

# Optimal Hermite collocation applied to a one-dimensional convection-diffusion equation using an adaptive hybrid optimization algorithm.

## 1 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 or downstream 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. Very often this optimization problem is one where the feasible region is disjoint and as such requires a specialized optimization search technique. An original hybrid method that utilizes a specialized adaptive genetic algorithm followed by a hill-climbing approach is used to search for the optimal mesh refinement for a number of models differentiated by their velocity fields. The adaptive genetic algorithm is used to determine a mesh refinement that is close to a locally optimal mesh refinement. Following the adaptive genetic algorithm, a hill-climbing approach is used to determine a local optimal feasible mesh refinement. In all cases the optimal mesh refinements determined with this hybrid method are equally optimal to or a significant improvement over mesh refinements determined through direct search methods.

**Keywords:** integer programming, collocation, mesh refinement, hybrid optimization

**Classification:** Technical paper

## 2 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 element theory 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 a convection driven flow field.

Improvements in the method of collocation applied to the convection-diffusion equation occurred in the mid 1980's. Upstream collocation on Hermite cubics as compared to orthogonal collocation with respect to effective diffusion was found to improve the accuracy of the solution to the convection-diffusion equation (Dougherty and Pinder, 1982). Soon after

this, 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 (Allen, 1983). 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).

Determining the solution to partial differential equations using collocation methods reduces to solving linear systems that are in general non-symmetric and non-diagonally dominant. For this reason one focus on improving the efficiency of collocation methods has been to develop approaches that simplify the associated linear system. One such technique involved block iterative methods for the collocation equations of elliptic PDEs defined on a rectangular domain and subject to uncoupled mixed boundary conditions (Lai *et al.*, 1994). More recent advances with this focus have involved techniques that reduce the number of unknowns associated with each node in the discretization. One such method uses polynomials to approximate the first-order derivatives of the solution to the flow and transport equations over a construct about each node in the discretization (Fedele *et al.*, 2004). Another method that is structured on multiquadrics radial basis functions effectively changes the traditional global collocation approach to one that is made locally over a set of overlapping domains of influence (Vertnik and Sarler, 2006). In this approach, small linear systems with dimension equal to the number of nodes in each domain must be solved. This method and further advances in this methodology have been illustrated both on the convection problem in porous media as well as in more complex systems (Sarler *et al.*, 2004; Vernik and Sarler, 2006). Further studies using the radial basis functions include the double boundary collocation Hermitian approach for the solution of steady state convection-diffusion problems (LaRocca and Powers, 2007). In this approach, at the boundary collocation points the boundary condition and the governing partial differential equation are required to be satisfied simultaneously, resulting in higher precision results, especially for prediction of the fluxes at the boundaries.

While finding techniques that decrease the computational intensity associated with collocation techniques is one focus of interest, others involved in the study of the methods of collocation have focused on the accuracy and precision of the resultant solutions. A coordinate transform approach that couples solutions using a finite difference approach to a method that utilizes Hermite collocation is developed to address one dimensional differential problems with steep solutions (Lang and Sloan, 2002). Recent advances in finding accurate solutions for three dimensional boundary value problems using the method of collocation have involved applying genetic algorithms to determine optimal arrangements of collocation points (Zieniuk *et al.*, 2005).

In general numerical methods used for solving differential equations improve in accuracy with 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 intensive 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.

An adaptive genetic algorithm followed by a hill-climbing algorithm is used in this work to determine an 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 or downstream 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.

### 3 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:

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

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.

The method of collocation is one that falls under the category of the "method of weighted residuals" in the field of finite element analysis. In finite element analysis, the weighted sum of specified base functions over each element is used to approximate the solution to a differential equation. To solve the one-dimensional convection-diffusion equation using the method of collocation two points of collocation must be specified for each element in the discretization. These points of collocation are fixed for each element prior to the application of the method of collocation and affect the accuracy of the determined approximation to the solution. As such the collocation points are decision variables in the search for an optimally accurate approximation to the convection-diffusion equation 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 can 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 or downstream 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 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 velocity of the groundwater over each element,  $v_1, v_2, \dots, v_p$ , 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 collocation locations,  $\zeta_i$ , will fall within bounds set in Brill's 2006 paper.

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  $m_k$ ,  $\zeta_k$  is determined to be the root of a quadratic equation.

All other  $\zeta_i$  (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  $\zeta_i$  values can be determined as long as  $m_i$  is known as well as either (1)  $\zeta_{i-1}$  and  $m_{i-1}$  are known or (2)  $\zeta_{i+1}$  and  $m_{i+1}$  are known. In both cases,  $\zeta_i$  is determined to be the root of a polynomial of potentially high degree. Numerical methods must be used to determine 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  $\zeta_i$  value.

To summarize, once an  $m_k$  value is found such that  $\zeta_k$  falls within the assigned bounds set for all  $\zeta_i$  values, one can fix all other  $m_i$  values where  $i \neq k$  and determine the associated  $\zeta_i$  values. The goal is to find values of  $m_i$  such that (1)  $\sum_{i=1}^p m_i$  is minimized and (2)  $\zeta_i$  satisfy their given constraints.

## 4 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 for maximum accuracy and minimum computational intensity associated with finding the solution to the convection-diffusion equation using the method of collocation is given by:

minimize:

$$\sum_{i=1}^p m_i \quad (2)$$

subject to:

$$G(\zeta_k, m_k) = 0 \quad (3)$$

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

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

$$|\zeta_i| \leq \frac{1}{2} - \frac{1}{\sqrt{12}} \quad (6)$$

$$m_i \in \mathbf{Z}^+ \text{ and } \zeta_i \in \mathbf{C} \quad \forall \quad i = 1..p \quad (7)$$

where  $m_i$  is the number of new elements, i.e. the mesh refinement in element  $i$ ,  $\mathbf{Z}^+$  denotes the set of positive integers and  $\mathbf{R}$  denotes the set of real numbers. If  $m_i = 1$ , then no mesh refinement occurs. The value  $k$  is a fixed integer that is equal to the largest index of the element with the largest value of velocity in the physical model. The value  $p$  is the number of elements in the original physical model. The value of  $\zeta_i$  is the amount of upstream weighting, when  $\zeta_i$  is determined to be greater than zero, or downstream weighting, when  $\zeta_i$  is determined to be less than zero, applied to the convective term, and hence is related to the collocation location for the convective term over element  $i$ . The functions  $G$ ,  $F$  and  $H$  refer to the relationships prescribed by Brill in order to obtain maximum accuracy in the application of the method of collocation for this problem. These functions are described in detail in Appendix A.

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

Through a direct search method Brill has determined discretizations that result in highly accurate approximations (Brill, 2006). This work, however, was conducted on a problem where there were initially only four initial elements in the finite-element mesh. As the number of initial elements considered in this problem increases, the dimensionality of the problem increases and a direct search method is not an efficient search technique for determining a discretization that results in highly accurate solutions.

## 5 Optimization Solver: Hybrid Search Technique

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  $\zeta_i$  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  $G$ ,  $F$  and  $H$  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,  $\zeta_i$ , the violation is determined to be the sum of the Euclidian distances from the violating  $\zeta_i$  values to the closest boundary value placed upon all  $\zeta_i$  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,  $\phi$ , then added to the sum of the  $m_i$  values to obtain the value of the transformed objective function. This modification of the optimization problem is expressed as:

Objective:

$$\min \sum_{i=1}^p (m_i + \phi * \max(0, \xi_i)) \quad (8)$$

Subject to:

$$G(\zeta_k, m_k) = 0 \quad (9)$$

$$F_i(\zeta_i(m_i, \zeta_{i+1}, m_{i+1})) = 0 \quad (10)$$

$$H_i(\zeta_i(m_i, \zeta_{i-1}, m_{i-1})) = 0 \quad (11)$$

$$m_i \in \mathbf{Z}^+ \text{ and } \zeta_i \in \mathbf{C} \quad \forall \quad i = 1..p \quad (12)$$

where  $\xi_i$  is given by:

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

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

While it may seem that the objective function is now a function of continuous variables, namely  $\zeta_i$ , this is not the case. Recall that the  $\zeta_i$  values are dependent upon the integer values  $m_i$ . This optimization problem remains of the integer type; however, the transformation performed has rendered the objective function nonlinear.

The method used to solve this optimization problem is a hybrid method that consists of cycling through two search phases: an adaptive genetic algorithm (AGA) and a local gradient search algorithm (GS). The objective of the first phase, the AGA 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. Parameters that define the search space of the genetic algorithm within the AGA phase are updated at the start of each cycle of the search. By changing these parameters the optimization algorithm is able to improve its efficiency in determining disjoint feasible regions where improved locally optimal solutions may exist. For the problem considered in this paper two regions are disjoint if the minimum distance between these regions is greater than one. A connected region is one where the distance between any point in the region and its nearest neighbor is not greater than one. The search algorithm terminates when the solutions to successive cycles does not result in an improved locally optimal solution. Because of the discrete nature of this optimization problem and the nonlinearity 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.

### 5.0.1 Adaptive Genetic Algorithm

Each cycle of the hybrid optimization solver begins with the application of an adaptive genetic algorithm. To begin the adaptive genetic algorithm (AGA) (Holland, 1992) 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

Table 1: Creation of a member of a child population of length  $p = 15$  from a single crossover event where the random crossover location is at  $s = 5$ . The child is then mutated with a mutation rate 20%, which affects 3 elements of the child vector.

	Integer String of length $p = 15$														
Parent 1	<b>3</b>	<b>8</b>	<b>4</b>	<b>14</b>	<b>5</b>	8	2	3	1	6	4	9	1	2	3
Parent 2	4	9	5	7	2	4	<b>9</b>	<b>2</b>	<b>26</b>	<b>5</b>	<b>7</b>	<b>4</b>	<b>8</b>	<b>17</b>	<b>2</b>
Child	<b>3</b>	<b>8</b>	<b>4</b>	<b>14</b>	<b>5</b>	4	<b>9</b>	<b>2</b>	<b>26</b>	<b>5</b>	<b>7</b>	<b>4</b>	<b>8</b>	<b>17</b>	<b>2</b>
Mutated Child	<b>3</b>	<b>8</b>	21	<b>1</b>	<b>5</b>	4	<b>9</b>	7	<b>26</b>	<b>5</b>	<b>7</b>	1	<b>8</b>	<b>3</b>	<b>2</b>

of values. In the first cycle of this solver, 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. In subsequent cycles of this solver, the upper bound is set equal to the sum of elements of the most recently determined local optimal solution,  $\sum_i^n m_i^*$ . 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  $|\zeta_k| \leq \frac{1}{2} - \frac{1}{\sqrt{12}}$ . It is the adaptive nature of the upper bound of the genetic algorithm that increases the efficiency of this algorithm for determining locally optimal solutions in disjoint feasible regions.

Once a parent population has been determined at the start of each cycle 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  $\zeta_i$  constraints (Equation 11).

After a parent population has been determined and a fitness value has been assigned to each member of the parent population, a new population is derived from a subset of the parent population. This new population, referred to as the child population or the next generation, will contain the same number of members as the parent population. The derivation of the child population entails a number of processes.

In determining the child population from the parent population, elitism is employed, i.e. some members of the parent population that have the best fitness values become members of the child population.

Members of the child population not obtained through elitism are derived from a subset of the parent population with the highest fitness values (including those members that were involved in elitism). These members of the child population are created through recombination of the given set of parents. This recombination involves crossover and mutation.

In one single point crossover event, two parents,  $\mathbf{m}^{\mathbf{P1}}$  and  $\mathbf{m}^{\mathbf{P2}}$ , are randomly selected from that subset of the parent population that meets the fitness criteria set by the modeler for recombination. In crossover, a new vector of  $p$  integers, the child,  $\mathbf{m}^{\mathbf{C}}$ , is created by taking a combination of the two parent strings in the following manner. The first  $s$  elements of  $\mathbf{m}^{\mathbf{C}}$  are equal to the first  $s$  elements of  $\mathbf{m}^{\mathbf{P1}}$ , while the remaining  $p - s$  elements of  $\mathbf{m}^{\mathbf{C}}$  are equal to the final  $p - s$  elements of  $\mathbf{m}^{\mathbf{P2}}$  (Table 1).

There is only one crossover location,  $s$ , in this algorithm, however, this crossover location is randomly selected for each crossover event. Random crossover events are conducted until

the child population has as many members as the parent population.

Crossover events and elitism do not introduce new integer values into the population. To introduce new integer values into each string of  $m_i$ 's in each population, random integer values are changed or mutated after each crossover event (Table 1). The number of integers that undergo mutation is determined through a mutation rate set by the modeler. Members of the child population that have been determined through elitism do not undergo mutation.

The child population generated from the original parent population is referred to as the first generation. After this first child population is obtained, this population acts as a parent population for a new generation or new child population. The  $n^{th}$  generation is used to obtain the  $n + 1^{st}$  generation, whose fitness values for the best fit members are equal to, or surpass the fitness values of the generation from which they were derived.

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  $\mathbf{m}$  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.

## 5.1 Local Search

Preliminary results have indicated that the nature of this optimization problem is such that the optimal values of  $m_1, m_2, \dots, m_p$  may vary by orders of magnitude (Brill, 2006). Due to this variation, convergence to a local optimal solution using a AGA is inhibited. At later stages of the AGA, the populations approach uniformity, i.e. the members of the population are very similar and crossover events are not effective in generating members of a child population with improved fitness values. 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  $\mathbf{m}$ , and (3) the likelihood of the mutation occurring at an  $m_i$  value that leads to substantial improvement of the fitness of  $\mathbf{m}$ . The conditions for beneficial mutation occurring at later stages of this AGA are not frequent, and convergence to a local optimal is slow. For this reason, a local search algorithm is developed for this problem.

While the AGA is not an efficient method for determining a local optimal  $\mathbf{m}$  for this problem, the AGA is very effective at determining a region where a locally optimal solution exists. After the AGA has determined a region where a locally 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 designed for this integer problem that contains scaling issues and known trajectories for improvement in the feasible region. The local search determines the fitness values of a number of  $\mathbf{m}$  vectors within the vicinity of the solution determined with the AGA. This set will contain at most  $p$  neighboring  $\mathbf{m}$  vectors determined in the following manner. First a step size that is directly related to the

Table 2: An example of a neighboring set of  $\mathbf{m}$  where  $p = 4$ .

$\mathbf{m} =$	2	42	3	1
Step size	1	5	1	1
<b>Neighbors</b>				
Neighbor (1)	1	42	3	1
Neighbor (2)	2	37	3	1
Neighbor (3)	2	42	2	1
Neighbor (4)		none		

size of  $m_i$  is determined for each  $i$  direction. Each member of the neighboring set is the same as the  $\mathbf{m}$  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 AGA 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 magnitude of the step size is reduced to the greatest integer closest to the original magnitude 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 AGA cannot be improved in the  $j^{th}$  direction, and no neighboring vector is included in the neighboring set (Table 2).

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 AGA.

## 5.2 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 twelve 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 six subgroups differ in the velocity values assigned to each node. These twelve example problems along with the optimal mesh refinement values are listed in Table 3 and Table 4. Additionally, four one-dimensional problems with eight equally spaced nodes and where  $D = 1$  are examined (Table 5).

The penalty weight for the problem where  $D = 5$  is  $\phi = 100$  and where  $D = 1$  is  $\phi = 200$ . 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 are listed in Table 6. The mutation rate is 20% for this problem, which is a very high value for most genetic algorithms. In this problem the

Table 3: Values of parameters in the computational examples.  $D = 5$ .

Example Number	$D$	$j$	$v_j$	Direct Search $m_j$	Hybrid Search $m_j$	Solution Frequency	Hybrid Search $\zeta_j$	Hybrid Maximum Error
1	5	1	100.0	41	24	100%	$-1.259 * 10^{-5}$	$2.187 * 10^{-14}$
		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	0.1	1	1	100%	$-2.097 * 10^{-1}$	$6.661 * 10^{-16}$
		2	100.0	24	24		$-1.259 * 10^{-5}$	
		3	10.0	1	1		$-8.085 * 10^{-4}$	
		4	1.0	1	1		$1.288 * 10^{-1}$	
3	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		$2.074 * 10^{-3}$	
4	5	1	0.1	2	1	100%	$-3.886 * 10^{-2}$	$1.669 * 10^{-8}$
		2	1.0	2	1		$3.781 * 10^{-4}$	
		3	10.0	2	2		$-1.090 * 10^{-4}$	
		4	100.0	17	17		$-3.552 * 10^{-5}$	
5	5	1	0.1	1	1*	72%	$-2.089 * 10^{-1}$	$2.153 * 10^{-8}$
		2	100.0	43	42		$2.514 * 10^{-6}$	
		3	0.1	1	1		$-2.235 * 10^{-2}$	
		4	100.0	41	42		$-2.345 * 10^{-6}$	
6	5	1	100.0	9	1	100%	$1.908 * 10^{-2}$	$2.142 * 10^{-14}$
		2	0.1	9	1		$-2.097 * 10^{-1}$	
		3	100.0	42	24		$-1.259 * 10^{-5}$	
		4	0.1	9	1		$2.100 * 10^{-1}$	

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

Table 4: Values of parameters in the computational examples.  $D = 1$ .

Example Number	$D$	$j$	$v_j$	Direct Search $m_j$	Hybrid Search $m_j$	Solution Frequency	Hybrid Search $\zeta_j$	Hybrid Maximum Error
7	1	1	100.0	108	54	100%	$-1.396 * 10^{-4}$	$1.121 * 10^{-4}$
		2	0.1	16	1		$2.076 * 10^{-1}$	
		3	10.0	16	2		$-4.804 * 10^{-3}$	
		4	1.0	16	1		$1.819 * 10^{-1}$	
8	1	1	0.1	1	1	100%	$-2.061 * 10^{-1}$	$1.142 * 10^{-4}$
		2	100.0	97	54		$-1.396 * 10^{-4}$	
		3	10.0	11	2		$-4.743 * 10^{-3}$	
		4	1.0	1	1		$1.798 * 10^{-1}$	
9	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		$1.798 * 10^{-3}$	
10	1	1	0.1	1	1	91%	$-1.729 * 10^{-1}$	$5.3291 * 10^{-15}$
		2	1.0	1	1		$1.476 * 10^{-3}$	
		3	10.0	4	4		$-5.855 * 10^{-4}$	
		4	100.0	36	36		$-4.790 * 10^{-4}$	
11	1	1	0.1	1	1*	32%	$-2.108^{-1}$	$3.962 * 10^{-8}$
		2	100.0	145	139		$6.887 * 10^{-6}$	
		3	0.1	1	1		$-3.949 * 10^{-3}$	
		4	100.0	139	145		$-7.131 * 10^{-6}$	
12	1	1	100.0	18	1	100%	$-4.000 * 10^{-2}$	code failed to produce a solution
		2	0.1	18	1		$-2.061 * 10^{-1}$	
		3	100.0	111	54		$-1.396 * 10^{-4}$	
		4	0.1	18	1		$2.076 * 10^{-1}$	

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

Table 5: Values of parameters in the computational examples.  $D = 1$ .

Example Number	$D$	$j$	$v_j$	Hybrid Search $m_j$	Solution Frequency	Hybrid Search $\zeta_j$	Hybrid Maximum Error
13	1	1	0.1	1*	6%	$-2.076 * 10^{-1}$	$5.854 * 10^{-12}$
		2	1.0	1		$1.993 * 10^{-3}$	
		3	10.0	4		$-1.629 * 10^{-4}$	
		4	100.0	59		$-1.246 * 10^{-6}$	
		5	0.1	1		$3.374 * 10^{-3}$	
		6	1.0	2		$-7.161 * 10^{-5}$	
		7	10.0	7		$-1.627 * 10^{-5}$	
		8	100.0	63		$-1.087 * 10^{-5}$	
14	1	1	0.1	1	100%	$1.472 * 10^{-1}$	$1.195 * 10^{-12}$
		2	0.1	1		$-1.458 * 10^{-1}$	
		3	1.0	1		$1.384 * 10^{-3}$	
		4	1.0	1		$-1.333 * 10^{-3}$	
		5	10.0	4		$-5.459 * 10^{-5}$	
		6	10.0	5		$3.002 * 10^{-5}$	
		7	100.0	44		$-3.509 * 10^{-5}$	
		8	100.0	43		$-3.429 * 10^{-5}$	
15	1	1	100.0	1*	70%	$3.257 * 10^{-2}$	$7.812 * 10^{-1}$
		2	0.1	1		$-1.991 * 10^{-1}$	
		3	10.0	7		$9.757 * 10^{-5}$	
		4	1.0	1		$-1.044 * 10^{-3}$	
		5	100.0	45		$-2.991 * 10^{-5}$	
		6	0.1	1		$1.063 * 10^{-1}$	
		7	10.0	2		$-8.272 * 10^{-4}$	
		8	1.0	1		$4.635 * 10^{-2}$	
16	1	1	100.0	1	100%	$3.256 * 10^{-2}$	$9.436 * 10^{-1}$
		2	100.0	38		$-4.976 * 10^{-5}$	
		3	0.1	1		$2.099 * 10^{-1}$	
		4	0.1	1		$-2.074 * 10^{-1}$	
		5	10.0	2		$-7.751 * 10^{-4}$	
		6	10.0	2		$-6.283 * 10^{-5}$	
		7	1.0	1		$1.611 * 10^{-2}$	
		8	1.0	1		$-1.537 * 10^{-2}$	

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

Table 6: Values of parameters in the optimization algorithm.

---

Total number of optimization runs	100
<b>Genetic algorithm:</b>	
Population size	50
Mutation rate	20%
Elitism percent	10%
Parent percent used for offspring	60%
Number of iterations	70
<b>Local search:</b>	
Maximum number of iterations	100

---

string of integers to be optimized contains either four or eight elements, and so a mutation rate of less than 25% is sufficiently small for optimization.

## 6 RESULTS

Optimal mesh refinements determined with the hybrid optimization algorithm are listed in Tables 3, 4 and 5. The hybrid method was run for each example 100 times, each with a different randomly generated initial parent population. In all instances where a direct search method has been previously examined for the problem the solutions determined with the hybrid algorithm are equal to or an improvement over previous results (Brill 2006).

The value of  $D$  is set equal to 5 in Examples 1 through 6. This differs from the value of  $D$  set in Examples 7 through 12, which is 1. Other than this difference for Examples 1 through 12 Example 1 is the same as Example 7, Example 2 is the same as Example 8, etc. The value of  $D$  is set equal to 1 in Examples 13 through 16, and the velocity fields are variations of Example 10 and Example 7.

The results from Examples 1 and 6 are a vast improvement over the direct search method. One phase cycle of the hybrid algorithm inclusive of the AGA and the GS was utilized to determine these solutions. The solution for Example 1 is  $\mathbf{m} = (24, 1, 1, 1)$  and the solution for Example 6 is  $\mathbf{m} = (1, 1, 24, 1)$ . In both of these examples, the solutions were determined in 100% of the runs. The maximum errors of the numerical solutions determined with these discretizations are  $2.187 * 10^{-14}$  and  $2.142 * 10^{-14}$  respectively.

The direct search results for Example 2 and 3 are identical to the results determined using the hybrid algorithm in 100% of the runs. The maximum errors are  $6.661 * 10^{-16}$  and  $6.772 * 10^{-15}$  respectively. One phase cycle of the hybrid algorithm was utilized to determine these solutions.

In Example 5, a number of different but equal mesh refinement combinations were determined 72% of the time using the hybrid algorithm. These combinations include:  $\mathbf{m} = (1, 44, 1, 40)$ ;  $\mathbf{m} = (1, 43, 1, 41)$ ;  $\mathbf{m} = (1, 42, 1, 42)$  determined in 20%, 26% and 26% of the runs respectively. All of these solutions result in a highly accurate finite element approximation to the

convection-diffusion equation. The maximum errors associated with these discretizations are  $2.153 * 10^{-8}$ ,  $9.996 * 10^{-9}$  and  $1.300 * 10^{-9}$  respectively. These results were reached in one to nine phase cycles of the hybrid search algorithm.

The hybrid algorithm results in a modest improvement over the direct search method in Example 4. One to two phase cycles of the hybrid search method were utilized to obtain these results. 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  $\mathbf{m} = (1, 1, 2, 17)$ . The error associated with the improved hybrid solution is  $1.669 * 10^{-8}$ .

Some of the optimal discretizations determined with the hybrid method for the example problems where  $D = 5$  were obtained with the assistance of the adaptive abilities of the hybrid algorithm. To understand why the adaptive features of the hybrid algorithm were employed it is useful to examine the geometry of the feasible region for the optimization problem. For illustrative purposes the geometry of Example 4 is examined further.

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  $m_2 = 1$  the feasible region of the objective function for Example 4 can be viewed as a function of  $m_3$  and  $m_4$  only (Figure 1). In Figure 1, the asterisks represent feasible solutions to the optimization problem. In this example problem the feasible region is separated into two disjoint regions. The region where the optimal solution  $\mathbf{m} = (1, 1, 2, 17)$  exists is very small compared to the region where suboptimal solutions exist. In some instances the hybrid algorithm fails to find the global optimal after one phase cycle because the feasible region is disjoint and because the region where the global optimal exists is very small in the scope of the problem. By limiting the search space of the genetic algorithm in the second phase cycle of the hybrid algorithm, the likelihood of the random search method to determine a new feasible region is increased.

The genetic algorithm employed in the hybrid method is a random start method and for this reason the likelihood of random combinations of  $m_i$  values falling within the region of the global optimal solution is quite small. The first phase cycle of the AGA successfully determines the large region where a feasible solution exists, and the local search successfully determines the local optimal solution within this region. The second phase cycle effectively locates the smaller feasible region where the global optimal solution exists by eliminating all but the boundary of the large feasible region from consideration in the genetic algorithm. Because of the random nature of the genetic algorithm and the potentially small sizes of the disjoint feasible regions where globally optimal solutions may exist, it is possible that the AGA will not successfully locate truly optimal solutions. By limiting the search space of the AGA, the effectiveness of the algorithm is enhanced.

Example 4 clearly illustrates the benefits of using the multi-phase adaptive hybrid algorithm developed for this problem to determine a global optimal solution.

*Take in Figure 1 here.*

The results of the hybrid optimization algorithm for determining a minimal mesh refinement for Examples 7 through 12 are summarized in Table 4.

The results in Example 9 match the solution determined in the direct search method. In

Example 9 the results were determined in one phase cycle of the hybrid algorithm. The solution for Example 9 is  $\mathbf{m} = (1, 1, 54, 1)$  and was determined in 100% of the runs. The maximum error of the numerical solution given this discretizations is  $1.776 * 10^{-15}$ .

The results from Examples 7, 8 and 12 are a great improvement over the direct search method. These results were determined after one phase cycle of the hybrid algorithm. The respective solutions for Example 7, 8 and 12 are  $\mathbf{m} = (54, 1, 2, 1)$ ,  $\mathbf{m} = (1, 54, 2, 1)$  and  $\mathbf{m} = (1, 1, 54, 1)$ , all obtained in 100% of the runs. The maximum errors of the numerical solutions to Examples 7 and 8 are  $1.121 * 10^{-4}$  and  $1.142 * 10^{-4}$ . The computer code that would generate the numerical solution given the discretization for Example 12 crashes due to unknown reasons.

In Example 11, a number of different but equal optimal mesh refinement combinations were determined 32% of the time using one to eleven phase cycles of the hybrid algorithm. These solutions are equal to the solution determined in the direct search method. The discretizations as well as the associated maximum errors in the numerical solutions include:  $\mathbf{m} = (1, 145, 1, 139)$  with maximum error of  $2.243 * 10^{-8}$ ;  $\mathbf{m} = (1, 144, 1, 140)$  with maximum error of  $1.098 * 10^{-8}$ ;  $\mathbf{m} = (1, 143, 1, 141)$  with maximum error of  $1.543 * 10^{-14}$ ;  $\mathbf{m} = (1, 142, 1, 142)$  with maximum error of  $1.054 * 10^{-8}$ ;  $\mathbf{m} = (1, 141, 1, 143)$  with maximum error of  $2.064 * 10^{-8}$ ;  $\mathbf{m} = (1, 140, 1, 144)$  with maximum error of  $3.033 * 10^{-8}$ ; and  $\mathbf{m} = (1, 139, 1, 145)$  with maximum error of  $3.962 * 10^{-8}$ . These refinements occur in 1%, 7%, 3%, 4%, 1%, 6% and 10% of the runs so that the total number of equivalent solutions make up 32% of the runs.

Just as Example 4 was one such example for the  $D = 5$  case where the hybrid algorithm required more than one phase cycle to determine a solution, so did the solution for Example 10 where  $D = 1$ . Between one and 4 phase cycles of the hybrid method were utilized to obtain the discretization of  $\mathbf{m} = (1, 1, 4, 36)$  associated with a maximum error of  $5.3291 * 10^{-15}$  in 91% of the runs. By fixing  $m_1 = 1$  and  $m_2 = 1$  the feasible region of the objective function is viewed as a function of  $m_3$  and  $m_4$  in Figure 2. Just as in Example 4, the feasible region for Example 10 is separated into disjoint regions. In this example, there are three disjoint regions depicted in Figure 2. The adaptive hybrid method was able to locate the smallest of the three disjoint regions where the global optimal feasible solution exists.

*Take in Figure 2 here.*

Examples 13 through 16 illustrate the ability of this hybrid algorithm to determine mesh refinements that result in accurate solutions to our convective-diffusion equation using the method of collocation with upstream or downstream weighting. These example problems contain elements of Examples 10 and 11. The characteristics of the objective functions for each of this problems is quite difficult to observe given that the dimensionality of the problems has risen from four to eight. The error analysis of the results, however are perfectly reasonable to determine and are used as a measure of the effectiveness of the hybrid algorithm to determine reasonable results.

The results of the hybrid algorithm for Examples 13 through 16 are summarized in Table 5. In Example 13 the hybrid method determines three equal solutions:  $\mathbf{m} = (1, 1, 4, 59, 1, 2, 7, 63)$  with a maximum error of  $5.854 * 10^{-12}$  in 4% of the runs;  $\mathbf{m} = (1, 1, 4, 58, 1, 2, 7, 64)$  with a maximum error of  $4.815 * 10^{-12}$  in 1% of the runs;  $\mathbf{m} = (1, 1, 3, 53, 1, 2, 7, 55)$  with a

maximum error of  $1.815 * 10^{-11}$  in 1% of the runs. One to two phase cycles of the hybrid method were called to determine these solutions. While the equal discretizations determined above account for only 6% of the runs, it should be noted that the other 94% of the runs resulted in five different solutions that each contained just one more discretization refinement, for example  $\mathbf{m} = (1, 1, 4, 62, 1, 2, 7, 61)$ .

In Example 14 the hybrid method determined the solution  $\mathbf{m} = (1, 1, 1, 1, 4, 5, 44, 43)$  with a maximum error of  $1.195 * 10^{-12}$  in 100% of the runs. One to four phase cycles were required in the optimization runs.

The hybrid method applied to Example 15 determined the equivalent solutions of  $\mathbf{m} = (1, 1, 7, 1, 45, 1, 2, 1)$ ,  $\mathbf{m} = (1, 1, 8, 1, 44, 1, 2, 1)$  and  $\mathbf{m} = (1, 1, 9, 1, 43, 1, 2, 1)$  with maximum errors of  $7.812 * 10^{-1}$ ,  $7.729 * 10^{-1}$  and  $7.721 * 10^{-1}$  respectively in 37%, 20% and 13% of the runs for a total of 70% of the runs. One to eight phase cycles were employed in the hybrid method to obtain these solutions.

In Example 16 the hybrid method determined the solution  $\mathbf{m} = (1, 38, 1, 1, 2, 2, 1, 1)$  with a maximum error of  $9.436 * 10^{-1}$  in 100% of the runs. Only one phase cycle was required in all of the optimization runs.

## 7 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 or downstream 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. Refinement of the solution to the genetic algorithm was done through a newly developed local search technique designed to handle the variations in scale for this integer problem.

Due to possibility of disjoint feasible regions in the presented optimization problem, an adaptive hybrid method was developed to solve this problem. The adaptive hybrid method utilizes both an adaptive genetic algorithm (ADA) as well as a local gradient search (GS) technique developed specifically for this problem. This adaptive algorithm consisted of completing phase cycles of the AGA and the GS, with each cycle containing a search space that is limited by the results of the most recently completed phase cycle.

This adaptive hybrid method was successful at determining discretization solutions that were equal to or better than previous results determined through the direct search method. In cases where direct search methods have not been utilized to determine mesh refinements due to the high dimensionality of the problems, this method effectively determined mesh refinements that resulted in approximate solutions with reasonably small errors.

The geometry of the feasible regions of problems where multiple phase cycles were employed was examined. 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 hybrid method utilized multiple phase cycles to determine a solution, the global optimal solution was found to be in a small disjoint region. Because genetic algorithms are random search methods, the likelihood of finding the small disjoint regions where the global optimal solution exists is dependent upon the size of the search space. The adaptive hybrid algorithm effectively determines disjoint feasible regions because at each new phase cycle, the search space for the AGA is reduced, thereby increasing the likelihood of determining small disjoint feasible regions where the optimal solution may exist.

## 8 Appendix A

**To find  $\zeta_k$ .**

To determine a feasible  $\zeta_k$  from  $m_k$  where  $k$  refers to the largest index of the largest  $v_i$  in the model, find the roots of the following polynomial,  $G(\zeta_k)$ :

$$G(\zeta_k) = [\beta_k - \beta_k \exp(\beta_k)]\zeta_k^2 + [4 + \beta_k - 4 \exp(\beta_k) + \beta_k \exp(\beta_k)]\zeta_k + \frac{[\beta_k^2 + 6\beta_k + 12] - \exp(\beta_k)[\beta_k^2 - 6\beta_k + 12]}{6\beta_k}$$

In this equation  $\beta_i$  are the Peclet numbers for each element  $i$  and are functions of the mesh refinement of element  $i$ ,  $m_i$ , the velocity prescribed to element  $i$ ,  $v_i$ , the diffusion coefficient that is constant over the entire model,  $D$  and the maximum number on initial elements,  $p$ :

$$\beta_i = \frac{v_i}{Dpm_i}$$

**To find  $\zeta_i$  in terms of  $m_i, m_{i+1}$  and  $\zeta_{i+1}$ .**

To determine  $\zeta_i$  in terms of  $m_i, m_{i+1}$  and  $\zeta_{i+1}$ , find the roots of the following polynomial,  $F(\zeta_i)$ :

$$F(\zeta_i) = B_{i+1}[(\beta_i^2 + 6\beta_i + 12 + 6\beta_i\zeta_i(4 + \beta_i + \beta_i\zeta_i))^{m_i}][\beta_i^2\zeta_i^2 + 4\beta_i\zeta_i + 2] \\ - 2\beta_i pm_i [1 + \beta_i\zeta_i][(\beta_i^2 + 6\beta_i + 12 + 6\beta_i\zeta_i(4 + \beta_i + \beta_i\zeta_i))^{m_i}] \\ - B_{i+1}[(\beta_i^2 - 6\beta_i + 12 + 6\beta_i\zeta_i(4 - \beta_i + \beta_i\zeta_i))^{m_i}][\beta_i^2\zeta_i^2 + 4\beta_i\zeta_i + 2]$$

where

$$B_{i+1} = \frac{\left[ \frac{\rho_{i+1}}{\lambda_{i+1}^{m_{i+1}} - 1} \right] [\beta_i pm_i \exp(\beta_i m_i)]}{\left[ \frac{\beta_{i+1} pm_{i+1}}{\exp(\beta_{i+1} m_{i+1}) - 1} \right] [\exp(\beta_i m_i) - 1]}$$

and

$$\rho_{i+1} = \left[ \frac{2\beta_{i+1}}{h_{i+1}} \right] \left[ \frac{1 + \beta_{i+1}\zeta_{i+1}}{\beta_{i+1}^2\zeta_{i+1}^2 + 4\beta_{i+1}\zeta_{i+1} + 2} \right]$$

$$\lambda_{i+1} = \frac{\beta_{i+1}^2 + 6\beta_{i+1} + 12 + 6\beta_{i+1}\zeta_{i+1}(4 + \beta_{i+1} + \beta_{i+1}\zeta_{i+1})}{\beta_{i+1}^2 - 6\beta_{i+1} + 12 + 6\beta_{i+1}\zeta_{i+1}(4 - \beta_{i+1} + \beta_{i+1}\zeta_{i+1})}$$

**To find  $\zeta_{i+1}$  in terms of  $m_{i+1}, m_i$  and  $\zeta_i$ .**

To determine  $\zeta_{i+1}$  in terms of  $m_{i+1}, m_i$  and  $\zeta_i$ , find the roots of the following polynomial,  $H(\zeta_{i+1})$ :

$$\begin{aligned} H_i(\zeta_{i+1}) = & C_i [(\beta_{i+1}^2 + 6\beta_{i+1} + 12 + 6\beta_{i+1}\zeta_{i+1}(4 + \beta_{i+1} + \beta_{i+1}\zeta_{i+1}))^{m_{i+1}}] [\beta_{i+1}^2\zeta_{i+1}^2 + 4\beta_{i+1}\zeta_{i+1} + 2] \\ & - [\beta_{i+1}^2 - 6\beta_{i+1} + 12 + 6\beta_{i+1}\zeta_{i+1}(4 - \beta_{i+1} + \beta_{i+1}\zeta_{i+1})]^{m_{i+1}} \left[ \frac{2\beta_{i+1}}{h_{i+1}} \right] [1 + \beta_{i+1}\zeta_{i+1}] \\ & - C_i [\beta_{i+1}^2 - 6\beta_{i+1} + 12 + 6\beta_{i+1}\zeta_{i+1}(4 - \beta_{i+1} + \beta_{i+1}\zeta_{i+1})]^{m_{i+1}} [\beta_{i+1}^2\zeta_{i+1}^2 + 4\beta_{i+1}\zeta_{i+1} + 2] \end{aligned}$$

where

$$C_i = \frac{\left[ \frac{\rho_i \lambda_i^{m_i}}{\lambda_i^{m_i} - 1} \right] [\beta_{i+1} p m_{i+1}]}{\left[ \frac{\beta_i p m_i \exp(\beta_i m_i)}{\exp(\beta_i m_i) - 1} \right] [\exp(\beta_{i+1} m_{i+1}) - 1]}$$

## References

- [1] Allen, M.B. (1983), "How Upstream Collocation Works", *International Journal of Numerical Methods in Engineering*, Vol. 19, pp. 1753-1763.
- [2] Brill, S.H. (2006), "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.
- [3] deBoor, C.(2001), *A Practical Guide to Splines*, Rev. ed., Springer-Verlag, New York, Vol. 27.
- [4] Dogherty, D. and G.F. Pinder (1983), "A brief noe on upwind collocation", *International Journal for Numerical Methods in Fluids*, Vol. 3(3), pp. 307-313.
- [5] Douglas, J. and T. Dupont (1973), "A finite element collocation method for quasilinear parabolic equations", *Mathematics of Computation*, Vol. 27, No. 3, pp. 17-28.

- [6] Fedele, F., M. McKay, G.F. Pinder and J. Guarnaccia (2004), “Single-degree of freedom Hermite collocation for multiphase flow and transport in porous media”, *International Journal for Numerical Methods in Fluids*, Vol. 44(12), pp. 1337-1354.
- [7] Holland, J. (1992), “Genetic algorithm”, *Scientific American*, pp. 66-72.
- [8] Lai, Y., A. Hadjidimos, E. Houstis and J. Rice (1994), “General interior Hermite collocation methods for second-order elliptic partial differential equations”, *Applied Numerical Mathematics*, Vol. 16(1-2), pp. 183-200.
- [9] Lang, A.W. and D.M. Sloan (2002), “Hermite collocation solution of near-singular problems using numerical coordinate transformation based on adaptivity”, *Journal of Computational and Applied Mathematics*, Vol. 140(1-2), pp. 499-520.
- [10] LaRocca, A. and H. Powers (2007), “A double boundary collocation Hermitian approach for the solution of steady state convection-diffusion problems”, *Computers and Mathematics with Applications*, Vol 55(9), pp. 1950-1960.
- [11] Prenter P.M. (1975), “Splines and Variational Methods”, John Wiley & Sons, New York.
- [12] Sarler, B. J. Perko, C.S. Chen (2004), “Radial basis function collocation method solution of natural convection in porous media”, *International Journal of Numerical Methods for Heat and Fluid Flow*, Vol. 14(2), pp. 187-212.
- [13] Vertnik, R. and B. Sarler (2006), “Meshless local radial basis function collocation method for convective-diffusive solid-liquid phase change problems”, *International Journal of Numerical Methods for Heat and Fluid Flow*, Vol. 16(5), pp. 617-640.
- [14] Zieniuk, E., K. Szerszen and A. Boltuc (2005), “Genetic algorithms applied to optimal arrangement of collocation points in 3D potential boundary-value problems”, *Information Processing and Security Systems*, Springer, pp. 113-122.

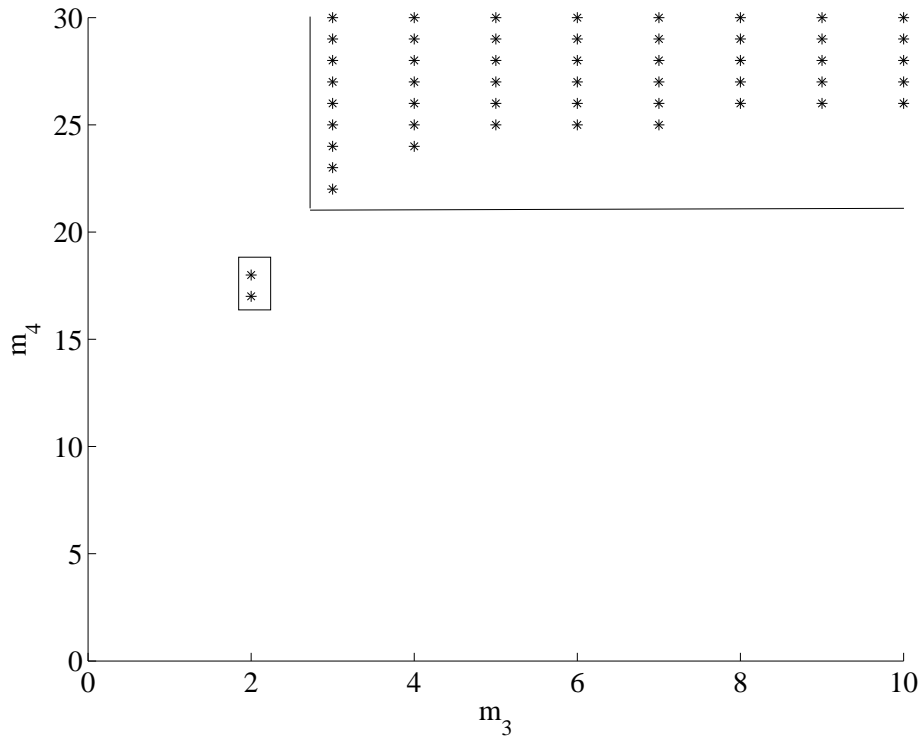


Figure 1: Example 4:  $m_1 = 1, m_2 = 1$ . Asterisks represent feasible solutions and the boxes indicate connected feasible regions.

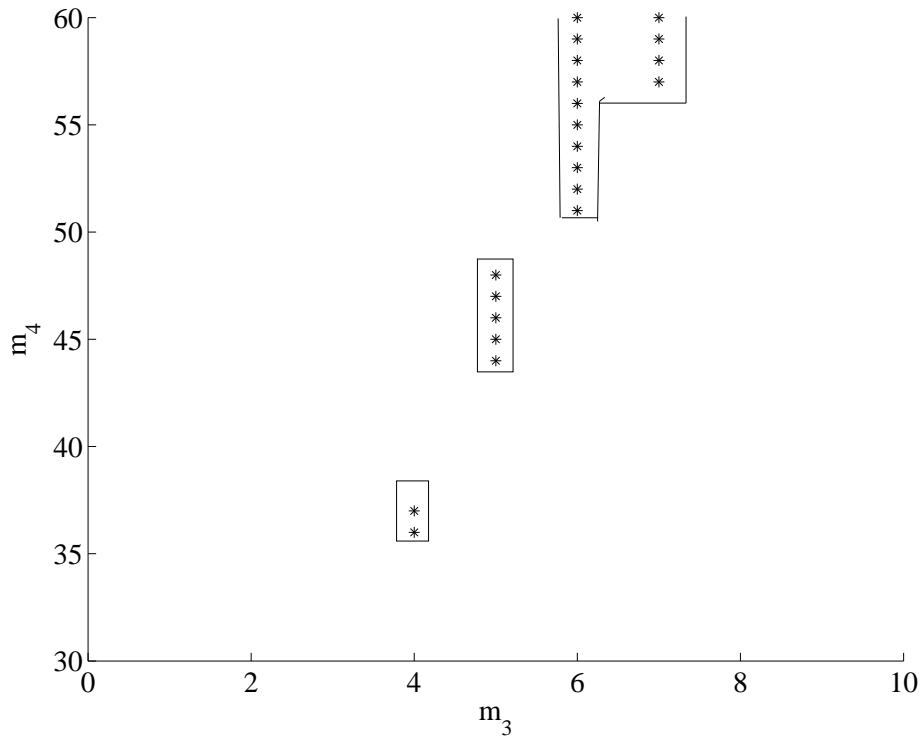


Figure 2: Example 10:  $m_1 = 1, m_2 = 1$ . Asterisks represent feasible solutions and the boxes indicate connected feasible regions.