Hybrid Evolutionary Algorithm with Hermite Radial Basis Function Interpolants for Computationally Expensive Adjoint Solvers

Y.S. ONG asysong@ntu.edu.sg
School of Computer Engineering, Block N4, Nanyang Technological University, Nanyang Avenue, Singapore 639798

K.Y. LUM kaiyew_lum@nus.edu.sg
Temasek Laboratories, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260

P.B. NAIR pbn@soton.ac.uk
Computational Engineering and Design Group, School of Engineering Sciences, University of Southampton, Highfield, Southampton SO17 1BJ, England

Abstract. In this paper, we present an evolutionary algorithm hybridized with a gradient-based optimization technique in the spirit of Lamarckian learning for efficient design optimization. In order to expedite gradient search, we employ local surrogate models that approximate the outputs of a computationally expensive Euler solver. Our focus is on the case when an adjoint Euler solver is available for efficiently computing the sensitivities of the outputs with respect to the design variables. We propose the idea of using Hermite interpolation to construct gradient-enhanced radial basis function networks that incorporate sensitivity data provided by the adjoint Euler solver. Further, we conduct local search using a trust-region framework that interleaves gradient-enhanced surrogate models with the computationally expensive adjoint Euler solver. This ensures that the present hybrid evolutionary algorithm inherits the convergence properties of the classical trust-region approach. We present numerical results for airfoil aerodynamic design optimization problems to show that the proposed algorithm converges to good designs on a limited computational budget.

Keywords: Hybrid Evolutionary Algorithm, Hermite Radial Basis function, Gradient-based Approximation, Computationally Expensive Adjoint Solver

1. Introduction

In recent years, evolutionary algorithms (EAs) have been applied with a great degree of success to aerodynamic design optimization [1-3]. Its popularity lies in the ease of implementation and the ability to arrive close to the global optimum design. However, the high computational costs associated with analysis codes employing Navier Stokes or Euler computational fluid dynamics (CFD) solvers pose a serious impediment to the successful application of EAs to aerodynamic design optimization. This is primarily because a single function evaluation may take many minutes to hours of computer time and EAs typically take thousands of function evaluations to locate a near optimal solution.

Many researchers have examined strategies that make use of approximate models in lieu of exact models to reduce the computational cost of gradient-based optimization algorithms. Since gradient-

based algorithms make use of line searches to locate a new iterate, the issue of range of validity of the approximation models or the control of approximation errors can be easily addressed using *ad hoc* move limits or a *trust region* framework [4-7]. In contrast, since EAs make use of probabilistic recombination operators, controlling the accuracy of approximate fitness predictions is not as straightforward as in gradient-based optimization algorithms.

More recently, much interest has focused on the development of strategies for integrating surrogate models with evolutionary search techniques to tackle the computational cost issue; see, for example, references [1-2, 8-14]. This makes perfect sense since EAs often require thousands of function evaluations to locate a near optimal solution and hence, one obvious way to significantly reduce the computational cost of EAs is to employ computationally cheap surrogate models in lieu of computationally expensive exact models during fitness evaluations. Surrogate models are essentially metamodels or approximation models of the original objective and constraint functions, which are often constructed using techniques in the machine learning and statistics literature such as polynomial response surface methods, neural networks, radial basis functions and Kriging [15-17].

The work on integrating surrogate models with evolutionary search in Ong et al. [1,10,14] represents recent attempts to develop strategies for integrating surrogate models with EAs. In addition, local surrogate models are used in place of global models for problems with large number of variables, since it become increasingly difficult to construct accurate global approximation models due to the *curse of dimensionality*. This paper presents an improvement of our earlier approach [1] to tackle problems where the sensitivities of the objective and constraint functions can be cheaply computed. For example, in the domain of CFD, it is possible to efficiently compute the sensitivities using adjoint methods. Further, when exact sensitivities are available, it also becomes possible to guarantee convergence of the optimization search that makes use of approximation models during search.

In this paper, we present a surrogate-assisted hybrid EA that utilizes the sensitivities of the objective and constraint functions. In diverse contexts, hybrid EAs are also commonly known as Memetic Algorithms, Baldwinian EAs and Lamarkian EAs [3]. For each individual in an EA population, we apply a trust-region enabled gradient-based optimization technique in the spirit of Lamarckian learning. The key idea proposed is to employ Hermite interpolation techniques to construct gradient-enhanced radial basis function networks to speed up local search. Our motivation behind this approach is two-folds – (1) the surrogate model is more accurate than that based on function values only and (2) the hybrid evolutionary approach inherits the convergence property of the classical trust-region approach.

We consider the problem of aerodynamic design, where the derivatives of the objective and constraint functions with respect to the design variables can be efficiently computed using the adjoint method. Numerical results are presented for airfoil shape optimization problems to demonstrate that the gradient-enhanced surrogate modeling strategy enables EAs to converge faster to the optimal solution on a limited computational budget.

The remainder of this paper is organized as follows. We begin with a brief overview of aerodynamic sensitivity analysis using the adjoint method in Section II. Section III presents an overview of how sensitivity information available from adjoint solvers can be leveraged to construct

gradient enhanced radial basis function networks. Section IV presents our evolutionary framework for optimization of computationally expensive CFD codes using surrogate models. Section V presents empirical results on airfoil design problems. Finally section VI summarizes the main conclusions.

2. Adjoint Methods for Aerodynamic Sensitivity Analysis

In this section, we present an overview of the adjoint Euler solver for aerodynamic shape sensitivity analysis. The adjoint approach was first applied by Jameson [18] to design optimization of transonic airfoils. Based on optimal control theory applied to systems of partial differential equations [19], it treats the problem of shape optimization as a constrained optimization problem. Here, the constraint appears in the form of the Euler or Navier-Stokes flow equations. For differentiable problems, this method yields the gradient of the cost function at any given design point.

Compared to traditional finite difference approximations, the computational cost of the adjoint method is only one solution cycle of the flow equation, plus one solution cycle of the adjoint equation which has roughly the same computational cost as the flow equation. Furthermore, the computational cost of sensitivity analysis using the adjoint method is independent of the number of design variables. In a typical steepest descent method, the gradient information can therefore be employed to efficiently search for local minima, as seen in earlier works on airfoil design optimization [20]. The method has more recently been extended to wing and wing-body design optimization [21-23]. For a more detailed account of sensitivity analysis techniques used in design, see [37].

The adjoint method can be applied to inverse design problems, where the cost function is the quadratic sum of the difference between a design's surface pressure and a target pressure distribution. In direct design problems, the cost functions are aerodynamic performance indices such as the drag coefficient, or the drag-to-lift ratio. In the present paper, we shall exploit the efficiency of the continuous adjoint method within our surrogate-assisted evolutionary optimization strategy, using an adjoint Euler solver for the inverse-pressure airfoil design problem. The following outlines the fundamentals of the continuous adjoint method in the Lagrange viewpoint. For a more detailed account of adjoint approaches, see [39].

2.1 The Continuous Adjoint Method

To illustrate the continuous adjoint approach for sensitivity analysis, consider the following constrained minimization problem:

$$\min_{w,S} I(w,S) \quad \text{subject to} \quad R(w,S) = 0 \,. \tag{1}$$

In the general adjoint method, w and S can be treated as design variables; in the following sections, they take on specific physical meanings. The scalar cost function I and the constraint function R are assumed to be differentiable, and R is an element of a Hilbert space with the inner product $< \cdot,\cdot >$. Then, the associated *Lagrangian* is given by

$$\mathcal{L} = I(w,S) + < \psi, R(w,S) > , \tag{2}$$

where ψ is the *Lagrange multiplier*, also known as the *adjoint* or *auxiliary variable*, and has the same dimension as *R*.. The main idea of introducing ψ is to reduce the problem to that of an unconstrained problem, with an additional set of equations to solve, i.e., the *adjoint equations.* This can be obtained by the first-order necessary condition for optimality, which states that at the optimum (local or global), the variation $\delta\mathcal{L}$ is zero for all admissible and independent variations δw and δS :

$$\delta\mathcal{L} = \frac{\partial I}{\partial w}\delta w + \frac{\partial I}{\partial S}\delta S + < \psi, \frac{\partial R}{\partial w} > \delta w + < \psi, \frac{\partial R}{\partial S} > \delta S = 0 . \tag{3}$$

If ψ is chosen to satisfy the following *adjoint equation*,

$$< \psi, \frac{\partial R}{\partial w} > + \frac{\partial I}{\partial w} = 0 \tag{4}$$

then, variations of the Lagrangian depend only on variations in *S*, i.e.,

$$\delta\mathcal{L} = \left(I_S(w,S) + < \psi, R_S(w,S) > \right)\delta S . \tag{5}$$

Equation (5) is simply the sensitivity of the Lagrangian with respect to shape changes. As it is an algebraic expression that depends analytically on ψ , one can compute the sensitivities by solving (4). Next, we shall apply the above general theory to an aerodynamic design problem.

2.2 Variational Calculus on the Euler Equations

In aerodynamic shape optimization problems, *w* represents the flow variables, *S* the shape design variables, and *R* the governing equations. Now, consider inviscid flow over an airfoil that can be modeled by the two-dimensional Euler equations in the physical domain as

$$\frac{\partial w}{\partial t} + \text{div}\ \ g(w) = 0 . \tag{6}$$

Denoting by x_i $(i = 1,2)$ the coordinates in the physical domain, consider the following transformation to a computational domain with coordinates ξ_i :

$$K_{ij} = \frac{\partial x_i}{\partial \xi_j}\ ,\ J = |K|\ ,\ S = JK^{-1}\ ,\ S_{ij} = J\frac{\partial \xi_i}{\partial x_j}\ ,\ (i, j = 1,2) .$$

Transformation of (6) yields the contravariant equation

$$\frac{\partial W}{\partial t} + \text{div}\, G(w,S) = 0 , \tag{7}$$

where $W = Jw$, and $G_i(w,S) = S_{ij} g_j(w)$. As the computational domain is defined by a fixed grid, the transformation matrix *S* is directly related to the shape of the airfoil, and for the moment can be considered as the shape design variables. Hence, transforming to the computational domain elicits the relation between the flow variables and the shape, i.e., they must satisfy the following constraint at steady-state:

$$\text{div}\, G(w,S) = 0 . \tag{8}$$

Moreover, variations of *G* take the form

$$\delta G_i = S_{ij}\frac{\partial g_j}{\partial w}\delta w + g_j \delta S_{ij} \overset{\Delta}{=} C_i \delta w + g_j \delta S_{ij}\,. \tag{9}$$

2.3 Adjoint of the Euler Equations for Inverse Pressure Design Problems

As is evident from (5), the gradient expression depends on the cost-function Jacobian $\partial I/\partial S$. We shall consider the inverse design problem, which involves minimizing the difference between the surface pressure p of a given airfoil and a desired pressure profile p_d. This problem may be considered in two different aspects. Firstly, it may be viewed as a test problem, where the desired pressure distribution p_d is computed from a known shape. Convergence to this known shape can thus serve as a validation of the proposed algorithm. Secondly, the inverse design problem also has a practical purpose, as the designer generally has an idea of the desired pressure profile that yields good aerodynamic performance. For example, in transonic design, a shock front or sharp pressure gradient on the upper surface generally leads to undesirably high pressure drag that degrades the efficiency of the airfoil. A typical approach to inverse pressure design is to 'smoothen' the pressure distribution on the upper-surface in a way that maintains the area under the curve, so as to maintain the lift force generated by the airfoil.

Thus, the inverse pressure design problem can be formulated as a minimization problem of the form

$$I(w,S) = \frac{1}{2}\int_{\text{wall}} (p - p_d)^2 d\sigma \quad \text{subject to (8).} \tag{10}$$

The Lagrangian in the present problem is thus given by

$$\mathcal{L} = \frac{1}{2}\int_{\text{wall}} (p - p_d)^2 d\sigma + \int_\Omega \psi^T \,\text{div}\, G \, d\Omega\,. \tag{11}$$

Here, Ω is the domain of computation. The continuous adjoint formulation as a natural extension when in the case the of p.d.e. constraints. In effect, taking variations of (11) and using Gauss' formula on the second term, and replacing δG by its expressions in (9), we obtain the following:

$$\delta\mathcal{L} = \int_{\text{wall}} (p - p_d)\delta p d\sigma - \int_\Omega \frac{\partial \psi^T}{\partial \xi_i}\left(C_i \delta w + g_j \delta S_{ij}\right) d\Omega + \int_{\partial\Omega} \psi^T \delta G_i \cdot d\sigma_i$$

Application of the adjoint method consists in choosing ψ as a function on the domain Ω and its boundary conditions such that the variations of the flow field, δw and δp, are eliminated in the expression of $\delta\mathcal{L}$, in the volume and boundary integrals. We omit the intermediate steps and refer the interested reader to [20]. The derivation yields the p.d.e. adjoint equation (12), and the adjoint wall boundary conditions (13):

$$\frac{\partial \psi}{\partial t} - C_i^T \frac{\partial \psi}{\partial \xi_i} = 0 \text{ over } \Omega, \tag{12}$$

$$J\left(\psi_2 \frac{\partial \xi_2}{\partial x_1} + \psi_3 \frac{\partial \xi_2}{\partial x_2}\right) = (p - p_d) \quad \text{on the wall boundary.} \tag{13}$$

The conditions of the adjoint variables in the far-field boundary are free, and are computationally determined via standard characteristics analysis of propagating waves [20]. Finally, one obtains the following sensitivity expression for the Lagrangian's variations:

$$\delta\mathcal{L} = -\int_{\Omega} \frac{\partial \psi^T}{\partial \xi_i} g_j \delta S_{ij} d\Omega - \int_{\text{wall}} p(\psi_2 \delta S_{21} + \psi_3 \delta S_{22}) d\xi_1 . \tag{14}$$

It is now obvious that in order to obtain sensitivity information, one only needs to solve (12) with (13), which is a linear differential equation of the same dimension as the Euler equations. Hence, the cost of gradient calculation is only one adjoint computation in addition to the contravariant Euler equations (7).

2.4 Shape Parameterization and Gradient Calculation

As mentioned earlier, S_{ij} are the local coordinate-change variables that are directly but not explicitly related to the definition of the airfoil shape. The next step is to arrive at expressions for the gradients of the Lagrangian with respect to variations in the shape, which is defined by a parametric representation. Here, we consider the function-series representation proposed by Hicks and Henne [24] which gives the upper and lower surfaces of the airfoil as respective linear combinations of a finite number of basis functions:

$$y_{\text{upper}}(x) = y^0_{\text{upper}}(x) + \sum_{j=1}^{N} z_j h_j(x) \ , \ y_{\text{lower}}(x) = y^0_{\text{lower}}(x) + \sum_{j=N+1}^{2N} z_j h_j(x) \tag{15}$$

where y^0_{upper} and y^0_{lower} are the upper and lower surfaces of a baseline shape, x is the position along the chord, and the z_j are design variables. Essentially, the search space is a set of variants of the baseline shape. The greater the number of design variables ($2N$), the larger the set of shapes represented. For example, Figures 1 and 2 shows a series of 12 such functions, and an airfoil it represents. The sensitivities of the objective function with respect to the shape design variables can be readily computed using (14). Finally, the gradient of the Lagrangian can be simply written as

$$\nabla\mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial z_1}, \frac{\partial \mathcal{L}}{\partial z_2}, \ldots, \frac{\partial \mathcal{L}}{\partial z_{2N}} \right). \tag{16}$$

Note that the same method of gradient calculation is applicable to other representations of airfoil geometry, such as B-splines and PARSEC [2]. The aerodynamic shape representations using basis functions, especially Hicks-Henne functions, have been widely used as they yield smooth shapes with few parameters, making this representation an efficient one for design optimization, although it has been shown that representations using B-splines are more accurate [40].

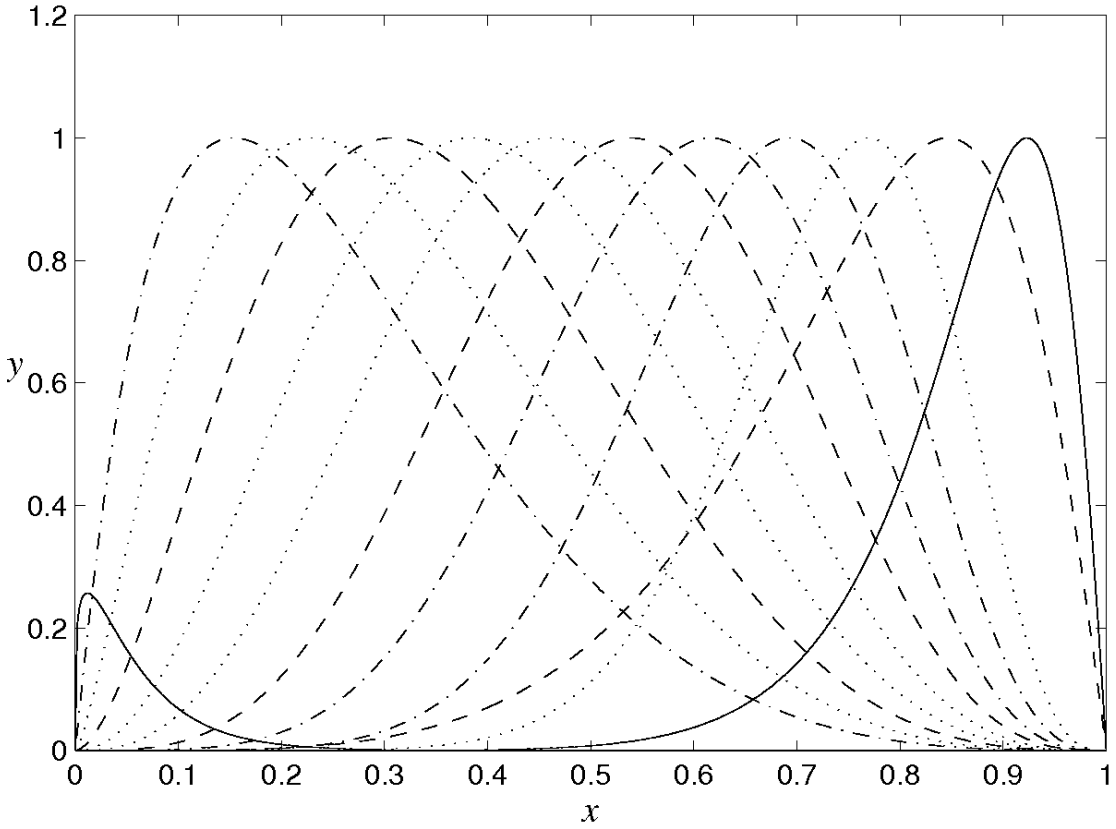

Figure 1. Series of 12 Hicks-Henne basis functions.

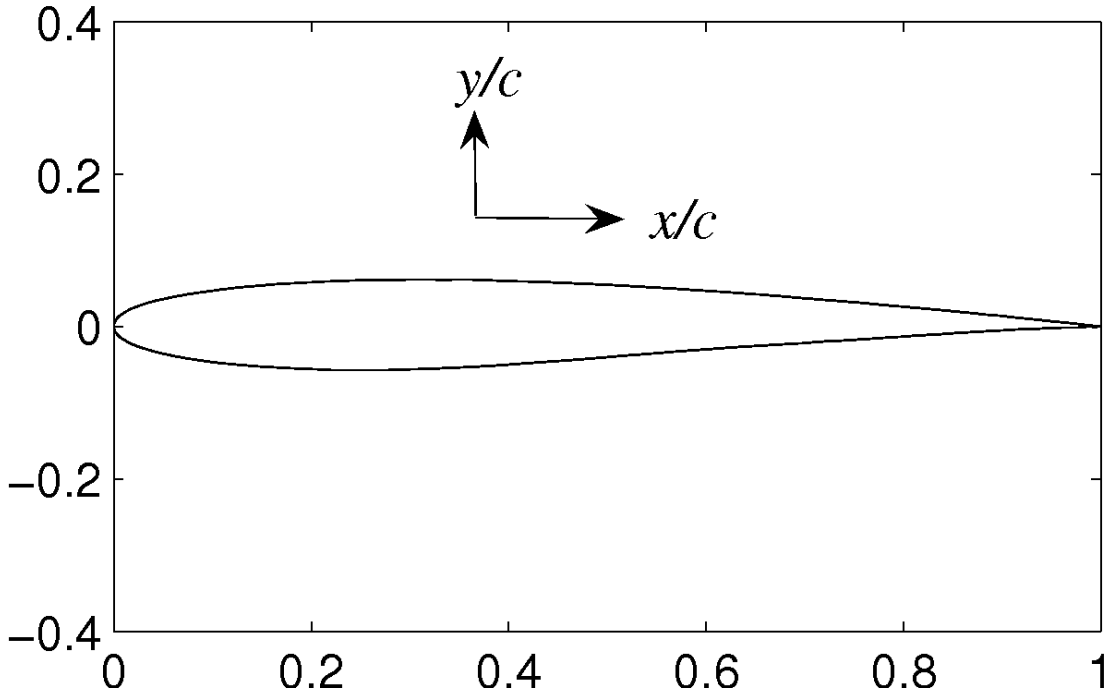

Figure 2. Airfoil represented by a series of 12 Hicks-Henne functions on each surface (normalized by chord length *c*).

3. Surrogate Modeling Using Hermite Interpolation

A variety of techniques exist in the literature for constructing surrogate models of simulation codes. These include response surface methodology [25], artificial neural network methods [15], multivariate regression splines [26], and Kriging [16]. A detailed comparison of some metamodeling techniques can be found in Jin et al. [27]. In the present investigation, our objective is to construct interpolating surrogate models using radial basis functions (RBFs) that employ sensitivity information provided by the adjoint CFD solver for enhanced accuracy [37]. We use the idea of Hermite interpolation to construct gradient enhanced RBF approximations. The idea of Hermite interpolation is not new and the theoretical foundations of this approach in the context of function approximation can be found in the literature; see, for example, references [28-31]. In this section, we outline how Hermite interpolation can be implemented using RBFs when sensitivity information is cheaply available via an adjoint CFD solver.

Let us denote the training dataset by $\{\mathbf{z}^i, f(\mathbf{z}^i), \nabla f(\mathbf{z}^i)\}, i = 1,2,...,m$, where $\mathbf{z}^i \in \Re^d$ denotes the input vector, $f(\mathbf{z}^i)$ denotes the output to be approximated and $\nabla f = \{\partial f / \partial z_1, \partial f / \partial z_2, ..., \partial f / \partial z_d\} \in \Re^d$ denotes the partial derivatives of the output $f(\mathbf{z})$ with respect to the components of the input vector. Then, a Hermite interpolant for approximating $f(\mathbf{z})$ can be written in terms of a set of RBFs as follows:

$$\hat{f}(\mathbf{z}) = \sum_{i=1}^{m} \beta_i \phi(\| \mathbf{z} - \mathbf{z}^i \|) + \sum_{i=1}^{m} \sum_{j=1}^{d} \tilde{\beta}_{ij} \frac{\partial \phi}{\partial z_j}(\| \mathbf{z} - \mathbf{z}^i \|), \tag{17}$$

where $\phi(\| \mathbf{z} - \mathbf{z}^i \|)$ is a radial basis function which is differentiable at least twice. β_i and $\tilde{\beta}_{ij}$, where $i = 1,2,...,m, \;\; j = 0,1,2,...,d,$ are a set of $m(d+1)$ undetermined weights.

Since the training dataset contains $f(\mathbf{z})$ and $\nabla f(\mathbf{z})$ at m points, we can arrive at a total of $m(d+1)$ linear algebraic equations to compute the undetermined coefficients in the RBF model. The first set of *m* equations using the function values corresponding to the points $\mathbf{z}^i, i = 1,2,....,m$ can be written as

$$\hat{f}(\mathbf{z}^i) = f(\mathbf{z}^i), i = 1,2,....,m . \tag{18}$$

An additional set of *md* equations can be derived by using the derivative information available in the training dataset, which gives

$$\nabla \hat{f}(\mathbf{z}^i) = \nabla f(\mathbf{z}^i), i = 1,2,....,m . \tag{19}$$

To implement the above conditions, we first differentiate (17) with respect to the variable z_k, which gives

$$\frac{\partial \hat{f}(\mathbf{z})}{\partial z_k} = \sum_{i=1}^{m} \beta_i \frac{\partial}{\partial z_k} \phi(\| \mathbf{z} - \mathbf{z}^i \|) + \sum_{i=1}^{m} \sum_{j=1}^{d} \tilde{\beta}_{ij} \frac{\partial}{\partial z_k} \frac{\partial \phi}{\partial z_j}(\| \mathbf{z} - \mathbf{z}^i \|) . \tag{20}$$

Given a set of m data points for a problem with d variables, we arrive at a total of $m(d+1)$ linear algebraic equations using (18-20), which can be compactly written as

$$\mathbf{A}\boldsymbol{\beta} = \mathbf{y},$$

where

$$\boldsymbol{\beta} = \{\beta_1, \tilde{\beta}_{11}, \tilde{\beta}_{12}, ..., \tilde{\beta}_{1d}, \beta_2, \tilde{\beta}_{21}, \tilde{\beta}_{22}, ..., \tilde{\beta}_{2d}, ..., \beta_m, \tilde{\beta}_{m1}, \tilde{\beta}_{m2}, ..., \tilde{\beta}_{md}\} \in \Re^{m(d+1)},$$

and

$$\mathbf{y} = \{f(\mathbf{z}^1), \frac{\partial f}{\partial z_1}(\mathbf{z}^1), \frac{\partial f}{\partial z_2}(\mathbf{z}^1), ..., \frac{\partial f}{\partial z_d}(\mathbf{z}^1), ..., f(\mathbf{z}^m), \frac{\partial f}{\partial z_1}(\mathbf{z}^m), \frac{\partial f}{\partial z_2}(\mathbf{z}^m), ..., \frac{\partial f}{\partial z_d}(\mathbf{z}^m)\} \in \Re^{m(d+1)}.$$

The coefficient matrix $\mathbf{A} \in \Re^{m(d+1)\times m(d+1)}$ can be written in partitioned form in terms of m submatrices as follows:

$$\mathbf{A} = \begin{bmatrix} \boldsymbol{\Phi}_{11} & \boldsymbol{\Phi}_{12} & ... & \boldsymbol{\Phi}_{1m} \\ \boldsymbol{\Phi}_{21} & \boldsymbol{\Phi}_{22} & ... & \boldsymbol{\Phi}_{21} \\ ... & ... & ... & ... \\ \boldsymbol{\Phi}_{m1} & \boldsymbol{\Phi}_{m2} & ... & \boldsymbol{\Phi}_{mm} \end{bmatrix},$$

where

$$\boldsymbol{\Phi}_{ij} = \begin{bmatrix} \phi(\| \mathbf{z}^i - \mathbf{z}^j \|) & \frac{\partial \phi}{\partial z_1}(\| \mathbf{z}^i - \mathbf{z}^j \|) & .... & \frac{\partial \phi}{\partial z_d}(\| \mathbf{z}^i - \mathbf{z}^j \|) \\ \frac{\partial \phi}{\partial z_1}(\| \mathbf{z}^i - \mathbf{z}^j \|) & \frac{\partial^2 \phi}{\partial z_1^2}(\| \mathbf{z}^i - \mathbf{z}^j \|) & ... & \frac{\partial^2 \phi}{\partial z_1 \partial z_d}(\| \mathbf{z}^i - \mathbf{z}^j \|) \\ ... & ... & ... & ... \\ \frac{\partial \phi}{\partial z_d}(\| \mathbf{z}^i - \mathbf{z}^j \|) & \frac{\partial^2 \phi}{\partial z_d \partial z_1}(\| \mathbf{z}^i - \mathbf{z}^j \|) & ... & \frac{\partial^2 \phi}{\partial z_d^2}(\| \mathbf{z}^i - \mathbf{z}^j \|) \end{bmatrix} \in \Re^{(d+1)\times(d+1)}$$

It can be noted from the above derivation that in order to implement Hermite interpolation the RBF ϕ must be differentiable at least twice. Note that in comparison to standard interpolating RBF models, the size of the resulting system of equations for Hermite interpolation depends on the total number of design variables. As a result, the computational cost of Hermite RBF interpolation becomes significant when the number of training points and design variables are increased. However, in the present research, we only construct local surrogate models using a subset of the training dataset. Hence, the size of the system of equations to be solved for the weight vector $\boldsymbol{\beta}$ turns out to be modest. We use Gaussian RBFs (which are infinitely differentiable) to construct surrogate models, i.e., $\phi(\| \mathbf{z}^i - \mathbf{z}^j \|) = \exp\left(- \frac{\| \mathbf{z}^i - \mathbf{z}^j \|^2}{2\sigma^2} \right)$, where $\|.\|$ denotes the L_2 norm and σ is a hyperparameter which is chosen using a cross-validation procedure.

4. Hybrid Evolutionary Algorithm for Computationally Expensive Adjoint Solvers

In this section, we present a brief overview of our proposed hybrid evolutionary algorithm for computationally expensive adjoint solvers. In particular, we consider bound constrained nonlinear programming problems of the form:

Minimize: $f(\mathbf{z})$

Subject to: $\mathbf{z}_l \leq \mathbf{z} \leq \mathbf{z}_u$ (21)

where $\mathbf{z} \in \Re^d$ denotes the vector of design variables, and $\mathbf{z}_l$ and $\mathbf{z}_u$ are vectors of lower and upper bounds on the design variables.

The basic structure of our hybrid EA which makes use of Hermite RBF interpolants is shown in Figure 3. The hybrid evolutionary algorithm begins with the initialization of a database using a set of designs, either randomly, or using design of experiments techniques or *a priori* knowledge if it exists. Exact adjoint CFD analysis is then carried out for each design point and its objective function value $f(\mathbf{z})$ together with its partial derivatives with respect to the components of the design vector are archived in the database.

Procedure: Hybrid Evolutionary Algorithm for Computationally Expensive Adjoint Solvers
BEGIN

Initialize:
- **Generate a database containing a population of designs. (*Optional: upload a historical database if exists*)**
- **Specify *m* and** $k_{\max}$

While (computational budget not exhausted)
  Evaluate all individuals in the population using the exact adjoint CFD analysis code.
  For (each non-duplicated individual in population)
  - Apply trust-region enabled gradient-based optimization algorithm to each non-duplicated individual in the population by interleaving the exact and local Hermite RBF interpolation models.
  - Update the database with any new design points, $\mathbf{z}^k$, generated during the trust-region iterations with exact objectives, f^k, and its partial derivatives, ∇f^k.
  - Replace the individuals in the population with the locally improved solution in the spirit of Lamarckian learning.

  End For
  Apply standard EA operators to create a new population.
End While
END

Figure 3. Outline of Hybrid Evolutionary Algorithm for Computationally Expensive Adjoint Solvers using Hermite RBF interpolants.

Subsequently, for each non-duplicated new design point in the EA population, a gradient-based optimization algorithm is applied with this point as the initial guess. The main idea used here is to

interleave the computationally expensive CFD solver with a surrogate model to reduce the computational cost of gradient-based optimization. We use a trust-region approach to manage the interplay between the exact and approximate model [1,32]. This involves the solution of a sequence of subproblems of the form:

$$\begin{aligned} &\text{minimize} && \tilde{f}^k(\mathbf{z}) \\ &\text{subject to} && \mathbf{z}_l \le \mathbf{z}_l^k \le \mathbf{z} \le \mathbf{z}_u^k \le \mathbf{z}_u \end{aligned} \tag{22}$$

where $k = 0,1,2,...,k_{\max}$ is the subproblem number or the trust-region iteration counter. $\mathbf{z}_l^k = \mathbf{z}_c^k - \Delta^k$ and $\mathbf{z}_u^k = \mathbf{z}_c^k + \Delta^k$ are the lower and upper bounds, respectively. $\mathbf{z}_c^k$ is the initial guess for $\mathbf{z}$ at the kth trust-region iteration and Δ^k is the trust-region radius used to control the move limits on the design variables. $\tilde{f}^k(\mathbf{z})$ is a local surrogate model of the original objective function $f(\mathbf{z})$. Note that we use the superscript k to indicate that the design variable bounds as well as the surrogate model are updated at each iteration.

During the solution of each subproblem, a *local surrogate model*, $\tilde{f}^k(\mathbf{z})$, is created dynamically. The training dataset $\mathfrak{D}^k$ used for constructing the surrogate model is formed from the nearest m neighboring points of the initial guess, $\mathbf{z}_c^k$, which are chosen by searching a central database containing an archive of all previous designs. The metric used here to establish similarity between design points during database search is the Euclidean distance measure. We use local surrogate models since earlier studies have shown that they provide more accurate approximations compared to standard global surrogate models [1]. Further, we construct local surrogate models using the Hermite RBF interpolation approach presented earlier. The motivation for this arises from our observations that the accuracy of the gradient-enhanced surrogate model is superior to standard RBF approximations that do not make use of sensitivity information. Further, as discussed later, the use of gradient information also guarantees convergence of the proposed hybrid EA.

After solving each subproblem in (22), the trust region size, Δ^k, is updated based on how well the surrogate model predicts the k^{th} local optimum, $\mathbf{z}_{lo}^k$. A figure of merit for the performance of the surrogate model, ρ^k, is calculated as

$$\rho^k = \frac{f(\mathbf{z}_c^k) - f(\mathbf{z}_{lo}^k)}{\tilde{f}(\mathbf{z}_c^k) - \tilde{f}(\mathbf{z}_{lo}^k)} \tag{23}$$

The preceding equation provides a measure of the actual versus predicted change in the function values at the k^{th} local optimum. ρ^k is then used to update the trust region radius, Δ^k, as follows:

$$\begin{aligned} \Delta^{k+1} &= 0.25\,\Delta^k, && \text{if } \rho^k \le 0.25, \\ &= \Delta^k, && \text{if } 0.25 < \rho^k < 0.75, \\ &= \zeta\,\Delta^k, && \text{if } \rho^k \ge 0.75, \end{aligned} \tag{24}$$

where $\zeta = 2$, if $\| \mathbf{z}_{lo}^{k} - \mathbf{z}_{c}^{k} \|_{\infty} = \Delta^{k}$ or $\zeta = 1$, if $\| \mathbf{z}_{lo}^{k} - \mathbf{z}_{c}^{k} \|_{\infty} < \Delta^{k}$

After solving the *k*th subproblem, the exact objective function value at the local optimum $f(\mathbf{z}_{lo}^{k})$ and its partial derivatives $\nabla f(\mathbf{z}_{lo}^{k})$ are used in the augmented dataset $\mathfrak{D}^{k+1}$ to construct an updated surrogate model for the subsequent trust-region iteration. Note that, as indicated in Figure 3, there is a user specified parameter *m*, which is the maximum number of points used to construct a surrogate model. This parameter controls the degree to which the surrogate model is local - in the limit when *m* equals the total number of points in the database we arrive at a global surrogate model.

The initial guess for the *k+1* th iteration is subsequently found using

$$\mathbf{z}_{c}^{k+1} = \mathbf{z}_{lo}^{k}, \qquad \text{if } \rho^{k} > 0,$$

$$= \mathbf{z}_{c}^{k}, \qquad \text{if } \rho^{k} \leq 0.$$

The trust-region iterations are terminated when $k \geq k_{\max}$, where $k_{\max}$ is the maximum number of exact function and gradient evaluations for each individual in the EA population which is set *a priori*. At the end of the trust-region iterations, the exact fitness of the locally optimized design point is determined. If the exact fitness of this point is found to be better than its initial value, then Lamarckian learning proceeds. Lamarckian learning forces the genotype to reflect the result of improvement through placing the locally improved individual back into the population to compete for reproductive opportunities. In addition, the optimized design point along with its exact fitness and sensitivities are appended to the database. As indicated in Figure 3, the hybrid EA is terminated when the computational budget specified by the user is exhausted.

4.1 Convergence Property

Clearly, the performance of the proposed hybrid EA depends on the ability of the approximation models to accurately predict the objective function. Hence it is of theoretical interest to make some general mathematical statements about the convergence properties of the proposed hybrid EA which uses surrogate models. We would like to note that here by convergence we mean the assurance that the iterates produced by an optimization algorithm working with the surrogate models, started at an arbitrary initial iterate, will converge to a stationary point or local optimizer of the original problem. As discussed in Alexandrov et al. [4, 32], convergence can be guaranteed provided the following two consistency conditions are satisifed by the surrogate model at the initial guess, $\mathbf{z}_{c}^{k}$.

$$\tilde{f}(\mathbf{z}_{c}^{k}) = f(\mathbf{z}_{c}^{k})$$

$$\nabla \tilde{f}(\mathbf{z}_{c}^{k}) = \nabla f(\mathbf{z}_{c}^{k}) \tag{25}$$

For the case of using standard RBF surrogate models in our earlier studies [1], only the zero-order consistency condition is satisfied at the initial guess. However, in the case of using Hermite RBF

interpolants as proposed here, both consistency conditions are satisfied at the initial guess. This implies that our proposed hybrid EA inherits the convergence property of the classical trust-region approach.

It is worth noting here that the convergence property is of theoretical interest only since the local search strategy in the proposed hybrid evolutionary algorithm is often not run to convergence in real-world design problems. Even so, for the example problems considered later, we find that the algorithm is capable of convergence. Further, we find that since the proposed framework satisfies both the zero- and first-order consistency conditions, the convergence rate is significantly faster for solving computationally expensive adjoint solvers compared to our earlier work [1] where only the zero-order consistency condition is satisfied by the surrogate model.

5. Numerical Results

In this section, we present results obtained using the proposed memetic algorithm on airfoil design problems when adjoint CFD solvers are available. In particular, we consider the inverse pressure design problems described in section II. The airfoil geometry is characterized using 24 design variables with the NACA 0012 or NACA 0015 airfoils as baselines, as shown in Figure 4.

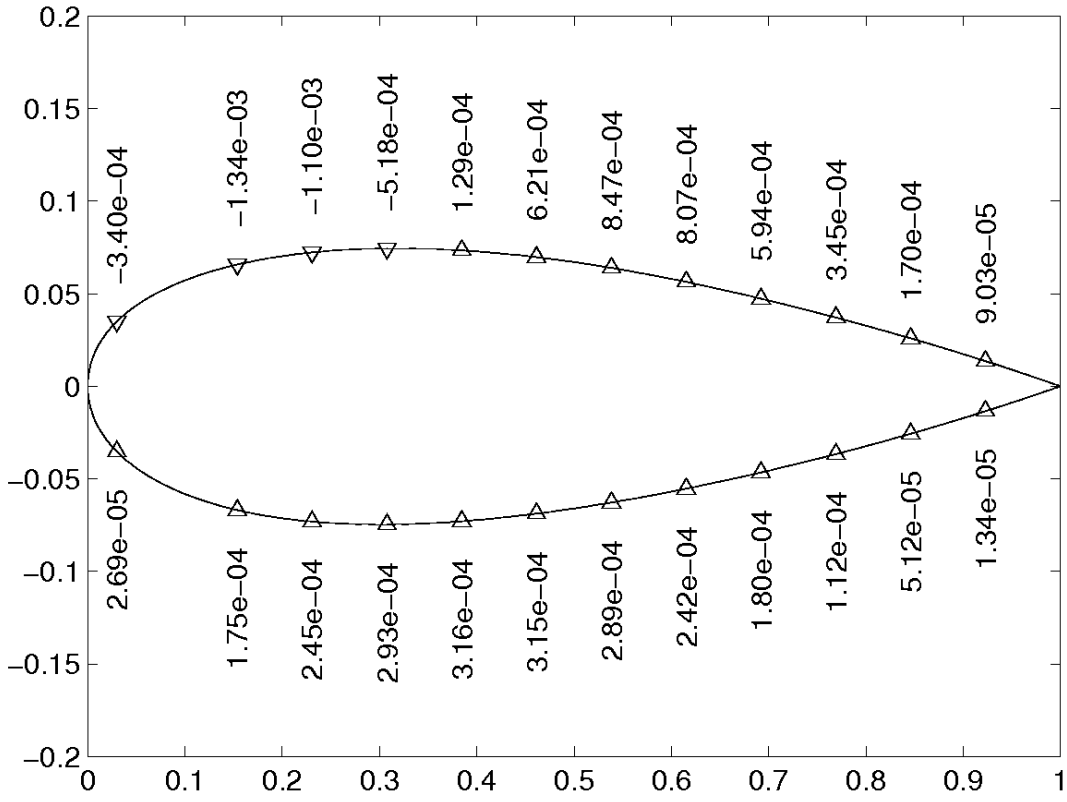

Figure 4. Airfoil geometry characterized using 24 design variables with the NACA 0012 as baseline.

Here, a single analysis of a typical airfoil geometry using an Euler CFD solver takes around 10 minutes on a Pentium IV processor, while an exact adjoint CFD calculation takes around 15 minutes. In comparison, surrogate model construction using Hermite RBF interpolation takes only a few seconds when around a hundred points are used for training. When dealing with computationally

expensive adjoint solvers that cost more than a few minutes of CPU time per function evaluation, this training cost is negligible. To start the evolutionary search process, we first construct a space filling set of initial designs using Latin hypercube sampling (LHS). LHS is a stratified sampling technique where the random variable distributions are divided into equal probability intervals and the population of individuals is randomly selected from within each interval. This serves to generate a distribution of space filling set collections of initial GA population from the large multidimensional search space. The exact objective function value and its sensitivities are computed for these points to create an initial database.

In our numerical studies, we employ a standard binary coded parallel genetic algorithm (GA) for the search. 10-bit encoding is used in representing each search dimension. A linear ranking algorithm with selection pressure of 1.5 is used for selection. The population size is kept at 20 for the problems considered. Uniform crossover and mutation operators are applied at probabilities 0.6 and 0.01, respectively. The search termination criteria are set to a maximum computational budget of a thousand exact evaluations or convergence to the global optimum.

Local search for each individual in the GA population is carried out using the feasible sequential quadratic programming solver [33]. A well-known strength of evolutionary algorithm is the ability to partition the population of individuals among multiple compute nodes. Hence it is important that the intrinsic parallelism of EA is retained in our work. Here, parallelism is attained in our parallel GA using Grid Computing technologies [34]. The use of Grid technologies enables us to tap on large compute power and achieve better utilization of remote solvers during the design search. In our numerical studies, the adjoint codes implementing the objective functions were wrapped as Grid services [35]. Further, the EA codes used here are gridified using standard Grid Remote Procedure Call (Gridrpc) and Globus such that execution of the adjoint Euler code on computing clusters that spans across at geographically distributed locations may be attained remotely [36].

Next, we present numerical results for the case when local surrogate models are constructed using Hermite RBF (HRBF) interpolation. These results are compared with the approach presented in Ong et al. [1] where only standard RBF (SRBF) interpolants are used as surrogate models and a standard GA implementation. In the hybrid GA, the parameters *m* (number of nearest neighbors used to construct the local surrogate model) and $k_{\max}$ (maximum number of trust-region iterations) are set to 40 and 3, respectively. These parameter configurations were selected based on our previous experiences. Further, it is to be noted here that one design cycle for the standard and SRBF-assisted GA is one run of the Euler CFD solver. For the case of the HRBF-assisted GA, one design cycle is one run of the adjoint CFD solver which includes efforts to compute the sensitivities of the objective function.

In this study, several criteria have been defined to measure the performance. These are listed in Table 1. Among these criteria, *CPU time* is used to measure the computational cost of the algorithms in wall-clock time. *Average*, *Average gap*, *Best*, *Gap* and *Success rate* serve as the criteria for measuring the solution quality of the algorithms. In our experimental studies, the pool of heterogeneous computing clusters in the Nanyang Campus Grid has been employed to facilitate parallel execution of the solvers [38].

Table 1. Criteria for measuring performance.

Criterion	Definition
Average Evaluations	Average number of exact evaluations consumed by the EA before the search termination criteria sets in.
Average	Average fitness value of the solutions obtained across all five simulation runs.
Average gap	*Average gap* = (*Average* - *go*) Difference between the Average value and the global optimum, where *go* is the global optimum value of the fitness function.
Best	Best solution obtained among all the EA runs.
Gap	*gap* = (*bf* - *go*) Difference between the Best-found and the global optimum, where *go* is the global optimum of the fitness function.
Success rate	Number of times the algorithm converges to the global optimum out of all five simulation runs under limited computational budget. Here convergence is assumed to occur when the fitness reaches 3 significant figures.

5.1 Case Study I — Subsonic Inverse Pressure Design Problem

We first consider a subsonic inverse pressure design problem. In Case Study I, the target pressure profile is generated from the NACA 0012 airfoil, which itself is also the baseline shape. Hence, there exists for this problem a global solution corresponding to $z_1 = \ldots = z_{24} = 0$. This constitutes a good test problem for validating the convergence property of the proposed hybrid EA, since the optimal design is known in advance. The free-stream conditions in this problem are subsonic speed of Mach 0.5, and zero angle of attack (AOA), corresponding to symmetric pressure profiles on the upper and lower walls.

Using 24 design variables and the cost function in (10), the empirical results of the HRBF–assisted hybrid GA, the SRBF–assisted hybrid GA and standard GA for Case Study I are summarized in Table 2 and Figures 7-8. On this particular problem, the global minimum given by cost function (10) equals zero, i.e., *go* = 0.0, see Table 1. The results in Table 2 indicate that the HRBF–assisted hybrid GA outperforms both the SRBF–assisted hybrid GA and standard GA algorithm significantly in terms of solution quality and computational cost on the subsonic inverse pressure design problem. The *Average* criterion shown in Table 2 indicates that both HRBF–assisted hybrid GA and SRBF–assisted hybrid GA has significantly better convergence rate than the standard GA across all five independent runs conducted. In particular, the HRBF–assisted hybrid GA converges to the global optimum correctly at a greater precision and at significantly a lower computational cost than both the SRBF–assisted hybrid GA and standard GA algorithms, i.e., see *Average Evaluations* criterion in Table 2. Further, the 100% S*uccess Rate* of the HRBF–assisted hybrid GA also demonstrates the superior convergences ability of the algorithm, indicating that the algorithm was capable of converging to the global across all five independent runs conducted. On the whole, the HRBF–assisted hybrid GA displays the best performances on all other criteria, see Table 2.

Figure 6 reveals the final airfoil geometries obtained on one of the five simulation runs using HRBF-assisted hybrid GA, SRBF-assisted hybrid GA and standard GA. The NACA 0012 target shape is also included in the plot for comparison. It can be observed that the HRBF-assisted hybrid GA converges very close to the target shape than the other two algorithms considered.

Table 2. Simulation Results for Case Study I

	Algorithm	*Average Evaluations*	*Average, Standard Dev*	*Average gap*	*Best*	*Gap*	*Success rate*
Case Study I	GA	1000	0.007832, 0.003131	0.007832	0.005950	0.005950	0%
	SRBF–GA	1000	0.001709, 0.000074	0.001709	0.000949	0.000949	20%
	HRBF–GA	**320.8**	**0.000331, 0.000023**	**0.000331**	**0.000315**	**0.000315**	**100%**

The rationale between the superiority of the HRBF-assisted hybrid GA over the SRBF-assisted hybrid GA may be explained by the fact that employing Hermite interpolation techniques to construct gradient-enhanced radial basis function networks generally produce more accurate surrogate models than those based on function values only. This in turn helps generate significant improvements in the design optimization search. To illustrate this, we show in Figure 6, the typical prediction errors, $\left| f(\mathbf{z}_{lo}^1) - \hat{f}(\mathbf{z}_{lo}^1) \right|$, and local improvements, $f(\mathbf{z}_c^1) - f(\mathbf{z}_{lo}^1)$, produced at the 1st iteration of the trust-region approach (i.e., $k = 1$) when HRBF and SRBF are used for surrogate modeling are presented for several GA individuals. It can be seen from Figure 6 that the HRBF–assisted hybrid GA produces smaller prediction errors and this in turn leads to a larger local search improvements than the SRBF–assisted hybrid GA. This trend explains the faster convergence rate of the HRBF-assisted hybrid GA.

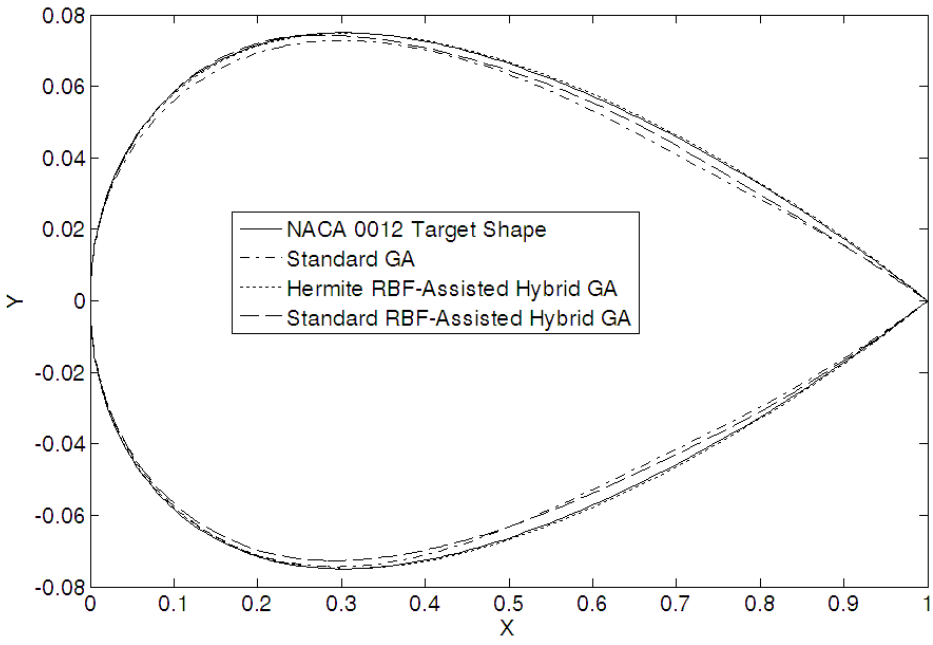

Figure 5. Comparison of best airfoil geometries obtained after approximately 30 hours using HRBF-assisted hybrid GA, SRBF-assisted hybrid GA and standard GA with the NACA 0012 target shape for Case Study I.

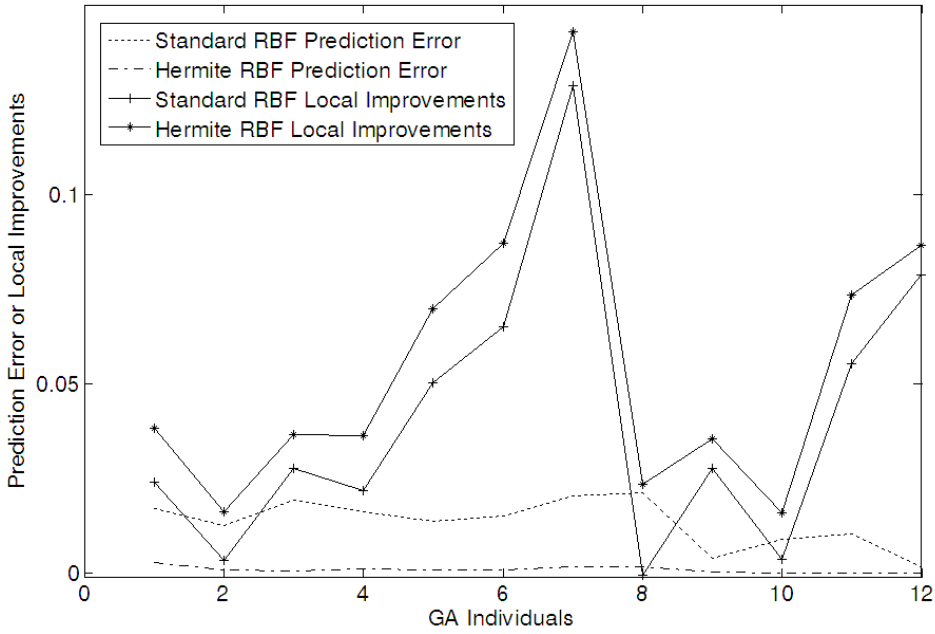

Figure 6. Prediction errors, $\left| f(\mathbf{z}_{lo}^1) - \hat{f}(\mathbf{z}_{lo}^1) \right|$, and local improvements, $f(\mathbf{z}_c^1) - f(\mathbf{z}_{lo}^1)$ using SRBF and HRBF surrogate models.

5.2 Case Study II — Transonic Inverse Pressure Design Problem

In the second design problem, we consider the NACA 0015 airfoil as baseline, at a transonic speed of Mach 0.7 and an angle of attack of 4 degrees. Under these conditions, a shock front appears on the upper surface of the airfoil in the form of a step jump in the pressure profile. Consequently, pressure drag increases and degrades the efficiency of the design as discussed in Section 2. Note that for the angle of attack considered, the flow remains laminar, for which modeling using the inviscid Euler equations is still valid ignoring skin friction.

Figure 7 shows the pressure profile of the NACA 0015 airfoil in the case study conditions. A sharp pressure gradient can be seen on the upper surface. For this problem, we consider the more practical goal of 'smoothening' out the shock by synthesizing a target pressure profile (shown as dotted line in Figure 7) with gentler pressure gradients, but roughly the same area under the curve as for the NACA 0015. In this way, a design that closely matches the target pressure will have lower drag than the NACA 0015 but with similar lift performance. In this problem, there is obviously no 'target shape', as the desired pressure distribution profile is synthetic and not produced by a known shape.

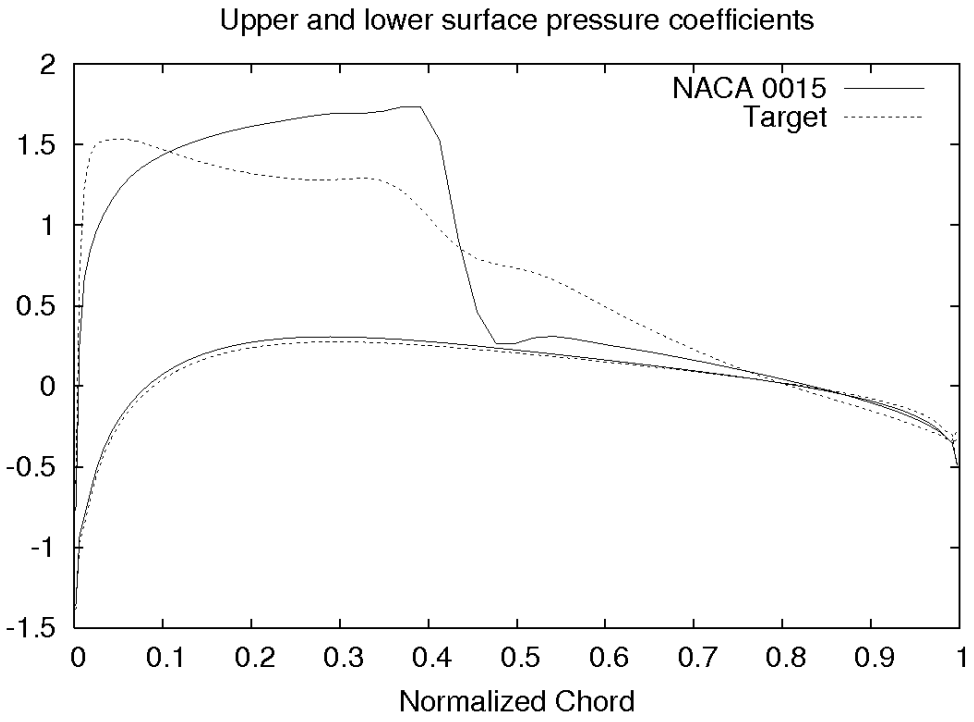

Figure 7. Pressure profile at Mach Number 0.7 and 4-degree angle of attack: sharp pressure gradient on the NACA 0015 versus a synthetic target profile for Case Study II.

The convergence trends of the HRBF-assisted hybrid GA, SRBF–assisted hybrid GA and the standard GA are shown in Figure 8. The search traces indicate that the HRBF-assisted hybrid GA converges to the best obtained cost function value of 0.041 after 222 design cycles or exact evaluations based on the adjoint Euler solver. This is equivalent to 333 design cycles or exact evaluations for HRBF-assisted hybrid GA which uses the computationally cheaper Euler solver. In addition, both the SRBF–assisted hybrid GA and standard GA failed to reach this value even after 1000 exact evaluations. To make a fair comparison of the three algorithms, we shall examine the respective designs obtained at the end of 333 design cycles. Figure 9 shows the resulting pressure profiles of designs obtained using the standard GA and the SRBF-assisted hybrid GA. It can be seen that the shock front is merely shifted forward in both cases but not eliminated. Figure 10 shows the resulting pressure profile of the optimal design obtained using the HRBF-assisted hybrid GA. Clearly, the shock front has disappeared and a good match with the target pressure distribution is achieved.

The practicality of Case Study II can be elucidated by examining the aerodynamic performances (normalized values) of the 3 different designs versus those of the baseline. As shown in Table 3, this exercise consists in drag reduction. The final design produced by the HRBF-assisted hybrid GA, which best matches the target profile, has the lowest drag corresponding to a 60% reduction, albeit with marginal loss in lift. Another performance index is the drag/lift ratio (*D/L*). The target pressure profile entails lower *D/L* ratio, which improves airplane efficiency.

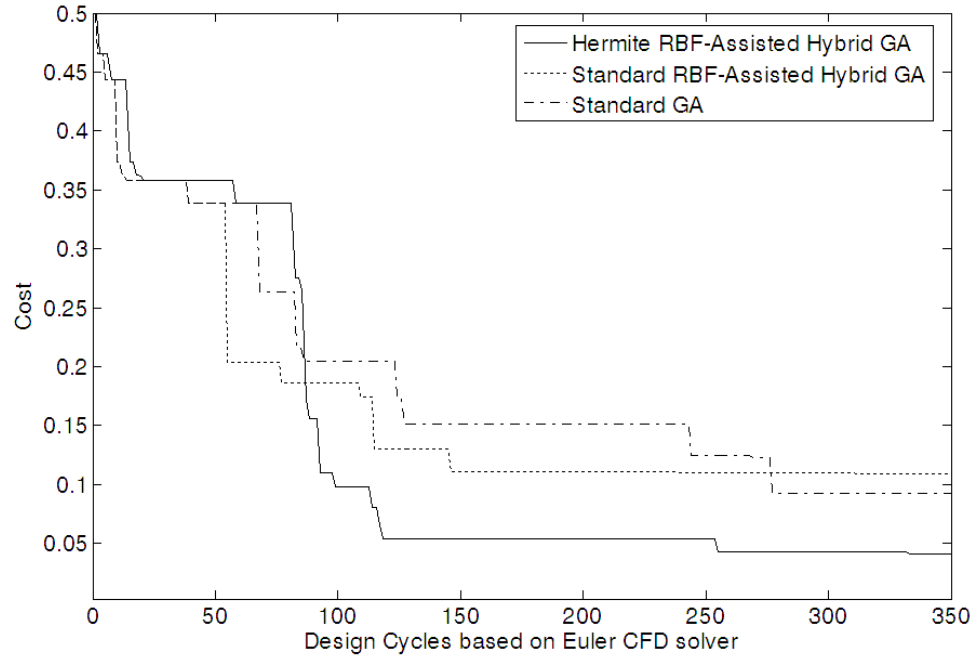

Figure 8. Convergence trends of HRBF-assisted hybrid GA, SRBF-assisted hybrid GA and standard GA for the transonic inverse pressure design problem, Case Study II.

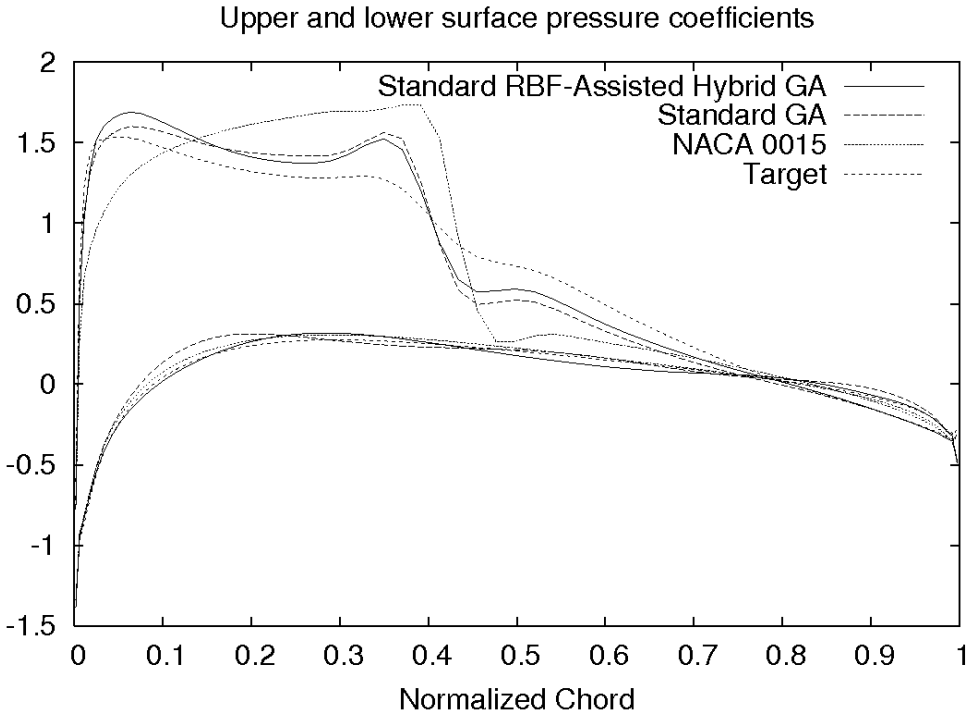

Figure 9. Pressure distribution profiles of optimal designs obtained after 333 design cycles (based on the Euler solver) using standard GA and SRBF-assisted hybrid GA optimization compared with the original NACA 0015 airfoil and the target pressure profile for Case Study II.

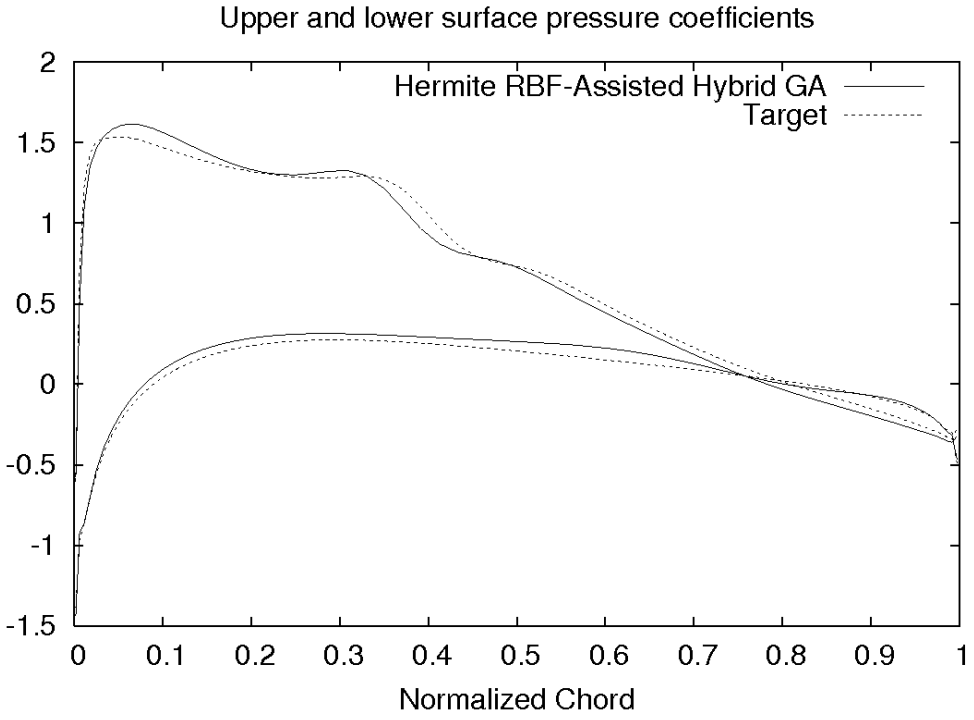

Figure 10. Pressure distribution profile of optimal design obtained using HRBF-assisted hybrid GA optimization after 333 design cycles (based on the Euler solver) for Case Study II.

Table 3. Aerodynamic performances (normalized values) of optimal airfoil geometries obtained using Standard GA, SRBF-assisted hybrid GA and HRBF-assisted hybrid GA for Case Study II

Inverse Pressure Design	**Drag *D***	**Lift *L***	***D*/*L* ratio**
NACA 0015 (baseline)	0.0316	0.6212	0.0509
Standard GA	0.0154	0.5834	0.0263
SRBF-assisted hybrid GA	0.0142	0.6253	0.0228
HRBF-assisted hybrid GA	0.0122	0.5657	0.0215

5.3 Discussion

In this section, numerical studies were presented for the evolutionary design optimization of two airfoil aerodynamic design problems with computational expensive adjoint Euler solvers. The results show that using Hermite interpolation techniques to construct gradient-enhanced radial basis function networks in the proposed algorithm help to generate a more accurate surrogate model than that based on function values only. The numerical results presented also show that the HRBF-assisted hybrid GA is capable of converging to the global optimum accurately on a limited computational budget. Most importantly, the proposed algorithm contributes significant speedup in the evolutionary search. This is evident for the airfoil design problems considered where the HRBF-assisted hybrid GA results in faster convergence when compared to both SRBF-assisted hybrid GA and the Standard GA.

6. Concluding Remarks

In this paper, we have presented a hybrid evolutionary algorithm that leverages Hermite radial basis function interpolants for optimization of computationally expensive adjoint solvers on a limited computational budget. Our focus was on aerodynamic design problems where the sensitivities of the objective function can be efficiently computed using adjoint methods. The key idea was to employ Hermite interpolation techniques to construct gradient-enhanced radial basis function networks so that more accurate surrogate models can be constructed than those based on function values only. Besides expediting the search process, the proposed hybrid evolutionary algorithm also guarantees convergence since a trust-region approach is used to interleave the exact analysis model with a surrogate model using local search.

Numerical studies were presented for evolutionary design optimization of airfoil aerodynamic design problems analyzed using computationally expensive adjoint solvers. Case Studies show that our proposed algorithm converges to good designs on a much lower computational budget. Most importantly, our studies also clearly show that the use of Hermite interpolants results in faster convergence compared to standard radial basis function approximations on the airfoil design problems considered.

Acknowledgments

This work has been funded in part under the A*STAR SERC Grant No. 052 015 0024 administered through the National Grid Office. The authors would like to thank all members of the Grid-based PSE for Engineering of Materials (GPEM) projects and the Emerging Research Laboratory, at School of Computer Engineering, Nanyang Technological University for their support in making this work possible.

References

1. Y. S. Ong, P. B. Nair and A. J. Keane, "Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modeling", *American Institute of Aeronautics and Astronautics Journal*, Vol. 41, No. 4, 2003, pp. 687-696.
2. K. C. Giannakoglou, "Design of Optimal Aerodynamic Shapes using Stochastic Optimization Methods and Computational Intelligence", *Progress in Aerospace Sciences,* 38, 2001, pp. 43-76.
3. Y. S. Ong and A. J. Keane, "Meta-Lamarckian Learning In Memetic Algorithm", *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 2, pp. 99-110, April 2004.

4. N. Alexandrov, J. E. Dennis, R. M. Lewis and V. Torczon, "A Trust Region Framework for Managing the use of Approximation Models in Optimization," *Structural Optimization*, Vol. 15, No. 1, 1998, pp. 16-23.
5. A. J. Booker, J. E. Jr Dennis, P. D. Frank, D. B. Serafini, V. Torczon and M. W. Trosset, "A Rigorous Framework for Optimization of Expensive Functions by Surrogates", *Structural Optimization*, Vol. 17, No.1, 1998, pp. 1-13.
6. D. B. Serafini, "A Framework for Managing Models in Nonlinear Optimization of Computationally Expensive Functions", PhD Thesis, Rice University, 1998.
7. T. W. Simpson, A. J. Booker, D. Ghosh, A. A. Giunta, P. N. Koch and R. J. Yang, "Approximation Methods in Multidisciplinary Analysis and Optimization: A Panel Discussion", *Proceedings of the Third ISSMO/AIAA Internet Conference on Approximations in Optimization*, pp. 14-25, 2002.
8. M. A. El-Beltagy, P. B. Nair and A. J. Keane, "Metamodelling Techniques For Evolutionary Optimization of Computationally Expensive Problems: Promises and Limitations," *Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufman*, pp. 196-203, 1999.
9. K. H. Liang, X. Yao and C. Newton, "Evolutionary Search of Approximated N-dimensional Landscapes," *International Journal of Knowledge-Based Intelligent Engineering Systems*, Vol. 4, No. 3, 2001, pp. 172-183.
10. Z. Z. Zhou, Y. S. Ong, P. B. Nair, A. J. Keane and K. Y. Lum, "Combining Global and Local Surrogate Models to Accelerate Evolutionary Optimization", *IEEE Transactions On Systems, Man and Cybernetics - Part C*, Vol. 36, No. 6, pp. 814 - 823, November 2006.
11. Y. S. Ong, P. B. Nair, A. J. Keane and K. W. Wong, "Surrogate-Assisted Evolutionary Optimization Frameworks for High-Fidelity Engineering Design Problems", *Knowledge Incorporation in Evolutionary Computation*, editor: Y. Jin, Studies in Fuzziness and Soft Computing Series, Springer Verlag, pp. 307 - 331, 2004.
12. Y. Jin, M. Olhofer and B. Sendhoff, "A Framework for Evolutionary Optimization with Approximate Fitness Functions," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 5, 481-494, 2002.

13. Y. Jin, "A Comprehensive Survey of Fitness Approximation in Evolutionary Computation", *Soft Computing Journal,* vol. 9, no. 1, pp. 3-12, 2005.
14. Y. S. Ong, P. B. Nair and K. Y. Lum, "Max-Min Surrogate-Assisted Evolutionary Algorithm for Robust Aerodynamic Design", *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 4, pp. 392-404, August 2006.
15. C. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, 1995.
16. J. Sacks, W. J. Welch, T. J. Mitchell and H. P. Wynn, "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, pp. 409-435, 1989.
17. C. K. I. Williams and C. E. Rasmussen, "Gaussian Processes for Regression", *Advances in Neural Information Processing Systems*, edited by D. S. Touretsky, M. C. Mozer, M. E. Hasselmo, MIT Press, 1996.
18. A. Jameson, "Aerodynamic Design via Control Theory", *Journal of Scientific Computing*, Vol. 3, No. 3, 1988, pp.233-260.
19. J. L. Lions, *Optimal Control Of Systems Governed by Partial Differential Equations*, S. K. Mitter (translation), Springer-Verlag, Berlin , New York, 1971.
20. A. Jameson and J. Reuther, "Control Theory Based Airfoil Design Using the Euler Equations", AIAA 94-4272-CP, 1994.
21. G. W. Burgreen and O. Baysal, "Three-Dimensional Aerodynamic Shape Optimization of Wings Using Discrete Sensitivity Analysis", *AIAA Journal*, Vol. 34, No. 9, 1996, pp.1761-1770.
22. J. Reuther, A. Jameson, J. J. Alonso, M. J. Rimlinger and D. Saunders, "Constrained Multipoint Aerodynamic Shape Optimization Using Adjoint Formulation and Parallel Computers", AIAA Paper 97-0103, January 1997.
23. A. Jameson and J. C. Vassberg, "Computational Fluid Dynamics for Aerodynamic Design: Its Current and Future Impact", AIAA 2001-0538, January 2001.
24. R. M. Hicks and P. A. Henne, "Wing Design by Numerical Optimization", *Journal of Aircraft*, Vol. 15, No. 7, 1978, pp. 407-412.
25. R. H. Myers and D. C. Montgomery, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, John Wiley & Sons Inc., 1995.

26. Friedman, J. H., Multivariate adaptive regression splines, *Annals of Statistics*, No. 19, 1991, pp. 1-141.
27. R. Jin, W. Chen and T. W. Simpson, "Comparative Studies of Metamodeling Techniques under Multiple Modeling Criteria", *Structural and Multidisciplinary Optimization*, Vol. 23, No. 1, 2001, pp. 1-13.
28. Hardy, R. L., "Theory and Applications of the Multiquadric-Biharmonic Method", *Computer and Mathematics with Applications*, Vol. 19, 1990, pp. 163-208.
29. Zhongmin, W., "Hermite-Birkhoff Interpolation of Scattered Data by Radial Basis Functions", *Approximation Theory and Applications,* Vol. 8, 1992, pp. 1-10.
30. Narcowich, F. J. and Ward, J. D., "Generalized Hermite Interpolation via Matrix-Valued Conditionally Positive Definite Functions", *Mathematics of Computation*, Vol. 63, No. 208, 1994, pp. 661-687.
31. Fasshauer, G., "Hermite Interpolation with Radial Basis Functions on Spheres,'' *Advances in Computational Mathematics,* Vol. 10, 1999, pp. 81-96.
32. J. F. Rodriguez, J. E. Renaud and L. T. Watson, "Convergence of Trust Region Augmented Lagrangian Methods Using Variable Fidelity Approximation Data", *Structural Optimization*, Vol. 15, No. 3-4, 1998, pp. 141-156.
33. C. T. Lawrence and A. L. Tits, "A Computationally Efficient Feasible Sequential Quadratic Programming Algorithm," *SIAM Journal on Optimization*, Vol. 11, No. 4, 2001, pp. 1092-1118.
34. I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", *International Journal of Supercomputer Applications*, vol. 15, no. 3, 2001.
35. Q. T. Ho, Y. S. Ong and W. T. Cai, "Gridifying Aerodynamic Design Problem Using GridRPC", *Second Grid and Cooperative Computing: Second International Workshop* 2003, Shanghai, China, LNCS, Springer-Verlag Heidelberg, Part I, Subject: Computer Science, Volume 3032 / 2004, pp. 83 – 90, April 2004.
36. H. K. Ng, D. Lim, Y. S. Ong, B. S. Lee, L. Freund, S. Parvez and B. Sendhoff, "A Multi-Cluster Grid Enabled Evolution framework for Aerodynamic Airfoil Design Optimization", *International Conference on Natural Computing*, LNCS 3611, pp. 1112-1121, Springer-verlag, August 2005, L. P. Wang, K. Chen and Y. S. Ong, Editors.

37. A. J. Keane and P. B. Nair, *Computational Approaches for Aerospace Design*, Chapter 4, Joh-Wiley and Sons, 2005.

38. Nanyang Campus Grid, http://ntu-cg.ntu.edu.sg/

39. M. B. Giles and N. A. Pierce, "An Introduction to the Adjoint Approach to Design", *Flow, Turbulence and Combustion*, vol. 65, 2000, pp. 393-415.

40. W. B. Song and A. J. Keane, "A Study of Shape Parameterisation Methods for Airfoil Optimisation", *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2004, AIAA 2004-4482, p 2031-2038.