Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 2184658, 7 pages
https://doi.org/10.1155/2017/2184658

Research Article

Adaptive Mesh Iteration Method for Trajectory Optimization Based on Hermite-Pseudospectral Direct Transcription

Humin Lei,[1] Tao Liu,[1] Deng Li,[2,3] Jikun Ye,[1] and Lei Shao[1]

[1] *Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China*
[2] *College of Education, Shaanxi Normal University, Xi'an 710062, China*
[3] *Nanyang No. 2 High School, Nanyang 473000, China*

Correspondence should be addressed to Tao Liu; ltloveder@163.com

Received 20 February 2017; Accepted 3 May 2017; Published 18 June 2017

Academic Editor: Vladimir Turetsky

An adaptive mesh iteration method based on Hermite-Pseudospectral is described for trajectory optimization. The method uses the Legendre-Gauss-Lobatto points as interpolation points; then the state equations are approximated by Hermite interpolating polynomials. The method allows for changes in both number of mesh points and the number of mesh intervals and produces significantly smaller mesh sizes with a higher accuracy tolerance solution. The derived relative error estimate is then used to trade the number of mesh points with the number of mesh intervals. The adaptive mesh iteration method is applied successfully to the examples of trajectory optimization of Maneuverable Reentry Research Vehicle, and the simulation experiment results show that the adaptive mesh iteration method has many advantages.

1. Introduction

There are general two kinds of methods to solve the optimal control problems, indirect method and direct method [1–3]. The indirect method uses the Pontryagin minimum principle and the first-order optimal necessary conditions to convert the optimal problem to two or more points boundary value problem. This method could obtain the exact optimal solution. The advantages of the indirect method are obtaining the high precision solution, satisfying the optimality of the first-order necessary conditions. The disadvantages are the small convergence radius and highly sensitive costate which is difficult to estimate. In recently years, direct methods have been widely applied for the numerical solution of nonlinear optimal control problems [4, 5]. The state and control are discretized at a series of suitable points in a direct method, then the continuous-time optimal control is converted into a finite dimensional nonlinear programming problem (NLP), and the next procedure is using the NLP solver software to solve the NLP [6].

Among the popular direct methods are the pseudospectral method and Hermite-Simpson method [7, 8]. The pseudospectral method has the advantage of high rate of convergence [9–11], so this paper develops the theory about the Legendre pseudospectral method. There are also some considerable interest in developing theory related to Hermite-Simpson method due to its reasonable accuracy with highly sparse Hessian and constraint Jacobians matrix. Williams provides a framework for arbitrary order and arbitrary number of intervals for implementation on digital computers [12]. That method allows the trading of the mesh points within each interval with the number of intervals; thus, better accuracy can be achieved by increasing mesh points for smooth problems, while increasing the number of intervals to achieve better accuracy for nonsmooth problems. However, it is a fact that smooth regions and nonsmooth regions together exist in one solution of the problem, so it is difficult to trade the number of mesh points with the number of mesh intervals when solving a complicated problem. Motived by the desire to trade the number of mesh points with the number of mesh intervals, both the number of mesh points within each mesh interval and the number of mesh intervals are allowed to vary in the method described in this paper. Furthermore,

the method also can improve computational efficiency by reducing the size of the mesh.

2. Trajectory Optimization Problems

Without loss of generality, consider the following optimal control problems with inequality path constraints:

$$J = M\left(x(-1), t_0, x(+1), t_f\right) + \frac{t_f - t_0}{2}\int_{-1}^{+1} L\left(x(\tau), u(\tau), t\left(\tau, t_0, t_f\right)\right) d\tau, \tag{1}$$

subject to the constraints

$$\begin{aligned} &\frac{dx}{d\tau} = \frac{t_f - t_0}{2} f\left(x(\tau), u(\tau), t\left(\tau, t_0, t_f\right)\right), \\ &C\left(\mathrm{x}(\tau), \mathrm{u}(\tau), t\left(\tau, t_0, t_f\right)\right) \le 0, \\ &B\left(\mathrm{x}(-1), t_0, \mathrm{x}(-1), t_f\right) \le 0, \end{aligned} \tag{2}$$

where the term $x(\tau) \in R^{n_x}$ denotes the state and the term $u(\tau) \in R^{n_u}$ denotes the control. In (1)-(2), the time domain $\tau \in [-1, +1]$ is transformed from the time domain $t \in [t_0, t_f]$ by the following affine transformation [12]:

$$\begin{aligned} &\tau = \frac{2t}{t_f - t_0} - \frac{t_f + t_0}{t_f - t_0}, \\ &t = t(\tau) = \frac{1}{2}\left(\tau\left(t_f - t_0\right) + t_0 + t_f\right), \end{aligned} \tag{3}$$

where the terms t_0 and t_f represent initial time and terminal time, respectively. The basic idea of the approach in this paper is based on interpolating functions for state and costate on Legendre-Gauss-Lobatto (LGL) quadrature nodes [4, 7]. As the LGL nodes points are distributed over the interval $[-1, 1]$, so it will be useful to transform the time domain.

3. Adaptive Mesh Iteration Methodology

3.1. Numerical Discretization. The domain $\tau \in [-1, +1]$ is divided into K mesh subintervals S_k when using mesh refinement. Then we have

$$\bigcup_{k=1}^{K} S_k = [-1, +1], \quad S_k = \left[T_{k-1}, T_k\right]. \tag{4}$$

The mesh points have the property $-1 = T_0 < T_1 < \cdots < T_k = +1$. The state in the subintervals S_k is approximated by the Hermite interpolating polynomial with nth order [11]

$$\mathbf{x}(\tau) = \mathbf{a}_0 + \mathbf{a}_1\tau + \cdots + \mathbf{a}_n\tau^n, \quad \tau \in [-1, +1]. \tag{5}$$

Then the cost function is approximated by Gauss-Lobatto quadrature rule as follows:

$$J \approx M\left(\mathbf{x}_1^{(1)}, t_0, \mathbf{x}_N^{(K)}, t_f\right) + \frac{t_f - t_0}{2}\sum_{k=1}^{K}\frac{h_i}{2}\sum_{j=1}^{N_k}\omega_j^{(k)} L\left(\mathbf{x}_j^{(k)}, \mathbf{u}_j^{(k)}, t\left(\tau_i^{(k)}\right)\right), \tag{6}$$

where the term $\omega_j^{(k)}$ is the same as ω_j in [12].

3.2. Approximation of Solution Error. The estimate method for the relative error is similar to the error estimate obtained for numerically solving a different equation through using the modified Euler Runge-Kutta scheme [13, 14]. Suppose the NLP on mesh S_k, $k = 1, \ldots, K$, with N_k HLGL points has been solved. The ensuing mesh with $M_k = N_k + 1$ HLGL points $(\hat{\tau}_1^{(k)}, \ldots, \hat{\tau}_{M_k}^{(k)})$, where $\hat{\tau}_1^{(k)} = \tau_1^{(k)} = T_{k-1}$ and $\hat{\tau}_{M_k}^{(k)} = T_k$. Assume further that $(x(\hat{\tau}_1^{(k)}), \ldots, x(\hat{\tau}_{M_k}^{(k)}))$ are the values of the state approximation at $(\hat{\tau}_1^{(k)}, \ldots, \hat{\tau}_{M_k}^{(k)})$. We then have

$$\begin{aligned} \hat{\mathbf{x}}^{(k)}\left(\hat{\tau}_j^{(k)}\right) &= \hat{\mathbf{x}}^{(k)}\left(\tau_{k-1}\right) + \frac{t_f - t_0}{2} \\ &\cdot \sum_{l=1}^{M_k}\left(\frac{h_l}{2} f\left(\hat{\mathbf{x}}^{(k)}\right)\left(\hat{\tau}_j^{(k)}\right), u^{(k)}\left(\hat{\tau}_j^{(k)}\right), t\left(\hat{\tau}_i^{(k)}, t_0, t_f\right)\right), \\ &\qquad j = 1, \ldots, M_k. \end{aligned} \tag{7}$$

The absolute error and the relative error approximations at $(\hat{\tau}_1^{(k)}, \ldots, \hat{\tau}_{M_k}^{(k)})$ of the state are defined, respectively, as

$$\begin{aligned} &E_i^{(k)}\left(\hat{\tau}_l^{(k)}\right) = \left|\hat{\mathbf{x}}_i^{(k)}\left(\hat{\tau}_l^{(k)}\right) - \mathbf{x}_i^{(k)}\left(\hat{\tau}_l^{(k)}\right)\right|, \\ &e_i^{(k)}\left(\hat{\tau}_l^{(k)}\right) = \frac{E_i^{(k)}\left(\hat{\tau}_l^{(k)}\right)}{1 + \max_{j\in[1,\ldots,N_k+1],\ k\in[1,\ldots,K]}\left|\mathbf{x}_i^{(k)}\left(\tau_l^{(k)}\right)\right|}, \\ &\qquad l = 1, \ldots, M_k,\ i = 1, \ldots, n_x. \end{aligned} \tag{8}$$

The maximum relative error in S_k is then defined as

$$e_{\max}^{(k)} = \max_{i\in[1,\ldots,n_x],\ l\in[1,\ldots,M_k+1]} e_i^{(k)}\left(\hat{\tau}_j^{(k)}\right). \tag{9}$$

3.3. Nonsmooth Solution Location. If a mesh interval has met the accuracy tolerance, that is, $e_{\max}^{(k)} \le \varepsilon$, where ε is the desired relative tolerance, then mesh size is reduced by decrease of the number of collocation points or merging adjacent mesh intervals; otherwise the mesh size needs to be modified by increasing points or dividing that mesh interval into several subintervals. Let $\kappa^{(k)}(\tau)$ be the curvature of the ith component of the state in mesh interval k, as

$$\kappa^{(k)}(\tau) = \frac{\left|\ddot{X}_i^{(k)}(\tau)\right|}{\left|\left(1 + \dot{X}_i^{(k)}(\tau)^2\right)^{3/2}\right|}. \tag{10}$$

Let $\overline{\kappa}^{(k)}$ and $\kappa_{\max}^{(k)}$ be the mean and maximum value of $\kappa^{(k)}(\tau)$, respectively. Then define r_k as the ratio of the maximum to the mean curvature:

$$r_k = \frac{\kappa_{\max}^{(k)}}{\overline{\kappa}^{(k)}}. \tag{11}$$

3.4. Increasing the Number of Mesh Points within a Mesh Interval. When $e_{\max}^{(k)} > \varepsilon$ and if $r_k \le r_{\max}$, where $r_{\max}$ is a user-defined parameter, the curvature is considered uniform in this mesh interval and then the number of collocation points

should be increased in interval k. Let $N_k^{(M)}$ and $N_k^{(M+1)}$ denote the number of collocation points in interval k at mesh M and $M+1$, respectively, where M is the mesh refinement iteration. The number of points $N_k^{(M+1)}$ at mesh $M+1$ is calculated by the formula

$$N_k^{(M+1)} = N_k^{(M)} + P_k, \quad P_k = \text{ceil}\left[\log_{10}\left(\frac{e_{\max}^{(k)}}{\varepsilon}\right)\right]. \tag{12}$$

It is noted in (12) that the ratio of the maximum to the error tolerance has a direct effect on the increases of the polynomial degree in mesh interval. An upper limit $N_{\max}$ is set for the maximum allowable polynomial degree to make sure that the number of collocation points does not grow to an unreasonably large value. If $N_k^{(M+1)} > N_{\max}$ (i.e., $N_k^{(M+1)}$ exceeds the maximum allowable polynomial degree), then the mesh interval S_k must be divided into equally spaced subintervals.

3.5. Generation of New Mesh Segment. Assume $e_{\max}^{(k)} > \varepsilon$ and $r_k > r_{\max}$, and then the kth mesh interval should be refined. The following procedure is the strategy for mesh interval division. Firstly, the predicted polynomial determines the number of all the collocation points in the new subinterval. Secondly, the number of collocation points should be no fewer than the minimum allowable number. In other words, whenever dividing a mesh interval, each interval will contain at least $N_{\min}$ collocation points. Third, the new number of mesh intervals, B_k, is given by the formula

$$B_k = \text{ceil}\left[B_u \log_{10}\left(\frac{e_{\max}^{(k)}}{\varepsilon}\right)\right], \tag{13}$$

where B_u is a user-defined positive integer. In this process, it is ensured that the number of new intervals should be at least two. Thus, the number of new subintervals, denoted as B_k, can be rewritten as

$$B_k = \max\left\{2, \text{ceil}\left[B_u \log_{10}\left(\frac{e_{\max}^{(k)}}{\varepsilon}\right)\right]\right\}. \tag{14}$$

3.6. Reducing the Number of Collocation Points in a Mesh Interval. The relative error of the mesh interval is less than the desired relative tolerance, and $r_k < r_{\max}$, and then the number of collocation points should be decreased. The number of points $N_k^{(M+1)}$ at mesh $M+1$ is calculated by the formula

$$N_k^{(M+1)} = N_k^{(M)} - P_k', \quad P_k' = \text{ceil}\left(\log_{10}\sqrt{\frac{e_{\max}^{(k)}}{\varepsilon}}\right). \tag{15}$$

3.7. Merging Adjacent Mesh Subintervals. Before the adjacent mesh subintervals merging, it is necessary to decrease the number of each interval according to the method in Section 3.5 and then generally estimate the number of mesh interval points. If $N_{k+1} \neq N_k$, the mesh interval $S_{k+1} = [T_k, T_{k+1}]$ and mesh interval $S_k = [T_{k-1}, T_k]$ cannot be merged because the highest polynomial orders of the two adjacent mesh intervals are not equal. All the matching points of the original two mesh intervals are combined, and the conditions for the merging of the two mesh subintervals are mainly three:

(1) The two mesh subintervals must be adjacent.

(2) The relative error estimates of the two grid intervals are not more than ε.

(3) The relative error of the new mesh interval after the merger is not larger than ε.

3.8. Penalty Function Method for Solving NLP. For convenience, the cost function and the path constraints are adjoined together, so that we obtain the augmented cost function

$$\tilde{J} \approx J + \rho \sum_{k=1}^{K}\sum_{j=1}^{N_k} \max\left\{0, C\left(X_j^{(k)}, U_j^{(k)}, t\left(\tau_i^{(k)}\right)\right)\right\}, \tag{16}$$

where the term ρ represents penalty factor and $C_j^{(k)} = C(X_j^{(k)}, U_j^{(k)}, t(\tau_i^{(k)}))$. When the penalty factor is greater than a certain threshold, the solution of problem is equal to the solution of original problem. The max operator has a great difference on the time spent of solving the problem, so it is time-consuming to solve the discretized NLP by using the existing mature gradient-based optimization algorithm. This paper develops the smoothing approximation function method proposed in [15].

$$p_j^{(k)}(\varepsilon, \rho) = \begin{cases} 0, & C_j^{(k)} < -\dfrac{\varepsilon}{\rho n}, \\ \dfrac{\rho n}{2\varepsilon} C_j^{(k)} + C_j^{(k)} + \dfrac{\varepsilon}{2\rho n}, & -\dfrac{\varepsilon}{\rho n} \le C_j^{(k)} < 0, \\ C_j^{(k)} + \dfrac{\varepsilon}{2\rho n}, & C_j^{(k)} > 0. \end{cases} \tag{17}$$

Obviously, the function p is first-order differentiable with respect to C and has a good approximation to the max operation. When the smoothing factor $\varepsilon \to 0$, then we have $p_j^{(k)}(\varepsilon, \rho) = \max\{0, C_j^{(k)}\}$. The optimal index is converted into

$$\tilde{J} \approx J + \rho \sum_{k=1}^{K}\sum_{j=1}^{N_k} p_j^{(k)}(\varepsilon, \rho). \tag{18}$$

The gradient-based NLP algorithm also needs to compute the gradient information of the objective function for the optimal variable. Define the Hamiltonian function of optimal control problem:

$$H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}) = L + \boldsymbol{\lambda}^T f + p(\varepsilon, \rho). \tag{19}$$

3.9. Mesh Refinement Method. The schematic of adaptive mesh iteration method is shown in Figure 1. The adaptive mesh iteration method is summarized as follows.

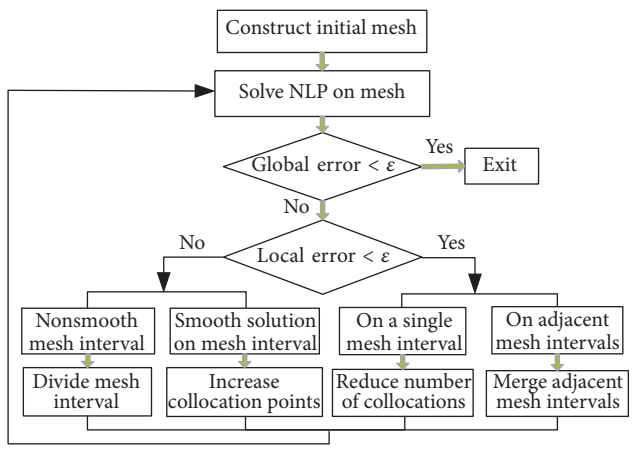

Figure 1: Schematic of adaptive mesh iteration method.

Step 1. Set $M = 0$ and supply initial mesh, $S = \bigcup_{k=1}^{K} S_k = [-1, +1]$, where $\bigcap_{k=1}^{K} S_k = \varnothing$.

Step 2. Solve NLP on current mesh S.

Step 3. Compute maximum relative error $e_{\max}^{(k)}$ in S_k, $k = 1, \ldots, K$, if $e_{\max}^{(k)} \leq \varepsilon$ for all $k = 1, \ldots, K$ or $M > M_{\max}$, and then quit. Otherwise, proceed to Step 4.

Step 4. If $e_{\max}^{(k)} > \varepsilon$, $k = 1, \ldots, K$, proceed to Step 5; otherwise, proceed to Step 6.

Step 5. Compute the ratio between the maximum and the mean curvature r_k in S_k, and if $r_k \leq r_{\max}$, set the number of collocation points according to (15); else divide the interval S_k into B_k subintervals, where B_k is given by (14). Then proceed to Step 7.

Step 6. For the single mesh interval, reduce the number of collocation points. Merge the adjacent mesh intervals if they satisfy the merger conditions.

Step 7. Set $M \overset{=}{=} M + 1$, and return to Step 2.

4. Simulation Analysis

The order and intervals of the method in [12] are fixed in each simulation, while those of the method described in this paper are variable. For the convenience of narration, the mesh refinement method in [12] is called FMRM (fixed mesh refinement method), and the method in Section 3 is called AMIM (adaptive mesh iteration method). The term M denotes the mesh refinement iteration, and $M = 0$ means the mesh initialization, and the term N and term K denote the total collocation points and interval number, respectively. The number of collocation points within each interval of the two methods is at least 2. The maximum of all mesh interval allowable error values is ε, where $\varepsilon = 10^{-6}$. When the mesh is initialization, the whole mesh is divided into 10 intervals, and each interval has a number of 2 collocation points. The value of term $r_{\max}$ is 1.2, and the maximum number of collocation points with each interval is 12. The penalty factor ρ is 100 [15]. The simulation results of this paper were performed on a 3.4 GHz Intel Core i7 CPU computer and MATLAB Version R2013.

Table 1: Hypersonic boost-glide vehicle date and physical constants.

Quantity	Numerical values
Mass (kg)	1200
W/S (kg/m^2)	420
R_e (m)	6.378×10^6
ρ_0 (kg/m^3)	1.225
μ (m^2/s^2)	3.986×10^{14}
β (m)	7200

Consider the following trajectory optimization problem taken from [16], and the state equations for a three-dimensional point mass vehicle which is commonly used in midcourse guidance systems of a Maneuverable Research Reentry Vehicle (MaRRV) are listed as follows:

$$\begin{aligned}
\dot{r} &= v \sin \gamma, \\
\dot{\phi} &= \frac{v \cos \gamma \sin \chi}{r \cos \theta}, \\
\dot{\theta} &= \frac{V \cos \gamma \cos \chi}{r}, \\
\dot{v} &= -\frac{D}{m} - g \sin \gamma, \\
\dot{\gamma} &= \frac{1}{v}\left[L \cos \beta - \left(g - \frac{v^2}{r}\right) \cos \gamma\right], \\
\dot{\chi} &= \frac{L \sin \beta}{m v \cos \gamma} + \frac{v}{r} \cos \gamma \sin \chi \tan \theta,
\end{aligned} \tag{20}$$

where r is the radial position and the terms ϕ and θ are the latitude and longitude, respectively. The term v is the total velocity, and the terms γ and χ denote the flight path angle and azimuth angle, respectively. The symbol m represents the vehicle mass and g is the gravity (μ/r^2 with μ being the gravitational parameter). The atmospheric density with altitude is assumed to be exponential model as

$$\rho(r) = \rho_0 e^{-(r - R_e)/H}, \tag{21}$$

where ρ_0 and H are constants in Table 1.

The lift and drag are defined as

$$\begin{aligned}
L &= q_d S C_L(\alpha), \\
D &= q_d S C_D(\alpha),
\end{aligned} \tag{22}$$

where

$$\begin{aligned}
C_L(\alpha) &= -0.034 + 0.93\alpha, \\
C_D(\alpha) &= 0.037 - 0.01\alpha + 0.736\alpha^2 + 0.936\alpha^3.
\end{aligned} \tag{23}$$

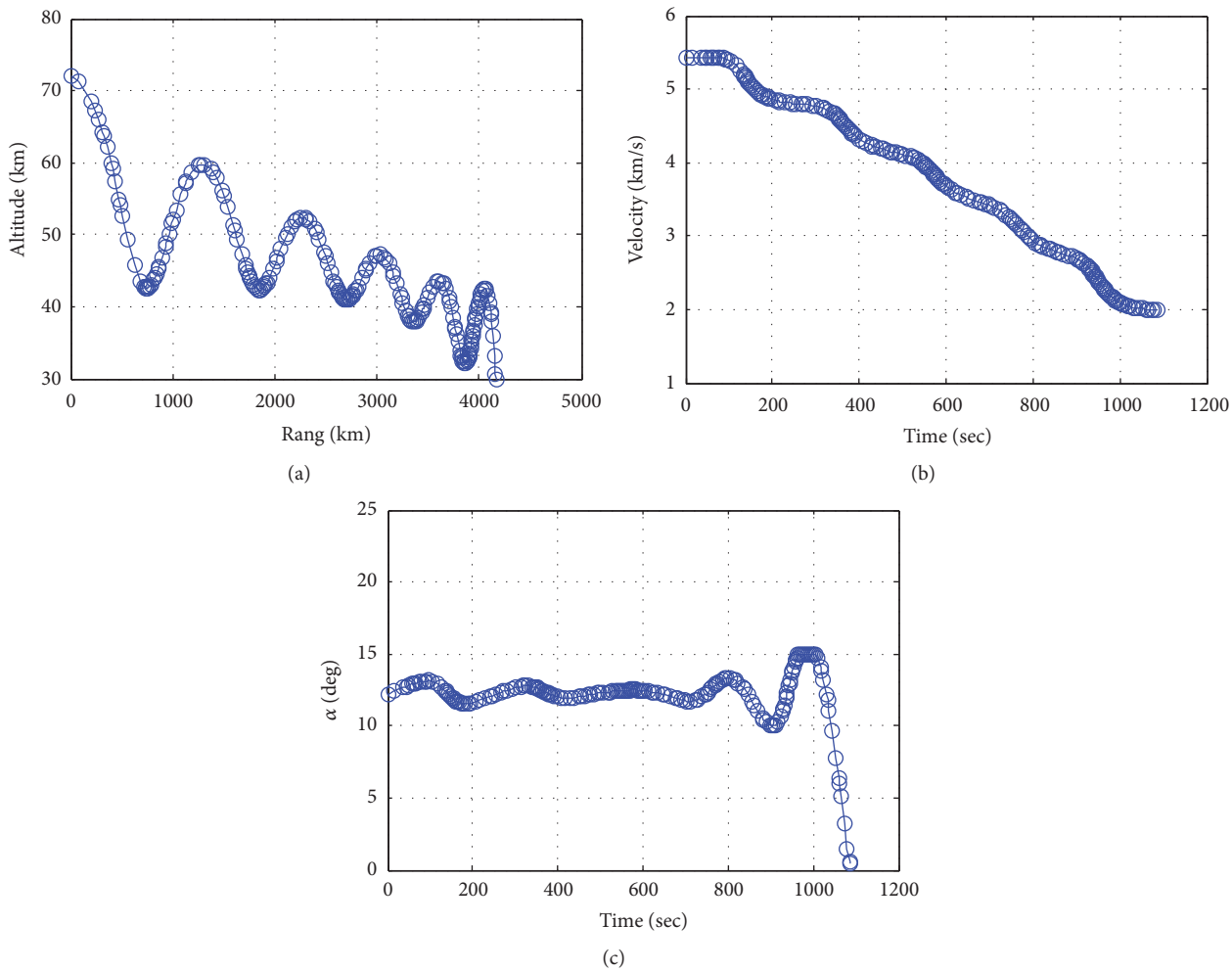

Figure 2: (a) Altitude versus range. (b) Velocity versus time. (c) Attack angle versus time.

The objective is to maximize the range, so the cost function is given by

$$J = \min\left\{-\theta_f\right\}. \tag{24}$$

Figures 2(a), 2(b), and 2(c) show the final solutions of the optimal control problem. It can be seen from Figures 2(a), 2(b), and 2(c) that the solution to the trajectory is relatively smooth, though there seems to be a rapid change in the attack angle in Figure 2(c) near $t = 1000$ s. It is noted that the control variable is limited to $[-5\,\text{deg}, 15\,\text{deg}]$ and that the solutions showed in Figure 2(c) satisfy that constraint. The results demonstrate the ability of approaches described in this paper for solving optimal control problems with inequality path constraints. In particular, it can be seen that the attack angle remains of 15 deg near $t = 1000$ s. One might suppose that if the attack angle is not constrained, its value would likely be larger than 15 deg.

Figure 3 shows the collocation points distribution of the solutions obtained using AMIM. When $M = 0$, the initial mesh is constructed. From $M = 0$ to $M = 2$, the mesh points are added largely, while the increment is relatively small from $M = 2$ to the iteration process terminated. As Figures 4(a) and 4(b) show about the trajectories iteration process and attack angle iteration process, respectively, it can be seen that

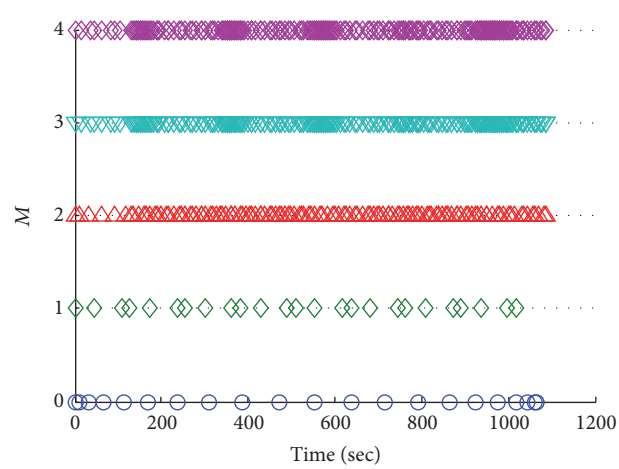

Figure 3: Mesh point history.

Table 2: Compare results using VOIand FOI methods.

Case	Methods	n	K	Mean times/s	N	M	Cost function
1	**AMRM**	**(2, 11)**	***65***	***6.3***	***214***	***4***	***−4165.3***
2	FMRM	5	50	*19.6*	*201*	*8*	*−4050.2*
3	FMRM	5	60	*33.8*	*241*	*7*	*−4103.7*
4	FMRM	5	65	*41.3*	*261*	*6*	*−4128.5*
5	FMRM	7	40	*25.2*	*241*	*7*	*−4085.2*
6	FMRM	7	45	*38.1*	*271*	*6*	*−4114.6*
7	**FMRM**	**7**	**55**	***47.8***	***331***	***5***	***−4163.4***
8	FMRM	9	25	*23.0*	*201*	*8*	*−4016.1*
9	FMRM	9	30	*36.4*	*241*	*7*	*−4073.9*
10	FMRM	9	35	*49.7*	*281*	*5*	*−4122.8*

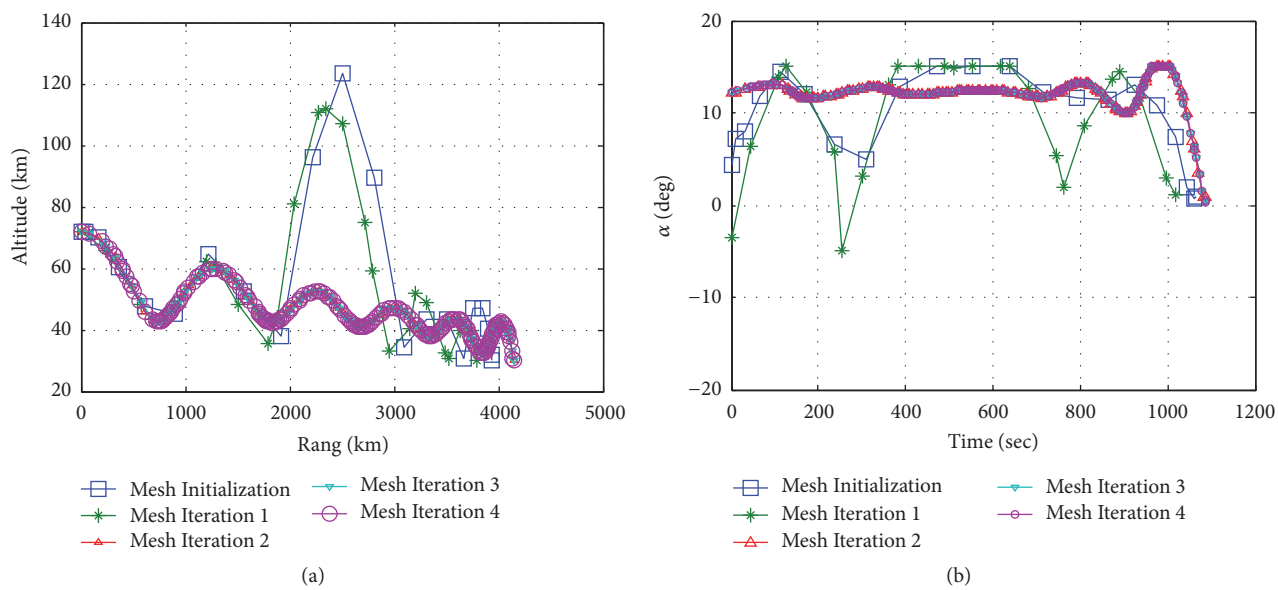

Figure 4: (a) Trajectories iteration. (b) Attack angle iteration.

there is a great difference from mesh initialization to Mesh Iteration 1 and Mesh Iteration 2 of the trajectories; moreover, the gap between Mesh Iteration 2 and Mesh Iterations 3–4 is very small, and the situation of attack angle iteration process shown in Figure 4(b) is the same as the trajectories process, which means that the solutions are gradually converged on the actual solution with each mesh refinement iteration. It is noted that a higher number of mesh points are needed to ensure the convergence of the solutions, and that explains why the increments in Figure 3 from mesh initialization to Mesh Iteration 2 and from Mesh Iteration 2 to Mesh Iteration 4 are different.

A comparison of the implementation of the AMRM and FMRM with different higher-order and intervals solutions are given in Table 2. Comparisons are made in terms of computation time, the number of total mesh points N and the number of mesh intervals K and the number of mesh refinement iterations M, and the cost function values. In each case, the initial guesses are randomized controls, with randomized state. A total of 100 samples are used to produce the results in this paper. The terminology AMRM (2, 11) refers to the AMRM where the number of mesh points within each interval can vary between 2 and 11; furthermore, the number of mesh intervals in AMRM can vary as well. All the simulations parameters are shown in black roman font in Table 2, and all the results are shown in black italics. As can be seen from the results listed in Table 2, the AMRM result in the smallest overall times compared with other cases. The reason is that the AMRM has the properties of reducing the unnecessary points and intervals, while FMRM (other cases) does not have this kind of property but only keeps the number of mesh points and intervals fixed until the simulation terminated. It is a fact that computation times mostly depend on the number of mesh refinement iterations and mesh size, while the growth of computation time for cases 4, 7, and 10 is due mostly to the increment number of mesh points and mesh intervals. Interestingly, the number of mesh intervals in case 1 is not set parameter but the result from the simulation, where the numbers of mesh intervals in cases 2–10 are set parameters. Case 7 using 7 mesh points

with 55 intervals gives a terminal cost function of −4163.4 km, which is the most optimality one in cases 2–9. The results show, as expected, that using larger mesh size results in improvements in accuracy at the expense of more runtime, due to the denser Jacobians. For FMRM, better accuracy can be achieved by increasing mesh points for smooth problems, while increasing the number of intervals to achieve better accuracy for nonsmooth problems [12]. However, it is a fact that smooth regions and nonsmooth regions together exist in one solution of the problem, so it is difficult to trade the number of mesh points and the number of mesh intervals when solving a complicated problem. The simulations show that the AMEM can trade the number of mesh points within intervals with the number of mesh intervals and obtain an accurate solution with a relatively small mesh size.

5. Conclusions

This paper develops an adaptive mesh iteration method for solving optimal control problems. The method has the ability to trade the number of mesh points with number of mesh intervals compared to other mesh refinement methods. The mesh size is guided by a previously derived maximum relative error and the ratio of the maximum to the mean curvature of the mesh intervals. The number of mesh intervals is increased in regions where solution is nonsmooth, while the mesh points increased in regions where the solution is smooth. Furthermore, the mesh size can be decreased either by reducing the mesh points or by combining adjacent mesh intervals which share the same number of mesh points. The method applied successfully trajectory optimization of MaRRV from the open literature and compared with other mesh refinement methods. The simulation results demonstrate the superiority of the AMRM.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by National Natural Science Foundation of China (61573374, 61503408).

References

[1] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, 1998.

[2] I. M. Ross and F. Fahroo, "A perspective on methods for trajectory optimization," in *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, pp. 1–7, AIAA Press, Monterey, CA, USA, 2002.

[3] G. Li and X. Liu, "Optimal control problems with incomplete and different integral time domains in the objective and constraints," *Nonlinear Dynamics*, vol. 78, no. 4, pp. 2883–2895, 2014.

[4] G. Elnagar, M. A. Kazemi, and M. Razzaghi, "The pseudospectral Legendre method for discretizing optimal control problems," *IEEE Transactions on Automatic Control*, vol. 40, no. 10, pp. 1793–1796, 1995.

[5] A. L. Herman and B. A. Conway, "Direct optimization using collocation based on high–order Gauss–Lobatto quadrature rules," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 3, pp. 592–599, 1996.

[6] P. E. Gill, E. Wong, W. Murray, and M. A. Saunders, "User's guide for SNOPT Version 7.4: software for large–scale nonlinear programming," Numerical Analysis Report, Department of Mathematics, University of California, San Diego, CA, USA, 2005.

[7] F. Fahroo and I. M. Ross, "Costate estimation by a Legendre pseudospectral method," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 2, pp. 270–277, 2001.

[8] I. M. Ross and F. Fahroo, "Legendre pseudospectral approximations of optimal control problems," in *Lecture Notes in Control and Information Sciences*, vol. 295, pp. 327–342, Springer–Verlag, Berlin, Germany, 2003.

[9] G. W. Reddien, "Collocation at gauss points as a discretization in optimal control," *SIAM Journal of Control and Optimization*, vol. 17, no. 2, pp. 1–6, 1979.

[10] G. T. Huntington, D. Benson, and A. V. Rao, "A comparison of accuracy and computational efficiency of three pseudospectral methods," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, pp. 1–25, AIAA Press, South Carolina, SC, USA, 2007.

[11] I. M. Ross and M. Karpenko, "A review of pseudospectral optimal control: from theory to flight," *Annual Reviews in Control*, vol. 36, no. 2, pp. 182–197, 2012.

[12] P. Williams, "Hermite–legendre–gauss–lobatto direct transcription in trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 4, pp. 1392–1395, 2009.

[13] H. M. Lei, T. Liu, J. Li, and Z. P. Jiang, "Adaptive mesh refinement of hp pseudospectral method using size reduction," *Control Theory and Applications*, vol. 33, no. 8, pp. 1061–1067.

[14] D. Garg, M. A. Patterson, C. Francolin et al., "Direct trajectory optimization and costate estimation of finite–horizon and infinite–horizon optimal control problems using a Radau pseudospectral method," *Computational Optimization and Applications*, vol. 49, no. 2, pp. 335–358, 2011.

[15] Y. Q. Hu, X. G. Liu, and A. K. Xue, "A penalty method for solving inequality path constrained optimal control problems," *Acta Automatica Sinica*, vol. 39, no. 12, pp. 1996–2001, 2013.

[16] I. R. S. Tauqeer ul, L. S. He, and Naseemullah., "Vehicle performance tradeoff study for a small size lifting reentry vehicle," in *Proceedings of 2013 10th International Bhurban Conference on Applied Sciences & Technology*, Islamabad, Pakistan, 2013.