WATER RESOURCES RESEARCH, VOL. 27, NO. 1, PAGES 31-36, JANUARY 1991

Convex Interpolation for Gradient Dynamic Programming

EFI FOUFOULA-GEORGIOU

St. Anthony Falls Hydraulic Laboratory, Department of Civil and Mineral Engineering, University of Minnesota, Minneapolis

Local approximation of functions based on values and derivatives at the nodes of a discretized grid are often used in solving problems numerically for which analytical solutions do not exist. In gradient dynamic programming (Foufoula-Georgiou and Kitanidis, 1988) the use of such functions for the approximation of the cost-to-go function alleviates the "curse of dimensionality" by reducing the number of discretization nodes per state while obtaining high-accuracy solutions. Also, efficient Newton-type schemes can be used for the stage-wise optimization, since now the approximation functions have continuous first derivatives. Our interest is in the case where the cost-to-go function is convex. However, the interpolants may not always be convex, introducing numerical problems. In this paper we address the problem of interpolating nodal values and derivatives of a one-dimensional convex function with a convex interpolant so that potential computational difficulties due to approximation-induced nonconvexity are avoided, and an efficient convergence to global instead of local optimal controls is guaranteed at every single-stage optimization.

1. INTRODUCTION

One of the oldest and most important algorithms in optimal control theory is discrete dynamic programming (DDP) [cf. *Bellman*, 1957]. This approach requires the discretization of the state space (and, in most applications, of the control space) and the solution of an optimization problem for each of the grid points. However, due to the exponential increase of the computer memory and computation time requirements with the number of state and control variables ("curse of dimensionality"), the applicability of DDP is limited to small-dimensional systems.

Several methods have been proposed over the years to overcome the dimensionality difficulties of DDP. One of the most well-studied methods is differential dynamic programming [*Mayne*, 1966; *Jacobson and Mayne*, 1970]. This method and other successive approximation methods (see, for example, the review article of *Yakowitz* [1982]), despite their success for deterministic optimization, are not directly applicable to stochastic optimal control problems. The main reason is that, because of the stochasticity of the input, a single state trajectory cannot now be projected with certainty. Instead, the whole optimal control policy over all states is needed, so that minimization of the expected cost can be obtained through integration over a range of states. Thus to date, the conventional DDP method remains the only universal approach to stochastic optimal control problems. This imposes a severe restriction on the dimensionality of the systems that can be solved under an explicit (and not implicit) stochastic framework.

A straightforward way of alleviating the "curse of dimensionality" associated with DDP is to use higher-order approximations of the cost-to-go function so that solutions of a desired accuracy can be obtained with a fewer number of nodes per state. Note that if n denotes the dimension of the problem (i.e., the number of states) and N_i denotes the number of nodes of the ith state, the high-speed memory requirements of conventional DDP are

$$\prod_{i=1}^{n} N_i$$

Thus reduction of N_i by a factor of 2 induces a reduction in storage requirements by a factor of 2^n. The idea of higher-order approximation and dynamic programming was initially explored by *Bellman and Dreyfus* [1962, Ch. 12], who used orthogonal polynomials for the cost-to-go functions so that storage of only the polynomial coefficients was needed instead of storage of the values of the cost function at all the nodal points of the multidimensional state space grid. This global approximation, however, has the main disadvantage that, at least for fast-changing functions, oscillatory approximations may be obtained. Also, if a function is hard to approximate in a particular domain of the state space, a poor approximation will result over the whole domain. *Daniel* [1976] and *Birnbaum and Lapidus* [1978] recognized the importance of using local approximations and explored the use of multidimensional B splines [e.g., *Schultz*, 1973]. Although splines provide approximations with continuous first and second derivatives, the first derivatives at the nodes are not preserved. For optimal control problems, however, it is the values of the derivatives (and not the values of the function) that are used for the computation of the optimal control. Also, in many cases, the optimal knots of the splines must be determined (a time-consuming process), or estimates of the derivatives must be provided so that a good spline approximation is obtained.

Recently, *Kitanidis and Foufoula-Georgiou* [1987] and *Foufoula-Georgiou and Kitanidis* [1988] studied a new computational algorithm in which the cost-to-go function was approximated within each element of the discretized state space using Hermite interpolation, that is, the lowest-order polynomials which preserve the values of the function and the values of its derivatives with respect to the state variables at all the nodes of the discretization grid. This class of algorithms was termed "gradient dynamic programming" because the gradient of the cost-to-go function is preserved at the state nodal points and is explicitly used in the calculations. The methodology and equations of gradient dynamic programming (GDP) were developed by *Foufoula-Georgiou and Kitanidis* [1988] in a general framework,

Paper number 90WR02032.
0043-1397/91/90WR-02032$05.00

permitting thus the incorporation of any interpolating function as long as it preserves the values of the cost-to-go function and its derivatives with respect to all the state variables at all the discretization nodes. Hermite interpolation polynomials provide approximations of high accuracy (see, for example, the error analysis of *Kitanidis and Foufoula-Georgiou* [1987]) and are computationally efficient especially for multidimensional spaces (see the discussion by *Kitanidis and Foufoula-Georgiou* [1988]).

A major handicap of optimization problems involving function approximation is that the approximants of convex functions may turn out to be nonconvex. This leads to numerical problems in terms of convergence of the optimization algorithms. In this paper we first study the convexity of the Hermite interpolation polynomials and give conditions for convex interpolation with this type of functions. It is shown that these conditions are much more restrictive than the conditions for the existence of any convex interpolant of a convex function. We then consider an alternative class of interpolating functions which preserve the values of the function and the values of its derivatives and are always convex when, of course, the data at the end points of the interpolating interval allow a convex interpolating function to exist.

Our study pertains to the one-dimensional interpolation problem as it is used in the context of gradient dynamic programming. It is hoped that our analysis will motivate research toward the development of a multidimensional convex interpolation theory.

2. A Brief Summary of Gradient Dynamic Programming

Let N denote the number of decision times (stages), n the dimension of the state vector $\mathbf{x}$, m the dimension of the control vector $\mathbf{u}$, and r the dimension of a random forcing function (input) $\mathbf{w}$. Also, let $\mathbf{x}(k)$ be the state vector at the beginning of period k, and $\mathbf{u}(k)$, $\mathbf{w}(k)$ the control and random input vectors, respectively, during period k. Without loss of generality we may assume that the random vectors $\mathbf{w}(k)$, $k = 1, \ldots, N$ are independent of each other. Serially correlated inputs can be accounted for through state augmentation.

Let the system dynamics be described by the state transition function $\mathbf{T}_k$ such that

$$\mathbf{x}(k+1) = \mathbf{T}_k[\mathbf{x}(k), \mathbf{u}(k+1), \mathbf{w}(k+1)] \qquad k=0, 1, \ldots, N-1 \tag{1}$$

A typical set of constraints will consist of lower and upper bounds on the control and state variables

$$\mathbf{u}^{\min}(k+1) \le \mathbf{u}(k+1) \le \mathbf{u}^{\max}(k+1)$$

$$\mathbf{x}^{\min}(k+1) \le \mathbf{x}(k+1) = \mathbf{T}_k(\mathbf{x}(k), \mathbf{u}(k+1), \mathbf{w}(k+1)) \le \mathbf{x}^{\max}(k+1) \qquad k = 0, 1, \ldots, N-1$$

We restrict our attention to linear state transition equations and linear constraints.

The objective of a discrete-time stochastic optimal control problem is to find the sequence of optimal controls $\{\mathbf{u}^*(k)\}$, $k = 1, \ldots, N$ which minimizes the performance criterion

$$J = \underset{\mathbf{w}(1), \mathbf{w}(N)}{E} \sum_k \{C_k[\mathbf{x}(k), \mathbf{u}(k+1)]\} + F_N[\mathbf{x}(N)] \tag{2}$$

given an initial state vector $\mathbf{x}(0)$. The performance criterion (objective function) consists of the sum of the single-stage cost functions $C_k[\mathbf{x}(k), \mathbf{u}(k+1)]$ over the whole operating horizon and a terminal cost $F_N[\mathbf{x}(N)]$. The expectation is taken with respect to the random vectors $\mathbf{w}(1), \ldots, \mathbf{w}(N)$.

Let $F_k := F_k[\mathbf{x}(k)]$ denote the cumulative cost associated with the state vector $\mathbf{x}(k)$ and the optimal control policy from k to the end of the operating horizon. We will refer to this function as the cost-to-go at stage k. Then, the functional equation of the system takes the form

$$F_{k-1}[\mathbf{x}(k-1)] = \min_{\mathbf{u}(k)} \{C_{k-1}[\mathbf{x}(k-1), \mathbf{u}(k)] + \underset{\mathbf{w}(k)}{E} F_k[\mathbf{T}(\mathbf{x}(k-1), \mathbf{u}(k), \mathbf{w}(k))]\} \qquad k = 1, \ldots, N \tag{3}$$

On the basis of the principle of optimality [*Bellman*, 1957], any multistage optimization problem which is separable in stages may be decomposed through dynamic programming into a sequence of single-stage optimization problems. Below we describe the gradient dynamic programming methodology at a typical stage.

The state space is discretized and represented by a finite number of nodes. Let $\langle \mathbf{x}(k) \rangle$ denote a discretized value of the state vector $\mathbf{x}(k)$. Assume that at stage k the values of the cost-to-go function $F_k(\langle \mathbf{x}(k) \rangle)$ and the values of its first derivatives $\nabla_x F_k = dF_k(\mathbf{x})/d\mathbf{x}|_{\mathbf{x}=\langle \mathbf{x}(k) \rangle}$ are known for all the grid points, that is, all nodal state vectors $\langle \mathbf{x}(k) \rangle$. These values are known at the last operation period, $k = N$, and can be explicitly updated from stage to stage as the algorithm moves backward in time [see *Foufoula-Georgiou and Kitanidis*, 1988]. The GDP involves the following steps: (1) approximation of the cost-to-go function F_k with piecewise polynomials which preserve F_k and $\nabla_x F_k$ at all the grid points, (2) computation of the optimal control $\mathbf{u}_k^*[\langle \mathbf{x}(k-1) \rangle]$ associated with the nodal state vector $\langle \mathbf{x}(k-1) \rangle$, using a Newton-type constrained optimization method [*Luenberger*, 1984], (3) computation of the Jacobian of $\mathbf{u}_k^*[\langle \mathbf{x}(k-1) \rangle]$ with respect to the state vector $\langle \mathbf{x}(k-1) \rangle$, and (4) computation of the values of $F_{k-1}[\langle \mathbf{x}(k-1) \rangle]$ and $\nabla_x F_{k-1} = dF_{k-1}(\mathbf{x})/d\mathbf{x}|_{\mathbf{x}=\langle \mathbf{x}(k-1) \rangle}$. Once this is done for all possible state vectors $\langle \mathbf{x}(k-1) \rangle$, the solution to the single-stage optimization problem has been completed. The procedure is then repeated for all stages. Finally, a forward "sweep" starting at the given vector $\mathbf{x}(0)$ will determine the sequence of optimal controls $\{\mathbf{u}^*(k)\}$, $k = 1, \ldots, N$. A detailed discussion of this method and an application to the stochastic optimal control of a four-reservoir system can be found in the paper by *Foufoula-Georgiou and Kitanidis* [1988].

In step (1) of the above algorithm the function F_k needs to be interpolated. Although F_k is convex, problems may arise due to the nonconvexity of the interpolating function. It is precisely the problem of convex interpolants that is addressed in this paper. Thus from the whole dynamic programming scheme schematically shown for the one-dimensional case in Figure 1 we will concentrate only on one element of the discretized state space, as shown in the insert of Figure 1. In particular, let x_{e-1} and x_e denote the lower and upper nodes of the discrete interval of interest and $G(x)$ denote the interpolation function such that it preserves the

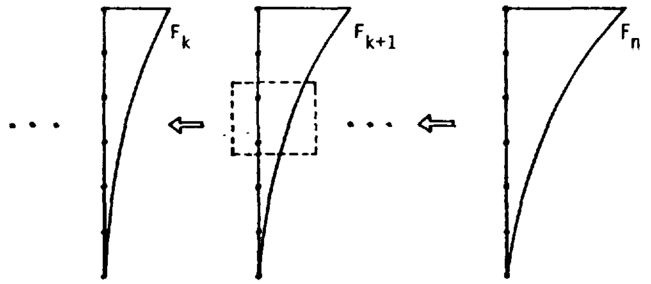

Fig. 1. One-dimensional schematic representation of the backward-moving discrete dynamic programming method. The area in the box is shown in detail in Figure 2.

values $F_k(x_{\ell-1})$, $F_k(x_\ell)$, $dF_k(x)/dx|_{x_{\ell-1}}$, and $dF_k(x)/dx|_{x_\ell}$. This element is shown in detail in Figure 2*a* and with further notational simplifications in Figure 2*b*. In the sequel we will use the terminology of Figure 2*b*.

3. Hermite Interpolation and Requirements for Convexity

In this section we study the necessary and sufficient conditions for the convexity of the one-dimensional Hermite interpolating polynomial.

Proposition 1. Let $g(x)$ be a third-degree polynomial $g(x) = \alpha x^3 + \beta x^2 + \gamma x + \delta$, $x \in [x_1, x_2]$, $x_2 = x_1 + \Delta$, such that at the ends of the interval $[x_1, x_2]$ satisfies the conditions

$$g(x_1) = f_1 \qquad g'(x_1) = f_1' \tag{4a}$$

$$g(x_2) = f_2 \qquad g'(x_2) = f_2' \tag{4b}$$

Then $g(x)$ is convex if and only if the following conditions hold:

$$0 \le f_2 - f_1 - \Delta f_1' \tag{5a}$$

$$\frac{3}{2} \le \frac{\Delta(f_2' - f_1')}{f_2 - f_1 - \Delta f_1'} \le 3 \tag{5b}$$

Proof. Without loss of generality, we simplify the problem by defining the function

$$h(x) = g(x + x_1) - (f_1 + x f_1') \qquad x \in [0, \Delta] \tag{6}$$

It is easy to see that $h(x)$ is of the form

$$h(x) = x^2(\alpha x + \beta) \tag{7}$$

$$0 = h(0) = h'(0) \tag{8a}$$

$$h := h(\Delta) \qquad d := h'(\Delta) \tag{8b}$$

Below we derive the convexity conditions for $h(x)$ in the interval $[0, \Delta]$ which imply the respective conditions for $g(x)$. We note that (h, d, Δ) is related to (f_1, f_2, f_1', f_2') by

$$h = f_2 - f_1 - \Delta f_1' \tag{9}$$

$$d = f_2' - f_1' \tag{10}$$

From (7) and (8) we obtain the parameters α and β in terms of (h, d, Δ) as

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = -\frac{1}{\Delta^4} \begin{bmatrix} 2\Delta & -\Delta^2 \\ -3\Delta^2 & \Delta^3 \end{bmatrix} \begin{bmatrix} h \\ d \end{bmatrix} \tag{11}$$

Clearly, $h \ge 0$ is a necessary condition for $h(x)$ to be convex, and this gives (5*a*). Elementary analysis shows that $h(x)$ is convex if and only if either $\alpha \ge 0$ and $\beta \ge 0$ or $\alpha < 0$ and the inflection point x_0, defined as the point where $h''(x_0) = 0$, is such that

$$x_0 = -\beta/3\alpha \ge \Delta \tag{12}$$

These cases are illustrated in Figures 3*a* and 3*b*, respectively. Using (11), the first of the above conditions becomes

$$2h/\Delta \le d \le 3h/\Delta \tag{13}$$

while the second becomes

$$3h/2\Delta \le d < 2h/\Delta \tag{14}$$

Combining (13) and (14), we obtain that $h(x)$ is convex if and only if

$$\frac{1}{2}\frac{3h}{\Delta} \le d \le \frac{3h}{\Delta} \tag{15}$$

Using (9) and (10), (5*b*) follows.

Thus Hermite interpolation will result in a convex interpolant when the nodal values of the cost-to-go function and its derivatives satisfy condition (15). If this condition is not satisfied, the resulting interpolation function will be nonconvex, and the optimization procedure may experience numerical difficulties in terms of efficiently converging to a global optimum.

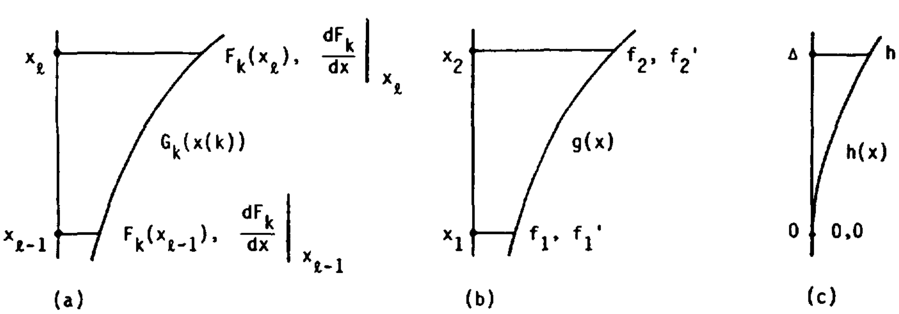

Fig. 2. An isolated element of the discretized state vector at stage *k*. (*a*) Definition of the variables used in the general description of gradient dynamic programming. (*b*) and (*c*) Simplified terminology used in the convex interpolation analysis. More details on the definition of the functions are given in the text.

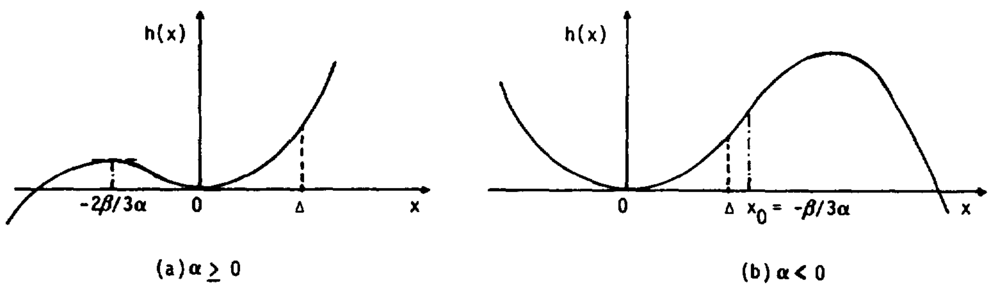

Fig. 3. Two cases of a cubic polynomial which is convex in the interval [0, Δ].

4. An Alternative Class of Functions for Convex Interpolation

In this section we propose and study a class of functions, as an alternative to cubic polynomials, which are continuously differentiable; they preserve the values and the derivatives at the nodal points, and they are always convex when the interpolation data allow a convex interpolant function to exist. We first give necessary and sufficient conditions for the existence of a convex interpolating function.

Proposition 2. Let $g(x)$ be a convex function with continuous first derivative and such that at the ends of an interval, $[x_1, x_2 = x_1 + \Delta]$ satisfies the condition

$$g(x_1) = f_1 \qquad g'(x_1) = f_1' \tag{16a}$$

$$g(x_2) = f_2 \qquad g'(x_2) = f_2' \tag{16b}$$

Then the following hold:

$$0 \le f_2 - f_1 - \Delta f_1' \tag{17a}$$

$$0 \le f_1 - f_2 + \Delta f_2' \tag{17b}$$

Also, if the data (f_1, f_2, f_1', f_2') satisfy (17), then there exists an interpolating function $g(x)$ which satisfies (16) and which is convex.

Proof. Without loss of generality, we work with the function $h(x)$, $x \in [0, \Delta]$, defined in (6). Note that $g(x)$ is convex if and only if $h(x)$ is. Clearly, $h = f_2 - f_1 - \Delta f_1' \ge 0$ is a necessary condition. Provided $h \ge 0$, $h(x)$ is convex if and only if

$$\Delta d \ge h \tag{18}$$

Sufficiency follows by noting that if (18) holds, a piecewise linear and convex interpolating function exists. A convex interpolating function with continuous first derivative can be obtained by smoothing the piecewise linear function where the first derivative is discontinuous.

To show the necessity of the condition (18), consider a convex interpolating function with continuous first derivative. By virtue of the mean value theorem there exists a point $0 < x_0 < \Delta$ such that

$$h'(x_0) = h/\Delta$$

Since by convexity $h'(x)$ is monotonically nondecreasing in $[0, \Delta]$, $d = h'(\Delta) \ge h/\Delta$. This establishes (18). Finally, (17*b*) follows directly from (18).

Having established the necessary and sufficient conditions in terms of the data (f_1, f_2, f_1', f_2') for the existence of a convex interpolating function, we proceed to present a class of convex interpolating functions that can always fit the given data.

Proposition 3. Let (f_1, f_2, f_1', f_2') satisfy (17). The function

$$g_1(x) = f_1 + (x - x_1) f_1' + a_1(x - x_1)^{b_1} \tag{19}$$

where

$$a_1 = \frac{f_2 - f_1 - \Delta f_1'}{\Delta^{b_1}} \tag{20}$$

$$b_1 = \frac{\Delta(f_2' - f_1')}{f_2 - f_1 - \Delta f_1'} \tag{21}$$

is convex in the interval $[x_1, x_2]$, $x_2 = x_1 + \Delta$, and is such that at the ends of the interval

$$g_1(x_1) = f_1 \qquad g_1'(x_1) = f_1' \tag{22a}$$

$$g_1(x_2) = f_2 \qquad g_1'(x_2) = f_2' \tag{22b}$$

Proof. Note that (17) implies that $a_1 \ge 0$ and $b_1 \ge 1$. Then $g_1(x)$ is convex in the interval $[x_1, x_1 + \Delta]$ because

$$g_1''(x) = a_1 b_1 (b_1 - 1)(x - x_1)^{b_1 - 2} \tag{23}$$

is a nonnegative function. Conditions (22*a*) and (22*b*) can be verified by direct substitution.

Remark. Note that condition (17*b*) can be written as

$$1 \le \frac{\Delta(f_2' - f_1')}{f_2 - f_1 - \Delta f_1'} = b_1 \tag{24}$$

By comparing conditions (17) and (5) it is interesting to observe how restrictive (5), which guarantees the convexity of the cubic Hermite polynomials, is. On the other hand, the above class of functions $g_1(x)$ can always be used to fit admissible data, that is, data (f_1, f_2, f_1', f_2') originating from a convex function. However, in case $b_1 < 2$, the function $g_1(x)$ has unbounded second derivative at $x = x_1$, as follows from (23). Since evaluation of the second derivative of the interpolating function at the end points of the interval $[x_1, x_2]$ is needed in the equations of gradient dynamic programming [see *Foufoula-Georgiou and Kitanidis*, 1988], we construct below a different convex interpolant which can be used in case $b_1 < 2$.

Proposition 4. Let (f_1, f_2, f_1', f_2') satisfy (17). The function

$$g_2(x) = f_2 + (x_2 - x) f_2' + a_2(x_2 - x)^{b_2} \tag{25}$$

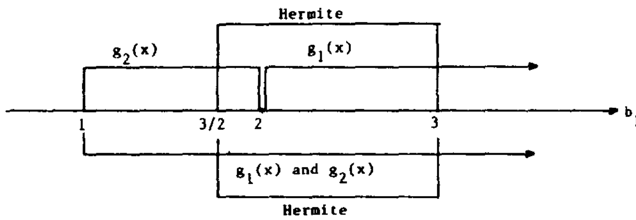

Fig. 4. Regions of convexity and bounded second derivatives (marked above the b_1 axis) and regions of convexity (marked below the b_1 axis) for the Hermite interpolating polynomial and the proposed functions $g_1(x)$ and $g_2(x)$.

where

$$a_2 = \frac{f_1 - f_2 + \Delta f_2'}{\Delta^{b_2}} \quad (26)$$

$$b_2 = \frac{\Delta(f_2' - f_1')}{f_1 - f_2 + \Delta f_2'} \quad (27)$$

is convex in the interval $[x_1, x_2 = x_1 + \Delta]$ and is such that at the ends of the interval

$$g_2(x_1) = f_1 \qquad g_2'(x1) = f_1' \quad (28a)$$

$$g_2(x_2) = f_2 \qquad g_2'(x_2) = f_2' \quad (28b)$$

Proof. In this case, (17) implies that $a_2 \geq 0$ and $b_2 \geq 1$. Then in the interval $[x_2 - \Delta, x_2]$ the second derivative

$$g_2''(x) = a_2 b_2(b_2 - 1)(x_2 - x)^{b_2 - 2} \quad (29)$$

is a nonnegative function.

We now show that $g_2(x)$ has bounded second derivative whenever $g_1(x)$ fails and vice versa.

Lemma. Define

$$h_{12} := f_2 - f_1 - \Delta f_1' \quad (30a)$$

$$h_{21} := f_1 - f_2 + \Delta f_2' \quad (30b)$$

Then,

$$1 \leq b_1 \leq 2 \Leftrightarrow 2 \leq b_2 \leq \infty \Leftrightarrow 0 \leq h_{12} \leq h_{21} \leq \infty$$

Similarly,

$$1 \leq b_2 \leq 2 \Leftrightarrow 2 \leq b_1 \leq \infty \Leftrightarrow 0 \leq h_{21} \leq h_{12} \leq \infty$$

Proof. The proof follows from direct algebraic manipulations.

Remark. From the above lemma it is observed that if $b_1 \leq 2$ (in which case $g_1''(x_1)$ is unbounded), $b_2 \geq 2$, and thus $g_2(x)$ has bounded second derivative at $x = x_1$. Similarly, if $b_2 \leq 2$ (in which case $g_2''(x)$ is unbounded), $b_1 \geq 2$, and thus $g_1(x)$ has bounded second derivative at $x = x_2$. Therefore depending on the value of b_1, one can choose between Hermite interpolating polynomials and one of the proposed functions $g_1(x)$ or $g_2(x)$ according to Figure 4 such that the convexity requirements are satisfied and the interpolating function has bounded second derivatives. In intervals where more than one interpolant satisfies all requirements one might choose the function with the smaller second derivative.

In optimization problems, when the cost-to-go function at a stage is either "too smooth" or "fast changing," the Hermite polynomials may lead to nonconvex interpolation (as explained in proposition 1). This is especially true when input or state constraints introduce discontinuity in the derivative of the cost-to-go function at the various stages. In such cases, problems might arise with the convergence of Newton-type optimization algorithms. In the context of one-dimensional optimal control problems these difficulties can be alleviated by the use of the convex interpolating functions proposed in propositions 3 and 4.

5. CONCLUDING REMARKS

Local approximations of functions based on values and derivatives at the nodes of a discretized grid are often used in solving problems numerically for which analytical solutions do not exist. In discrete dynamic programming, the use of continuously differentiable interpolating functions such as polynomials for the cost-to-go function alleviates the curse of dimensionality by reducing the number of discretization nodes per state, while obtaining high-accuracy solutions. It also permits the use of Newton-type schemes for the single-stage optimization problem (see, for example, *Kitanidis and Foufoula-Georgiou* [1987] and *Foufoula-Georgiou and Kitanidis* [1988]).

Hermite interpolating polynomials have found extensive application in finite element methods [e.g., *Zienkiewicz*, 1971] for the numerical solution of partial differential equations arising in many engineering and scientific problems. Few examples of the vast spectrum of finite element applications include potential flow problems [e.g., *Frind*, 1977], boundary value problems [e.g., *Villadsen and Stewart*, 1967], contaminant transport [e.g., *van Genuchten et al.*, 1977], and stress analysis [e.g., *Anderson et al.*, 1968], to mention only a few of the applications and related references. However, the advantage of using Hermite interpolation both in the context of discrete optimization and in the context of numerically solving partial differential equations will be fully realized when these polynomials result on a convex interpolation of convex functions.

In this paper we address the problem of interpolating nodal values and derivatives of a convex cost-to-go function with a convex interpolant so that computational difficulties due to approximation-induced nonconvexity are avoided and an efficient convergence to global instead of local optima is guaranteed. We study conditions of the Hermite interpolation functions to be convex and also provide a class of convex functions that can interpolate any set of admissable nodal values and derivatives.

Our analysis is for the one-dimensional case. The conditions of convexity of the multidimensional Hermite interpolation are fairly complicated, as is the study of the feasibility and construction of multidimensional convex interpolants. From the practical standpoint the usefulness of these methods is expected to be much greater in problems of large dimensionality. It is hoped that the present study will motivate research toward the development of a convex multidimensional interpolation theory.

Acknowledgments. This work was supported in part by the National Science Foundation under grant BSC-8957469. I would like to thank Tryphon Georgiou and one of the referees for their helpful input.

REFERENCES

Anderson, R. G., B. M. Irons, and O. C. Zienkiewicz, Vibration and stability of plates using finite elements, *Int. J. Solids Struct.*, *4*, 1031–1055, 1968.

Bellman, R. E., *Dynamic Programming*, Princeton University Press, Princeton, N. J., 1957.

Bellman, R. E., and S. E. Dreyfus, *Applied Dynamic Programming*, Princeton University Press, Princeton, N. J., 1962.

Birnbaum, I., and L. Lapidus, Studies in approximation methods, I, Splines and control via discrete dynamic programming, *Chem. Eng. Sci.*, *33*, 415–426, 1978.

Daniel, J. W., Splines and efficiency in dynamic programming, *J. Math. Anal. Appl.*, *54*, 402–407, 1976.

Foufoula-Georgiou, E., and P. K. Kitanidis, Gradient dynamic programming for stochastic optimal control of multidimensional water resources systems, *Water Resour. Res.*, *24*(8), 1345–1359, 1988.

Frind, E. O., An isoparametric Hermitian element for the solution of field problems, *Int. J. Numer. Methods Eng.*, *11*(6), 945–962, 1977.

Jacobson, D., and D. Mayne, *Differential Dynamic Programming*, Academic, San Diego, Calif., 1970.

Kitanidis, P. K., and E. Foufoula-Georgiou, Error analysis of conventional discrete and gradient dynamic programming, *Water Resour. Res.*, *23*(5), 845–858, 1987.

Kitanidis, P. K., and E. Foufoula-Georgiou, Reply, *Water Resour. Res.*, *24*(12), 2105–2106, 1988.

Luenberger, D. G., *Introduction to Linear and Nonlinear Programming*, 2nd ed., Addison-Wesley, Reading, Mass., 1984.

Mayne, D., A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems, *Int. J. Control*, *3*(1), 85–95, 1966.

Schultz, M. H., *Spline Analysis*, 156 pp., Prentice-Hall, Englewood Cliffs, N. J., 1973.

van Genuchten, M. T., G. F. Pinder, and E. O. Frind, Simulation of two-dimensional contaminant transport with isoparametric Hermitian finite elements, *Water Resour. Res.*, *13*(10), 451–458, 1977.

Villadsen, J. V., and W. E. Stewart, Solutions of boundary value problems by orthogonal collocation, *Chem. Eng. Sci.*, *22*, 1485–1501, 1967.

Yakowitz, S., Dynamic programming applications in water resources, *Water Resour. Res.*, *18*(4), 673–696, 1982.

Zienkiewicz, O. C., *The Finite Element Method in Engineering Science*, McGraw-Hill, New York, 1971.

E. Foufoula-Georgiou, St. Anthony Falls Hydraulic Laboratory, Department of Civil and Mineral Engineering, University of Minnesota, Minneapolis, MN 55414.

(Received May 16, 1990;
revised August 29, 1990;
accepted September 13, 1990.)