OPTIMAL HERMITE COLLOCATION APPLIED TO A ONE-DIMENSIONAL CONVECTION-DIFFUSION EQUATION USING A HYBRID OPTIMIZATION ALGORITHM

KAREN L. RICCIARDI[1] AND STEPHEN H. BRILL[2]

[1] University of Massachusetts Boston, Department of Mathematics
[2] Boise State University, Department of Mathematics

Abstract

The Hermite collocation method of discretization can be used to determine highly accurate solutions to the steady state one-dimensional convection-diffusion equation (which can be used to model the transport of contaminants dissolved in groundwater). This accuracy is dependent upon sufficient refinement of the finite element mesh as well as applying upstream weighting to the convective term through the determination of collocation locations which meet specified constraints. Due to an increase in computational intensity of the application of the method of collocation associated with increases in the mesh refinement, minimal mesh refinement is sought. A hybrid method that utilizes a genetic algorithm and a hill-climbing approach is used to search for the optimal mesh refinement for a number of models differentiated by their velocity fields. The genetic algorithm is used to determine a mesh refinement that is close to a locally optimal feasible mesh refinement. Following the genetic algorithm, a hill-climbing approach is used to determine a truly local optimal mesh refinement that is feasible. In most cases the mesh refinements determined with this hybrid method are equally optimal or a significant improvement over previous mesh refinements determined through direct search methods.

1. INTRODUCTION

The method of collocation, while simple to implement, has not traditionally been the preferred finite element approach taken when determining a numerical solution to groundwater flow and transport models. Advances in this method were made early in the development of finite elements when it was found that for certain differential equations, the collocation locations that minimize local discretization error occur at the points of Gaussian quadrature for each element in the finite element mesh [Prenter, 1975] [deBoor, 2001]. While this advancement found that the approximate solution at the collocation locations minimized local discretization error, it did not solve the difficulties inherent in oscillatory artifacts associated with the application of collocation to solve for transport migration in an convection driven flow field.

Improvements in the method of collocation applied to the convection-diffusion equation occurred in the mid 1980's [Allen, 1983]. Allen found that by using upstream weighting on the convective term, i.e. assigning a different collocation location for different terms in the convection-diffusion finite element approximation, oscillations could be reduced in the numerical approximation. These results were taken one step further when it was shown that by careful refinement of the finite element mesh, the approximation could be substantially improved for accuracy [Brill, 2006].

Most numerical methods used for solving differential equations improve in accuracy after mesh refinement. The computational intensity of numerical methods is almost always directly related to the discretization of the solution space, i.e. the mesh complexity. Useful numerical solvers are computationally intense and require the use of computer technology to implement. While computer technology is capable of performing numerical tasks, the technology is limited and hence, it is advantageous to the modeler to minimize the required computations for a given algorithm. The mesh refinement exercise for Brill's optimized finite-element application is one such example where it is advantageous to obtain a mesh refinement that contains the smallest number of nodes such that the constraints placed on the system are satisfied.

A genetic algorithm followed by a hill-climbing algorithm is used in this work to determine a locally optimal refinement of the steady-state finite element mesh used to approximate the solution to the convection-diffusion equation in one dimension. This optimization algorithm determines a new finite element mesh for the proposed system as well as the upstream collocation locations for the convective term. In doing such, the method of collocation in conjunction with the determined mesh refinement can be used to efficiently determine an accurate solution to the convection-diffusion equation.

2. COLLOCATION AND CONSTRAINTS

The convection-diffusion equation consists of a differential equation with two terms, one that describes transport due to convection and another that describes transport due to diffusion:

$$-D\frac{d^2u}{dx^2} + v\frac{du}{dx} = 0. \tag{1}$$

The parameter D describes the diffusion, which is assumed to be constant over the entire region considered. The parameter v is the velocity of the groundwater and may vary over the entire region; however it is assumed to be a constant value on each element in the finite element mesh. The function $u(x)$ is the concentration of the contaminant in the groundwater and is the function to be approximated using the method of collocation.

When using the method of collocation to approximate the solution to the convection-diffusion equation in one-dimension, it has been shown that upstream weighting applied to the convective term will improve the accuracy of the solution and minimize numerical artifacts associated with the method of collocation [Allen, 1983]. Through refinement of the finite element mesh in a precise manner, along with upstream weighting, the numerical artifacts associated with the method of collocation can be eliminated and the results of approximation are highly accurate [Brill, 2006].

Brill has successfully determined the required mathematical relationships between the locations of the nodes of a finite element mesh and the collocation locations for the convective and diffusive terms for each element so that the method of collocation results in highly accurate numerical approximations to the true solution of our convection-diffusion equation [Brill, 2006]. The parameters of these relationships include (1) an initial number of elements for the model, p, (2) the constant velocity of the groundwater over each element, $v_1, v_2, ..., v_p$, and (3) the diffusivity for the contaminant, D, which is constant over the entire model. The refinement of the original mesh prescribed will be such that the i^{th} element will be divided into m_i equally spaced elements and the amount of upstream weighting applied to the convective term, ζ_i, will fall within bounds set in Brill's 2006 paper, i.e. $|\zeta_i| \leq \frac{1}{2} - \frac{1}{\sqrt{12}}$.

To search for the minimum mesh refinement, begin by determining the largest index associated with the element assigned the highest velocity value for the groundwater model. This index is called the k^{th} index. It is over this element that there is a single equation that relates the element's refinement, m_k, to the collocation locations over this element, $\pm\frac{1}{\sqrt{12}} - \zeta_k$ [Brill, 2006]. Given a fixed m_k, ζ_k is determined to be the root of a quadratic equation, namely ζ_k satisfies $G(\zeta_k, m_k) = 0$.

All other ζ_i (where $i \neq k$) values can be determined from fixed m_k and m_i values through the use of equations derived by Brill. These equations relate consecutive elements in such a way that the unknown ζ_i values can be determined as long as m_i is known as well as either (1) ζ_{i-1} and m_{i-1} are known or (2) ζ_{i+1} and m_{i+1} are known. In both cases, ζ_i is determined to be the root of a high order polynomial, and so ζ_i satisfies (1) $F_i(\zeta_i(m_i, \zeta_{i+1}, m_{i+1})) = 0$ or (2) $H_i(\zeta_i(m_i, \zeta_{i-1}, m_{i-1})) = 0$. Numerical methods must be used to determined such a root. It is sufficient that only one root of this polynomial be within the bounds set by Brill. This root is the assigned ζ_i value.

3. OPTIMIZATION PROBLEM

The convection-diffusion equation can be solved with very high accuracy using the method of collocation provided certain relationships exist between the location of the nodes of the finite-element mesh and the collocation location assigned to the convective term in the convection-diffusion equation (Brill 2006). Obtaining the relationships necessary for the application of collocation with highly accurate solutions often requires refinement of a crude finite element mesh that may have been defined with the intent to capture physical distinctions within the model, such as spatially variable changes in parameter values. Because computational intensity involved in the application of the method of collocation is related to the number of elements prescribed in the model, it is advantageous to determine a minimal mesh refinement that optimizes the numerical accuracy of the approximating solution.

The optimization problem is to find the vector $\mathbf{m} = (m_1, m_2, ..., m_p)$ such that the sum of the components of $\mathbf{m}$ is minimized while the ζ_i values all adhere to the prescribed relationships given by the functions G, F_i and H_i and that the ζ_i values all fall within the required upper and lower bounds. The boundaries placed upon the ζ_i values vary nonlinearly as a function of $\mathbf{m}$, so while the objective function for this problem is linear, the non-linearity of the constraints results in a feasible region that is non-convex.

The constrained optimization problem is transformed into an unconstrained optimization problem so that global optimization techniques can be employed. Violations of those constraints that pertain to the ζ_i values are permitted, however, when this occurs, a penalty term is added to the value of the objective function.

If it is found that given a fixed $\mathbf{m}$ at least one of the polynomials G, F_i and H_i does not have a real root, then a violation value is determined for that $\mathbf{m}$. Examining the complex roots for each polynomial with the smallest magnitude, ζ_i, the violation is determined to be the sum of the Euclidian distances from the violating ζ_i values to the closest boundary value placed upon all ζ_i values, namely $\frac{1}{2}-\frac{1}{\sqrt{12}}$ or $-\frac{1}{2}+\frac{1}{\sqrt{12}}$. Once the sum of the violations has been determined, this value is multiplied by a penalty weight of 100, then added to the sum of the m_i values to obtain the value of the transformed objective function. This optimization problem is expressed as:

Objective:

$$\min \sum_{i=1}^{p} (m_i + 100 * \max(0, \xi_i)) \tag{2}$$

Subject to:

$$G(\zeta_k, m_k) = 0 \tag{3}$$

$$F_i(\zeta_i(m_i, \zeta_{i+1}, m_{i+1})) = 0 \tag{4}$$

$$H_i(\zeta_i(m_i, \zeta_{i-1}, m_{i-1})) = 0 \tag{5}$$

$$m_i \in \mathbf{Z}^+ \text{ and } \zeta_i \in \mathbf{C} \quad \forall \quad i = 1..p \tag{6}$$

where ξ_i is given by:

$$\xi_i = \begin{cases} 0, & \text{if } \zeta_i \in \mathbf{R} \text{ and } |\zeta_i| \le \frac{1}{2} - \frac{1}{\sqrt{12}}; \\ \sqrt{\left(\text{Re}(\zeta_i) - \left(\frac{1}{2} - \frac{1}{\sqrt{12}}\right)\right)^2 + (\text{Im}(\zeta_i))^2}, & \text{if } \zeta_i \in \mathbf{C}, \zeta_i \notin \mathbf{R} \text{ and } \text{Re}(\zeta_i) \ge 0; \\ \sqrt{\left(\text{Re}(\zeta_i) - \left(-\frac{1}{2} + \frac{1}{\sqrt{12}}\right)\right)^2 + (\text{Im}(\zeta_i))^2}, & \text{if } \zeta_i \in \mathbf{C}, \zeta_i \notin \mathbf{R} \text{ and } \text{Re}(\zeta_i) < 0. \end{cases}$$

and $\mathbf{C}$ denotes the set of complex numbers.

4. OPTIMIZATION SOLVER: HYBRID SEARCH ALGORITHM

The method used to solve this optimization problem is a hybrid method that consists of two search phases: a genetic algorithm (GA) and a local gradient search algorithm. The objective of the first phase, the GA phase, is to obtain a solution that is within close proximity of a local optimal solution. In the second phase, the GS phase, a local search is conducted to obtain the local optimal solution in the region determined. Because of the discrete nature of this optimization problem and the non-linearity in this problem, there are no current optimization techniques developed that ensure that the determined local solution is the global solution; however, this does not preclude one from using the available techniques to determine a reasonable solution.

4.1. **Genetic algorithm.** To begin the genetic algorithm (GA) value encoding is employed so that a parent population of size n of the $\mathbf{m}$ vectors is created by randomly selecting positive integer values for each m_i variable within some prescribed range of values [Holland, 1992]. The upper bound is based upon preliminary analysis of the problem to determine a value that, when setting all elements of the $\mathbf{m}$ vector equal to it, will result in a feasible solution. The lower bound, for all m_i where $i \neq k$ is 1. And the lower bound for m_k is set equal to the minimum value of m_k such that $|\zeta_k| \le \frac{1}{2} - \frac{1}{\sqrt{12}}$.

Once an initial population has been determined, a fitness value is assigned to each member of the population. This fitness value is defined to be the sum of the m_i values plus a penalty term associated with any violations of the ζ_i constraints (Equation 2). Using the parent population as a starting point, subsequent populations are generated through a genetic algorithm to obtain a solution that is within the proximity of a locally optimal solution. Elitism is employed to preserve the optimal members of each generation. Following elitism, members of a new generation are determined from the previously generated population through crossover and mutation. For this problem one randomly selected crossover location is used to generate each member of the new generation.

The genetic algorithm ends when a maximum number of generations has been derived. Because of the random nature of the genetic algorithm, no assurance can be made that the algorithm results in a global optimal solution. Only an exhaustive search of **m** vectors with incrementally larger magnitudes can make this assurance. As the as the number of elements considered in the physical model increases, however, an exhaustive search is not feasible.

4.2. **Local search.** Convergence to a local optimal solution using a GA is inhibited due to the variable magnitudes of the elements of the optimal **m** vectors. At later stages of the GA, improvements in fitness become reliant upon mutation. Values of m_i that are relatively low are closer to their optimal value than m_i values which are large, and hence, the most beneficial mutation events are dependent upon (1) the mutation rate, which is typically set to be a low value, (2) the mutation direction being one that results in an improvement of the fitness of **m**, and (3) the likelihood of the mutation occurring at an m_i value that leads to substantial improvement of the fitness of **m**. The conditions for beneficial mutation occurring at later stages of this GA are not frequent, and convergence to a local optimal is slow. For these reasons, a local search algorithm is developed for this problem.

While the GA is not an efficient method for determining a local optimal **m** for this problem, the GA is very effective at determining a region where a local optimal solution exists. After the GA has determined a region where a local optimal solution exists, the newly developed local search is used to determine the true local optimal.

The local search that has been developed is specifically developed for this integer problem with scaling issues and know trajectories for improvement in the feasible region. The local search determines the fitness values of a number of **m** vectors in the downward gradient direction of the solution determined with the GA. This set will contain at most p neighboring **m** vectors determined in the following manner. First a step size that is directly related to the size of m_i is determined for each i direction. Each member of the neighboring set is the same as the **m** determined through the GA except the m_j^{th} element. The m_j^{th} element is set equal to m_j from the solution to the GA minus the j^{th} step size. If this step results in a vector where either the fitness value has not improved or the original constraints for the optimization problem are violated then the step size is reduced by one half. The search for a better-fit-feasible neighbor that varies in the j^{th} element is implemented with the reduced step size. Reduction of the step size continues until a neighboring vector is determined to have an improvement in fitness and zero penalty. If the j^{th} step size is determined through the reduction process to be less than one, this implies that the solution to the GA cannot be improved in the j^{th} direction, and no neighboring vector is included in the neighboring set.

Once the neighboring set has been determined, the vectors within the set are ranked according to their fitness values. The best fit solution in the neighboring set is then determined. The region of this new best solution is examined in the local search, which continues until the neighboring set contains no vectors. The last best fit solution is the locally optimal solution in the region of the solution determined through the GA.

4.3. **Example problems.** The search for an optimal mesh refinement for the application of the method of collocation to solve our convection-diffusion equation is applied to six one-dimensional problems with (initially) four equally spaced nodes. These models are broken up into two groups that differ in the values of the diffusivity constants. The first group has a diffusivity constant of 5, while the second group has a diffusivity constant of 1. The three subgroups differ in the velocity values assigned to each node. The six example problems along with the optimal mesh refinement values are listed in Table 1.

The penalty weight for the problem where $D = 5$ is 100 and where $D = 1$ is 200. Violations in the constraints will be multiplied by this value and added to the objective function so that this optimization problem can be viewed as an unconstrained optimization problem.

The parameters of the genetic algorithm and local search are: population size: 50; mutation rate: 20%; elitism percent: 10%; parent percent used for offspring: 60%; number of iterations in GA: 70; total number of GA runs: 100; number of iterations in local search: 100. The mutation rate is 20% for this problem, which is a very high value for most GAs. In this problem the string of integers to be optimized contains only four elements, and so a mutation rate of less than 25% is sufficiently small for optimization.

5. RESULTS

Optimal mesh refinements determined with the hybrid optimization algorithm are listed in Table 1. The hybrid method was run for each example 100 times, each with a different randomly generated initial parent population. In most instances the solutions determined with the hybrid algorithm are equal to or better than previous results determined through a direct search [Brill, 2006]. In a few instances the results determined through the hybrid method produced a solution that is not an improvement over the direct search method.

The value of D is set equal to 5 in Examples 1 through 3. This differs from the value of D set in Examples 4 through 6, which is 1. Other than this difference, Example 1 is the same as Example 4, Example 2 is the same as Example 5 and Example 3 is the same as Example 6.

The results from Examples 1 and 4 are a vast improvement over the direct search method. The solution for Example 1 is $\mathbf{m} = (24, 1, 1, 1)$ obtained in 100% of the runs, and the maximum error of the numerical solution determined with this discretization is $2.187 * 10^{-14}$. The three equal solution are determined in 86% of the runs for Example 4. These solutions are $\mathbf{m} = (44, 1, 10, 1)$ obtained in 60% of the runs, $\mathbf{m} = (45, 1, 9, 1)$ obtained in 17% of the runs and $\mathbf{m} = (46, 1, 8, 1)$ obtained in 9% of the runs. The maximum errors for each of these discretizations are $5.246 * 10^{-6}$, $7.021 * 10^{-6}$ and $9.229 * 10^{-6}$ respectively.

TABLE 1. Values of parameters in the computational examples. $D = 5, 1$.

Example Number	D	j	v_j	Direct Search m_j	Hybrid Search m_j	Solution Frequency	Hybrid Search ζ_j	Hybrid Maximum Error
1	5	1	100.0	41	24	100%	$-1.258 * 10^{-5}$	$2.187 * 10^{-14}$
		2	0.1	8	1		$2.100 * 10^{-1}$	
		3	10.0	8	1		$-8.473 * 10^{-4}$	
		4	1.0	8	1		$1.324 * 10^{-1}$	
2	5	1	10.0	1	1	100%	$6.149 * 10^{-4}$	$6.772 * 10^{-15}$
		2	0.1	1	1		$-2.097 * 10^{-1}$	
		3	100.0	24	24		$-1.259 * 10^{-5}$	
		4	1.0	1	1		$2.074 * 10^{-3}$	
3	5	1	0.1	2	1	100%	$-1.680 * 10^{-1}$	$1.669 * 10^{-8}$
		2	1.0	2	1		$1.660 * 10^{-3}$	
		3	10.0	2	3		$-7.563 * 10^{-5}$	
		4	100.0	17	22		$-1.635 * 10^{-5}$	
4	1	1	100.0	108	44*	86%	$-2.598 * 10^{-4}$	$5.246 * 10^{-6}$
		2	0.1	16	1		$4.789 * 10^{-1}$	
		3	10.0	16	10		$-3.742 * 10^{-4}$	
		4	1.0	16	1		$2.082 * 10^{-3}$	
5	1	1	10.0	1	1	100%	$8.567 * 10^{-3}$	$1.776 * 10^{-15}$
		2	0.1	1	1		$-2.061 * 10^{-1}$	
		3	100.0	54	54		$-1.396 * 10^{-4}$	
		4	1.0	1	1		$1.798 * 10^{-3}$	
6	1	1	0.1	1	1*	99%	$-2.081 * 10^{-1}$	$1.665 * 10^{-15}$
		2	1.0	1	2		$3.990 * 10^{-3}$	
		3	10.0	4	6		$-2.432 * 10^{-4}$	
		4	100.0	36	50		$-1.762 * 10^{-4}$	

* Multiple solutions of equal value are determined. See text for these solutions.

The direct search results for Example 2 and 5 are identical to the results determined using the hybrid algorithm in 100% of the runs. These discretizations are $\mathbf{m} = (1, 1, 24, 1)$ and $\mathbf{m} = (1, 1, 54, 1)$ and the maximum errors are $6.772 * 10^{-15}$ and $1.776 * 10^{-15}$ respectively.

The hybrid algorithm fails to determine a solution that is equal to or an improvement over the direct search method in Examples 3 and 6. In Example 3 the direct search method determines an optimal solution of $\mathbf{m} = (2, 2, 2, 17)$ while the hybrid method determines the optimal solution to be $\mathbf{m} = (1, 1, 3, 22)$. The maximum error of this discretization is $1.669 * 10^{-8}$. In Example 6 the direct search method determines an optimal solution of $\mathbf{m} = (1, 1, 4, 36)$ while the hybrid method determines the optimal solution to be $\mathbf{m} = (1, 2, 6, 50)$ in 77% of the runs and $\mathbf{m} = (1, 1, 6, 51)$ in 22% of the runs. The maximum errors of these discretizations are $1.665 * 10^{-15}$ and $1.850 * 10^{-6}$ respectively. In one out of 100 runs of the hybrid optimization algorithm applied to Example 6, a solution of $\mathbf{m} = (1, 1, 5, 44)$ with a maximum error of $1.731 * 10^{-5}$ was determined. This solution, while an improvement over the other 99 runs, is still not an improvement over the solution determined with the direct search.

To understand why the hybrid algorithm was not able to determine a solution equal to or better than a solution determined with the direct search method, it is useful to examine the geometry of the feasible region for the optimization problem.

Because the objective function is dependent upon m_1, m_2, m_3 and m_4, the relationship between the independent variables and the objective function cannot be fully viewed in one figure. By fixing $m_1 = 1$ and $m_2 = 1$ in both Example 3 and Example 6 the feasible regions of the objective function can be viewed as a function of m_3 and m_4 only (Figure 1, 2). In Figure 1 and 2, the asterisks represent feasible solutions to the optimization problem. In these example problems the feasible region is separated into multiple disjoint regions. In Example 3 (Figure 1), the region where the optimal solution $\mathbf{m} = (2, 2, 2, 17)$ exists is very small compared to the region where the solution $\mathbf{m} = (1, 1, 3, 22)$ exists. In Example 6 (Figure 2), the feasible region is divided into three disjoint regions. While the hybrid method was able to locate one of the small disjoint regions in one of the runs for Example 6, namely the solution $\mathbf{m} = (1, 1, 5, 44)$, the hybrid method did not successfully locate the global optimal solution located in the smallest of the three disjoint feasible regions. In both Example 3 and Example 6, the hybrid algorithm fails to find the global optimal because the feasible region is disjoint and because the region where the global optimal exists is very small in the scope of the problem.

Because the hybrid method is a random start method, the likelihood of random combinations of m_i values falling within the region of the global optimal solution is quite small. The GA successfully determines the large region where a feasible solution exists, and the local search successfully determines the local optimal solution within this region; however neither the GA or the local search are able to locate the smaller feasible region where the global optimal solution exists. If the GA were permitted to search indefinitely through mutation for the global optimal solution, there is a chance that the region of the global optimal solution would be found. This chance, however, is small and so the GA would need to conduct a very large number of mutations before this region would be found. This process would be very inefficient, and is not explored.

6. CONCLUSIONS

To determine an optimal mesh refinement so that the method of collocation can determine highly accurate solutions to the steady state one-dimensional convection-diffusion, one must solve a nonlinear integer optimization problem. The results of this optimization problem provide both the mesh refinement as well as the upstream collocation locations for the convective term for each element of the finite element mesh.

The constrained optimization problem was transformed into an unconstrained optimization problem. This was done by allowing violations of the constraints, however, when the violations occurred, the objective function was penalized. This transformation allowed for the application of a random search algorithm, namely a genetic algorithm.

Due to variations in scale of the resultant optimal solutions, the GA was limited in its search capabilities, and so the GA was terminated after a maximum number of iterations. Refinement of the solution was done through a newly developed local search technique designed to handle the variations in scale for integer problems.

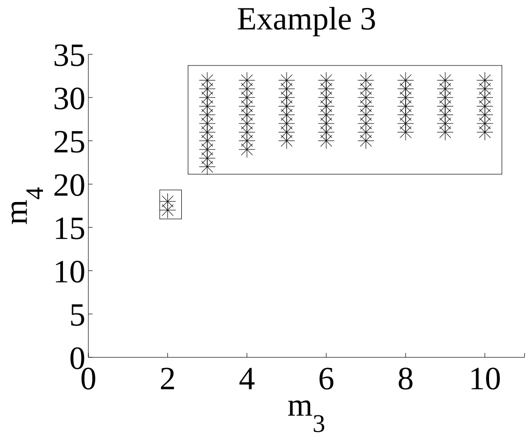

FIGURE 1. Example 3: $m_1, m_2 = 1$. Asterisks represent feasible solutions.

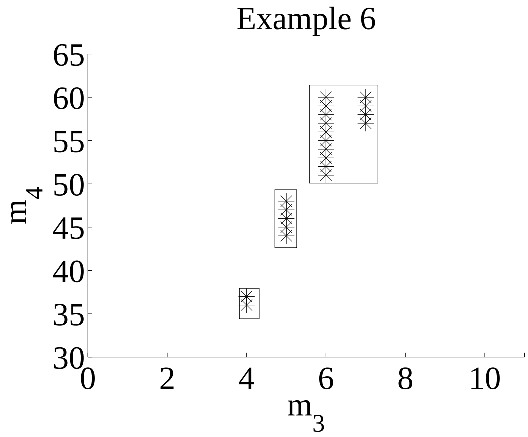

FIGURE 2. Example 6: $m_1, m_2 = 1$. Asterisks represent feasible solutions.

This hybrid of a GA and a local search algorithm was successful at determining discretization solutions that were equal to or better than previous results determined through the direct search method. Cases where the hybrid method failed to improve upon the direct search method were explored more fully. The results of this exploration indicate that in some instances this optimization problem is one with disjoint feasible regions. In the two instances where the direct search method determined the global optimal and the hybrid method failed, the global optimal solution was found to be in a small disjoint region. Because GAs are random search methods, the likelihood of finding the small disjoint regions where the global optimal solution exists is very small. Alternative optimization techniques should be explored to overcome this challenge.

References

Allen, 1983. Allen, M.B., "How Upstream Collocation Works", *International Journal of Numerical Methods in Engineering*, Vol. 19, pp. 1753-1763, 1983.

Brill, 2006. Brill, S.H., "Optimal Collocation Solution of the One-Dimensional Steady-State Convection-Diffusion Equation with Variable Coefficients", *Journal of Computational and Numerical Analysis and Applications*, in press (2006).

deBoor, 2001. deBoor, C., *A Practical Guide to Splines*, Rev. ed., Springer-Verlag, New York, v. 27, 2001.

Holland, 1992. Holland, J. "Genetic algorithm", *Scientific American*, July 1992, pp. 66-72.

Prenter, 1975. Prenter P.M., "Splines and Variational Methods",John Wiley & Sons, New York, 1975.