2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Symbolic-Numeric Computation of Integrals in Successive Galerkin Approximation of Hamilton-Jacobi-Bellman Equation

Tomoyuki Iori

Abstract— **This paper proposes an efficient symbolic-numeric method to compute integrals in the successive Galerkin approximation (SGA) of the Hamilton-Jacobi-Bellman (HJB) equation. By approximating its solution with a linear combination of basis functions, the HJB equation is reduced to a linear equation comprising integrals that include the basis functions. By choosing the Hermite polynomials as the basis functions, their recursive structure is inherited by the integrals. The recurrence relations of the integrals are computed using the symbolic computation and Mellin transform of differential operators. The integrals can then be computed using recursive substitutions, which are more accurate and require less computational cost than numerical integrations. A numerical example is provided to demonstrate the efficiency of the proposed method compared to other numerical integration methods.**

I. Introduction

For general nonlinear systems, an optimal feedback controller with an infinite horizon is obtained by a solution to the Hamilton-Jacobi-Bellman (HJB) equation [1]. For linear systems with a quadratic cost, the HJB equation can be reduced to a matrix equation called the Riccati equation, which can be efficiently solved. However, for nonlinear cases, the HJB equation is formulated as a nonlinear partial differential equation (PDE), which is difficult to solve analytically. Hence, numerous numerical approximation techniques have been proposed for solving the HJB equation.

One possibility is to approximate a solution to the HJB equation using the Taylor-series approximation [2]–[4]. By substituting a truncated Taylor-series expansion to the HJB equation, a finite number of algebraic equations for its coefficients can be obtained. A recursive closed form of the coefficients is proposed in [4] under several conditions.

The stable manifold method [5] is one of the most promising techniques. In this method, a sequence of trajectories that converges to a trajectory on the stable manifold is iteratively computed. After a sufficient number of iterations, the trajectory can be viewed as an approximation of the Hamiltonian flow on the stable manifold, and thus, it approximately provides the optimal feedback controller.

Another approach is to iteratively solve a linear PDE called the generalized HJB (GHJB) equation; this is termed a successive approximation approach (SAA) [6]–[8]. One of the most popular SAAs is the successive Galerkin approximation (SGA), in which the GHJB equation is reduced to a linear equation by approximating the solution with a linear combination of basis functions. The solutions to the SGA were proven to converge to a solution to the HJB equation as the number of basis functions and iterations tend to infinity [6].

One of the issues in the SGA is the computation of integrals associated with the inner product of functions. Although the integrals can be computed offline, their computational costs may not be handled with a reasonable amount of computational resources. Moreover, as the number of basis functions increases, the nonlinearity of the integrands usually increases, rendering integration more difficult and inaccurate. Therefore, an efficient method for computing the integrals in SGA has to be developed.

In recent years, symbolic computation of differential operators has been intensively studied [9]–[12]. Its applications can be found in statistics [13], [14], moment problems [15], and filtering problems [16]. Furthermore, many related algorithms have also been implemented in the computer algebra systems (CASs), such as `Singular` [17], `Macaulay2` [18], and `Risa/Asir` [19]. In the symbolic computation of differential operators, a specific type of PDEs and their solutions called holonomic functions are the primary focus. In particular, a set of PDEs satisfied by the integral of a holonomic function can be computed from certain PDEs satisfied by the integrand using symbolic computation. This situation is significantly different from integrating nonlinear functions directly because the integral of a nonlinear function can rarely be computed symbolically. Moreover, differential operators can be converted into difference operators via the Mellin transform [10], [12], which is related to the multivariate z-transform. This allows us to treat recurrence relations using the symbolic computation of differential operators.

In this paper, we propose a method to compute the recurrence relations satisfied by the integrals in the SGA. A solution to the HJB equation is approximated by a linear combination of Hermite polynomials, whose z-transforms are holonomic. In addition, the plant model is assumed to consist of holonomic functions. These problem settings allow us to compute a set of recurrence relations satisfied by the integrals. Once the recurrence relations are computed, any number of integrals can be readily computed by recursive substitutions if the first several integrals are computed numerically. Moreover, in contrast to the earlier methods [7], [8], the proposed method exactly considers the nonlinearities of the plant model in the computation of the integrals.

Notations: Let $\mathbf{N}$ be the set of positive integers and $\mathbf{R}$ be the field of real numbers. For discrete variables

This work was partly supported by JSPS KAKENHI Grant Numbers JP21K21285 and 22K17855.

T. Iori is with the Department of Information and Physical Sciences, Graduate School of Information Science and Technology, Osaka University, 1-5 Yamadaoka, Suita, Osaka 565–0871, Japan `t-iori@ist.osaka-u.ac.jp`

$i = (i_1, \ldots, i_p) \in \mathbf{N}^p$ and continuous variables $x = (x_1, \ldots, x_n) \in \mathbf{R}^n$, a function F of i and x is denoted by $F(i; x)$. If $n = 0$ or $p = 0$, $F(i; x)$ is simply written as $F(i)$ or $F(x)$, respectively. The symbols $\mathsf{S}_{i_j}, \mathsf{S}_{i_j}^{-1}$ $(j \in \{1, \ldots, p\})$, and ∂_{x_j} $(j \in \{1, \ldots, n\})$ denote the shift operators for i_j and differential operator for x_j, respectively. That is, $\mathsf{S}_{i_j} \bullet F(i; x) = F(i_1, \ldots, i_j + 1, \ldots, i_p; x)$, $\mathsf{S}_{i_j}^{-1} \bullet F(i; x) = F(i_1, \ldots, i_j - 1, \ldots, i_p; x)$, and $\partial_{x_j} \bullet F(i; x) = \partial F / \partial x_j(i; x)$, where $\bullet$ denotes the action of a differential or difference operator on a function. Let $\mathbf{R}[x]$ be the set of all polynomials in $x_1, \ldots, x_n$ with coefficients in $\mathbf{R}$. The symbol $\mathbf{R}[x]\langle \partial_x \rangle$ denotes the noncommutative ring of differential operators with coefficients in $\mathbf{R}[x]$. Similarly, $\mathbf{R}[i]\langle \mathsf{S}_i \rangle$ denotes the noncommutative ring of difference operators with coefficients in $\mathbf{R}[i]$. The symbols $\mathbf{R}[x]\langle \partial_x \rangle$ and $\mathbf{R}[i]\langle \mathsf{S}_i \rangle$ are also denoted by $\mathcal{D}_n$ and O_p, respectively, if the indeterminates x and i are clearly specified according to the context. If $\mathcal{P} \bullet F = 0$ for $\mathcal{P} \in \mathcal{D}_n$, $\mathcal{P}$ is said to *annihilate* a function F and F is a *solution* to $\mathcal{P}$. If $\mathsf{E} \bullet F = 0$ for $\mathsf{E} \in \mathsf{O}_p$, E is said to *annihilate* F and F is a *solution* to E.

II. Problem Setting

Consider the optimal control problem of a nonlinear control-affine system:

$$\dot{x} = f(x) + g(x)u$$

with a performance index defined as follows:

$$J(x, u) \coloneqq \int_0^\infty q(x(t)) + \|u(t)\|_R^2 dt,$$

where $x \in \mathbf{R}^n$ and $u \in \mathbf{R}^m$ denote the state and input of the system, respectively. The nonlinear functions $f \colon \mathbf{R}^n \to \mathbf{R}^n$, $g \colon \mathbf{R}^n \to \mathbf{R}^{n \times m}$, and $q \colon \mathbf{R}^n \to \mathbf{R}$ are assumed to consist of holonomic functions, which will be defined in Section III. It is also assumed that $q(x) \geq 0$ $(x \in \mathbf{R}^n)$ and $R \in \mathbf{R}^{m \times m}$ is positive definite. The value function $V^*(x_0) \coloneqq \inf_u J$ satisfies a nonlinear PDE called the HJB equation:

$$\mathrm{HJB}(V^*) \coloneqq \frac{\partial V^*}{\partial x} f + q - \frac{1}{4} \frac{\partial V^*}{\partial x} g R^{-1} g^\top \frac{\partial V^*}{\partial x}^\top = 0,$$

and the optimal feedback law $u^*(x)$ is expressed by

$$u^*(x) = -\frac{1}{2} R^{-1} g^\top(x) \frac{\partial V^*}{\partial x}^\top (x).$$

In the SGA, a solution to the HJB equation is approximated using a finite number of basis functions $\Phi(x) \coloneqq [\phi_1(x) \ \cdots \ \phi_N(x)]^\top$ as follows:

$$V(x) \coloneqq \sum_{i=1}^N v_i \phi_i(x) = \Phi^\top(x) v \quad (v_i \in \mathbf{R}). \tag{1}$$

The coefficient vector $v \coloneqq [v_1 \ \cdots \ v_N]^\top$ is determined by iteratively solving

$$\left\langle \mathrm{GHJB}(V^{(l)}, u^{(l)}), \phi_k \right\rangle = 0 \quad (k = 1, \ldots, N), \tag{2}$$

$$u^{(l+1)}(x) = -\frac{1}{2} R^{-1} g^\top(x) \frac{\partial \Phi}{\partial x}^\top (x) v^{(l)}, \tag{3}$$

where $V^{(l)} = \Phi(x)^\top v^{(l)}$, $\mathrm{GHJB}(V, u) = \frac{\partial V}{\partial x}(f + gu) + q + \|u\|_R^2$, and $\langle f(x), g(x) \rangle \coloneqq \int_{\mathbf{R}^n} w(x) f(x) g(x) dx$ is an inner product of functions with a weighting function $w(x) > 0$ $(x \in \mathbf{R}^n)$. For a particular input $u(x)$, the linear PDE $\mathrm{GHJB}(V, u) = 0$ for $V(x)$ is called the GHJB equation [6].

By substituting (1) and (3) into (2), it can be reduced to a set of linear equations for $v^{(l)}$:

$$\mathrm{LE}_k(v^{(l)}, v^{(l-1)}) \coloneqq v^{(l)\top} \left(d_k - \frac{1}{2} Q_k v^{(l-1)} \right) + c(k) + \frac{1}{4} \left\| v^{(l-1)} \right\|_{Q_k}^2 = 0 \quad (k = 1, \ldots, N), \tag{4}$$

where $Q_k = \{a(i, j, k)\}$ is an $N \times N$ symmetric matrix, $d_k = \{b(i, k)\}$ is an N-dimensional vector, and $a(i, j, k)$, $b(i, k)$, and $c(k)$ are defined as follows:

$$\begin{aligned} a(i, j, k) &\coloneqq \left\langle \frac{\partial \phi_i}{\partial x} g R^{-1} g^\top \frac{\partial \phi_j}{\partial x}^\top, \phi_k \right\rangle, \\ b(i, k) &\coloneqq \left\langle \frac{\partial \phi_i}{\partial x} f, \phi_k \right\rangle, \quad c(k) \coloneqq \langle q(x), \phi_k(x) \rangle . \end{aligned} \tag{5}$$

In the followings, for a given $N \in \mathbf{N}$, A_N, B_N, and C_N denote the sets of integrals required to define (4):

$$\begin{aligned} A_N &\coloneqq \{a(i, j, k) \mid i, j, k \in \{1, \ldots, N\}\}, \\ B_N &\coloneqq \{b(i, k) \mid i, k \in \{1, \ldots, N\}\}, \\ C_N &\coloneqq \{c(k) \mid k \in \{1, \ldots, N\}\}. \end{aligned}$$

Once the integrals A_N, B_N, and C_N are computed, the SGA procedure can be performed by solving the linear equation (4) iteratively, which is summarized in Algorithm 1.

The elements of A_N, B_N, and C_N are defined as integrals of nonlinear functions, which are difficult to compute analytically. However, if the basis functions have some recurrence relations, such as those of orthogonal polynomials, the elements of A_N, B_N, and C_N may have similar recurrence relations. Such recurrence relations can be computed using the symbolic computation of differential operators, which is briefly introduced in the next section.

III. Holonomic Functions and Ideals

This section introduces definitions and lemmas related to the symbolic computation of differential operators with reference to [10], [11]. In this section, $x = [x_1 \ \cdots \ x_n]^\top \in \mathbf{R}^n$, $s = [s_1 \ \cdots \ s_p]^\top \in \mathbf{R}^p$, and $i = [i_1 \ \cdots \ i_p]^\top \in \mathbf{N}^p$. Let $\mathcal{D}_n$ and $\mathcal{D}_{n+p}$ denote $\mathbf{R}[x]\langle \partial_x \rangle$ and $\mathbf{R}[x, s]\langle \partial_x, \partial_s \rangle$, respectively. Furthermore, let $\mathcal{D}_n\langle i, \mathsf{S}_i, \mathsf{S}_i^{-1} \rangle$ denote $\mathcal{D}_n$-algebra generated by i, $\mathsf{S}_i = [\mathsf{S}_{i_1} \ \cdots \ \mathsf{S}_{i_p}]^\top$, and $\mathsf{S}_i^{-1} = [\mathsf{S}_{i_1}^{-1} \ \cdots \ \mathsf{S}_{i_p}^{-1}]^\top$.

Algorithm 1 Successive Galerkin approximation

Input: Initial guess $v^{(0)}$, sets of integrals A_N, B_N, and C_N, error tolerance ϵ, max iteration $l_{\max}$
Output: Approximation of value function $V^{(l)}(x)$

```
l ← 0
while sqrt(Σ_{k=1}^N LE_k^2(v^(l), v^(l))) > ε and l < l_max do
    Solve linear equations (4) and compute v^(l+1)
    l ← l + 1
return V^(l) = Φ^⊤(x) v^(l)
```

Consider a set of PDEs for an unknown function $f(x)$.

$$\mathcal{P}_1 \bullet f = \cdots \mathcal{P}_d \bullet f = 0, \tag{6}$$

where $\mathcal{P}_1, \ldots, \mathcal{P}_d \in \mathcal{D}_n$. For any solution f of (6) and differential operator $Q \in \mathcal{D}_n$, $Q\mathcal{P}_j \bullet f = Q \bullet 0 = 0$ holds. This leads us to consider a set of differential operators $I \coloneqq \{Q_1\mathcal{P}_1 + \cdots Q_d\mathcal{P}_d \in \mathcal{D}_n \mid Q_1, \ldots, Q_d \in \mathcal{D}_n\} \subset \mathcal{D}_n$, which is called the left ideal generated by $\mathcal{P}_1, \ldots, \mathcal{P}_d$. In this paper, the adjective "left" is omitted because all ideals in this paper are left ideals.

Definition 1 (Holonomic ideal of $\mathcal{D}_n$): An ideal $I \subset \mathcal{D}_n$ is holonomic if the quotient $\mathcal{D}_n/I$, which can be viewed as a left $\mathcal{D}_n$-module, has dimension n.

For a definition of the dimensions of the left $\mathcal{D}_n$-modules, see [9], [11], [20]. Holonomic ideals play an important role in the symbolic computation of differential operators, especially to solve PDEs [9].

The solutions to holonomic ideals are called holonomic functions and include most nonlinear functions that appear in systems and control theory [16], [21].

Definition 2 (Holonomic function): An analytic function $f(x)$ is said to be holonomic if f is annihilated by a holonomic ideal $I \subset \mathcal{D}_n$; that is, $\mathcal{P} \bullet f = 0$ for all $\mathcal{P} \in I$.

Holonomic functions are closed under multiplication and integration; that is, the products and integrals of holonomic functions are holonomic [10].

Lemma 1: For two holonomic functions $f(x)$ and $g(x)$, their product $(f \cdot g)(x)$ is holonomic.

Lemma 2: Suppose a holonomic function $f(x)$ is *rapidly decreasing* with respect to x_1; that is, $\lim_{x_1 \to \infty} x_1^i \partial_{x_1}^j f(x) = 0$ for any nonnegative integers i and j. Then, the integral $\int_{-\infty}^{\infty} f(x)dx_1$ is holonomic as a function of $x_2, \ldots, x_n$.

The recurrence relations for discrete variables can be converted into PDEs and manipulated by the symbolic computation of differential operators through the Mellin transform.

Definition 3 (Mellin transform between ∂ and S): The mappings $\mu\colon \mathcal{D}_{n+p} \to \mathcal{D}_n\langle i, \mathsf{S}_i, \mathsf{S}_i^{-1}\rangle$:

$$\mu(s_j) = \mathsf{S}_{i_j}, \quad \mu(\partial_{s_j}) = -i_j \mathsf{S}_{i_j}^{-1} \tag{7}$$

and $\hat{\mu}\colon \mathcal{D}_n\langle i, \mathsf{S}_i\rangle \to \mathcal{D}_{n+p}$:

$$\hat{\mu}(i_j) = -\partial_{s_j} s_j = -s_j\partial_{s_j} - 1, \quad \hat{\mu}(\mathsf{S}_{i_j}) = s_j \tag{8}$$

are called the *Mellin transform* and *inverse Mellin transform*, respectively.

The Mellin transform is related to the multivariate z-transform $\mathcal{Z}$ as $\hat{\mu}(i) \bullet \mathcal{Z}[f] = \mathcal{Z}[if]$, $\hat{\mu}(\mathsf{S}_i) \bullet \mathcal{Z}[f] = \mathcal{Z}[\mathsf{S}_i f]$, where $\mathcal{Z}[f](s,x)$ is defined for $f(i;x)$ as

$$\mathcal{Z}[f](s,x) \coloneqq \sum_{i_1=0}^{\infty} \cdots \sum_{i_p=0}^{\infty} f(i;x) s_1^{-i_1-1} \cdots s_p^{-i_p-1}.$$

IV. Recurrence Relations of Integrals

In this section, a symbolic computation method is proposed to derive the recurrence relations of the integrals $a(i,j,k)$, $b(i,k)$, and $c(k)$ defined in (5). This is achieved by computing differential operators that annihilate the z-transforms of the integrals and converting them into recurrence relations using the inverse Mellin transform.

Hereafter, we suppose $n = 1$ for simplicity and that the i-th basis function $\phi_i(x)$ in (1) is the Hermite polynomial $H(i;x)$ of degree i. For multidimensional cases, each basis function can be defined as the product of the Hermite polynomials $H(i_1;x_1)H(i_2;x_2)\cdots H(i_n;x_n)$. The following discussion can be applied with appropriate modifications, which will be part of future work.

As each integral in (5) is linear in each basis function with subscripts i, j, or k, the z-transforms of (5) can be obtained as follows:

$$\mathcal{Z}[a](s_1,s_2,s_3) = \int_{-\infty}^{\infty} \frac{w(x)g^2(x)\mathscr{H}_x(s_1,x)\mathscr{H}_x(s_2,x)\mathscr{H}(s_3,x)}{R}dx, \tag{9}$$

$$\mathcal{Z}[b](s_1,s_3) = \int_{-\infty}^{\infty} w(x)f(x)\mathscr{H}_x(s_1,x)\mathscr{H}(s_3,x)dx, \tag{10}$$

$$\mathcal{Z}[c](s_3) = \int_{-\infty}^{\infty} w(x)q(x)\mathscr{H}(s_3,x)dx, \tag{11}$$

where $\mathscr{H} \coloneqq \mathcal{Z}[H]$ and $\mathscr{H}_x \coloneqq \mathcal{Z}[\partial_x H]$. If each integrand in (9)–(11) is holonomic and rapidly decreasing with respect to x, then each z-transform is holonomic from Lemma 2. In this case, we can compute differential operators that annihilate the z-transforms (9)–(11) using symbolic computation (for detailed algorithms, see [9]–[11]), which yields the recurrence relations of the integrals $a(i,j,k)$, $b(i,k)$ and $c(k)$ via the Mellin transform (7).

The factors of the integrands f, g, and q are assumed to be holonomic, and the weighting function w can be designed such that it is holonomic and all the integrands are rapidly decreasing with respect to x. Hence, from Lemma 1, the z-transforms (9)–(11) as well as their integrands are holonomic if $\mathscr{H}$ and $\mathscr{H}_x$ are holonomic.

First, to show that $\mathcal{Z}[H]$ is holonomic, we consider the difference-differential equations satisfied by $H(i;x)$:

$$(\partial_x^2 - 2x\partial_x + 2i) \bullet H(i;x) = 0, \tag{12}$$

$$(\mathsf{S}_i^2 - 2x\mathsf{S}_i + 2(i+1)) \bullet H(i;x) = 0. \tag{13}$$

The inverse Mellin transform (8) yields the following PDEs:

$$(\partial_x^2 - 2x\partial_x - 2s\partial_s - 2) \bullet \mathscr{H}(s,x) = 0, \tag{14}$$

$$(s^2 - 2xs - 2s\partial_s) \bullet \mathscr{H}(s,x) = 0. \tag{15}$$

It can be readily verified using a CAS that (14) and (15) generate a holonomic ideal. Hence, from Definition 2, $\mathscr{H}(s,x)$ is holonomic.

Lemma 3: The z-transform $\mathscr{H} = \mathcal{Z}[H]$ is holonomic.

Next, to show that $\mathscr{H}_x$ is holonomic, we use the equality $\partial_x H(i;x) = 2iH(i-1;x)$ satisfied by $H(i;x)$. Using this equality combined with (12) and (13), we first obtain difference-differential equations satisfied by $\partial_x H$ as follows.

From (12) with $i = i - 1$, we obtain

$$\begin{aligned} 0 = 2i \cdot 0 &= 2i \cdot \left\{(\partial_x^2 - 2x\partial_x + 2(i-1)) \bullet H(i-1;x)\right\} \\ &= \left\{\partial_x^2 - 2x\partial_x + 2(i-1)\right\} \bullet 2iH(i-1;x) \\ &= \left\{\partial_x^2 - 2x\partial_x + 2(i-1)\right\} \bullet \partial_x H(i;x), \end{aligned} \tag{16}$$

where we use the fact that $2i$ and $\partial_x^2 - 2x\partial_x + 2(i-1)$ commute. On the other hand, $(\mathsf{S}_i^2 - 2x\mathsf{S}_i + 2i) \bullet H(i-1;x) = 0$ holds

from (13) with $i = i - 1$. By multiplying both sides by $2(i+1)(i+2)$ from the left, we obtain

$$\begin{aligned} 0 &= 2(i+1)(i+2)(\mathsf{S}_i^2 - 2x\mathsf{S}_i + 2i) \bullet H(i-1;x) \\ &= \left\{(i+1)\mathsf{S}_i^2 - 2x(i+2)\mathsf{S}_i + 2(i+1)(i+2)\right\} \\ &\qquad \bullet \partial_x H(i;x), \end{aligned} \tag{17}$$

where equalities $(i+2)\mathsf{S}_i^2 = \mathsf{S}_i^2 i$ and $(i+1)\mathsf{S}_i = \mathsf{S}_i i$ are used to derive the second line. Eventually, we obtain the differential operators that annihilate $\mathscr{H}_x(s,x)$ as the inverse Mellin transforms of (16) and (17); that is,

$$\hat{\mu}(\partial_x^2 - 2x\partial_x + 2(i-1)) = \partial_x^2 - 2x\partial_x - 2s\partial_s - 4, \tag{18}$$

$$\begin{aligned} &\hat{\mu}((i+1)\mathsf{S}_i^2 - 2x(i+2)\mathsf{S}_i + 2(i+1)(i+2)) \\ &= 2s^2\partial_s^2 + (2s^2x - s^3)\partial_s - 2s^2. \end{aligned} \tag{19}$$

We can confirm that the differential operators in (18) and (19) generate a holonomic ideal using a CAS and prove the following lemma:

Lemma 4: The z-transform $\mathscr{H}_x = \mathscr{Z}[\partial_x H]$ is holonomic.

As shown above, all factors of the integrands in (9)–(11), as well as the integrals, are holnomic. Consequently, the recurrence relations satisfied by $a(i,j,k)$, $b(i,k)$, and $c(k)$ can be obtained using the Mellin transform. The entire procedure is summarized in Algorithm 2.

V. Recursive Computation of Integrals

Once the recurrence relations are obtained from the outputs G_a, G_b, and G_c of Algorithm 2, the integrals can be computed by recursive substitution. As each recurrence relation may include a monomial of shift operators with a degree exceeding one, we must first compute a few integrals numerically. Moreover, for a certain set of indices, some coefficients of the shift operators in a difference operator may vanish, which indicates that the difference operator fails to provide a recurrence relation at the set of indices, as shown in the following example.

Example 1: Consider the following difference operator in $\mathbf{R}[i,j,k]\langle \mathsf{S}_i, \mathsf{S}_j, \mathsf{S}_k\rangle$:

$$\begin{aligned} &(-j+i+k+1)(-k-3-j+i)\mathsf{S}_k^2 \\ &\qquad + 4(k+2)(k+1)(i+j-k-1), \end{aligned} \tag{20}$$

which provides a recurrence relation for $a(i,j,k)$ as follows:

$$\begin{aligned} &(-j+i+k+1)(-k-3-j+i)a(i,j,k+2) \\ &\qquad + 4(k+2)(k+1)(i+j-k-1)a(i,j,k) = 0. \end{aligned} \tag{21}$$

For a generic triplet (i,j,k) such that the coefficients of a do not vanish, we can compute $a(i,j,k+2)$ from $a(i,j,k)$ using (21). As $a(i,j,k+1)$ cannot be computed directly from $a(i,j,k)$, we must provide two values $a(i,j,1)$ and $a(i,j,2)$ to compute $a(i,j,k)$ for $k \geq 3$. Moreover, for the triplet (i,j,k) such that the coefficient of $a(i,j,k+2)$ in (21) vanishes, (20) fails to provide any recurrence relation. In this case, we must compute $a(i,j,k+2)$ from other recurrence relations, or if not possible, compute it numerically.

To summarize the numerical part of the proposed method briefly, we introduce the following terminology. Let $f(i)$ be a function that maps $i = (i_1, \ldots, i_n) \in \mathbf{N}^p$ to a value $f(i) \in \mathbf{R}$, g be a difference operator in $\mathbf{R}[i]\langle \mathsf{S}_i\rangle$, and $\bar{F} \subset \mathbf{R}$ be a set of known values of $f(i)$. For $i^* \in \mathbf{N}^p$ at which $f(i^*)$ is unknown, we say that $f(i^*)$ is *computable by* g *from* $\bar{F}$ if there exists $\bar{i}$ such that the recurrence relation based on g at $i = \bar{i}$ includes only $f(i^*)$ and the elements of $\bar{F}$. Using the terminology, the numerical part of the proposed algorithm is summarized in Algorithm 3.

Example 2: Consider the difference operator (20) and $\bar{A} \coloneqq \{a(2,1,1), a(1,3,1)\}$. As (20) includes S_k^2, $a(2,1,3)$ and $a(1,3,3)$ are the candidates of unknown values computable from $\bar{A}$. Indeed, $a(2,1,3)$ is computable by (20) from $\bar{A}$ because (21) at $(\bar{i},\bar{j},\bar{k}) = (2,1,1)$ is $-9a(2,1,3) + 24a(2,1,1) = 0$. However, $a(1,3,3)$ is not because (21) is $48a(1,3,1) = 0$ at $(\bar{i},\bar{j},\bar{k}) = (1,3,1)$ and provides no information on the unknown value $a(1,3,3)$. Even in the latter case, $48a(1,3,1) = 0$ indicates that $a(1,3,1)$ is exactly zero, which is difficult to verify through numerical integration when indices i,j,k are large.

Algorithm 2 Derivation of recurrence relations satisfied by $a(i,j,k)$, $b(i,k)$, and $c(k)$

Input: Sets of differential operators $\mathcal{G}_f$, $\mathcal{G}_g$, $\mathcal{G}_q$, $\mathcal{G}_w \subset \mathcal{D}_1$ that annihilate $f(x)$, $g(x)$, $q(x)$, $w(x)$, respectively, and $\mathcal{G}_H$, $\mathcal{G}_{\partial H} \subset \mathcal{D}_2$ that annihilate $\mathscr{H}(s,x)$, $\mathscr{H}_x(s,x)$, respectively

Output: Set of difference operators $\mathsf{G}_a \subset \mathsf{O}_3$, $\mathsf{G}_b \subset \mathsf{O}_2$, and $\mathsf{G}_c \subset \mathsf{O}_1$ that annihilate $a(i,j,k)$, $b(i,k)$, and $c(k)$, respectively

1: Compute sets of differential operators $\mathcal{J}_a \subset \mathcal{D}_4$, $\mathcal{J}_b \subset \mathcal{D}_3$, and $\mathcal{J}_c \subset \mathcal{D}_2$ that annihilate the integrands of (9), (10), and (11), respectively, from inputs

2: Compute sets of differential operators $\mathcal{K}_a \subset \mathcal{D}_3$, $\mathcal{K}_b \subset \mathcal{D}_2$, and $\mathcal{K}_c \subset \mathcal{D}$ that annihilates $\mathscr{Z}[a]$, $\mathscr{Z}[b]$, and $\mathscr{Z}[c]$ from $\mathcal{J}_a$, $\mathcal{J}_b$, and $\mathcal{J}_c$, respectively

3: Compute G_a, G_b, and G_c from $\mathcal{K}_a$, $\mathcal{K}_b$, and $\mathcal{K}_c$, respectively, via Mellin transform (7)

VI. Numerical Example

This section presents a numerical example that demonstrates the efficiency of the proposed method. In the following demonstration, `Risa/Asir` and `Julia` were used to perform symbolic and numerical computations for the proposed method, respectively. In addition, `Chebfun` [22], a MATLAB package that provides an efficient numerical integration method, and `Maple` were used for comparison. All computations were performed on a PC(Intel(R) Core(TM) i9-10920X CPU @ 3.50GHz; RAM: 64 GB).

A. Problem setting

Consider a nonlinear optimal control problem for a nonlinear scalar system.

$$\dot{x} = \sin(x) + u(x), \quad J = \int_0^\infty \frac{1}{2}x^2(t) + \frac{1}{2}u^2(t)dt. \tag{22}$$

The nonlinear functions $\sin(x)$ and x^2 are holonomic and annihilated by differential operators $\partial_x^2 + 1$ and ∂_x^3, respectively.

To perform the SGA (Algorithm 1), the following integrals must be computed:

$$a(i,j,k) = \langle 2\partial_x H(i;x)\partial_x H(j;x), H(k;x)\rangle\,,$$

$$b(i,k) = \langle \partial_x H(i;x)\sin(x), H(k;x)\rangle\,,\ c(k) = \left\langle \frac{x^2}{2}, H(k;x)\right\rangle,$$

where the weighting function is set to $w(x) = \exp(-x^2)$ such that all integrands are rapidly decreasing.

B. Computation of integrals by proposed method

To obtain the recurrence relations satisfied by $a(i,j,k)$, $b(i,k)$, and $c(k)$, Algorithm 2 can be applied to the inputs $\mathcal{G}_f = \{\partial_x^2 + 1\}$, $\mathcal{G}_g = \{0\}$, $\mathcal{G}_q = \{\partial_x^3\}$, $\mathcal{G}_w = \{\partial_x + 2x\}$, $\mathcal{G}_H$ being the set of (14) and (15), and $\mathcal{G}_{\partial H}$ being the set of (18) and (19). For example, $\mathcal{J}_a \subset \mathbf{R}[x, s_1, s_2, s_3]\langle \partial_x, \partial_{s_1}, \partial_{s_2}, \partial_{s_3}\rangle$ is obtained as a set of 19 differential operators, including

$$\begin{aligned}&\partial_x + 2s_1\partial_{s_1}^2 + (2s_1x - s_1^2 + 2)\partial_{s_1}\\ &\quad + 2s_2\partial_{s_2}^2 + (2s_2x - s_2^2 + 2)\partial_{s_2} + s_3 - 2s_1 - 2s_2, \quad (23)\end{aligned}$$

where the remaining elements have been omitted owing to space limitations. It can be observed that (23) is invariant under the interchange of s_1 and s_2, which is consistent with the fact that the integrand of (9) is invariant under the interchange of s_1 and s_2. From the output of Algorithm 2, we obtain G_a and G_b as listed in Table I and $\mathsf{G}_c = \{\mathsf{S}_k^3\}$. Note that $c(k)$ in this numerical example can be computed analytically as $c(k) = 0$ except for $c(2) = \sqrt{\pi}$, which clearly satisfies $\mathsf{S}_k^3 \bullet c(k) = c(k+3) = 0$ for $k \in \mathbf{N}$.

From the recurrence relations obtained from G_a and G_b, the sets of integrals A_N and B_N are computed using Algorithm 3 up to degree $N = 14$. During the operation of Algorithm 3, certain integrals must be evaluated numerically. In this example, 10 integrals: $a(1,1,1) = a(1,1,2) = a(2,1,2) = a(1,2,2) = a(4,1,1) = a(4,1,2) = a(1,4,1) = a(1,4,2) = 0$, $a(2,1,1) = a(1,2,1) = 56.7$ for A_N and 11 integrals: $b(1,1) = 2.76$, $b(1,2) = b(2,1) = b(3,2) = b(4,1) = b(5,3) = 0$, $b(1,3) = -2.76$, $b(3,1) = 24.8$, $b(3,3) = 108$, $b(4,2) = 144$, $b(5,1) = -96.6$ for B_N were numerically evaluated, which is less than 1% of the total number of elements of A_{14} and B_{14}, that is, $14^3+14^2 = 2940$.

Algorithm 3 Recursive computation of A_N, B_N, and C_N

Input: Number of basis functions $N \in \mathbf{N}$ and sets of difference operators $\mathsf{G}_a \subset \mathsf{O}_3$, $\mathsf{G}_b \subset \mathsf{O}_2$, and $\mathsf{G}_c \subset \mathsf{O}_1$ that annihilate $a(i,j,k)$, $b(i,k)$, and $c(k)$, respectively
Output: Integrals A_N, B_N, and C_N

function CalcNewElem(f, i, G, $\bar{F}$)
  if There exists $\mathsf{g} \in \mathsf{G}$ such that $f(i)$ is computable by g from $\bar{F}$ **then**
    Compute $f(i)$ from g and $\bar{F}$
  else Evaluate $f(i)$ numerically
  Append $f(i)$ to $\bar{F}$
  return $\bar{F}$
Compute $a(1,1,1)$, $b(1,1)$, and $c(1)$ numerically
$i \leftarrow 1$, $j \leftarrow 1$, and $k \leftarrow 1$
$\bar{A} \leftarrow \{a(1,1,1)\}$, $\bar{B} \leftarrow \{b(1,1)\}$, and $\bar{C} \leftarrow \{c(1)\}$
while $k < N$ **do**
  while $i < N$ **do**
    while $j < N$ **do**
      $\bar{A} \leftarrow$ CalcNewElem(a, (i,j,k), G_a, $\bar{A}$)
      $j \leftarrow j + 1$
    $\bar{B} \leftarrow$ CalcNewElem(b, (i,k), G_b, $\bar{B}$)
    $i \leftarrow i + 1$
  $\bar{C} \leftarrow$ CalcNewElem(c, k, G_c, $\bar{C}$)
  $k \leftarrow k + 1$
$A_N \leftarrow \bar{A}$, $B_N \leftarrow \bar{B}$, and $C_N \leftarrow \bar{C}$

C. Results and comparison

For comparison, A_N and B_N were computed using an existing integration method implemented in `Chebfun` and `Maple` with 10 significant digits. Computations with `Chebfun` were parallelized in MATLAB using 12 workers.

The error of each integral is evaluated as the absolute difference from the value obtained using `Maple`. Table II summarizes the errors in the proposed and existing methods. For A_N, the maximum and mean errors in the proposed method are large and identical to those in the existing method. This is because the absolute difference, as well as the integrals, grows rapidly with respect to the indices i, j, k. Indeed, both the proposed and existing methods yielded the maximum error at $i = j = k = 14$, whose magnitude was almost 10^{-10} times smaller than the integral 5.39×10^{29} computed by `Maple` and thus negligible. On the other hand, the median of the proposed method is considerably smaller than that of the existing method, indicating that the proposed method can compute most of the integrals with high accuracy. For B_N, the proposed method also yielded significantly smaller errors than the existing method.

Table III summarizes the computational times for all the methods. The total time of the proposed method is considerably smaller than that of `Maple`, despite its small error. In particular, the numerical part of the proposed method was finished within one second because it consisted of recursive substitutions. Although the computational time of the existing method was shorter than that of the proposed method, its results were inaccurate.

TABLE I: Difference operators annihilating integrals $a(i,j,k)$, $b(i,k)$, and $c(k)$

G_a	$(j+1)(-j+i+k+1)\mathsf{S}_i + (i+1)(-j+i-k-1)\mathsf{S}_j$
	$(k+1)(-k+i+j)\mathsf{S}_j - (j+2)(-j+i+k)\mathsf{S}_k$
	$(-j+i+k+1)(-k-3-j+i)\mathsf{S}_k^2 + 4(k+2)(k+1)(i+j-k-1)$
G_b	$i(2i-2k-5)\mathsf{S}_i\mathsf{S}_k + (i+1)\mathsf{S}_k^2 - 4(k+1)(i+1)(i-k-2)$
	$i(2i-2k-5)\mathsf{S}_i^2 + (i+2)(2i-2k+1)\mathsf{S}_k^2 + 4(i+2)(-k-1+i)(2i^2-4ik+2k^2-5i+3k-2)$
	$(k+1)(-2k+2i-7)\mathsf{S}_i + i(i+1)(-k-1+i)\mathsf{S}_k^3 + 4(i+1)(i^3 + (-3k-7)i^2 + (3k^2+13k+\frac{53}{4})i - k^3 - 6k^2 - 11k - 6)\mathsf{S}_k$

TABLE II: Errors of integrals

		Max	Mean	Median	Min
A_N	Proposed	3.41×10^{19}	1.43×10^{16}	8.31×10^{-11}	0.00
	`Chebfun`	3.41×10^{19}	1.43×10^{16}	3.42×10^{-2}	0.00
B_N	Proposed	2.48×10^{8}	2.30×10^{6}	3.41×10^{-8}	0.00
	`Chebfun`	1.28×10^{13}	2.13×10^{11}	1.10×10^{-4}	0.00

Remark 1: The computation of recurrence relations, which accounts for most of the computational time of the proposed method, does not depend on the number of basis functions N. This indicates that even for large N, the computational time of the proposed method does not increase significantly, in contrast to other methods.

The SGA (Algorithm 1) was performed using the integrals computed by each method. The initial guess $v^{(0)}$ is set to $v_k^{(0)} = 0$ except for $v_2^{(0)} = (1\sqrt{2})/8$, which corresponds to $V^{(0)}(x) = v_2^{(0)} H(2;x)$ and is equal to the value function $V(x) = (1+\sqrt{2})x^2/2$ of the linearized system $\dot{x} = x + u$ with J in (22) up to a constant. Note that for the integrals obtained by `Chebfun`, the SGA failed to converge owing to the errors included in the integrals.

Figure 1 shows $V^*(x)$ and $\mathrm{HJB}(V^*(x))$ computed from the integrals obtained by the proposed method, `Chebfun`, and `Maple`, where the constant term of each value function is adjusted such that $V^*(0) = 0$. For the proposed method, the region where $\mathrm{HJB}(V^*(x)) \simeq 0$ is considerably wider than those of the initial guess and existing method, which indicates that the proposed method computes the integrals accurately. Finally, the state and input trajectories for $x(0) = 4$ are shown in Fig. 2, which indicates that the feedback law obtained from $V^*(x)$ with $N = 14$ stabilizes the system while suppressing the magnitude of the input.

VII. Conclusion

In this paper, a symbolic-numeric computation method for integrals in the successive Galerkin approximation (SGA) is proposed. The SGA approximates a solution to the Hamilton-Jacobi-Bellman equation by iteratively solving a linear equation, which is defined by multiple integrals of nonlinear functions. Using the symbolic computation of differential operators and the Mellin transform of difference-differential operators, a set of recurrence relations satisfied by the integrals can be computed. After evaluating several integrals numerically, the recurrence relations allow us to compute all the other integrals using recursive substitutions.

Future work directions include extending the proposed method to the case of multidimensional systems. Moreover, the approximation with different basis functions, such as the Chebyshev polynomials, could be investigated.

TABLE III: Comparison of computational times [s]

	Proposed	Maple	Chebfun
Recurrence relations	4.48×10^{1}	N/A	N/A
Integrals	1.33×10^{-1}	1.62×10^{4}	2.41×10^{1}
Total	4.49×10^{1}	1.62×10^{4}	2.41×10^{1}

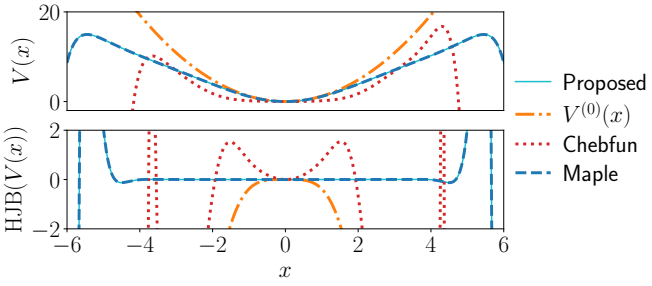
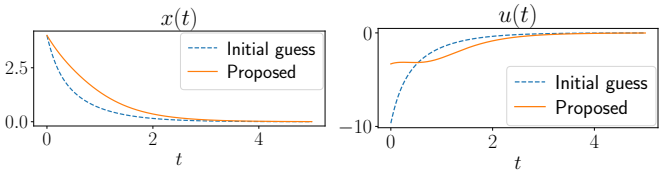


Fig. 1: Value functions and evaluation of HJB equation



(a) State (b) Input

Fig. 2: Trajectories for $x(0) = 4$

References

[1] J. A. E. Bryson and Y.-C. Ho *Applied Optimal Control*, John Wiley & Sons, 1st Edition, 1975.

[2] D. L. Lukes, "Optimal regulation of nonlinear dynamical systems," *SIAM J. Control*, vol. 7, no. 1, pp. 75–100, 1969.

[3] W. L. Garrard, "Suboptimal feedback control for nonlinear systems," *Automatica*, vol. 8, no. 2, pp. 219–221, 1972.

[4] H. Almubarak, N. Sadegh, and D. G. Taylor, "Infinite horizon nonlinear quadratic cost regulator," *Proc. Am. Control Conf.*, pp. 5570–5575, 2019.

[5] N. Sakamoto and A. J. van der Schaft, "Analytical approximation methods for the stabilizing solution of the Hamilton-Jacobi equation," *IEEE Trans. Autom. Control*, vol. 53, no. 10, pp. 2335–2350, 2008.

[6] R. W. Beard, G. N. Saridis, and J. T. Wen, "Galerkin approximations of the generalized Hamilton-Jacobi-Bellman equation," *Automatica*, vol. 33, no. 12, pp. 2159–2177, 1997.

[7] D. Kalise and K. Kunisch, "Polynomial approximation of high-dimensional Hamilton-Jacobi-Bellman equations and applications to feedback control of semilinear parabolic PDEs," *SIAM J. Sci. Comput.*, vol. 40, no. 2, pp. A629–A652, 2018.

[8] I. Maruta, S. Nishida, and K. Fujimoto, "A study on numerical solutions of Hamilton-Jacobi-Bellman equations based on successive approximation approach," *SICE J. Control Meas. Syst. Integr.*, vol. 13, no. 3, pp. 157–163, 2020.

[9] M. Saito, B. Sturmfels, and N. Takayama *Gröbner Deformations of Hypergeometric Differential Equations*, Springer-Verlag, 2000.

[10] T. Oaku, Y. Shiraki, and N. Takayama, "Algebraic algorithms for D-Modules and numerical analysis," *Computer Mathematics (Pro. ASCM 2003)*, vol. 10, pp. 23–39, 2003.

[11] T. Hibi ed. *Gröbner Bases: Statistics and Software Systems*, Springer Japan, 1st edition, 2013.

[12] T. Oaku, "Algorithms for D-modules, integration, and generalized functions with applications to statistics," *Adv. Stud. Pure Math.*, vol. 77, pp. 253–352, 2018.

[13] H. Nakayama, K. Nishiyama, M. Noro, K. Ohara, T. Sei, N. Takayama, and A. Takemura, "Holonomic gradient descent and its application to the Fisher-Bingham integral," *Adv. Appl. Math.*, vol. 47, no. 3, pp. 639–658, 2011.

[14] A. Kume and T. Sei, "On the exact maximum likelihood inference of Fisher-Bingham distributions using an adjusted holonomic gradient method," *Stat. Comput.*, vol. 28, no. 4, pp. 835–847, 2018.

[15] F. Bréhard, M. Joldes, and J.-B. Lasserre, "On a moment problem with holonomic functions," in *Proc. Int. Symp. Symb. Algebr. Comput. (ISSAC)*, pp. 66–73, 2019.

[16] T. Iori and T. Ohtsuka, "Nonlinear Bayesian filtering via holonomic gradient method with quasi moment generating function," *Asian J. Control*, vol. 25, no. 4, pp. 2655–1670, 2023.

[17] W. Decker, G.-M. Greuel, G. Pfister, and H. Schönemann, "Singular 4-2-0 —A computer algebra system for polynomial computations," 2019.

[18] D. R. Grayson and M. E. Stillman, "Macaulay2, a software system for research in algebraic geometry," 2020.

[19] M. Noro, N. Takayama, H. Nakayama, K. Nishiyama, and K. Ohara, "Risa/Asir: A computer algebra system," 2020.

[20] S. C. Coutinho *A Primer of Algebraic D-Modules*, Cambridge University Press, 1995.

[21] C. Koutschan, "Advanced applications of the holonomic systems approach," Ph.D. dissertation, Johannes Kepler University Linz, 2009.

[22] T.A. Driscoll, N. Hale, and L.N. Trefethen, eds., *Chebfun Guide*, Pafnuty Publications, Oxford, 2014.