to reduce the number of high-fidelity evaluations required. The most popular approach combining fidelities is through corrections of lower fidelity data to match the high-fidelity response. These corrections may be sought globally, or calculated locally for use in a small region. The calibration of the low-fidelity response may take the form of mappings of either the input space or the output response.

In the input space mapping approach, the inputs to the low-fidelity model are altered in an attempt to elicit matching between the low-fidelity and high-fidelity predictions. This approach is used by Thomas *et al.*[5] in the calculation of a parameter to quantify the goodness or agreement between models of two fidelities. Robinson *et al.*[6] expand the space mapping approach to account for design variables that appear in only one of the model fidelities, and combine it with a surrogate-based optimization.

In some cases, a globally accurate surrogate model is sought to approximate the high-fidelity response across the entire design space. These surrogate models may then serve to evaluate the objective for global, non-gradient based or local, gradient-based optimizations. The global approach is often accompanied by adaptive training point selection and an indicator of what fidelity should be evaluated[7–11] The advantage of this approach is that cheaper, lower fidelity analyses may be used where they are valid, and higher fidelity analyses substituted when required. A disadvantage is that fitting a surrogate model may become challenging as the dimensionality of the design problem increases. Furthermore, the behavior of a response may vary greatly across the design space, making it difficult to fit an accurate global surrogate.

Alternatively, an accurate surrogate model or bridge function may be sought only over a local region of the design space. This approach is usually taken when using methodologies such as the trust region framework applied to approximate models.[12–14] In the trust region framework, a series of optimizations are performed on an approximate problem (such as a surrogate model or corrected low-fidelity function). Each sub-optimization is constrained or bounded to remain within a limited region of the design space. Once the minimizer of the subproblem is found, the trust region is re-centered about the new design, and a new function approximation is formed. The approximate model is selected such that at the center of the trust region (the current design), the surrogate is consistent with the high-fidelity function up to the first derivatives. When the approximate models satisfies these conditions, sequential optimizations on the surrogates will converge provably to an optimum of the high-fidelity function.[12] The size of the trust region is updated with each iteration based upon the performance of the approximate model. If the approximation is a good predictor of the actual response, the trust region may be expanded. If it performs poorly, the

trust region should be contracted. A number of heuristics have been proposed to control the size of the trust region.[13,15–18] Similar heuristics could potentially also be extended to provide an indication of when to switch analysis fidelities.

II. Methodology

II.A. Trust Region Model Management Framework

A multifidelity optimization framework is constructed based on the Trust Region Model Management (TRMM) framework.[12] Trust regions allow for approximate models to be used in place of more expensive high-fidelity models over a limited region of the design space. Thus, a design optimization may be carried out as a sequence of optimizations of the approximate sub-problem. An advantage of this approach is that the sub-problem may be solved by any appropriate numerical optimizer without modification. Here, the limited memory, bounded, Broyden-Fletcher-Goldfarb-Shanno (L-BFGS-B) quasi-Newton method[19,20] is used. Though the algorithm implements a limited-memory estimate of the inverse Hessian, all optimization steps are retained, making the method effectively full BFGS.

The trust region methodology has the convenient property that the series of sub-optimizations is provably convergent to a local optimizer of the high-fidelity problem provided that the approximate problem maintains first-order consistency (i.e., matching of function value and gradients) with the high-fidelity problem.[13] This property is easily achieved by several surrogate modeling techniques. Aside from the construction of the approximate model, one of the challenges of the trust region approach is the selection of the trust region bounds. This limit may take the form of a constraint on the permissible move through the design space in a hypersphere, or as a set of design variable bounds on the subproblem, forming a hypercube. The latter approach is taken here.

For every outer loop iteration, the trust region is re-centered about the current design, and the sizes of the move limits or subproblem bounds are determined via a heuristic.[12] The decision to expand or contract the trust region is based on the ratio of the actual improvement to the expected improvement,

$$\rho^{(k)} = \frac{f\left(\mathbf{x}_c^{(k)}\right) - f\left(\mathbf{x}_*^{(k)}\right)}{\widehat{f}\left(\mathbf{x}_c^{(k)}\right) - \widehat{f}\left(\mathbf{x}_*^{(k)}\right)} \tag{1}$$

where k is the outer iteration count, f and $\hat{f}$ represent high-fidelity and surrogate model evaluations, respectively, and $\mathbf{x}_*$ and $\mathbf{x}_c$ are the location of the sub-optimum and trust region center, respectively. Ratios near unity indicate a high-quality approximate model, suggesting that larger optimization steps are warranted and the trust region should be expanded. In contrast, ratios far from unity indicate that the surrogate is poor, and optimization steps should be taken conservatively and the trust region contracted. Several variations of a heuristic have been proposed for the selection of the trust region size.[15–18] The following criteria are used here:

$$\Delta^{(k+1)} = \begin{cases} 0.5\Delta^{(k)} & \text{if } \rho^{(k)} \leq 0.25 \\ \Delta^{(k)} & \text{if } 0.25 \leq \rho^{(k)} \leq 0.75 \text{ or } 1.25 \leq \rho^{(k)} \\ \gamma\Delta^{(k)} & \text{otherwise,} \end{cases} \tag{2}$$

$$\gamma = \begin{cases} 2 & \text{if } \|x_*^{(k)} - x_c^{(k)}\|_\infty = \Delta^{(k)} \\ 1 & \text{if } \|x_*^{(k)} - x_c^{(k)}\|_\infty < \Delta^{(k)}. \end{cases} \tag{3}$$

The design optimization using trust regions begins with high- and low-fidelity function and gradient evaluations at the initial design point. An initial trust region is formed using a prescribed initial size. The approximate problem is formed (as discussed in Section II.D), and optimized using an optimization algorithm of choice. Once the sub-optimization has converged, the trust region is re-centered about the approximate optimum, and the high-fidelity objective and gradient are evaluated. Convergence of the outer loop optimization is checked. If the outer optimization has not converged, the ratio of actual versus predicted improvement is determined and the trust region expanded or contracted accordingly. Using the data at the new design point, an updated approximate problem is constructed, and the process is repeated until the termination criteria are reached.

II.B. Unified Multifidelity Quasi-Newton Method

While its simplicity makes TRMM easy to implement, it exhibits several weaknesses. First, the progression toward the high-fidelity optimum depends on the accuracy of the successive approximate models. While first-order consistency is enforced at the trust region center point to ensure convergence, the behavior near the trust region bounds may be misleading to the optimizer. To improve the estimate of the high-fidelity function, the trust region must either be reduced in size (limiting the amount of optimization that may be performed, which is the normal recourse), or consistency with the high-fidelity function must be enforced at additional points in the design space. The later solution may be implemented with some approximation methods such as kriging (discussed below) but increases the size of the model fitting problem. Furthermore, early in the optimization when most of the reduction in the objective function is achieved, data for improving the model fit is limited.

The inadequacy of early approximate models is compounded by the fact that information about the high-fidelity problem is not propagated from one sub-optimization to the next, aside from the scaling of the trust region. While data from previous evaluations may provide a better picture of the global design space, TRMM limits the scope to the current approximate model.

To address these issues, a unified, multifidelity quasi-Newton method is proposed. The new approach currently only extends the L-BFGS-B code, though the proposed method can be applied to all optimization algorithms which separate line searching from search direction finding. The method follows the basic algorithm outlined below, and is compared graphically to TRMM in Figure 1. While the unified algorithm may be intrusive to the optimizer implementation (the use of high-fidelity and approximate data must be managed between subroutines), the expectation is that the improved data use leads to faster convergence.

1. The method is started by calculating both the high- and low-fidelity objective responses and gradients at the initial design point. For BFGS, the inverse Hessian is initialized using the identity matrix.

2. Using the inverse Hessian estimate and high-fidelity gradient, the expected (bound constrained) optimal point is calculated as in typical quasi-Newton methods. This expected point minimizes the approximate quadratic model of the objective subject to the side constraints. The high- and low-fidelity functions and gradients are evaluated at this point.

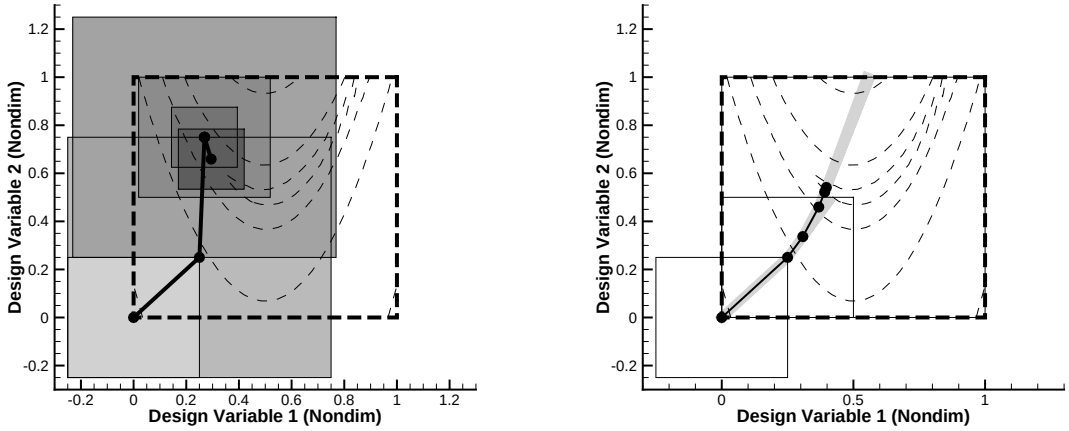
3. An objective function surrogate is constructed satisfying zero- and first-order consistency at both points using additive and multiplicative corrections to the low-fidelity function. Additional data in the vicinity of the current and expected points may be added if the surrogate model will accept it (e.g., as in kriging).

4. A line search is performed in the direction of the expected point using the surrogate model. The resulting point satisfies the strong Wolfe conditions (on the approximate model) or is limited by trust region bounds placed on the surrogate.

5. The high-fidelity response is evaluated at the new point produced by the line search. If the response is reduced, it is accepted; otherwise, it is rejected. In either case, the new data is used to update the inverse Hessian matrix. If the step has been rejected, the data are reversed to create a step moving from the rejected point to the current point so that the inverse Hessian update is applied at the starting point of the next iteration.

6. The expected optimal point and search direction are updated. The previously computed design point nearest to the expected point is used to form a new surrogate model. The procedure is repeated from Step 3 until the termination criteria are reached.

One weakness the proposed method shares with TRMM is that a trust region is still utilized, though in the new approach it simply limits the maximum allowable step size in the line search. The trust region is adjusted according to the same rules as the TRMM, depending on the performance of the approximate model relative to the high-fidelity function. The new approach may provide some alternatives for limiting the step length. For example, a key quantity in the strong Wolfe conditions is the dot product of the gradient and search direction. Comparing this product using gradients from the approximate function versus high-fidelity data may provide an indicator of model quality. However, this and other techniques are left for future research.



(a) TRMM

(b) Unified multifidelity

Figure 1. Comparison of TRMM to proposed unified approach. Dots with solid lines represent optimization path. Bold dashed lines show the side constraints, and light dashed lines are objective function contours. Gray boxes/lines display the approximate model domain for sub-optimization problems (TRMM) or line searches (Unified). Light solid lines illustrate trust region size for each iteration. Note that the trust region encompasses the entire domain starting with the second iteration for the unified approach.

II.C. High-Fidelity Optimization Warm Starting

In gradient-based, high-fidelity optimization, it is often the case that multiple function calls are required in the early line searches while an accurate internal approximation (the inverse Hessian in the case of BFGS) is being built. However, once sufficient data has been acquired, usually the first step requiring only one evaluation is accepted. Thus, at best, multifidelity algorithms will add the cost of a single approximate model build and evaluation on top of the high-fidelity evaluation in the final iterations. Thus, at some point it is desirable to transition to monofidelity optimization to alleviate this burden.

We wish to glean what information we can about the design space from the past multifidelity optimization rather than discarding it. In the new, unified approach, the inverse Hessian already embodies this information. However, TRMM views this as a fresh start from a new point.

The previous objective and gradient information may be fed into the limited-memory BFGS update without actually performing any optimization. Similar to the handling of rejected points in the unified algorithm, pseudo-steps are calculated from each prior point to the current initial design. Each provides an improved picture of the design space, particularly compared to the usual initial guess the identity matrix.

II.D. Multifidelity Surrogate Models

II.D.1. Weighted, Independent Polynomial Chaos Corrections

Because the required form of the correction from low-fidelity to high-fidelity is typically not known *a priori*, a weighted combination of additive and multiplicative forms may be used,[21]

$$\widetilde{f_h}(\mathbf{x}) = w_{Add}^{(k)}\left(f_l(\mathbf{x}) + \widehat{\delta}(\mathbf{x})\right) + w_{Mult}^{(k)}\left(f_l(\mathbf{x}) \cdot \widehat{\alpha}(\mathbf{x})\right). \tag{4}$$

(Here, the notation $\widetilde{(\cdot)}$ indicates the function is a combination of approximate models ($\widehat{(\cdot)}$), but the function itself is not being directly approximated using the surrogate modeling techniques.) To satisfy the conditions of provable convergence[12] to an optimum of the high-fidelity function, the corrected function must be consistent with the high-fidelity function up to the first derivatives at the center point of the trust region. In the weighted model approach, the additive and multiplicative terms are formed independently using linear polynomial chaos expansions which are a linear combination of orthogonal polynomial bases. These corrections each require $N + 1$ pieces of information, where N is the dimensionality of the design space. Sufficient data is gathered from the high- and low-fidelity function and gradient evaluations of a single design point. The two

correction terms are found by fitting the linear models described by

$$\alpha\left(\mathbf{x}\right) = \frac{f_h\left(\mathbf{x}\right)}{f_l\left(\mathbf{x}\right)}, \tag{5}$$

$$\nabla\alpha\left(\mathbf{x}\right) = \nabla\left(\frac{f_h\left(\mathbf{x}\right)}{f_l\left(\mathbf{x}\right)}\right), \tag{6}$$

$$\delta\left(\mathbf{x}\right) = f_h\left(\mathbf{x}\right) - f_l\left(\mathbf{x}\right), \tag{7}$$

and

$$\nabla\delta\left(\mathbf{x}\right) = \nabla\left(f_h\left(\mathbf{x}\right) - f_l\left(\mathbf{x}\right)\right). \tag{8}$$

Having formed the corrective functions, the selection of weighting coefficients remains. Here, we follow the Bayesian model averaging approach presented by Fischer and Grandhi[18] (an alternative approach is provided by Eldred et al.[16]),

$$w_{Add}^{(k)} = \frac{w_{Add}^{(k-1)}\psi_{Add}\left(\mathbf{x}\right)}{w_{Add}^{(k-1)}\psi_{Add}\left(\mathbf{x}\right) + w_{Mult}^{(k-1)}\psi_{Mult}\left(\mathbf{x}\right)}, \tag{9}$$

$$w_{Mult}^{(k)} = \frac{w_{Mult}^{(k-1)}\psi_{Mult}\left(\mathbf{x}\right)}{w_{Add}^{(k-1)}\psi_{Add}\left(\mathbf{x}\right) + w_{Mult}^{(k-1)}\psi_{Mult}\left(\mathbf{x}\right)}, \tag{10}$$

where the individual model likelihoods ψ_i depend on the maximum likelihood estimators of the model variances,

$$\psi_i\left(\mathbf{x}\right) = \left(\frac{1}{2\pi\hat{\sigma}_{i,\mathrm{mle}}^2}\right)^{\frac{n}{2}} \exp\left(-\frac{n}{2}\right), \tag{11}$$

and the maximum likelihood estimator of the variance of each model is

$$\hat{\sigma}_{i,\mathrm{mle}}^2 = \frac{\sum_{j=1}^{n}\left(f_h\left(\mathbf{x}_j\right) - \widehat{f_{h,i}}\left(\mathbf{x}_j\right)\right)^2}{n}. \tag{12}$$

In the initial iteration, the weights are selected to be $w_{Add}^{(0)} = w_{Mult}^{(0)} = 0.5$. In each subsequent iteration, the weights are updated using sample data in a region centered about the current design with design variable ranges spanning twice that of the previous trust region.

II.D.2. Unified Hybrid Polynomial Chaos Corrections

Bryson and Rumpfkeil[22] present a method for generating a unified, hybrid additive-multiplicative polynomial chaos correction. This approach eliminates the necessity of selecting weights for all but the first iteration. To start the optimization, initial weights may be selected to be equal, or an additional point may be provided, such as the center of the domain. Here, the expected point used in the first step of the quasi-Newton method is provided for consistency in comparisons.

In a traditional PCE, a function is approximated by the series

$$\widehat{f}\left(\mathbf{x}\right) = \sum_{k=0}^{P}\beta_k\Psi_k\left(\mathbf{x}\right), \tag{13}$$

where $\Psi_k\left(\mathbf{x}\right)$ are the selected multidimensional bases formed by

$$\Psi_k\left(\mathbf{x}\right) = \prod_{i=1}^{M}\Psi_{a_i^k}\left(x_i\right). \tag{14}$$

The multi-index a_i^k[23] indicates the order of the one-dimensional polynomial in the i^{th} direction for the k^{th} term in the total expansion. The coefficients β_k may be calculated non-intrusively by the collocation method (as is done here) or by the Galerkin spectral projection method.[24] The gradient and Hessian of $\Psi\left(\mathbf{x}\right)$ are readily available[9,25] to admit the use of derivative information in the surrogate modeling procedure.

For the inclusion of multifidelity data in the model fitting, it is assumed that the high-fidelity function is matched by multiplicative and additive corrections to the low-fidelity model,

$$f_h(\mathbf{x}) = f_l(\mathbf{x}) + \alpha(\mathbf{x}) f_l(\mathbf{x}) + \delta(\mathbf{x}). \tag{15}$$

These corrections themselves may be represented by PCEs. Substituting the PCE representation from Equation (13) for the high-fidelity and corrective functions, and rearranging Equation (15) yields

$$\widehat{f_h}(\mathbf{x}) = \sum_{k=0}^{P} \beta_k \Psi_k(\mathbf{x}), \tag{16}$$

$$\widehat{f_l}(\mathbf{x}) = \sum_{k=0}^{P} \beta_k \Psi_k(\mathbf{x}) - f_l(\mathbf{x}) \sum_{k=0}^{Q} \alpha_k \Psi_k(\mathbf{x}) - \sum_{k=0}^{R} \delta_k \Psi_k(\mathbf{x}). \tag{17}$$

If the low-fidelity model correlates well with the high-fidelity model, then the polynomial order of the correction terms should be small compared to the order of the high-fidelity PCE. The least squares-optimal coefficients for the additive and multiplicative corrections and the model of interest may be determined simultaneously using the regression procedure described by Bryson and Rumpfkeil.[22] Once the additive and multiplicative terms are determined, the high-fidelity function is approximated using Equation 4 with unitary weights.

For the trust region framework, the first-order consistency condition is satisfied by fitting the models to the function values and gradients of only two points, one of which is the current center point of the trust region. The second point is the most recently evaluated point if the step was rejected; otherwise, the previous accepted design is used.

II.D.3. Unified Kriging Corrections

Similarly, Han, Görtz, and Zimmermann[26] provide a method for simultaneously determining the best hybrid bridge function using gradient-enhanced kriging. The advantage of kriging over polynomial chaos expansion is that kriging is always interpolative, whereas polynomial chaos becomes a regression when overspecified. Thus, kriging shows promise to provide a more accurate correction of the low-fidelity function over a larger region of the design space by fitting more data points while maintaining the first-order consistency criterion.

The kriging derivation provided by Yamazaki and Mavriplis[27] is summarized here; for a more thorough treatment, the reader is referred to the original sources. For "ordinary" kriging (as opposed to "universal" kriging, where the underlying mean is a low-order polynomial), an interpolatory surrogate model is represented by

$$\widehat{f_k}(\mathbf{x}) = \mu_k + Z_k(\mathbf{x}), \tag{18}$$

where μ is the mean of the training data and $Z(\mathbf{x})$ is a stationary random process. The subscript k indicates the fidelity level, as data may be correlated between fidelities as well as within a fidelity. The random process $Z_k(\mathbf{x})$ satisfies the properties

$$\mathrm{E}(Z_k(\mathbf{x})) = 0, \tag{19}$$

$$\mathrm{Var}(Z_k(\mathbf{x})) = \sigma_k^2, \tag{20}$$

and

$$\mathrm{Cov}\left(Z_k\left(\mathbf{x}^{(i)}\right), Z_n\left(\mathbf{x}^{(j)}\right)\right) = \sigma_k^2 \sigma_n^2 R_{kn}\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right). \tag{21}$$

The correlation matrix, R, is the product of spatial correlation functions Φ in all dimensions evaluated for the distance between points,

$$R\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = \prod_{m=1}^{M} \Phi\left(\theta_m, d_m^{ij}\right), \tag{22}$$

$$d_m^{ij} = \|x_m^{(i)} - x_m^{(j)}\|_2. \tag{23}$$

Here, Wendland's C4 radial basis function[28] is implemented,

$$\Phi\left(\theta_m, d_m^{ij}\right) = \begin{cases} \left(1 - \theta_m d_m^{ij}\right)^6 \left(35(\theta_m d_m^{ij})^2 + 18\theta_m d_m^{ij} + 3\right) & \text{for } d_m^{ij} \le \frac{1}{\theta_m} \\ 0 \quad \text{otherwise} \end{cases}. \tag{24}$$

The correlations within a single fidelity level are the same as given in Equation 22, while the correlations between fidelity levels are multiplied by a relaxation factor to improve numerical conditioning when training points between fidelities are very near each other,

$$R_{kn} = \begin{cases} R, & k = n \\ \rho R, & k \neq n \end{cases} . \tag{25}$$

Here, ρ is taken to be 0.9999. Following Yamazaki and Mavriplis,[27] the means and variances of the multifidelity system may be determined analytically, but the parameters θ_m must be determined numerically. Using direct co-kriging, Yamazaki and Mavriplis[27] also provide for the inclusion of gradient information in the parameter determination by taking derivatives of the correlation function.

The bridge function for multifidelity kriging is similar to that of polynomial chaos in Equation 15, except that the multiplicative term α is a low-order polynomial (here, linear), and the additive term δ is a kriging model. A multi-step approach is taken in which a kriging model of the low-fidelity data is generated, followed by simultaneous estimation of the coefficients for the multiplicative and additive corrections. Similar to the model's means and variances, the coefficients $\alpha(\mathbf{x})$ may be determined analytically, and updated with each iteration during the estimation of θ_m.[26,29] The final high-fidelity approximation is then taken to be the combination of the low-fidelity kriging, multiplicative polynomial regression, and additive kriging bridge functions.

III. Results and Discussion

The multifidelity optimization methods described above are demonstrated on five side-constrained, analytic functions[30] with varying characteristics (polynomial, sinusoidal, exponentiation; unimodal and multimodal). For all the test cases, achieving a projected gradient norm of 10^{-1} has been found to indicate that the optimizer will drive the gradient norm to machine zero in a relatively small number of iterations. Thus, this tolerance is the nominal condition for transition to monofidelity optimization. The final convergence tolerance is 10^{-6}.

As a failsafe, the optimization will transition to high-fidelity if the trust region becomes too small, indicating disagreement between the high-fidelity and approximate models. The threshold criterion is a trust region at or below 1% of the design variable domain for three subsequent iterations. The initial trust region for both TRMM and unified quasi-Newton algorithms is 25% of the domain.

Each TRMM sub-optimization requests the gradient norm be reduced by one order of magnitude from the starting point. Tighter convergence tolerances are not required since the goal is to find a better candidate for the high-fidelity optimum, not to find the true optimum of the approximate model. The number of approximate function evaluations are also limited to twenty-one per sub-problem, which was sufficient in most cases to find a new candidate point. The L-BFGS-B code internally limits the line search to twenty function calls, and exits in an error status if a suitable point cannot be found. If the TRMM fails to produce a new point (typically due to the objective function being dominated by round-off error), the optimizer immediately transitions to monofidelity optimization.

While L-BFGS-B is a limited-memory formulation, the history is set sufficiently large to retain all prior step information, making the algorithm effectively full BFGS for both the TRMM sub-optimization and the multifidelity quasi-Newton optimization.

Each test case takes the average of five starting points—four design space corners and the low-fidelity optimum (for the multimodal low-fidelity Styblinski-Tang function, the optimum near the high-fidelity global optimum is used). The four corners used are

$$\begin{aligned} \mathbf{x}_1^{(0)} &= [0, 0, 0, \ldots, 0] \\ \mathbf{x}_2^{(0)} &= [1, 1, 1, \ldots, 1] \\ \mathbf{x}_3^{(0)} &= \left[0, 1, 0, \ldots, \frac{1 + (-1)^N}{2}\right] \\ \mathbf{x}_4^{(0)} &= \left[1, 0, 1, \ldots, \frac{1 + (-1)^{N-1}}{2}\right] . \end{aligned} \tag{26}$$

III.A. Analytic Test Functions

Five analytic functions in multiple dimensions serve as the test suite for comparing the different optimization approaches. Each has a corresponding "low-fidelity" function that mimics the overall global trends, but lacks local detail. All of the low-fidelity functions have minima in different locations than the high-fidelity functions, except for the Brown function where they are coincident. Two high-fidelity functions (Rosenbrock and Brown) have single minima, while the others are multimodal. The test suite encompasses a variety of mathematical forms, including polynomials, sinusoids, and exponents.

The first test case, the McCormick function (Eqn. 27), is limited to two dimensions and has two local minima in opposite quadrants of the design space. A second-order polynomial approximation (Eqn. 28) is used for the low-fidelity function.

$$\begin{aligned} f_{h,1}(\mathbf{x}) = &\sin(x_1 + x_2) + (x_1 - x_2)^2 - \frac{3}{2}x_1 + \frac{5}{2}x_2 + 1 \\ &-1.5 \leq x_1 \leq 4 \\ &-3 \leq x_2 \leq 3 \end{aligned} \tag{27}$$

$$f_{l,1}(\mathbf{x}) = (x_1 - x_2)^2 + 2x_1 + 2x_2 \tag{28}$$

The N-dimensional Styblinski-Tang function (Eqn. 29) has multiple local minima, with the single global minimum near the lower bounds of all the design variables. Here, the low-fidelity approximation is a product of a quartic and a cosine (Eqn. 30).

$$\begin{aligned} f_{h,2}(\mathbf{x}) = &\frac{1}{2}\sum_{i=1}^{N} x_i^4 - 16x_i^2 + 5x_i \\ &-5 \leq x_i \leq 5 \end{aligned} \tag{29}$$

$$f_{l,2}(\mathbf{x}) = \sum_{i=1}^{N} x_i^4 \cos\left(\frac{2\pi x_i}{5}\right) \tag{30}$$

The Brown function (Eqn. 31) has a single optimum at the origin, and is characterized by an extremely steep gradient in one corner of the design space, while the basin around the minimum is relatively flat. An approximation to this function is an exponential of the sum of squares of the design variables (Eqn. 32).

$$\begin{aligned} f_{h,3}(\mathbf{x}) = &\sum_{i=1}^{N-1} (x_i^2)^{x_{i+1}^2+1} + (x_{i+1}^2)^{x_i^2+1} \\ &-1 \leq x_i \leq 2 \end{aligned} \tag{31}$$

$$f_{l,3}(\mathbf{x}) = 10^{-3} + \sum_{i=1}^{N-1} \exp\left(x_i^2 + x_{i+1}^2\right) \tag{32}$$

An N-dimensional trigonometric function (Eqn. 33) becomes increasingly multimodal and challenging to navigate as dimensionality increases. Several local optima exist at the bottom of a global basin. The low-fidelity approximation is an N-dimensional quadratic function offset slightly from the true, global optimum (Eqn. 32).

$$\begin{aligned} f_{h,4}(\mathbf{x}) = &\sum_{i=1}^{N} \left[N - \sum_{j=1}^{N} \cos x_j + i\left(1 - \cos x_i - \sin x_i\right) \right]^2 \\ &-1.5 \leq x_i \leq 3 \end{aligned} \tag{33}$$

$$f_{l,4}(\mathbf{x}) = 10^{-3} + \sum_{i=1}^{N} 5\left(x_i - \frac{1}{4}\right)^2 \tag{34}$$

The multi-dimensional Rosenbrock function (Eqn. 35) is characterized by a steep valley with a slight gradient towards a single optimum. The function is approximated in low-fidelity by another fourth-order polynomial (Eqn. 36)

$$f_{h,5}(\mathbf{x}) = \sum_{i=1}^{N-1} (1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2 \qquad -2 \leq x_i \leq 2 \tag{35}$$

$$f_{l,5}(\mathbf{x}) = 1 + \sum_{i=1}^{N} 50 x_i^2 \left[(x_{i+1} - 2)^2 + 2 \right] \tag{36}$$

III.B. High-Fidelity Optimization Warm Starting

Warm starts of the different test functions demonstrate the impact of leveraging prior data to approximate the response function curvature for monofidelity optimization. To begin, the standard L-BFGS-B algorithm reduces the objective gradient norm to 10^{-1}. From this weakly-converged point, two new optimizations start—one with the proposed warm start technique, and one with a fresh start.

Figure 2 shows the savings produced by warm starting in terms of high-fidelity function calls averaged over the five starting points. For functions with attractive basins that are easily optimized, the impact is small, on the order of one or two function calls. However, for functions that are more difficult to navigate, such as the Trig and Rosenbrock functions, the optimizer clearly benefits from the improved starting condition. In particular, the Trig function becomes increasingly multimodal and difficult as dimensionality increases. Correspondingly, the warm start savings increase as the cold-started optimization requires more and more function calls to converge. Given the potential performance improvement and the risk of only minimally hindering the optimizer, the warm starting procedure is adopted for all results in this paper.

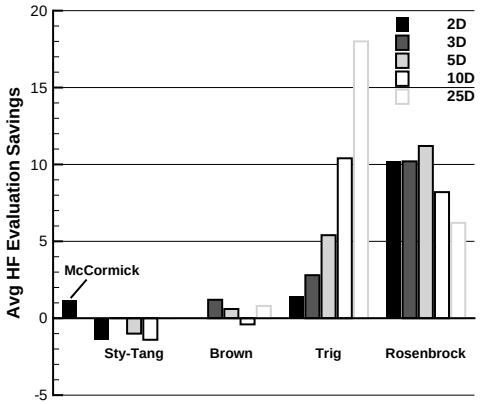


Figure 2. High-fidelity function evaluations saved by warm starting optimization with prior data, averaged over five different starting points. Negative values indicate an increase in function calls.

III.C. Hybrid versus Weighted Bridge Functions

The selection of bridge function drives the behavior of the approximate model and consequently the performance of the multifidelity algorithm. Regardless of the underlying mathematical form (kriging, polynomial, etc.), a decision must be made whether the calibration to the low-fidelity data should be additive, multiplicative, or a combination. The hybrid approach has the advantage of not requiring an a priori choice of the best form, but requires additional data beyond the function and gradient at the current point. Here, the two techniques of creating hybrid polynomial models from Sections II.D.1 and II.D.2 are compared. While the weighting approach of Fischer and Grandhi[18] was originally used for kriging models, the comparison made here is limited to PC.

Figure 3 shows the relative performance of TRMM optimization using weighted combinations versus hybrid corrections, in terms of high-fidelity function calls saved normalized by the cost of monofidelity optimization. In some cases, the weighted approach was slightly cheaper than the unified hybrid approach, but more significant are the cost increases associated with the poor performance of the weighed model for the Styblinski-Tang and Brown functions. Thus, the unified hybrid approach is utilized going forward for both TRMM and unified quasi-Newton optimization and for kriging and PC bridge functions.

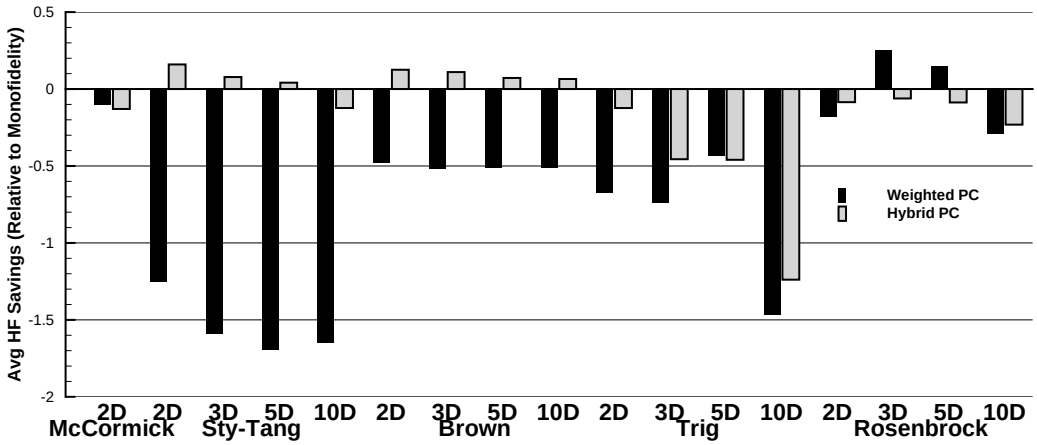


Figure 3. Comparison of high-fidelity function calls saved (normalized by cost of monofidelity optimization) for TRMM with hybrid versus weighed corrections, averaged over five starting points. Negative values indicate an increase in function evaluations.

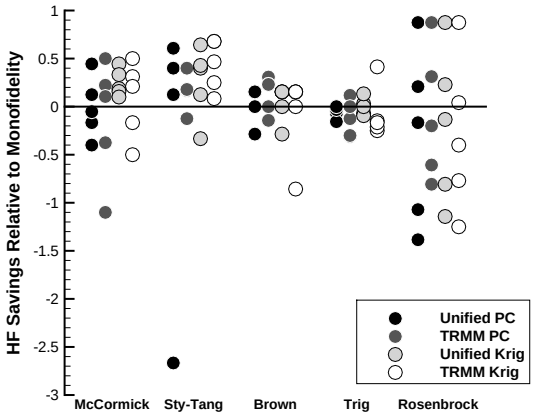
III.D. Unified Multifidelity Method versus TRMM

Comparisons of the unified and TRMM multifidelity techniques on the various test cases indicate mixed performance, dependent on function and starting point. The total number of cases where the new approach requires fewer (or the same) number of high-fidelity evaluations versus TRMM are tabulated in Tables 1 and 2 for PC and kriging surrogate corrections. (The kriging bridge functions are limited to five dimensions due to poor scalability.) Overall, the unified quasi-Newton technique with PC performed better than TRMM in 49% of cases and as well as TRMM in 18% of cases. Using kriging, the new approach performed better than TRMM in 55% of cases and as well as TRMM in 18% of cases. The superiority of the new method increased slightly in higher-dimensions, though further study is needed to determine if this trend continues for large-scale problems. Detailed tables of individual optimization runs are provided as an appendix.

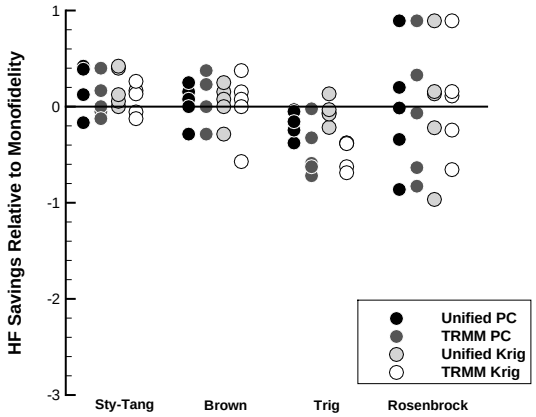
Table 1. Number of cases (out of 105) where the unified quasi-Newton approach required fewer (or the same number) of high-fidelity evaluations compared to TRMM using hybrid PC.

Function	2D	3D	5D	10D	25D	Total
McCormick	2 (0)	—	—	—	—	2 (0)
Sty-Tang	3 (1)	3 (1)	3 (1)	3 (1)	3 (1)	15 (5)
Brown	0 (1)	0 (2)	0 (2)	1 (2)	1 (1)	2 (8)
Trig	2 (0)	3 (0)	5 (0)	5 (0)	5 (0)	20 (0)
Rosenbrock	1 (1)	2 (1)	1 (2)	4 (1)	4 (1)	12 (6)
Total	8 (3)	8 (4)	9 (5)	13 (4)	13 (3)	51 (19)

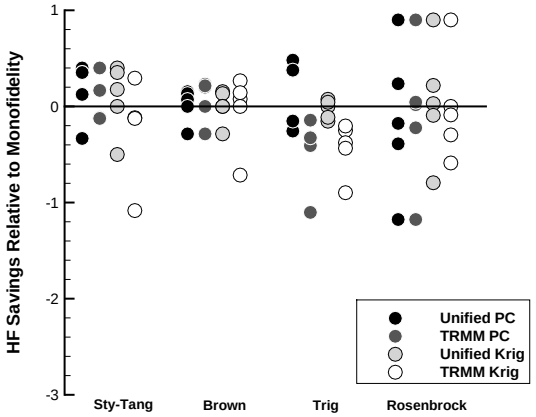
Figure 4 provides the savings of the multifidelity methods relative to monofidelity optimization in terms of high-fidelity evaluations. In most of the cases for the McCormick, Styblinski-Tang, and Brown functions, multifidelity optimization results in fewer high-fidelity evaluations, on average, compared to monofidelity optimization. Cases requiring more function evaluations are indicated by negative savings. This situation occurs most often with the Trig and Rosenbrock functions and in certain instances for the other three problems.



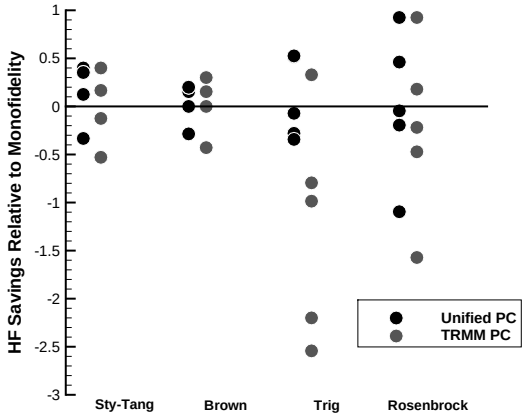
(a) 2D



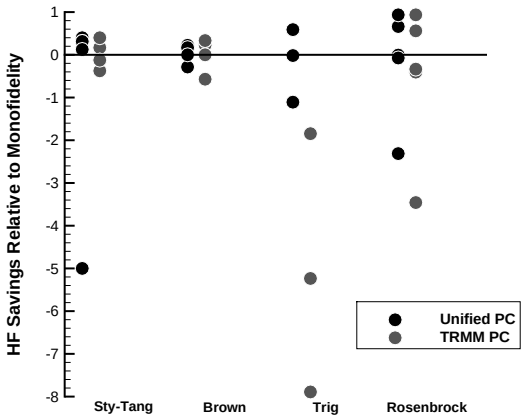
(b) 3D



(c) 5D



(d) 10D



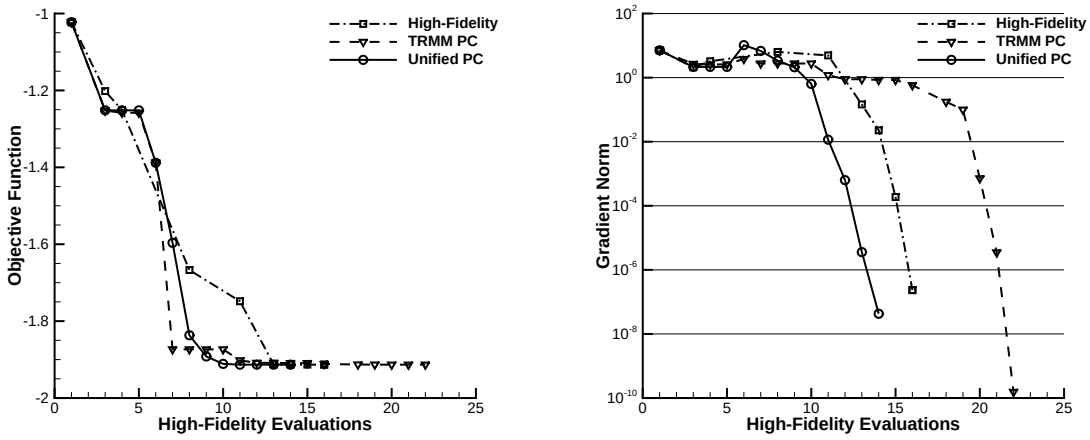
(e) 25D

Figure 4. High-fidelity function call savings of multifidelity optimization normalized by calls required for monofidelity optimization.

Table 2. Number of cases (out of 65) where the unified quasi-Newton approach required fewer (or the same number) of high-fidelity evaluations compared to TRMM using hybrid kriging.

Function	2D	3D	5D	Total
McCormick	2 (0)	—	—	2 (0)
Sty-Tang	0 (0)	4 (0)	4 (0)	8 (0)
Brown	1 (4)	1 (3)	2 (1)	4 (8)
Trig	4 (0)	5 (0)	5 (0)	14 (0)
Rosenbrock	3 (1)	2 (2)	3 (1)	8 (4)
Total	10 (5)	12 (5)	14 (2)	36 (12)

Optimizing the McCormick function with either multifidelity method using hybrid PC is more costly on average than using high-fidelity data alone, though only marginally so using the unified approach. Initially, the optimization progresses well. However, near the global optimum, the low-fidelity gradient opposes the high-fidelity trend and tries to push the optimizer toward the boundary. The algorithm rejects many candidate points because they result in a function increase, reducing the size of the trust region each time. This wasted effort leads to higher computational cost. A representative convergence history is provided in Figure 5.



(a) Objective function

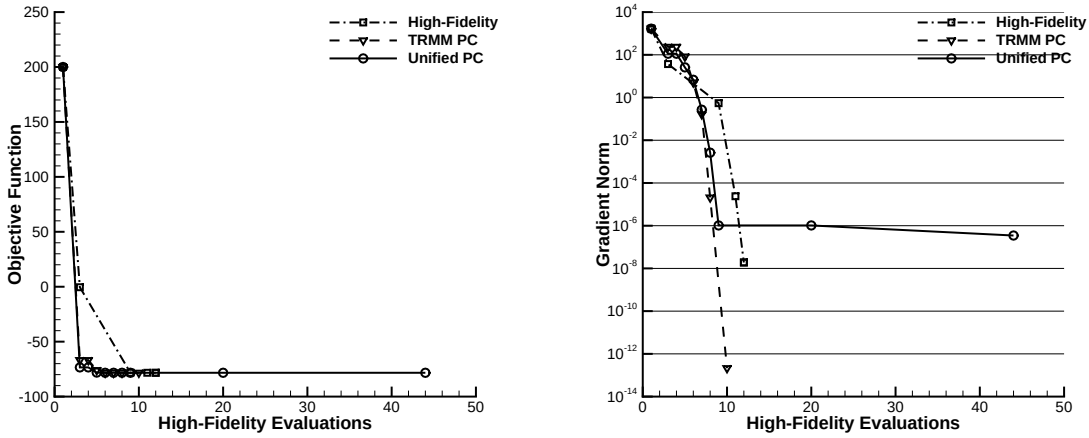
(b) Gradient norm

Figure 5. Representative convergence history for the McCormick function. The multifidelity methods cheaply reduce the objective function, but rejected points add to the overall cost as the optimizer fine tunes the design.

Using hybrid kriging as a bridge function, however, produces favorable results for the McCormick function. Because kriging can build an interpolatory model using more than two points, it better captures the error between the approximate and exact functions, and finds the optimum more efficiently. The preservation of the inverse Hessian estimate between iterations augments its efficiency.

The average savings for Styblinski-Tang indicate that the unified method outperforms TRMM in three, five, and ten dimensions. In two and twenty-five dimensions, four of the five starts reduce high-fidelity cost while one start dramatically increases it. In these cases, the multifidelity algorithm performed as expected and transitioned normally to monofidelity optimization. The optimizer then quickly reduced the gradient norm to between 10^{-5} and 10^{-6}. The L-BFGS-B method then required on the order of twenty to forty function evaluations to cross the convergence threshold. Thus, the poor performance is not attributable to the multifidelity algorithm, but rather to the high-fidelity algorithm for this particular problem. This behavior is highlighted in Figure 6. The same behavior is observed in select monofidelity cases, which contributed to some of the observed multifidelity savings.

For the Brown function, on average, TRMM with PC saves more function evaluations than the unified quasi-Newton method, whereas the opposite is true for kriging corrections. The challenge for the L-BFGS-B optimizer in this case is that the approximate quadratic model does not accurately reflect the exponential



(a) Objective function

(b) Gradient norm

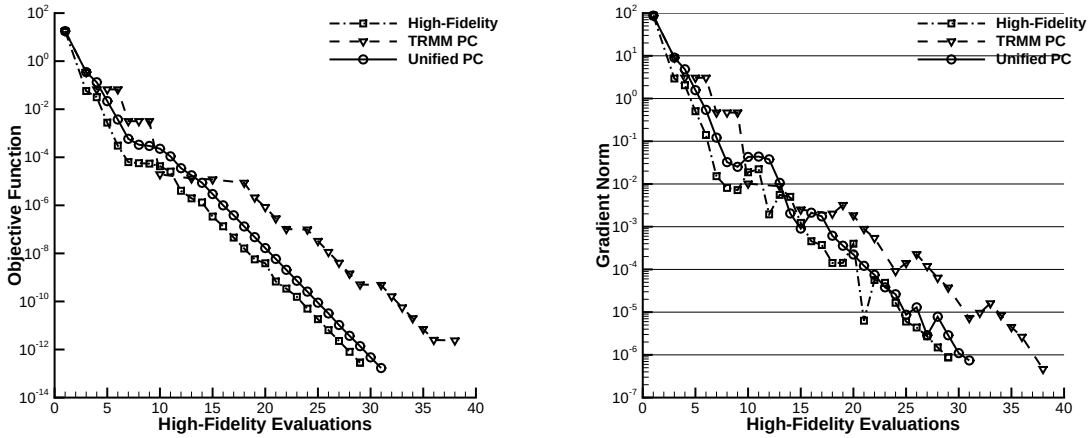
Figure 6. Representative convergence history for the Styblinski-Tang function. The monofidelity optimizer has difficulty driving the gradient norm below the convergence threshold in the final iterations, erasing multifidelity savings.

function behavior, leading to multiple, sequential steps in the same direction. This behavior is inherited by the unified multifidelity quasi-Newton method, and only one or two calls to the low-fidelity model are made per iteration. TRMM, on the other hand, performs more approximate optimization per iteration, finding the optimum more quickly for this problem. This additional approximate optimization actually leads to TRMM's worse performance when using kriging. While the unified approach constrains the search direction, TRMM allows the sub-optimization to explore any spurious oscillations that may arise in the approximate model, pushing the optimizer in less useful directions.

For the Trig function, the majority of function evaluations are spent fine tuning the design after the optimizer quickly finds the vicinity of the optimum as illustrated in Figure 7. In the early iterations, the unified multifidelity optimizer performs similarly to monofidelity optimization, and thus accrues only small gains or losses in terms of function evaluation savings. TRMM, on the other hand, behaves poorly for this problem, requiring very small steps before bailing out due to the minimum trust region size constraint. TRMM bails out for all starting points (PC and kriging) in three, five, ten, and twenty-five dimensions; it triggers the low-level convergence threshold for all starts in two dimensions. In comparison, the unified method bails out for five starts in twenty-five dimensions, four starts in ten dimensions, one start (PC) or none (kriging) in five dimensions, and achieved the desired convergence level for all starts in three and two dimensions.

As anticipated for Rosenbrock's function, the trust regions for both approaches fall below the minimum acceptable size in all cases once the optimizer finds the gently sloped, narrow valley. The multifidelity quasi-Newton method is better able to navigate the valley due to its better search direction, though, and performs significantly more optimization than TRMM before bailing out. In some cases, this leads to increased high-fidelity cost over TRMM, while in others the improved transition point decreases the overall cost.

While low-fidelity function evaluations are expected to be more affordable than their high-fidelity counterparts, their cost may not be negligible. Figure 8 reports the effort spent on approximate model evaluations. First, the low-fidelity cost ratio (relative to the high-fidelity evaluation) indicates the maximum cost allowable to not counteract any high-fidelity evaluation reductions. Negative ratios may arise when multifidelity optimization actually requires more high-fidelity function evaluations rather than fewer. These values are omitted as they suggest that the prediction generates resources rather than expending them. In the majority of cases, the unified approach allows more expensive approximate models than TRMM. However, the negative savings confound with the low-fidelity cost. Thus, the absolute number of low-fidelity evaluations may present a clearer picture of relative performance. In all the test problems, the unified approach requires fewer approximate model calls on average than TRMM, with differences of up to a full order of magnitude.



(a) Objective function

(b) Gradient norm

Figure 7. Representative convergence history for the trigonometric function. The majority of high-fidelity evaluations are spent navigating the nonlinearity at the bottom of the attractive basin.

IV. Summary and Future Work

A new, unified multifidelity optimization approach based on quasi-Newton algorithms has been presented. The approach maintains an estimate of the objective inverse Hessian and bases the search direction purely on high-fidelity data. Using calibrated low-fidelity data in the line search provides computational savings over traditional optimization methods. Any choice of surrogate model may be utilized provided that the approximate model matches the high-fidelity data up to first derivatives at the current design point. Interpolatory polynomial chaos and kriging models have been demonstrated. For PC, a hybrid additive-multiplicative form has been shown to be more effective than using weighted individual calibrations.

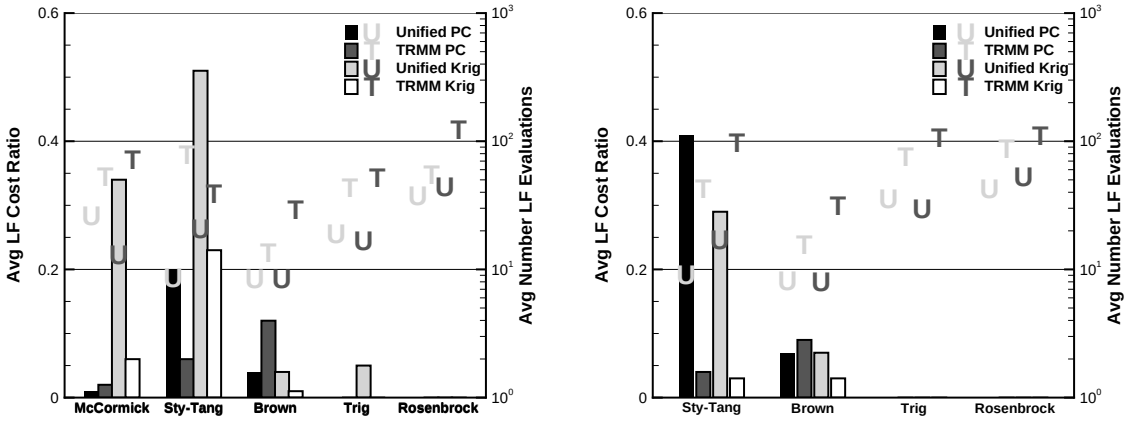
As part of the multifidelity framework, a warm starting technique has been demonstrated to transition from multifidelity to monofidelity optimization. The transition is motivated by the ability of the optimizer to pick good candidate points after sufficient data has been collected, after which point using approximate models only adds overhead and error. Using the high-fidelity data already gathered to initialize the approximate inverse Hessian has been shown to speed convergence in most cases.

The performance of the new method has been compared to a traditional Trust Region Model Management framework on a variety of analytic test functions. The new technique reached optimal points in fewer high-fidelity function evaluations than TRMM in 69% of cases. The unified method also required significantly fewer low-fidelity evaluations over an average of five starting points on all the test problems. This is a significant advantage when the cost of generating approximate data is not negligible. Additionally, TRMM does not preserve any information about the objective function from iteration to iteration (aside from adjusting the trust region size), forcing the optimizer to use very small step sizes for challenging or deceptive functions. Preserving the approximate inverse Hessian allows the unified approach to select better search directions and take longer steps.

The unified approach also maintains the notion of an expected optimal point. This concept allows the algorithm to be forward-looking when selecting data to augment the current design point in building corrective functions. In contrast, TRMM is backward-looking to the previous design, or calibration parameters must be estimated over a set of prior data.

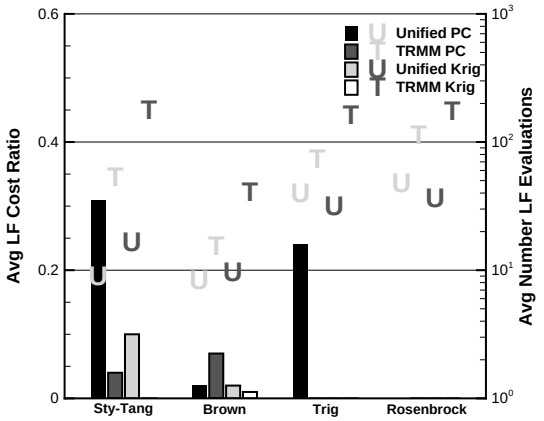
TRMM holds the advantage of allowing optimizers to appear as black boxes. The new approach, on the other hand, requires slight modifications to manage the use of high-fidelity and approximate data.

While the new, unified algorithm is currently implemented for unconstrained quasi-Newton optimization, application to constrained algorithms, such as sequential quadratic programming, will be explored in future work. The new technique also still depends on the trust region that is the crux of TRMM. Currently this constraint is determined heuristically. Methods of estimating an appropriate trust region based on data already gathered is an additional topic of future research.

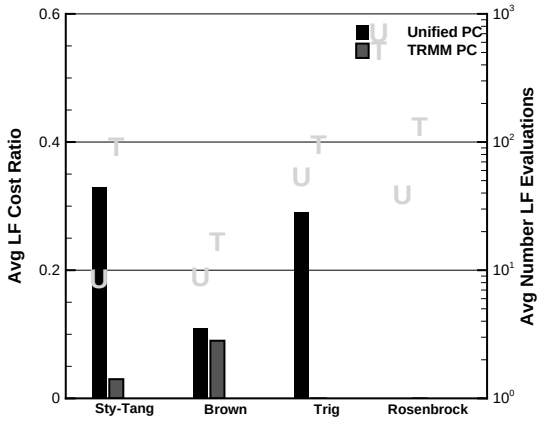


(a) 2D

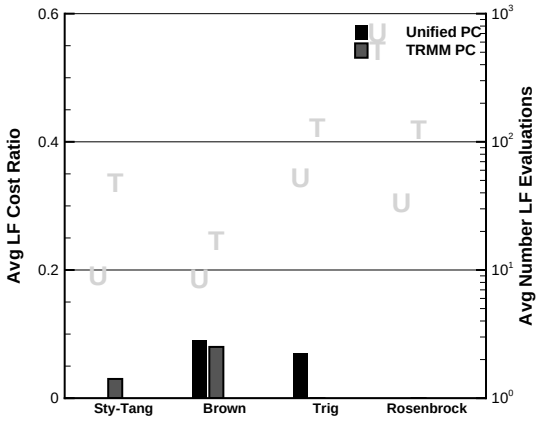
(b) 3D



(c) 5D



(d) 10D



(e) 25D

Figure 8. Low-fidelity function calls required for multifidelity optimization, averaged over five starting points. Bars show maximum allowable cost (left axis) of low-fidelity evaluation relative to high-fidelity savings over monofidelity optimization (negative savings omitted). Symbols indicate average number of low-fidelity evaluations required (right axis) for multifidelity optimization.

Acknowledgment

This work was supported by the Air Force Research Laboratory, Aerospace Systems Directorate under the Civilian Academic Degree Payment (CADP) program.

References

[1]Réthoré, P.-e., Fuglsang, P., Larsen, G. C., Buhl, T., Larsen, T. J., and Madsen, H. A., "TopFarm : Multi-fidelity Optimization of Offshore Wind Farm," *Twenty-first (2011) International Offshore and Polar Engineering Conference*, Vol. 8, 2011, pp. 516–524.

[2]Allaire, D. L., Willcox, K. E., and Toupet, O., "A Bayesian-Based Approach to Multifidelity Multidisciplinary Design Optimization," *13th AIAA/ISSMO Multidiscipinary Analysis & Optimization Conference*, No. September, 2010.

[3]March, A. and Willcox, K., "Provably Convergent Multifidelity Optimization Algorithm Not Requiring High-Fidelity Derivatives," *AIAA Journal*, Vol. 50, No. 5, 2012, pp. 1079–1089.

[4]Allaire, D. and Willcox, K., "A Mathematical and Computational Framework for Multifidelity Design and Analysis With Computer Models," *International Journal for Uncertainty Quantification*, Vol. 4, No. 1, 2014, pp. 1–20.

[5]Thomas, J., Alyanak, E., Agte, J., and Camberos, J., "Quantifying Uncertainty Across Fidelity Levels in the Design of Aerospace Systems," *AIAA Aviation*, AIAA, Los Angeles, CA, 2013.

[6]Robinson, T. D., Eldred, M. S., Willcox, K. E., and Haimes, R., "Surrogate-Based Optimization Using Multifidelity Models with Variable Parameterization and Corrected Space Mapping," *AIAA Journal*, Vol. 46, No. 11, 2008, pp. 2814–2822.

[7]Forrester, A. I. J. and Keane, A. J., "Recent advances in surrogate-based optimization," *Progress in Aerospace Sciences*, Vol. 45, 2009, pp. 50–79.

[8]Jo, Y. and Choi, S., "Variable - Fidelity Aerodynamic Design Using Gradient - Enhanced Kriging Surrogate Model with Regression," *52nd AIAA Aerospace Sciences Meeting*, No. January, National Harbor, MD, 2014, pp. 1–22.

[9]Boopathy, K. and Rumpfkeil, M. P., "Unified Framework for Training Point Selection and Error Estimation for Surrogate Models," *AIAA Journal*, Vol. 53, No. 1, 2015, pp. 215–234.

[10]Jo, Y., Yi, S., Choi, S., and Lee, D., "Variable-Fidelity Design Using Kriging Surrogate Model with Fidelity Indicator," *56th AIAA Structures, Structural Dynamics, and Materials Conference*, AIAA, 2015.

[11]Liem, R. P., Mader, C. A., and Martins, J. R. R. A., "Surrogate models and mixtures of experts in aerodynamic performance prediction for aircraft mission analysis," *Aerospace Science and Technology*, Vol. 43, 2015, pp. 126–151.

[12]Lewis, R. M., "A trust region framework for managing approximation models in engineering optimization," *AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 1996, pp. 1053–1055.

[13]Alexandrov, N. M., Dennis, J. E. J., Lewis, R. M., and Torczont, V., "A trust-region framework for managing the use of approximation models in optimization," *Structural Optimization*, Vol. 15, 1998, pp. 16–23.

[14]Alexandrov, N. M., Lewis, R. M., Gumbert, C. R., Green, L. L., and Newman, P. A., "Approximation and Model Management in Aerodynamic Optimization with Variable-Fidelity Models Introduction," *Journal of Aircraft*, Vol. 38, No. 6, 2001, pp. 1093–1101.

[15]Rodriguez, J. F., Renaud, J. E., and Watsen, L. T., "Convergence of trust region augmented Lagrangian methods using variable fidelity approximation data," *Structural Optimization*, Vol. 15, No. 3-4, jun 1998, pp. 141–156.

[16]Eldred, M. S., Giunta, A. A., and Collis, S. S., "Second-Order Corrections for Surrogate-Based Optimization with Model Hierarchies," *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, No. September, 2004, p. 4457.

[17]Gano, S. E., Renaud, J. E., Martin, J. D., and Simpson, T. W., "Update Strategies for Kriging Models for Use in Variable Fidelity Optimization," *46th AIAA Structures, Structural Dyamics & Materials Conference*, 2005.

[18]Fischer, C. C. and Grandhi, R. V., "A Surrogate-Based Adjustment Factor Approach to Multi-Fidelity Design Optimization," *17th AIAA Non-Deterministic Approaches Conference*, 2015.

[19]Zhu, C., Byrd, R. H., Lu, P., and Nocedal, J., "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization," *ACM Transactions on Mathematical Software*, Vol. 23, No. 4, 1997, pp. 550–560.

[20]Morales, J. L. and Nocedal, J., "Remark on Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-Scale Bound Constrained Optimization," *ACM Transactions on Mathematical Software*, Vol. 38, No. 1, 2011, pp. 7:1–7:4.

[21]Gano, S. E., Renaud, J. E., and Sanders, B., "Hybrid Variable Fidelity Optimization by Using a Kriging-Based Scaling Function," *AIAA Aviation*, Vol. 43, No. 11, 2005, pp. 2422–2430.

[22]Bryson, D. E. and Rumpfkeil, M. P., "Variable-Fidelity Surrogate Modeling of Lambda Wing Transonic Aerodynamic Performance," *54th AIAA Aerospace Sciences Meeting*, No. January, 2016, pp. 1–19.

[23]Eldred, M. S., Webster, C. G., and Constantine, P. G., "Evaluation of Non-Intrusive Approaches for Wiener-Askey Generalized Polynomial Chaos," *49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Vol. 2008, No. April, 2008, pp. 1–22.

[24]Ng, L. W. T. and Eldred, M. S., "Multifidelity Uncertainty Quantification Using Non-Intrusive Polynomial Chaos and Stochastic Collocation," *53rd AIAA Structures, Structural Dynamics, and Materials Conference*, No. April, AIAA, 2012, pp. 1–17.

[25]Li, Y., Anitescu, M., Roderick, O., and Hickernell, F., "Orthogonal Bases for Polynomial Regression With Derivative Information in Uncertainty Quantification," *International Journal for Uncertainty Quantification*, Vol. 1, No. 4, 2011, pp. 297–320.

[26]Han, Z.-h., Görtz, S., and Zimmermann, R., "Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function," *Aerospace Science and Technology*, Vol. 25, No. 1, 2013, pp. 177–189.

[27]Yamazaki, W. and Mavriplis, D. J., "Derivative-Enhanced Variable Fidelity Surrogate Modeling for Aerodynamic Functions," *AIAA Journal*, Vol. 51, No. 1, 2013, pp. 126–137.

[28]Wendland, H., "Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree," *Advances in Computational Mathematics*, Vol. 4, No. 1, 1995, pp. 389–396.

[29]Rumpfkeil, M. P. and Beran, P. S., "Construction of Multi-Fidelity Surrogate Models for Aerodynamic Databases," *9th International Conference on Computational Fluid Dynamics*, Istanbul, 2016.

[30]Jamil, M. and Yang, X.-S., "A Literature Survey of Benchmark Functions For Global Optimization Problems," *Int. Journal of Mathematical Modelling and Numerical Optimisation*, Vol. 4, No. 2, 2013, pp. 150–194.

Appendix: Tables of Multifidelity Optimization Results

Legend:

HF_{-6}	High-fidelity evaluations to reduce gradient norm to 10^{-6}
HF_{-1}	High-fidelity evaluations to reduce gradient norm to 10^{-1}
LF_{-1}	Low-fidelity evaluations to reduce gradient norm to 10^{-1}
Rel Save	High-fidelity evaluation savings relative to monofidelity cost
Max Cost	Ratio of high-fidelity savings to low-fidelity evaluations indicating relative cost to break even
†	Case failed to converge
	Blank cells indicate that a required case did not converge

Table 3. Results of multifidelity optimization with hybrid PC bridge for McCormick function.

Init Pt	2D Unified					2D TRMM				
	HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	14	11	20	0.13	0.10	22	16	81	−0.38	−0.07
2	10	7	29	0.44	0.28	9	6	45	0.50	0.20
3	21	19	31	−0.17	−0.10	14	10	33	0.22	0.12
4	20	18	30	−0.05	−0.03	17	12	50	0.11	0.04
5	14	11	21	−0.40	−0.19	21	16	53	−1.10	−0.21

Table 4. Results of multifidelity optimization with hybrid kriging bridge for McCormick function.

Init Pt	2D Unified					2D TRMM				
	HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	13	10	15	0.19	0.20	11	8	80	0.31	0.06
2	10	7	10	0.44	0.80	9	5	32	0.50	0.28
3	12	10	14	0.33	0.43	21	14	97	−0.17	−0.03
4	16	14	16	0.16	0.19	15	10	80	0.21	0.05
5	9	6	10	0.10	0.10	15	9	69	−0.50	−0.07

Table 5. Results of multifidelity optimization with hybrid PC bridge for Styblinski-Tang function.

Init Pt	2D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	2D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	44	8	9	−2.67	−3.56	10	8	28	0.17	0.07
2	9	7	10	0.40	0.60	9	7	30	0.40	0.20
3	11	8	9	0.61	1.89	23	17	155	0.18	0.03
4	11	8	9	0.61	1.89	23	17	158	0.18	0.03
5	7	5	6	0.13	0.17	9	7	23	−0.13	−0.04
Init Pt	3D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	3D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	14	8	9	−0.17	−0.22	10	8	28	0.17	0.07
2	9	7	10	0.40	0.60	9	7	30	0.40	0.20
3	11	8	11	0.42	0.73	20	15	51	−0.05	−0.02
4	11	8	9	0.39	0.78	18	13	81	0.00	0.00
5	7	5	6	0.13	0.17	9	7	23	−0.13	−0.04
Init Pt	5D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	5D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	16	8	9	−0.33	−0.44	10	8	28	0.17	0.07
2	9	7	10	0.40	0.60	9	7	30	0.40	0.20
3	11	8	11	0.35	0.55	19	14	85	−0.12	−0.02
4	11	8	9	0.35	0.67	19	14	104	−0.12	−0.02
5	7	5	6	0.13	0.17	9	7	23	−0.13	−0.04
Init Pt	10D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	10D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	16	8	9	−0.33	−0.44	10	8	28	0.17	0.07
2	9	7	10	0.40	0.60	9	7	30	0.40	0.20
3	11	8	9	0.35	0.67	26	21	223	−0.53	−0.04
4	11	8	9	0.35	0.67	26	17	154	−0.53	−0.06
5	7	5	6	0.13	0.17	9	7	23	−0.13	−0.04
Init Pt	25D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	25D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	72	8	9	−5.00	−6.67	10	8	28	0.17	0.07
2	9	7	10	0.40	0.60	9	7	30	0.40	0.20
3	11	8	11	0.31	0.45	18	14	74	−0.13	−0.03
4	11	8	9	0.31	0.56	22	14	84	−0.38	−0.07
5	7	5	6	0.13	0.17	9	7	23	−0.13	−0.04

Table 6. Results of multifidelity optimization with hybrid kriging bridge for Styblinski-Tang function.

Init Pt	2D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	2D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	16	14	34	−0.33	−0.12	11	8	48	0.08	0.02
2	9	7	10	0.40	0.60	8	6	32	0.47	0.22
3	16	15	43	0.43	0.28	9	6	48	0.68	0.40
4	10	8	11	0.64	1.64	9	6	47	0.68	0.40
5	7	5	6	0.13	0.17	6	4	20	0.25	0.10
Init Pt	3D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	3D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	12	11	20	0.00	0.00	10	7	37	0.17	0.05
2	9	7	10	0.40	0.60	13	9	91	0.13	0.02
3	11	8	12	0.42	0.67	14	8	70	0.26	0.07
4	17	14	37	0.06	0.03	19	14	189	−0.06	−0.01
5	7	5	6	0.13	0.17	9	7	99	−0.13	−0.01
Init Pt	5D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	5D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	18	10	19	−0.50	−0.32	25	20	302	−1.08	−0.04
2	9	7	10	0.40	0.60	13	1	222	0.13	0.01
2	9	7	10	0.40	0.60	†				
3	11	8	10	0.35	0.60	19	13	205	−0.12	−0.01
4	14	12	30	0.18	0.10	12	8	103	0.29	0.05
5	8	6	8	0.00	0.00	9	7	102	−0.13	−0.01

Table 7. Results of multifidelity optimization with hybrid PC bridge for Brown function.

Init Pt	2D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	2D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	9	6	6	−0.29	−0.33	8	6	15	−0.14	−0.07
2	11	9	15	0.15	0.13	9	6	10	0.31	0.40
3	11	9	10	0.15	0.20	10	7	21	0.23	0.14
4	11	9	10	0.15	0.20	10	7	21	0.23	0.14
5	1	1	1	0.00	0.00	1	1	0	0.00	0.00
Init Pt	3D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	3D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	9	6	6	−0.29	−0.33	9	6	17	−0.29	−0.12
2	11	9	15	0.15	0.13	10	7	15	0.23	0.20
3	12	8	9	0.25	0.44	10	7	25	0.38	0.24
4	12	9	10	0.08	0.10	10	7	21	0.23	0.14
5	1	1	1	0.00	0.00	1	1	0	0.00	0.00
Init Pt	5D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	5D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	16	8	9	−0.33	−0.44	10	8	28	0.17	0.07
2	9	7	10	0.40	0.60	9	7	30	0.40	0.20
3	11	8	11	0.35	0.55	19	14	85	−0.12	−0.02
4	11	8	9	0.35	0.67	19	14	104	−0.12	−0.02
5	7	5	6	0.13	0.17	9	7	23	−0.13	−0.04
Init Pt	10D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	10D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	9	6	6	−0.29	−0.33	10	6	17	−0.43	−0.18
2	11	9	15	0.15	0.13	11	7	16	0.15	0.13
3	16	10	11	0.20	0.36	14	7	25	0.30	0.24
4	16	10	11	0.20	0.36	14	7	25	0.30	0.24
5	1	1	1	0.00	0.00	1	1	0	0.00	0.00
Init Pt	25D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	25D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	9	6	6	−0.29	−0.33	11	6	17	−0.57	−0.24
2	11	9	15	0.15	0.13	10	7	16	0.23	0.19
3	14	9	10	0.22	0.40	12	7	25	0.33	0.24
4	15	10	11	0.17	0.27	12	7	26	0.33	0.23
5	1	1	1	0.00	0.00	1	1	0	0.00	0.00

Table 8. Results of multifidelity optimization with hybrid kriging bridge for Brown function.

Init Pt	2D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	2D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	9	6	6	−0.29	−0.33	13	9	42	−0.86	−0.14
2	11	9	15	0.15	0.13	11	7	19	0.15	0.11
3	11	9	10	0.15	0.20	11	8	34	0.15	0.06
4	11	9	10	0.15	0.20	11	8	52	0.15	0.04
5	1	1	1	0.00	0.00	1	1	0	0.00	0.00
Init Pt	3D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	3D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	9	6	6	−0.29	−0.33	11	7	50	−0.57	−0.08
2	11	9	15	0.15	0.13	11	6	30	0.15	0.07
3	12	8	9	0.25	0.44	10	7	52	0.38	0.12
4	12	9	9	0.08	0.11	12	8	26	0.08	0.04
5	1	1	1	0.00	0.00	1	1	0	0.00	0.00
Init Pt	5D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	5D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	9	7	10	−0.29	−0.20	12	7	52	−0.71	−0.10
2	11	9	15	0.15	0.13	12	7	64	0.08	0.02
3	13	9	13	0.13	0.15	11	6	47	0.27	0.09
4	14	10	10	0.00	0.00	12	7	40	0.14	0.05
5	1	1	1	0.00	0.00	1	1	0	0.00	0.00

Table 9. Results of multifidelity optimization with hybrid PC bridge for Trig function.

Init Pt	2D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	2D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	31	8	8	−0.07	−0.25	38	10	36	−0.31	−0.25
2	37	15	28	−0.16	−0.18	36	11	42	−0.13	−0.10
3	34	13	15	0.00	0.00	30	10	53	0.12	0.08
4	34	14	29	−0.03	−0.03	33	10	40	0.00	0.00
5	30	9	15	0.00	0.00	39	10	46	−0.30	−0.20
Init Pt	3D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	3D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	33	9	9	−0.03	−0.11	55	15	73	−0.72	−0.32
2	51	27	45	−0.38	−0.31	49	14	58	−0.32	−0.21
3	56	35	65	−0.24	−0.17	46	14	66	−0.02	−0.02
4	41	17	35	−0.05	−0.06	62	21	104	−0.59	−0.22
5	37	12	24	−0.16	−0.21	52	15	74	−0.63	−0.27
Init Pt	5D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	5D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	49	19	43	−0.26	−0.23	82	14	74	−1.10	−0.58
2	35	29	53	0.38	0.40	64	16	70	−0.14	−0.11
3	28	19	33	0.48	0.79	76	18	99	−0.41	−0.22
4	33	22	42	0.38	0.48	70	17	64	−0.32	−0.27
5	53	19	30	−0.15	−0.23	61	10	61	−0.33	−0.25
Init Pt	10D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	10D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	64	20	38	−0.28	−0.37	160	16	80	−2.20	−1.38
2	47	36	87	−0.34	−0.14	124	15	95	−2.54	−0.94
3	41	31	48	0.52	0.92	57	17	113	0.33	0.25
4	37	25	37	0.53	1.11	140	15	86	−0.79	−0.72
5	75	35	60	−0.07	−0.08	139	12	100	−0.99	−0.69
Init Pt	25D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	25D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	97	23	42	−1.11	−1.21	409	18	162	−7.89	−2.24
2	†	35	78			531	15	132		−3.36
3	51	28	49	0.59	1.47	350	14	102	−1.85	−2.23
4	55	31	55	0.59	1.44	†	16	184		
5	117	39	65	−0.02	−0.03	717	12	119	−5.23	−5.06

Table 10. Results of multifidelity optimization with hybrid kriging bridge for Trig function.

Init Pt	2D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	2D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	31	8	13	−0.07	−0.15	17	14	86	0.41	0.14
2	35	13	20	−0.09	−0.15	40	12	39	−0.25	−0.21
3	33	12	21	0.03	0.05	39	9	71	−0.15	−0.07
4	33	11	22	0.00	0.00	40	12	41	−0.21	−0.17
5	26	5	8	0.13	0.50	35	6	22	−0.17	−0.23
Init Pt	3D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	3D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	34	11	23	−0.06	−0.09	52	17	182	−0.63	−0.11
2	45	21	53	−0.22	−0.15	51	17	78	−0.38	−0.18
3	39	21	36	0.13	0.17	62	16	75	−0.38	−0.23
4	42	18	25	−0.08	−0.12	54	17	82	−0.38	−0.18
5	33	9	11	−0.03	−0.09	54	13	117	−0.69	−0.19
Init Pt	5D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	5D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	45	12	15	−0.15	−0.40	74	18	199	−0.90	−0.18
2	56	30	40	0.00	0.00	70	14	123	−0.25	−0.11
3	50	22	33	0.07	0.12	65	16	151	−0.20	−0.07
4	59	29	42	−0.11	−0.14	73	19	194	−0.38	−0.10
5	44	19	28	0.04	0.07	66	13	150	−0.43	−0.13

Table 11. Results of multifidelity optimization with hybrid PC bridge for Rosenbrock function.

Init Pt	2D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	2D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	62	38	75	−1.38	−0.48	47	14	71	−0.81	−0.30
2	35	18	37	−0.17	−0.14	36	11	50	−0.20	−0.12
3	38	17	34	0.21	0.29	33	15	104	0.31	0.14
4	58	17	39	−1.07	−0.77	45	11	44	−0.61	−0.39
5	3	3	3	0.88	7.00	3	1	3	0.88	7.00
Init Pt	3D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	3D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	54	23	43	−0.86	−0.58	53	12	108	−0.83	−0.22
2	36	15	28	0.20	0.32	48	14	134	−0.07	−0.02
3	71	46	101	−0.01	−0.01	47	12	113	0.33	0.20
4	55	18	37	−0.34	−0.38	67	13	75	−0.63	−0.35
5	3	3	3	0.89	8.33	3	1	3	0.89	8.33
Init Pt	5D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	5D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	74	39	74	−1.18	−0.54	74	13	109	−1.18	−0.37
2	42	16	32	0.24	0.41	54	19	213	0.02	0.00
3	80	46	80	−0.18	−0.15	65	12	113	0.04	0.03
4	75	21	52	−0.39	−0.40	66	13	130	−0.22	−0.09
5	3	3	3	0.90	9.00	3	1	3	0.90	9.00
Init Pt	10D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	10D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	88	22	42	−1.10	−1.10	108	17	183	−1.57	−0.36
2	42	14	27	0.46	1.33	64	12	120	0.18	0.12
3	91	35	75	−0.05	−0.05	106	17	202	−0.22	−0.09
4	86	32	46	−0.19	−0.30	106	19	152	−0.47	−0.22
5	3	3	3	0.93	12.33	3	1	3	0.93	12.33
Init Pt	25D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	25D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	159	23	36	−2.31	−3.08	214	21	190	−3.46	−0.87
2	50	16	35	0.66	2.80	65	11	100	0.56	0.83
3	154	25	47	−0.01	−0.02	215	18	203	−0.41	−0.31
4	160	26	47	−0.07	−0.23	199	15	127	−0.34	−0.39
5	3	3	3	0.94	14.67	3	1	3	0.94	14.67

Table 12. Results of multifidelity optimization with hybrid kriging bridge for Rosenbrock function.

Init Pt	2D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	2D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	47	32	54	−0.81	−0.39	46	18	175	−0.77	−0.11
2	34	16	44	−0.13	−0.09	42	11	170	−0.40	−0.07
3	37	33	68	0.23	0.16	46	14	72	0.04	0.03
4	60	15	52	−1.14	−0.62	63	12	189	−1.25	−0.19
5	3	3	3	0.88	7.00	3	1	3	0.88	7.00
Init Pt	3D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	3D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	57	44	117	−0.97	−0.24	48	14	138	−0.66	−0.14
2	39	25	71	0.13	0.08	40	14	112	0.11	0.04
3	59	31	41	0.16	0.27	59	16	162	0.16	0.07
4	50	16	34	−0.22	−0.26	51	16	132	−0.24	−0.08
5	3	3	3	0.89	8.33	3	1	3	0.89	8.33
Init Pt	5D Unified HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost	5D TRMM HF_{-6}	HF_{-1}	LF_{-1}	Rel Save	Max Cost
1	61	34	69	−0.79	−0.39	54	21	226	−0.59	−0.09
2	43	16	27	0.22	0.44	55	15	164	0.00	0.00
3	66	22	45	0.03	0.04	74	20	236	−0.09	−0.03
4	59	21	39	−0.09	−0.13	70	21	241	−0.30	−0.07
5	3	3	3	0.90	9.00	3	1	3	0.90	9.00

Table 13. Results of monofidelity optimization.

	2D Starting Point									
	1		2		3		4		5	
Function	HF_{-6}	HF_{-1}	HF_{-6}	HF_{-1}	HF_{-6}	HF_{-1}	HF_{-6}	HF_{-1}	HF_{-6}	HF_{-1}
McCormick	16	14	18	16	18	16	19	16	10	8
Sty-Tang	12	11	15	13	28	14	28	14	8	6
Brown	7	4	13	11	13	10	13	10	1	1
Trig	29	7	32	9	34	17	33	13	30	9
Rosenbrock	26	22	30	27	48	45	28	25	24	20
	3D Starting Point									
	1		2		3		4		5	
Function	HF_{-6}	HF_{-1}	HF_{-6}	HF_{-1}	HF_{-6}	HF_{-1}	HF_{-6}	HF_{-1}	HF_{-6}	HF_{-1}
Sty-Tang	12	11	15	13	19	16	18	15	8	6
Brown	7	5	13	11	16	12	13	11	1	1
Trig	32	8	37	16	45	21	39	16	32	11
Rosenbrock	29	26	45	42	70	67	41	37	28	23
	5D Starting Point									
	1		2		3		4		5	
Function	HF_{-6}	HF_{-1}	HF_{-6}	HF_{-1}	HF_{-6}	HF_{-1}	HF_{-6}	HF_{-1}	HF_{-6}	HF_{-1}
Sty-Tang	12	11	15	13	17	14	17	14	8	6
Brown	7	5	13	11	15	11	14	10	1	1
Trig	39	9	56	33	54	28	53	27	46	17
Rosenbrock	34	29	55	51	68	63	54	48	30	26
	10D Starting Point									
	1		2		3		4		5	
Function	HF_{-6}	HF_{-1}	HF_{-6}	HF_{-1}	HF_{-6}	HF_{-1}	HF_{-6}	HF_{-1}	HF_{-6}	HF_{-1}
Sty-Tang	12	11	15	13	17	14	17	14	8	6
Brown	7	5	13	11	20	13	20	13	1	1
Trig	50	13	35	24	85	52	78	50	70	33
Rosenbrock	42	33	78	71	87	82	72	66	40	34
	25D Starting Point									
	1		2		3		4		5	
Function	HF_{-6}	HF_{-1}	HF_{-6}	HF_{-1}	HF_{-6}	HF_{-1}	HF_{-6}	HF_{-1}	HF_{-6}	HF_{-1}
Sty-Tang	12	11	15	13	16	13	16	14	8	6
Brown	7	5	13	11	18	12	18	12	1	1
Trig	46	6	†	32	123	82	134	82	115	61
Rosenbrock	48	37	148	140	153	145	149	141	47	35