Legendre spectral collocation method for approximating the solution of shortest path problems

Emran Tohidi* and Omid Reza Navid Samadi

Department of Mathematics, Islamic Azad University, Zahedan Branch, Zahedan, Iran
(Received 8 August 2014; accepted 6 November 2014)

In the current article, we propose an accurate spectral approximation for solving the shortest path problems with boundary and interior barriers. For this goal, the shortest path problems are modelled as variational problems (VPs). Then, the Legendre polynomials are used as a basis for approximating the solution of these problems, and by using the Chebyshev–Gauss–Lobatto collocation points together with the Legendre–Gauss quadrature rule, the VPs will be changed into nonlinear programming problems (NLPs). The resulting NLPs are solved by the NLPSolve command in MAPLE software. Three numerical examples are provided for showing the robustness of the proposed method.

Keywords: shortest path problems; modelling; Legendre polynomials; Gauss–Lobatto collocation points; Gauss quadrature rules; nonlinear programming problem

1. Introduction
The shortest path problems (SPPs) have received considerable attention in engineering especially in robot industrial and recently in surgery planning. For instance, Latombe (1991) has gathered novel methods for path planning in the presence of obstacles and some extensions of them. Moreover, the problem of optimal path planning in Wang, Lane, and Falconer (2000) is considered as a semi-infinite constrained optimization problem. An important application of SPPs is in the Military industry for rocket range optimization. The task of planning trajectories for a rocket is one of the most applicable research subjects in the military research literature. Most of research works assume that the rocket has just boundary barriers in the model of its environment (see, for instance, Zamirian, Farahi, & Nazemi, 2007). However, less attention has been paid to the problem of interior together with boundary barriers in a known environment.

The optimal algorithms should be employed to find the lowest cost path from the robot (or rocket) start state to the goal state. Cost can be defined to be distance travelled, energy expended, time exposed to danger, etc. However, in this paper cost is defined to be distance travelled. Existing approaches such as Borzabadi, Kamyad, Farahi, and Mehne (2005) and Zamirian et al. (2007) planned traditional and classical methods such as measure theoretical approaches. These techniques usually need to transform the basic problem to an optimal control problem (OCP). These schemes have two disadvantages such as an increase in the dimension of associated algebraic problem and also the weakness of the approximate solution. For instance, in Borzabadi et al. (2005) the authors solve a collection of SPPs using a method based on measure theory. They considered the SPPs as optimization problems and then they convert these problems into OCPs by defining some artificial control functions. Using properties of some kind of measures, they obtained a linear programming problem (LPP) that their solutions give rise to constructing approximate optimal trajectory of the original problem. Defining the artificial control functions usually gives rise to the increase in the associated algebraic problem dimension. Moreover, the methods by Borzabadi et al. (2005) and Zamirian et al. (2007) deal with the local approximations (such as finite-difference methods), meanwhile global approximations have more accuracy with respect to the local approximations.

Because of the aforementioned reasons, we propose a global approximate method in which the dimension of the algebraic problem is very low. These are our motivations for presenting our basic idea. Therefore, in this paper by using Legendre polynomials as a basis for approximating the solutions we convert the SPPs into the associated nonlinear programming problems (NLPs). In other words, the infinite-dimensional SPPs will be transformed into the associated finite-dimensional NLPs in which the cost functionals are approximated by the Legendre–Gauss quadrature rule and the constraints of SPPs are collocated at the Chebyshev–Gauss–Lobatto (CGL) points. For clarity of
where the start state and the final state are respectively.

A general class of SPPs with boundary barriers \( \psi_1(s) \) and \( \psi_2(s) \) and interior barriers in the shape of circles with the centers \((\alpha_i, \beta_i), i = 1, 2, \ldots, k\), can be modelled by the following variational problem (VP):

\[
\begin{align*}
\text{Min} \quad & J = \int_{-1}^{1} \sqrt{1 + x^2(s)} \, ds \\
\text{s.t.} \quad & \psi_1(s) \leq x(s) \leq \psi_2(s), \quad \forall s \in [-1, 1], \\
& (s - \alpha_i)^2 + (x(s) - \beta_i)^2 \geq D_i^2, \quad i = 1, 2, \ldots, k, \\
& x(-1) = \alpha, \quad x(1) = \beta,
\end{align*}
\]

where the start state and the final state are \( \alpha \) and \( \beta \), respectively.

The rest of this paper is organized as follows. In the next section, some preliminaries about the spectral approximations especially Legendre polynomials will be provided. The basic idea of this paper, in which the VP (1) is changed into a NLP, is stated in Section 3. Three numerical examples are given in Section 4 for illustrating the efficiency and applicability of the proposed method. In Section 5, conclusions of the proposed method are provided. Finally, the MAPLE codes of the first example are given in the appendix.

2. Preliminaries

Spectral methods have proven to be powerful tools that are frequently employed in many fields of numerical analysis. They have a higher order of accuracy with respect to the finite-difference methods (FDMs) and finite element methods (FEMs) in the case of smooth solutions of any considered problem. For instance in Samadi and Tohidi (2012), spectral methods have shown their robustness with regard to the Bessel collocation scheme (Sahin, Yuzbasi, & Gulsu, 2011), homotopy perturbation method (Biazar & Ghazvini, 2009) and Block-by-Block technique (Katani & Shahmorad, 2010) for solving the system of Volterra integral equations. Also, in Bhrawy, Assas, Tohidi, and Alghamdi (2013) and Toutounian, Tohidi, and Kilicman (2013), Legendre and Fourier methods depict their spectral accuracy with regard to the reproducing kernel methods and other methods for solving Pantograph delay differential equations. In addition, spectral methods show their higher order accuracy in Doha and Bhrawy (2008, 2012), Doha, Bhrawy, and Abd-Elhameed (2009) and Doha, Bhrawy, and Ezz-Eldien (2011). Moreover, the efficiency of spectral methods has been proved for OCPs governed by Volterra integral equations in Tohidi and Samadi (2012). Among the spectral approximations, the Gegenbauer polynomials are used in many research works (Abd-Elhameed & Yousrri, 2014). The simple type of the Gegenbauer polynomials is the Legendre polynomials which are orthogonal in the interval \([-1, 1]\) and satisfy the following recurrence relation (Tohidi & Samadi, 2012):

\[
P_{i+1}(s) = \frac{2i+1}{i+1} sP_i(s) - \frac{i}{i+1} P_{i-1}(s), \quad i \geq 1,
\]

where \( P_0(s) = 1 \) and \( P_1(s) = s \).

The orthogonal property of Legendre polynomials is given by

\[
\int_{-1}^{1} P_i(s) P_j(s) \, ds = \begin{cases} 0, & i \neq j, \\ 2 \frac{2i+1}{2i+2}, & i = j. \end{cases}
\]

A function \( f(s) \), which is absolutely integrable within \(-1 \leq s \leq 1\), may be expressed in terms of a Legendre series as

\[
f(s) = \sum_{i=0}^{\infty} f_i P_i(s),
\]

where

\[
f_i = \frac{2i+1}{2} \int_{-1}^{1} f(s) P_i(s) \, ds.
\]

**Proposition 1** If we assume that the derivative of \( f(s) \) in Equation (2) is described by

\[
f'(s) = \sum_{i=0}^{\infty} g_i P_i(s),
\]

the relationship between the coefficients \( f_i \) in Equation (2) and \( g_i \) in Equation (3) can be obtained as follows:

\[
(2i+1)g_{i-1} - (2i-1)g_{i+1} - (2i-1)(2i+3)f_i = 0,
\]

\[
i = 1, 2, \ldots, \]

**Proof** The proof has been provided in Canuto, Hussaini, Quarteroni, and Zang (1987).

The Chebyshev–Gauss (CG), Chebyshev–Gauss–Rad au (CGR), and CGL collocation points lie on the open interval \( s \in (-1, 1) \), the half open interval \( s \in [-1, 1) \) or \( s \in (-1, 1] \), and the closed interval \( s \in [-1, 1] \), respectively. A depiction of these three sets of collocation points is shown in Figure 1 where it is seen that the CG points contain neither –1 nor 1, the CGR points contain only one of the points –1 or 1 (in this case, the point –1), and the CGL points contain both –1 and 1. In the procedure of collocating the constraints of VP (1), we use the CGL points for clarity of presentation and also for this reason that they cover the hole parts of the computational interval \([-1, 1]\). CGL points can be demonstrated in the following form:

\[
s_i = -\cos \left( \frac{i\pi}{N} \right), \quad i = 0, 1, \ldots, N.
\]
The Legendre–Gauss quadrature rule can be defined as follows (Krylov, 1962):

\[
\int_{-1}^{1} h(s) \, ds \approx \sum_{i=0}^{N} w_{i} h(t_{i}),
\]

where \( t_{i} \) for \( i = 0, 1, \ldots, N \) are the distinct roots of the \((N + 1)\)th Legendre polynomial \( P_{N+1}(s) \) and \( w_{i} = 2/(1 - t_{i}^{2}) P'_{N+1}(t_{i}). \)

3. Basic idea

We again consider the VP (1). Our aim is to discretize Equation (1) and then optimize itself. For this purpose, a discretization of the interval \(-1 = s_{0} < s_{1} < \cdots < s_{N} = 1\) is considered, where \( s_{i}'s \) are defined in Equation (5). The following expansions for approximating both \( x(s) \) and \( x'(s) \) are assumed:

\[
x(s) \approx x^{N}(s) = \sum_{i=0}^{N} a_{i} P_{i}(s),
\]

\[
x'(s) \approx x'(s) = \sum_{i=0}^{N} c_{i} P_{i}(s),
\]

where \( P_{i}(s) \) is the \( i \)th Legendre polynomial. The relationship between \( a_{i}'s \) and \( c_{i}'s \) is stated in Proposition 1. Now, we approximate the cost functional of VP (1) by the Legendre–Gauss quadrature rule (6). Therefore,

\[
\int_{-1}^{1} \sqrt{1 + (x'(s))^{2}} \, ds \approx \int_{-1}^{1} \sqrt{1 + \left( \sum_{i=0}^{N} c_{i} P_{i}(s) \right)^{2}} \, ds
\]

\[
= \int_{-1}^{1} F(c_{0}, \ldots, c_{N}, s) \, ds \approx \sum_{i=0}^{N} w_{i} F(c_{0}, \ldots, c_{N}, t_{i}),
\]

where \( F(c_{0}, \ldots, c_{N}, s) := \sqrt{1 + (\sum_{i=0}^{N} c_{i} P_{i}(s))^{2}} \) is a nonlinear function in terms of its variables \( c_{0}, c_{1}, \ldots, c_{N} \) and \( s \) and all of the \( t_{i}'s \) and \( w_{i}'s \) are defined in Equation (6). For the sake of clarity, we assume that

\[
G(c_{0}, c_{1}, \ldots, c_{N}) := \sum_{i=0}^{N} w_{i} F(c_{0}, \ldots, c_{N}, t_{i}).
\]

After approximating the cost functional, we use the CGL collocation points for discretizing the inequality state constraints, the VP (1) is changed into the following NLP:

\[
\begin{aligned}
\text{Min} & \quad G(c_{0}, c_{1}, \ldots, c_{N}) \\
\text{s.t.} & \quad \psi_{1}(s_{j}) \leq x(s_{j}) \leq \psi_{2}(s_{j}), \quad j = 0, 1, \ldots, N, \\
& \quad (s_{j} - a)_{j}^{2} + (x(s_{j}) - \beta)_{j}^{2} \geq D_{j}^{2}, \quad j = 0, 1, \ldots, N, \quad i = 1, 2, \ldots, k,
\end{aligned}
\]

\[
x^{N}(-1) = \sum_{i=0}^{N} a_{i} P_{i}(-1) = \alpha,
\]

\[
x^{N}(1) = \sum_{i=0}^{N} a_{i} P_{i}(1) = \beta,
\]

\[
[a_{1}, a_{2}, \ldots, a_{N}]^{T} = M[c_{0}, c_{1}, \ldots, c_{N}]^{T},
\]

where \( G(c_{0}, c_{1}, \ldots, c_{N}) \) is a nonlinear objective function, \( \hat{P}_{i}(0) = \hat{P}_{i}(s_{0}) = (-1)^{i} \) and \( \hat{P}_{i}(s_{N}) = 1. \)

Note that the last constraints of Equation (9) arise from the following relations:

\[
a_{i} = \left[ \frac{c_{i-1}}{2i - 1} - \frac{c_{i+1}}{2i + 3} \right], \quad i = 1, 2, \ldots, N,
\]

where \( c_{N+1} = c_{N} = 0. \)

Before presenting numerical examples, it should be noted that the convergence analysis of the proposed scheme can be done using some similar ideas provided in Doha, Abd-Elhameed, and Youssri (2014) and Tohidi and Samadi (2012). Since this subject needs some theoretical sections, we just provide the mentioned references.

4. Numerical examples

In this section, we conduct two numerical examples to illustrate the effectiveness of the proposed method. We use the method stated in the previous section to transform the main problem (1) into the equivalent NLP (9). All the problems are programmed in MAPLE 13 and run on a Laptop PC with 1.8 GHz and 2 GB RAM. It should be noted that, for solving the NLPs in MAPLE (i.e. the NLPSolve command), there are some options in which the NLPs are solved. If the NLP is univariate and unconstrained except for finite bounds, the quadratic interpolation method may be used. If the problem is unconstrained and the gradient of the objective function is available, the preconditioned conjugate gradient (PCG) method may be used. Otherwise, the sequential quadratic programming (SQP) method
can be used. According to the structure of our NLP in Equation (9), the SQP method is used.

Example 1 As the first example, we consider the following SPP with one lower boundary barrier (Hestenes, 1996):

$$\text{Min } J = \int_{-5/4}^{5/4} \sqrt{1 + x'^2(\tau)} \, d\tau$$

s.t. $x(\tau) \geq 1 - \tau^2, \forall \tau \in \left[ -\frac{5}{4}, \frac{5}{4} \right],$

$$x\left( -\frac{5}{4} \right) = 0, \quad x\left( \frac{5}{4} \right) = 0.$$

The optimal value of the objective functional is $J^* = 3.26911$ and the exact solution of this SPP is as follows:

$$x^*(\tau) = \begin{cases} \tau + \frac{5}{4}, & -\frac{5}{4} \leq \tau \leq -\frac{1}{2}, \\ 1 - \tau^2, & -\frac{1}{2} \leq \tau \leq \frac{1}{2}, \\ -\tau + \frac{5}{4}, & \frac{1}{2} \leq \tau \leq \frac{5}{4}. \end{cases}$$

Since the computational interval of this problem is $[-\frac{5}{4}, \frac{5}{4}]$, we should transform this interval into $[-1, 1]$ by changing variable $\tau = \frac{5}{2}s$. The readers can see this subject in the appendix using MAPLE codes. By assuming different values of $N$ such as 4, 8, 18 and 32, we solve this problem by the proposed idea. In Table 1, the approximated objective functionals $J^*_N$ are given for these values of $N$. It is seen from this table that we reach the exact objective functional in the case of $N = 32$. Also, a comparison between the numerical solution $x^N(s)$ ($N = 32$) together with the exact solution $x^*(s)$ at a uniform mesh is given in Table 2. From this table, one can conclude that our method has high order of accuracy and fits with the exact solution. Moreover, the history of the numerical solution $x^N(s)$ ($N = 32$) is depicted in Figure 2.

Example 2 As the second example, we consider the following SPP with two lower and upper boundary barriers (Zamirian et al., 2007):

$$\text{Min } J = \int_{3}^{6} \sqrt{1 + x'^2(\tau)} \, d\tau$$

s.t. $\sin(\tau - 0.25) \leq x(\tau) \leq 1 + \sin(\tau),$

$$x(3) = 0.75, \quad x(6) = 0.5.$$

It should be noted that this SPP has no exact solution. Similar to the previous example, we should transform the interval $[3,6]$ into $[-1, 1]$ by change of variable $\tau = \frac{9}{2}s + \frac{9}{2}$. We can solve this SPP by taking different values of $N$, but for showing the robustness of the presented technique with regard to the method of Zamirian et al. (2007), the considered problem has been solved by taking $N = 25$ and we reach $J^*_N = 3.2772$ ($N = 25$), meanwhile the optimal $J^*$ of Zamirian et al. (2007) is 3.4191. This confirms that our idea reaches the shorter path with respect to the method of Zamirian et al. (2007). The history of this optimal path $x^N(s)$ ($N = 25$) is depicted in Figure 3.

Example 3 As our final numerical example, we consider the following SPP with two boundary barriers and three

---

**Table 1. Numerical results of Example 1 for objective functionals.**

| N     | $J^*_N$   |
|-------|-----------|
| 4     | 3.26019   |
| 8     | 3.26492   |
| 18    | 3.26920   |
| 32    | 3.26911   |

**Table 2. Comparison of the exact solution $x^*(s)$ and $x^N(s)$ at the selected points for $N = 32$.**

| si  | $x^*(si)$ | $x^N(si)$ |
|-----|-----------|-----------|
| -1  | 0.0       | 0.0       |
| -0.8| 0.25      | 0.2503    |
| -0.6| 0.50      | 0.5007    |
| -0.4| 0.75      | 0.7505    |
| -0.2| 0.9375    | 0.9375    |
| 0.0 | 1.0       | 1.0       |
| 0.2 | 0.9375    | 0.9375    |
| 0.4 | 0.7505    | 0.7505    |
| 0.6 | 0.5       | 0.5007    |
| 0.8 | 0.25      | 0.2503    |
| 1.0 | 0.0       | 0.0       |
Similar to the previous problem, this SPP has no exact solution. Moreover, a transformation of the interval [3,6] into [−1, 1] by change of variable \( \tau = \frac{3}{2}s + \frac{9}{2} \) should be done. After doing this, we solve this SPP using our basic technique and reach the optimal objective functional \( J^*_N = 3.279366169 \) (\( N = 11 \)). The history of this optimal path \( x^N(s) \) (\( N = 11 \)) is depicted in Figure 4. From this figure, one can see that the numerical optimal path \( x^N(s) \) (\( N = 11 \)) does not meet the interior barriers. This confirms the efficiency and applicability of the presented idea.

5. Conclusion

The aim of this paper is to determine the optimal solution of SPPs with boundary and interior barriers by a direct method of solution based upon truncated Legendre series expansions together with the CGL points as collocation nodes. The method is based upon reducing a nonlinear SPP to an NLP. The unity of the weight function of orthogonality for the truncated Legendre series and the simplicity of the discretization are merits that make our idea very attractive. Moreover, only a small number of truncated Legendre series are needed to obtain a very satisfactory solution. The given numerical examples support this claim.

Acknowledgments

The authors would like to thank the referees for the valuable comments and nice suggestions which led to the final version of the manuscript.

Disclosure statement

The authors declare that they do not have any conflict of interest in their submitted manuscript.

References

Abd-Elhameed, W. M., & Youssri, Y. H. (2014). New ultraspherical wavelets spectral solutions for fractional Riccati differential equations. *Abstract and Applied Analysis*, 2014, 1–8.

Bhrawy, A. H., Assas, L. M., Tohidi, E., & Alghamdi, M. A. (2013). A Legendre–Gauss collocation method for neutral functional–differential equations with proportional delay. *Advances in Difference Equations*, 2013. doi:10.1186/1687-1847-2013-63

Biazar, J., & Ghazvini, H. (2009). He’s homotopy perturbation method for solving system of Volterra integral equations of the second kind. *Chaos Solitons Fractals*, 39, 770–777.

Borzabadi, A. H., Kamyad, A. V., Farahi, M. H., & Mehne, H.H. (2005). Solving some optimal path planning problems using an approach based on measure theory. *Applied Mathematics and Computation*, 170, 1418–1435.

Canuto, C., Hussaini, Y., Quarteroni, A., & Zang, T. A. (1987). *Spectral Methods in Fluid Dynamics (Scientific Computation)*. New York: Springer.

Doha, E. H., Abd-Elhameed, W. M., & Youssri, Y. H. (2014). New algorithms for solving third and fifth-order two point boundary value problems based on nonsymmetric generalized Jacobi Petrov–Galerkin method. *Journal of Advanced Research*, 2014. doi:10.1016/j.jare.2014.03.003
Doha, E. H., & Bhrawy, A. H. (2008). Efficient spectral-Galerkin algorithms for direct solution of fourth-order differential equations using Jacobi polynomials. *Applied Numerical Mathematics, 58*, 1224–1244.

Doha, E. H., & Bhrawy, A. H. (2012). An efficient direct solver for multidimensional elliptic Robin boundary value problems using a Legendre spectral-Galerkin method. *Computers & Mathematics with Applications, 64*, 558–571.

Doha, E. H., Bhrawy, A. H., & Abd-Elhameed, W. M. (2009). Jacobi spectral Galerkin method for elliptic Neumann problems. *Numerical Algorithms, 50*, 67–91.

Doha, E. H., Bhrawy, A. H., & Ezz-Eldien, S. S. (2011). Efficient Chebyshev spectral methods for solving multi-term fractional orders differential equations. *Applied Mathematical Modelling, 35*, 5662–5672.

Hestenes, M. R. (1996). *Calculus of variational and optimal control theory*. New York: Wiley.

Katani, R., & Shahmorad, S. (2010). Block by block method for the systems of nonlinear Volterra integral equations. *Applied Mathematical Modelling, 34*, 400–406.

Krylov, V. I. (1962). *Approximate calculation of integrals*. Mineola, New York: Dover.

Latombe, J. C. (1991). *Robot motion planning*. Boston: Kluwer.

Sahin, N., Yuzbasi, S., & Gulsu, M. (2011). A collocation approach for solving systems of linear Volterra integral equations with variable coefficients. *Computers & Mathematics with Applications, 62*, 755–769.

Samadi, O. R. N., & Tohidi, E. (2012). The spectral method for solving systems of Volterra integral equations. *Journal of Applied Mathematics and Computing, 40*, 477–497.

Tohidi, E., & Samadi, O. R. N. (2012). Optimal control of nonlinear Volterra integral equations via Legendre polynomials. *IMA Journal of Mathematical Control and Information, 30*, 67–83.

Toutounian, F., Tohidi, E., & Kilicman, A. (2013). Fourier operational matrices of differentiation and transmission: Introduction and applications. *Abstract and Applied Analysis, 2013*, 1–11.

Wang, Y., Lane, D. M., & Falconer, G. J. (2000). Two novel approaches for unmanned underwater vehicle path planning: Constrained optimisation and semi-infinite constrained optimisation. *Robotica, 18*, 123–142.

Zamirian, M., Farahi, M. H., & Nazemi, A. R. (2007). An applicable method for solving the shortest path problems. *Applied Mathematics and Computation, 190*, 1479–1486.
Appendix. MAPLE codes of the first example

```
restart;
with(orthopoly):with(Optimization):with(Student[Calculus1]):
r := 33; a := -5/4; b := 5/4; # r = N + 1
f 1 := (t) -> (1 - t^2):
Digits := 16:
t := (1/2) * (b - a) * s + (b + a) * (1/2):
F1 := f 1(t):
F1 := unapply(F1,s):
x0 := 0;x1 := 0:
c[r] := 0;c[r + 1] := 0:
for j from 1 to r - 1 do
A[j + 1] := c[j]/(2*j - 1) - c[j + 2]/(2*j + 3)
end do:
for j from 1 to r do
s[j] := -cos((j - 1) * Pi/(r - 1))
end do:
X := sum(A[i] * P(i - 1, s), i = 1..r):
X := unapply(X,s):
X1 := sum(c[i] * P(i - 1, s), i = 1..r):
S1 := seq(X(s[i]) >= F1(s[i]), i = 1..r):
S2 := {X(-1) - x0 = 0}:
S3 := {X(1) - x1 = 0}:
S := S1 union S2 union S3:
x := fsolve(P(r, s),):
PP := diff(P(r, s),):
PP := unapply(PP, s):
for j from 1 to r do
w[j] := 2/(1 - (x[j]^2)) * PP(x[j])^2
end do:
g := ((b - a) * (1/2)) * (1 + (2 * X1/(b - a))^2)^(1/2):
g := unapply(g,s):
z := sum(w[i] * g(s[i]), i = 1..r):
sol := NLPSolve(z, S):
assign(sol[2]):
X(s) := plot(1 - (25/16) * s^2, X(s), s = -1..1, color = [red, black], style = [line, line], font = [5, BOLD]);
```