

Sequential convex programming method using adaptive mesh refinement for entry trajectory planning problem



Xiang Zhou^a, Rui-Zhi He^a, Hong-Bo Zhang^{a,*}, Guo-Jian Tang^a, Wei-Min Bao^b

^a College of Aerospace Science and Engineering, National University of Defense Technology, Changsha 410073, China

^b China Aerospace Science and Technology Corporation, Beijing 100048, China

ARTICLE INFO

Article history:

Received 1 July 2020

Received in revised form 13 October 2020

Accepted 18 November 2020

Available online 26 November 2020

Communicated by Xingqun Zhan

Keywords:

Adaptive mesh refinement

Sequential convex programming

Convergence condition

Entry trajectory planning

ABSTRACT

Sequential convex programming method is one of the potential approaches to make trajectory generation onboard. However, the contradiction between the solution accuracy and the computational efficiency restricts its application. To overcome the difficulty, a novel sequential convex programming method based on customized adaptive mesh refinement is proposed in this paper. The main contribution includes two aspects. Firstly, a customized adaptive mesh refinement method is developed. The mesh points are adjusted adaptively based on the linearization error after each iteration, and thus the number of mesh points is changed. Secondly, an unconventional convergence condition is proposed, which can make the sequential convex programming method converges to a feasible solution of the original problem with fewer iterations. Taking the entry trajectory planning problem as an example, the simulation results show that the proposed method can decrease the number of mesh points while ensuring the feasibility of converged solution. Due to the fewer iterations, the computational time to solve the original problem is decreased. As a result, a good balance between solution accuracy and computational efficiency is achieved.

© 2020 Elsevier Masson SAS. All rights reserved.

1. Introduction

Entry trajectory planning is an essential procedure to achieve high precision guidance for entry vehicle with high lift-drag ratio [1,2]. How to solve the problem with higher accuracy and shorter computational time is a research focus in this field, which is also the basic technique of advanced guidance via trajectory generation onboard [3,4]. Because of the poor adaptability to different problems and the high sensibility for initial guess of the adjoint variables, the indirect method is hard to be applied for trajectory generation onboard. Generally, the entry trajectory planning problem is formulated as a continuous-time optimal control problem, and the direct method converts the continuous problem into a finite-dimensional nonlinear programming problem (NLP problem) by discretization, where a discrete converged solution is approximated as the solution of the original problem [5]. If the NLP problem has a convex formulation, and the number of discrete points is low, the direct method can solve the original problem with a higher computational efficiency [6]. In general, the direct

method for solving the entry trajectory planning problem includes the search method based on drag acceleration profile [7,8], the pseudospectral method [9,10], the convex programming method and so on.

Because of the good theoretical properties and high computation efficiency, in recent years, the convex programming method has attracted much attention of researchers [11–15]. For some highly nonlinear optimal control problems, the successive linearization technique is applied to convexify the highly nonlinear terms in the original problem. Thus, a sequence of convex subproblems are solved iteratively, and the corresponding method is known as the sequential convex programming method (SCP method) [16]. In Ref. [17–20], the different trajectory planning problems are solved by the SCP method, including orbit transfers, aerocapture trajectory planning, rocket landing, optimal rendezvous of unmanned vehicles and so on, which all have highly nonlinear terms. From Ref. [17,18], the SCP method has a higher computational efficiency under the same optimization results, compared with the pseudospectral method, which makes it have a potential for onboard application. Obviously, the converged solution of the SCP method is referred as the solution of the original problem [21]. Therefore, the convergence of the SCP method

* Corresponding author.

E-mail address: zhanghb1304@nudt.edu.cn (H.-B. Zhang).

is important to solve successfully the highly nonlinear trajectory planning problem. There are some influence issues of convergence, such as the initial trajectory, the radius of trust region constraint, the formulation of convex optimization subproblem and so on. In Ref. [21,22], several safe-guarding techniques are incorporated into the SCP method, namely virtual control and trust regions, which add another layer of algorithmic robustness, and are helpful the convergence of SCP method. In Ref. [23], a relaxation coefficient about trust region constraint is introduced to improve the numerical stability of the SCP method. In Ref. [24], two improved SCP methods are developed: line-search SCP and trust-region SCP, which improve the performance of the SCP method for entry trajectory planning. In Ref. [25,26], the SCP method has been applied to generate the reference trajectory online, and a better guidance performance is realized. The existed SCP methods mainly focus on how to improve the solution accuracy, but there is less research on improving computational efficiency, which is also an important measurement of the SCP method, and be easily effected by the number of mesh points.

Discretization has a greater effect on the solution accuracy and the computational efficiency [5,27]. Discretization consists of expressing the objective function, dynamical equations, path constraints as functions of a finite number of parameters denoted on a set of discrete points, which are also called mesh points. The number of mesh points determines the scale of convex subproblems to be solved, and the density of mesh points determines the approximation error of the discrete solution. Thus, a fine mesh is very important for improving computational efficiency and guarantying solution accuracy of the original problem [28].

For the continuous-time optimal control problem, the uniform mesh is used most widely, where the interval between two adjacent mesh points is constant. In Ref. [17,18,26], the uniform mesh is applied to discretize the continuous entry trajectory planning problem, and is invariable in the iteration process of the SCP method. The uniform mesh is very convenient to implement, but its disadvantage is the high computational burden when the high accuracy solution is desired. For the continuous-time highly nonlinear problem, the successive linearization and discretization process result in the approximation error [26]. The error is not uniform at all mesh points, and it can be adjusted to improve solution accuracy by varying the density of mesh points. In uniform mesh, the approximated error depends on the number of mesh points, and an accurate solution requires a larger number of mesh points. If the discretized convex problem has a very large scale, the computational efficiency will become very low. Therefore, the contradiction between the solution accuracy and computational efficiency is hard to overcome for the SCP method with uniform mesh.

Motivated by the above analysis, adaptive mesh refinement method has been developed to adjust the density of mesh points adaptively for balancing the contradiction. The prior direct methods in conjunction with adaptive mesh refinement can provide a highly accurate solution using the low computational burden for several complex trajectory planning problems [29–31], such as the hp adaptive pseudospectral method, where hp collocation methods are used in adaptive mesh refinement. In hp methods, the mesh points are chosen to be the roots of an orthogonal polynomial [32]. When an iteration solution is obtained at one iteration, the error at each mesh point is determined by the difference between an approximation of the time derivative of the state and the right-hand side of the dynamics equations. In the hp adaptive method, the accuracy is improved by adjusting the number of mesh intervals and the degree of the approximating polynomial in each mesh interval [33]. In Ref. [34–36], the pseudospectral-convex programming method is proposed for powered landing problem, and the method represents a good trade-off between solution accuracy

and computational efficiency. However, the method is only applied for the problem, where the original dynamics are linear and the path constraints are convex, so the SCP method is not necessary. In Ref. [37], a new drag-energy scheme, based on the use of pseudospectral method and SCP method, is proposed to solve the entry trajectory planning problem, which enables the capability to recompute the trajectories in real-time. But the non-uniform mesh in Ref. [37] is fixed in the iteration process, which cannot be adjusted adaptively according the current solution. Different from the pseudospectral-convex programming method, Ref. [38] proposes an adaptive mesh refinement method to solve Mars entry trajectory optimization problem, which inserts points by analyzing interpolation errors and removes points by using generalized data compression. The method discretizes an original continuous-time problem into a nonlinear programming problem, but the solution accuracy is found to be comparable with that obtained from indirect methods, and the computational efficiency is higher.

To the best of our knowledge, the SCP method using fixed and uniform mesh is widely used for highly nonlinear trajectory planning problem. The SCP method using the non-uniform mesh based on the pseudospectral method is also proposed recently. However, the density of mesh points is not adjusted adaptively in the iteration process, which means that the non-uniform mesh is unchanged for the SCP method. If the prescribed non-uniform mesh is bad, the computational efficiency and solution accuracy of the SCP method cannot be guaranteed.

According to the above analysis, a novel sequential convex programming method based on the customized adaptive mesh refinement is proposed to solve the entry trajectory planning problem in this paper. Firstly, the original entry trajectory planning problem is transformed to a convex programming problem by linearization; Secondly, a customized adaptive mesh refinement method is proposed, where the linearization error at each mesh point is quantified by the local integral. Thus, in the region with high error, the density of mesh point is increased to improve the accuracy, and in the region with low error, the density is decreased to improve the efficiency; Finally, a new convergence condition is defined, which can make the SCP method converge a feasible solution of the original problem with fewer iterations. Numerical simulations reveal that the proposed method can reduce the computational time to solve the original problem while ensuring the feasibility of the converged solution.

The remainder of the present paper is organized as follows. Section 2 states the two formulations of entry trajectory planning problem, including the original form and the convex form. Section 3 introduces the non-uniform discretization and the customized adaptive mesh refinement method. Section 4 introduces the solution procedure of the SCP method using proposed adaptive mesh refinement. In Section 5, numerical results are presented to demonstrate the effectiveness and features of the proposed method. Finally, the conclusions of the work are summarized in Section 6.

2. Problem formulation and convexification

2.1. Problem formulation

(1) Entry Dynamics

The three-dimensional motion model with time as an independent variable is established considering the rotation of the earth, which is given by

$$\begin{cases} \frac{dr}{dt} = V \sin \theta \\ \frac{d\lambda}{dt} = \frac{V \cos \theta \sin \sigma}{r \cos \phi} \\ \frac{d\phi}{dt} = \frac{V \cos \theta \cos \sigma}{r} \\ \frac{dV}{dt} = -D - g \sin \theta + \tilde{C}_V \\ \frac{d\theta}{dt} = \frac{L \cos \nu}{V} - \frac{g \cos \theta}{V} + \frac{V \cos \theta}{r} + C_\theta + \tilde{C}_\theta \\ \frac{d\sigma}{dt} = \frac{L \sin \nu}{V \cos \theta} + \frac{V \tan \phi \cos \theta \sin \sigma}{r} + C_\sigma + \tilde{C}_\sigma \end{cases} \quad (1)$$

where r is the radial distance from Earth's center to the mass center of the vehicle, λ is the longitude, ϕ is the latitude, V is the Earth-relative velocity (i.e., the velocity of the entry vehicle relative Earth), θ is the flight-path angle (FPA), σ is the heading angle of the relative velocity vector (HA), measured clockwise from the north, and ν is the bank angle. The r is scaled by the radius of the Earth $R_0 = 6378.137$ km, and the V is scaled by $\sqrt{g_0 R_0}$ where g_0 is the Earth gravitational acceleration at R_0 . The physical time t is scaled by $\sqrt{R_0/g_0}$.

The lift acceleration L and drag acceleration D are scaled by g_0 , which are expressed as follows

$$\begin{cases} L = \frac{1}{2M} \rho V^2 S_r C_L \\ D = \frac{1}{2M} \rho V^2 S_r C_D \end{cases} \quad (2)$$

where M is the mass of the vehicle; S_r is the reference area of the vehicle; C_L and C_D are the lift and drag coefficients respectively which are functions of the angle of attack and Mach number in this paper.

In Eq. (1), the Coriolis accelerations C_σ and C_θ , and the centrifugal accelerations \tilde{C}_σ , \tilde{C}_θ and \tilde{C}_V are given by

$$\begin{cases} C_\sigma = 2\omega_e (\sin \phi - \cos \sigma \tan \theta \cos \phi) \\ \tilde{C}_\sigma = \frac{\omega_e^2 r \cos \phi \sin \phi \sin \sigma}{V \cos \theta} \\ C_\theta = 2\omega_e \sin \sigma \cos \phi \\ \tilde{C}_\theta = \frac{\omega_e^2 r}{V} \cos \phi (\sin \phi \sin \theta \cos \sigma + \cos \phi \cos \theta) \\ \tilde{C}_V = \omega_e^2 r (\cos^2 \phi \sin \theta - \cos \phi \sin \phi \cos \sigma \cos \theta) \end{cases} \quad (3)$$

where the Earth self-rotation rate ω_e is scaled by $\sqrt{g_0/R_0}$.

In this paper, the atmospheric density model with exponent function is used, which is given by

$$\rho = \rho_0 e^{-h/h_s} \quad (4)$$

where $h_s = 7110$ m, $\rho_0 = 1.225$ kg/m³, h is the altitude.

(2) Control Variables

In this paper, the angle of attack is fixed as a piecewise linear function, which is given in Ref. [39]. Thus, the only control variable is the bank angle, which is less than 90° and larger than -90° generally. The amplitude constraint about the bank angle is given by

$$v_{\min} \leq |\nu| \leq v_{\max} \quad (5)$$

where $0 \leq v_{\min} < v_{\max} \leq 90^\circ$.

(3) Path Constraints

To meet the safety requirements of the entry vehicle and its facilities inside, there are inequality path constraints to be considered. The path constraints typically include the dynamic pressure q , aerodynamic load factor n , and heating rate \dot{Q} , which are respectively given as

$$\dot{Q} = k_Q \sqrt{g_0 R_0}^{3.15} \sqrt{\rho} V^{3.15} \leq \dot{Q}_{\max} \quad (6)$$

$$q = 0.5 g_0 R_0 \rho V^2 \leq q_{\max} \quad (7)$$

$$n = \sqrt{L^2 + D^2} \leq n_{\max} \quad (8)$$

where $k_Q = 9.4369 \times 10^{-5}$ in this paper, q_{\max} , n_{\max} and \dot{Q}_{\max} are the maximum allowable dynamic pressure, aerodynamic load, and heating rate, respectively.

(4) Boundary Constraints and Objective Function

In this paper, the boundary constraints are given by

$$\begin{cases} r(t_0) = r_0, \lambda(t_0) = \lambda_0, \phi(t_0) = \phi_0 \\ V(t_0) = V_0, \theta(t_0) = \theta_0, \sigma(t_0) = \sigma_0 \\ V(t_f) = V_f^* \end{cases} \quad (9)$$

where t_0 and t_f are the initial and terminal time respectively, V_f^* is the desired terminal velocity.

In this paper, the terminal velocity constraint is only enforced as an equality constraint, which is given in Eq. (9), but the terminal longitude, latitude and altitude constraints are represented by the three deviation relaxation terms, i.e. $|\lambda_f - \lambda_f^*|$, $|\phi_f - \phi_f^*|$ and $|r_f - r_f^*|$, which are augmented to the objective function J_0 defined in Eq. (10). The relaxation approach for terminal constraint has the advantage of ensuring that there is a feasible solution space for the original problem. If the three position constraints are directly added to the original problem as equality constraints, it is likely that there is no feasible solution to the original problem, and the planning method fails. Because the terminal position is beyond the reachable region of the vehicle. When the terminal position is located in the reachable region, the optimal value of the three relaxation terms in the augmented objective function will be zero, which is equivalent to the result of applying equality constraints about the three terminal positions. However, when the terminal position exceeds the reachable region of the vehicle, the optimal value of the three relaxation terms will not be zero. In this case, the optimal result is to find a trajectory that the terminal position is closest to the desired value.

$$J_0 = C_1 (|\lambda_f - \lambda_f^*| + |\phi_f - \phi_f^*|) + C_2 |r_f - r_f^*| \quad (10)$$

where λ_f^* is the desired terminal longitude, ϕ_f^* is the desired terminal latitude and r_f^* is the desired radial distance, which is non-dimensional. The C_1 and C_2 are weighted coefficients. When choosing the weight coefficient in this paper, there are two aspects to be mainly considered. Firstly, the numerical difference of each term in the objective function should not be too large. Secondly, the weight should be determined according to the priority of each term in the optimization. In order to meet the terminal location requirement, the longitude and latitude should be close to the expected value at the same time, so their coefficients should be the same to indicate the same importance or priority in the objective function.

Note that the terminal constraints of flight path angle and heading angle are not considered in this paper, because it is time consuming to determinate the appropriate desired values of the two constraints. The choice of the two terminal angles depends on the requirement of terminal guidance. The terminal guidance is not

considered, so the two terminal angles are unconstrained. If after considering the terminal constraints of the two angles, the solution space of the original problem is non-empty and continuous, and thus the following analysis is not affected.

Therefore, the original entry trajectory planning problem **P1** is formulated as a continuous-time optimal control problem, which is given as follows

$$\begin{aligned} \mathbf{P1}: \min_{\boldsymbol{v}} \quad & J_0 \\ \text{subject to: } & \text{Eq. (1)} \\ & v_{\min} \leq |v| \leq v_{\max} \\ & q \leq q_{\max}, n \leq n_{\max}, \dot{Q} \leq \dot{Q}_{\max} \\ & \begin{cases} r(t_0) = r_0, \lambda(t_0) = \lambda_0, \phi(t_0) = \phi_0 \\ V(t_0) = V_0, \theta(t_0) = \theta_0, \sigma(t_0) = \sigma_0 \\ V(t_f) = V_f^* \end{cases} \end{aligned} \quad (11)$$

2.2. Convexification

In a convex optimization problem, the objective function and inequality constraints must be convex or linear, and the equality constraints must be linear. Obviously, the problem **P1** is not a convex optimization problem, because of the highly nonlinear dynamics equations and path constraints. The convexification of problem **P1** is introduced below.

(1) New Controls

Motivated by Ref. [16], the change rate of bank angle is defined as the new controls in this paper, which is given as follows

$$u = \dot{\psi} \quad (12)$$

Thus, the amplitude constraint of the new controls is given by

$$\dot{\psi}_{\min} \leq u \leq \dot{\psi}_{\max} \quad (13)$$

where the minimum value $\dot{\psi}_{\min}$ and the maximum value $\dot{\psi}_{\max}$ of new controls are constants.

(2) Linearization of Dynamics Equations

Considering the new controls, the nonlinear dynamics in Eq. (1) is rewritten as

$$\dot{\boldsymbol{x}}(t) = \mathbf{f}(\boldsymbol{x}, t) + \mathbf{B}u + \mathbf{h}(\boldsymbol{x}, t) \quad (14)$$

where \boldsymbol{x} is the state vector, $\boldsymbol{x} = [r \ \lambda \ \phi \ V \ \theta \ \sigma \ v]^T$, and

$$\mathbf{f}(\boldsymbol{x}, t) = \begin{bmatrix} V \sin \theta \\ V \cos \theta \sin \sigma / (r \cos \phi) \\ V \cos \theta \cos \sigma / r \\ -D - g \sin \theta \\ -g \cos \theta / V + V \cos \theta / r \\ V \tan \phi \cos \theta \sin \sigma / r \\ 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (15)$$

$$\mathbf{h}(\boldsymbol{x}, t) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \tilde{C}_V \\ C_\theta + \tilde{C}_\theta \\ C_\sigma + \tilde{C}_\sigma \\ 0 \end{bmatrix} \quad (16)$$

In general, the initial time t_0 is fixed, but the terminal time t_f is free for most optimal control problems. In this paper, the

variable physical time interval $[t_0, t_f]$ is transformed to the fixed time interval $[0, 1]$ via the affine transformation $\tau = (t - t_0)/(t_f - t_0)$. The τ is called the mesh time, and has a domain $\tau \in [0, 1]$.

Then, applying the time interval transformation for Eq. (14), we have

$$\dot{\boldsymbol{x}}(\tau) = \xi (\mathbf{f}(\boldsymbol{x}, \tau) + \mathbf{B}u(\tau) + \mathbf{h}(\boldsymbol{x}, \tau)) \quad (17)$$

where the ξ is the time expand factor. The variable physical time interval $[t_0, t_f]$ is scaled by the new introduced variable ξ . For the convenience of the following description, define $\mathbf{F}(\boldsymbol{x}, \mathbf{u}, \tau) = \mathbf{f}(\boldsymbol{x}, \tau) + \mathbf{B}u(\tau) + \mathbf{h}(\boldsymbol{x}, \tau)$.

Compared with other terms in Eq. (17), the $\mathbf{h}(\boldsymbol{x}, \tau)$ is a small term, so the $\mathbf{h}(\boldsymbol{x}, \tau)$ is approximated directly by $\mathbf{h}(\boldsymbol{x}^{(k)}, \tau)$. Using the first order Taylor expansion, Eq. (17) is linearized about a reference trajectory, which is denoted as $(\boldsymbol{x}^{(k)}, \mathbf{u}^{(k)}, \xi^{(k)})$, and the linearized dynamics are obtained as follows

$$\begin{aligned} \dot{\boldsymbol{x}}(\tau) = & \xi^{(k)} \mathbf{A}(\boldsymbol{x}^{(k)}, \tau) \boldsymbol{x}(\tau) + \xi^{(k)} \mathbf{B}(\boldsymbol{x}^{(k)}, \tau) \mathbf{u}(\tau) \\ & + \mathbf{F}(\boldsymbol{x}^{(k)}, \mathbf{u}^{(k)}, \tau) \xi - \xi^{(k)} (\mathbf{A}\boldsymbol{x}^{(k)} + \mathbf{B}\mathbf{u}^{(k)}) \end{aligned} \quad (18)$$

where the superscript k represents the index of reference trajectories, and the expressions of \mathbf{A} are given as follows

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & a_{14} & a_{15} & 0 & 0 \\ a_{21} & 0 & a_{23} & a_{24} & a_{25} & a_{26} & 0 \\ a_{31} & 0 & 0 & a_{34} & a_{35} & a_{36} & 0 \\ a_{41} & 0 & 0 & a_{44} & a_{45} & 0 & 0 \\ a_{51} & 0 & 0 & a_{54} & a_{55} & 0 & a_{57} \\ a_{61} & 0 & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (19)$$

$$\begin{cases} a_{21} = -\frac{V^{(k)} \cos \theta^{(k)} \sin \sigma^{(k)}}{(r^{(k)})^2 \cos \phi^{(k)}} \\ a_{31} = -\frac{V^{(k)} \cos \theta^{(k)} \cos \sigma^{(k)}}{(r^{(k)})^2} \\ a_{41} = -D_r^{(k)} - g_r^{(k)} \sin \theta^{(k)} \\ a_{51} = \frac{L_r^{(k)} \cos \nu^{(k)}}{V^{(k)}} - \frac{g_r^{(k)} \cos \theta^{(k)}}{V^{(k)}} - \frac{V^{(k)} \cos \theta^{(k)}}{(r^{(k)})^2} \\ a_{61} = \frac{L_r^{(k)} \sin \nu^{(k)}}{V^{(k)} \cos \theta^{(k)}} - \frac{V^{(k)} \cos \theta^{(k)} \sin \sigma^{(k)} \tan \phi^{(k)}}{(r^{(k)})^2} \end{cases} \quad (20)$$

$$\begin{cases} a_{23} = \frac{V^{(k)} \cos \theta^{(k)} \sin \sigma^{(k)} \sin \phi^{(k)}}{r^{(k)} \cos^2 \phi^{(k)}} \\ a_{63} = \frac{V^{(k)} \cos \theta^{(k)} \sin \sigma^{(k)}}{r^{(k)} \cos^2 \phi^{(k)}} \end{cases} \quad (21)$$

$$\begin{cases} a_{14} = \sin \theta^{(k)} \\ a_{24} = \frac{\cos \theta^{(k)} \sin \sigma^{(k)}}{r^{(k)} \cos \phi^{(k)}} \\ a_{34} = \frac{\cos \theta^{(k)} \cos \sigma^{(k)}}{r^{(k)}} \\ a_{44} = -D_V^{(k)} \\ a_{54} = \frac{\cos \nu^{(k)} (L_V^{(k)} V^{(k)} - L^{(k)})}{(V^{(k)})^2} + \frac{g \cos \theta^{(k)}}{(V^{(k)})^2} + \frac{\cos \theta^{(k)}}{r^{(k)}} \\ a_{64} = \frac{\sin \nu^{(k)} (L_V^{(k)} V^{(k)} - L^{(k)})}{(V^{(k)})^2 \cos \theta^{(k)}} + \frac{\cos \theta^{(k)} \sin \sigma^{(k)} \tan \phi^{(k)}}{r^{(k)}} \end{cases} \quad (22)$$

$$\begin{cases} a_{15} = V^{(k)} \cos \theta^{(k)} \\ a_{25} = -\frac{V^{(k)} \sin \theta^{(k)} \sin \sigma^{(k)}}{r^{(k)} \cos \phi^{(k)}} \\ a_{35} = -\frac{V^{(k)} \sin \theta^{(k)} \cos \sigma^{(k)}}{r^{(k)}} \\ a_{45} = -g^{(k)} \cos \theta^{(k)} \\ a_{55} = \frac{g^{(k)} \sin \theta^{(k)}}{V^{(k)}} - \frac{V^{(k)} \sin \theta^{(k)}}{r^{(k)}} \\ a_{65} = \frac{L^{(k)} \sin \nu^{(k)} \sin \theta^{(k)}}{V^{(k)} \cos^2 \theta^{(k)}} - \frac{V^{(k)} \sin \theta^{(k)} \sin \sigma^{(k)} \tan \phi^{(k)}}{r^{(k)}} \end{cases} \quad (23)$$

$$\begin{cases} a_{26} = \frac{V^{(k)} \cos \theta^{(k)} \cos \sigma^{(k)}}{r^{(k)} \cos \phi^{(k)}} \\ a_{36} = -\frac{V^{(k)} \cos \theta^{(k)} \sin \sigma^{(k)}}{r^{(k)}} \\ a_{66} = \frac{V^{(k)} \cos \theta^{(k)} \cos \sigma^{(k)} \tan \phi^{(k)}}{r^{(k)}} \end{cases} \quad (24)$$

$$\begin{cases} a_{57} = -\frac{L^{(k)} \sin \nu^{(k)}}{V^{(k)}} \\ a_{67} = \frac{L^{(k)} \cos \nu^{(k)}}{V^{(k)} \cos \theta^{(k)}} \end{cases} \quad (25)$$

where

$$\begin{cases} D_r^{(k)} = -\beta D^{(k)} \\ D_V^{(k)} = \frac{2D^{(k)}}{V^{(k)}} \end{cases}, \quad \begin{cases} L_r^{(k)} = -\beta L^{(k)} \\ L_V^{(k)} = \frac{2L^{(k)}}{V^{(k)}} \end{cases}. \quad (26)$$

$$g^{(k)} = \frac{1}{(r^{(k)})^2}, g_r^{(k)} = -\frac{2}{(r^{(k)})^3} \quad (27)$$

and $\beta = R_0/h_s$.

Define $\mathbf{C} = \mathbf{F}(\mathbf{x}^{(k)}, \mathbf{u}^{(k)}, \tau)$ and $\mathbf{D} = -(\mathbf{A}\mathbf{x}^{(k)} + \mathbf{B}\mathbf{u}^{(k)})$, and we rewrite Eq. (18) as follows

$$\dot{\mathbf{x}}(\tau) = \xi^{(k)} \mathbf{A} \cdot \mathbf{x}(\tau) + \xi^{(k)} \mathbf{B} \cdot \mathbf{u}(\tau) + \mathbf{C} \cdot \xi + \mathbf{D} \cdot \xi \quad (28)$$

Along with the linearization of dynamics, a trust-region constraint is enforced and given as follows

$$|\mathbf{x} - \mathbf{x}^{(k)}| \leq \delta \quad (29)$$

where the “ \leq ” is a component-wise inequality, such as $|x_i - x_i^{(k)}| \leq \delta_i$, δ_i is the radius of trust-region constraints, and is a user-defined sufficient large constant, $i = 1, 2, \dots, 7$. The trust-region constraint has been demonstrated to be very effective in improving the convergence of SCP method.

(3) Linearization of Path Constraints

The highly nonlinear path constraints in Eq. (6)-Eq. (8) are linearized about the same reference trajectory. Because the three constraints are functions of r and V , the three linearized path constraints have the following linearized form

$$f_i(r^{(k)}, V^{(k)}) + f'_i(r^{(k)}, V^{(k)})[r - r^{(k)}; V - V^{(k)}] \leq f_{i,\max}, \quad i = 1, 2, 3 \quad (30)$$

where the subscript i represents the index of three path constraints. The f_i is defined as the left-hand side of Eq. (6)-Eq. (8), and the $f_{i,\max}$ represents the maximum value of three path constraints, and is a constant. The expressions of f'_i are given as follows

$$f'_1(r^{(k)}, V^{(k)}) = \begin{bmatrix} -\beta \frac{f_1}{2} & 0 \\ 0 & \frac{m}{V^{(k)}} \end{bmatrix} \quad (31)$$

$$f'_2(r^{(k)}, V^{(k)}) = \begin{bmatrix} -\beta f_2 & 0 \\ 0 & \frac{2f_2}{V_k} \end{bmatrix} \quad (32)$$

$$f'_3(r^{(k)}, V^{(k)}) = \frac{S_r}{Mg_0} \sqrt{C_L^2 + C_D^2} \begin{bmatrix} -\beta f_2 & 0 \\ 0 & \frac{2f_2}{V_k} \end{bmatrix} \quad (33)$$

where $m = 3.15$.

After convexification, a convex optimization problem **P2** is formulated, which is given as follows

$$\begin{aligned} \textbf{P2}: \min_{u, \eta} \quad & J = J_0 + C_3 \eta + C_4 \int_0^1 |u| d\tau \\ \text{subject to:} \quad & \text{Eq. (19)} \\ & |\mathbf{x} - \mathbf{x}^{(k)}| \leq \eta \delta \\ & \dot{v}_{\min} \leq u \leq \dot{v}_{\max} \\ & f_i(r^{(k)}, V^{(k)}) + f'_i(r^{(k)}, V^{(k)})[r - r^{(k)}; V - V^{(k)}] \\ & \leq f_{i,\max}, \quad i = 1, 2, 3 \\ & \mathbf{x}(\tau = 0) = [r_0 \ \lambda_0 \ \phi_0 \ V_0 \ \theta_0 \ \sigma_0 \ v_0]^T \\ & V(\tau = 1) = V_f^* \end{aligned} \quad (34)$$

In the problem **P2**, the relaxation coefficient η is introduced into the trust-region constraint, which is augmented to the objective function for reducing the optimization space and improve the convergence speed. The integral term about $|u|$ in objective function J is the main performance index, which can eliminate the oscillation of bank angle and make controls smoother. The C_3 and C_4 are weighted coefficients.

In the combined performance index of the problem **P2**, the term J_0 is primarily intended to ensure that the terminal position constraints are satisfied, including the longitude, the latitude and the altitude. The index J that really needs to be optimized is the integral term $\int_0^1 |u| d\tau$. The reason of the treatment is to increase the feasible solution space in problem formulation, so as to avoid the convergence difficulty caused by the narrow or nonexistent solution space of the original problem.

3. Discretization and adaptive mesh refinement

3.1. Discretization

The problem **P2** is a continuous-time optimal control problem, where the control variables and state variables are all continuous. For numerically solution, in this section, the continuous variables will be discretized with a kind of mesh in the domain $[0, 1]$, and all constraints are enforced at each mesh point. A mesh represents a distribution of mesh points or discrete points. After discretization, the continuous-time problem **P2** is transformed to a convex programming problem with finite optimization variables, which can be solved with high accuracy in polynomial time.

For the direct method, the mesh has a main effect on the solution accuracy and computation efficiency. The general uniform mesh cannot balance well the contradiction between the accuracy and the efficiency. In this paper, the spacing between the two adjacent mesh points is called as an interval. Meanwhile, the interval length has a connection with the linearization error of nonlinear dynamics equations. Because the linearization error along a reference trajectory is not uniform, the mesh points should be denser in the domain with high nonlinearity for reducing the linearization error. However, the uniform mesh cannot have the function. Therefore, the non-uniform mesh is studied in this paper.

The normalized mesh time τ is the independent variable of the problem **P2**. Suppose there exists two fixed mesh point at two boundaries, e.g. $\tau = 0$ and $\tau = 1$. Let N be the number of mesh points except for the two fixed mesh point. Note that N is not a constant in this paper. The total number of mesh points is $N + 2$, and the mesh point can be denoted as $\{\tau_0, \tau_1, \tau_2, \dots, \tau_N, \tau_{N+1}\}$. Thus, there are $N + 1$ intervals, which can be denoted as $\{\Delta\tau_1, \Delta\tau_2, \dots, \Delta\tau_N, \Delta\tau_{N+1}\}$ with $\tau_j = \tau_{j-1} + \Delta\tau_j$, and $1 \leq i \leq N + 1$. In non-uniform mesh, there must exist a pair of indexes at least, such as j_1 and j_2 , which make $\Delta\tau_{j_1} \neq \Delta\tau_{j_2}$.

According to the above definition, we have

$$\sum_{j=1}^{N+1} \Delta\tau_j = 1 \quad (35)$$

For non-uniform mesh, the reference number of mesh points is defined by

$$N_{\text{ref}} = \frac{1}{\min_j \Delta\tau_j} \quad (36)$$

which represents the number of uniform mesh points if the same density of mesh points is desired. The reference number can be used to evaluate the performance of adjusting mesh points, which will be used in simulations.

After the definition of non-uniform mesh, the linear discretized dynamics equations are obtained by the trapezoidal rule as follows

$$\mathbf{x}_j = \mathbf{x}_{j-1} + \left[\begin{array}{l} (\xi^{(k)} \mathbf{A}_{j-1}^{(k)} \mathbf{x}_{j-1} + \xi^{(k)} \mathbf{B} \mathbf{u}_{j-1} + \mathbf{C}_{j-1}^{(k)} \cdot \xi + \\ \mathbf{D}_{j-1}^{(k)} \cdot \xi^{(k)}) + (\xi^{(k)} \mathbf{A}_j^{(k)} \mathbf{x}_j + \xi^{(k)} \mathbf{B} \mathbf{u}_j + \\ \mathbf{C}_j^{(k)} \cdot \xi + \mathbf{D}_j^{(k)} \cdot \xi^{(k)}) \end{array} \right] \Delta\tau_j / 2 \quad (37)$$

where $\mathbf{x}_j = \mathbf{x}(\tau_j)$, $\mathbf{u}_j = \mathbf{u}(\tau_j)$, $\mathbf{A}_j^{(k)} = \mathbf{A}(\mathbf{x}^{(k)}(\tau_j))$, $\mathbf{C}_j^{(k)} = \mathbf{C}(\mathbf{x}^{(k)}(\tau_j))$, $\mathbf{D}_j^{(k)} = \mathbf{D}(\mathbf{x}^{(k)}(\tau_j))$, and the subscript j represents the index of mesh points. For each $j = 1, 2, \dots, N, N + 1$, the Eq. (37) is a linear algebra equation about the state variables \mathbf{x}_j , control variables \mathbf{u}_j and the time expand factor ξ , which satisfies the standard form of a convex programming problem.

The other constraints are enforced on each mesh point, and the objective function is rewritten with discretized variables. By the non-uniform discretization, the problem **P2** is transformed to a convex programming problem, denoted as problem **P3**, which is expressed as follows

$$\mathbf{P3}: \min J = J_0 + C_3 \eta + C_4 \sum_{j=0}^{N+1} |\mathbf{u}_j| \Delta\bar{\tau}$$

subject to: Eq. (37)

$$\begin{aligned} & |\mathbf{x}_j - \mathbf{x}_j^{(k)}| \leq \eta \delta \\ & (\dot{\mathbf{u}})_{\min} \leq \mathbf{u}_j \leq (\dot{\mathbf{u}})_{\max} \\ & f_i(r_j^{(k)}, V_j^{(k)}) + f'_i(r_j^{(k)}, V_j^{(k)}) [r_j - r_j^{(k)}; \\ & V_j - V_j^{(k)}] \leq f_{i,\max}, \quad i = 1, 2, 3 \\ & \mathbf{x}(\tau = 0) = [r_0 \ \lambda_0 \ \phi_0 \ V_0 \ \theta_0 \ \sigma_0 \ v_0]^T \\ & V(\tau = 1) = V_f^* \end{aligned} \quad (38)$$

where $\mathbf{x}_j = \mathbf{x}(\tau_j) = [r(\tau_j) \ \lambda(\tau_j) \ \phi(\tau_j) \ V(\tau_j) \ \theta(\tau_j) \ \sigma(\tau_j) \ v(\tau_j)]^T$, $\mathbf{u}_j = \mathbf{u}(\tau_j)$. In problem **P3**, the optimization variables include state variables $\{\mathbf{x}_1, \dots, \mathbf{x}_N, \mathbf{x}_{N+1}\}$, control variables $\{\mathbf{u}_1, \dots, \mathbf{u}_N, \mathbf{u}_{N+1}\}$, the relaxation coefficient η and the time expansion factor ξ .

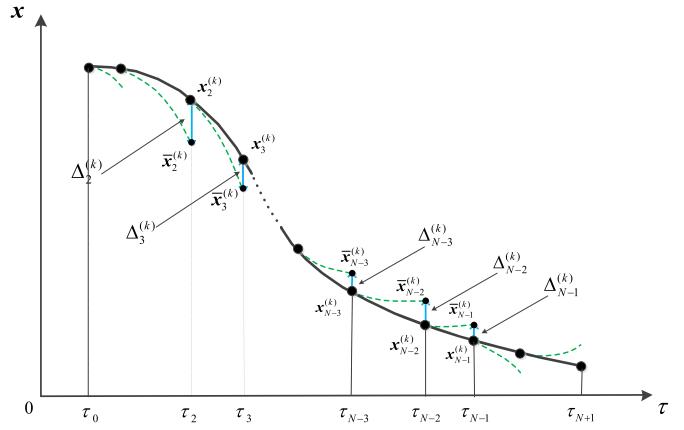


Fig. 1. Local error at each mesh point.

3.2. Adaptive mesh refinement

The SCP method is applied to solve the highly nonlinear problem **P1**. After discretization, the iteration solution is represented by a discrete form, so the mesh has influence on the accuracy of iteration solutions. In the iteration process, the accuracy of converged solution depends on the initial mesh when the mesh is fixed or unchanged. A good fixed mesh should be helpful to obtain a higher accuracy solution and improve the convergence of SCP method. However, how to choose a good initial mesh is very difficult. For handling the problem, a novel adaptive mesh refinement method is developed in this paper, which uses an initial uniform mesh, but can adjust the mesh adaptively based on the accuracy of iteration solutions in iteration process.

The solution error of each mesh point is the basis to adjust mesh points. In this paper, based on the iteration solution, the approximation error at each mesh point is defined by local integral, which is given as follows

$$\Delta_j^{(k)} = \|\mathbf{x}_j^{(k)} - \bar{\mathbf{x}}_j^{(k)}\|_2 \quad (39)$$

where the superscript k represents the index of iteration solutions or reference trajectories. $\Delta_j^{(k)}$ represents the approximation error at one mesh point, so it is also called as local error. $\mathbf{x}_j^{(k)}$ is the state vector from the iteration solution, and the $\|\cdot\|_2$ represents the 2-norm. The local integral state vector $\bar{\mathbf{x}}_j^{(k)}$ is defined by

$$\bar{\mathbf{x}}_j^{(k)} = \int_{\tau_{j-1}}^{\tau_j} \mathbf{F}(\mathbf{x}, \mathbf{u}, \tau) d\tau \quad (40)$$

where \mathbf{u} is obtained by linear interpolation on the interval $[u_{j-1}^{(k)}, u_j^{(k)}]$. $\bar{\mathbf{x}}_j^{(k)}$ represents the accurate state vector without linearization error, because it is obtained by integrating nonlinear dynamics $\mathbf{F}(\mathbf{x}, \mathbf{u}, \tau)$ using the fourth-order Runge-Kutta method. As shown in Fig. 1, the deviation between $\mathbf{x}_j^{(k)}$ and $\bar{\mathbf{x}}_j^{(k)}$ can be referred as an assessment of the linearization error at one mesh point j , which is defined by the local error $\Delta_j^{(k)}$ in this paper.

The high local error shows that the nonlinearity of original dynamics at corresponding mesh point is high, so it is necessary to increase the mesh density around the mesh point, because the old sparse mesh to approximate the continuous nonlinearity has resulted in a high error. On the contrary, at some mesh points with low local error, a dense mesh is not necessary, which increases computation burden and cannot improve the accuracy greatly, so the mesh density can be decreased around the mesh point.

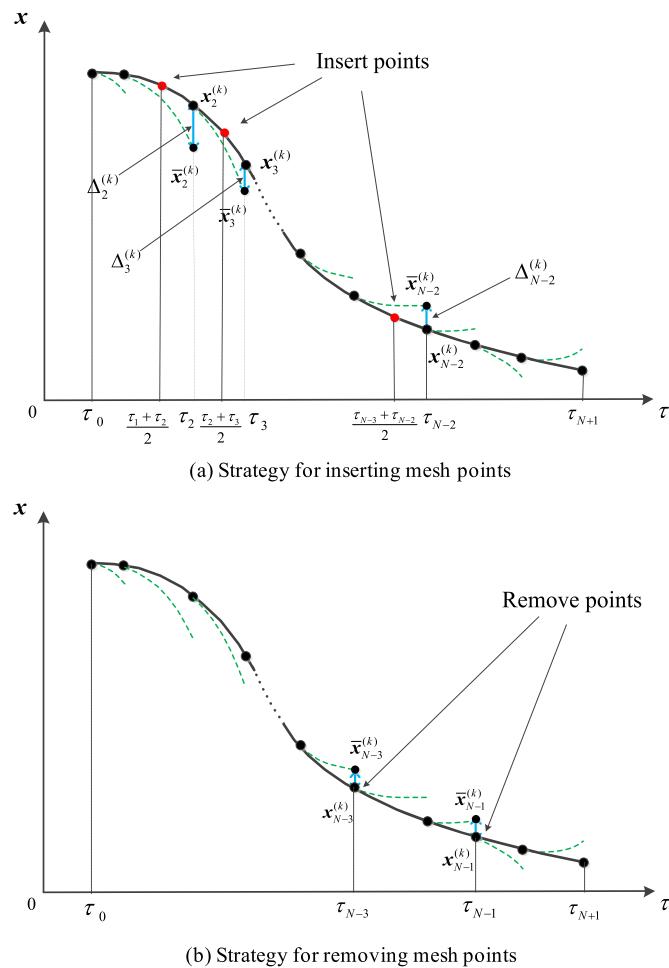


Fig. 2. Strategy for adjusting mesh points.

Based on local error at each mesh point, a strategy of inserting and removing mesh points is proposed. Take an arbitrary mesh point τ_j as an example. As shown in Fig. 2(a), if the local error at the mesh point τ_j is larger than a prescribed threshold ε_+ , which means the mesh in $[\tau_{j-1}, \tau_j]$ is so sparse that the local error is high, a new mesh point should be inserted into the middle of the interval $[\tau_{j-1}, \tau_j]$ to reduce the mesh interval. As shown in Fig. 2(b), if the local error at the mesh point τ_j is smaller than a prescribed threshold ε_- , which means the mesh density is enough to suppress the local error, the mesh point τ_j can be removed to reduce the computational burden of solving the convex subproblem. If the local error at the mesh point τ_j is smaller than ε_+ and larger than ε_- , the mesh point τ_j is retained. The mesh is redefined by performing the above procedure at all mesh points.

Once the mesh is redefined, the latest iteration solution on old mesh needs to be remapped on new mesh by linear interpolation, including the state vector and control vector, and the reference trajectory for next iteration is determined. Meanwhile, the new mesh will be used to describe next iteration solution.

By the proposed method, the mesh is redefined adaptively according to the accuracy of iteration solutions. In the domain with high nonlinearity, the distribution of mesh points will be denser regardless of the increasing computational burden, but in the domain with low nonlinearity, the distribution will sparser regardless of the decreasing accuracy. For balancing the contradiction between the solution accuracy and computation efficiency, during different distribution domains of nonlinearity, the proposed adaptive mesh refinement method tends to handle one of the two issues.

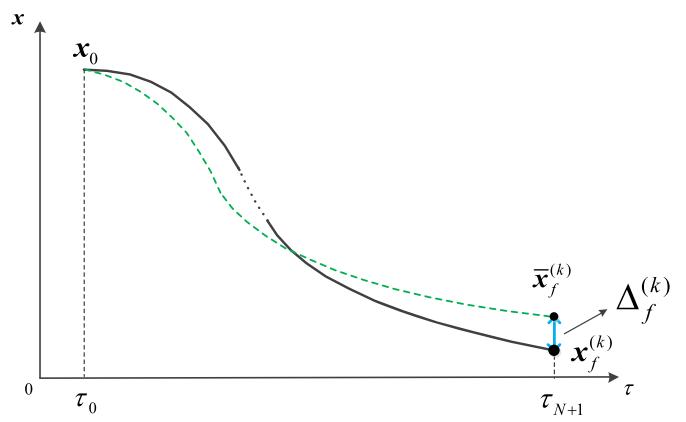


Fig. 3. Global error of reference trajectory.

The approximation error at each mesh point is obtained by local integral, which is the basis of adjusting mesh points adaptively. However, the convergence condition of SCP method must depend on a global accuracy assessment of iteration solutions. Thus, as shown in Fig. 3, the approximation error of iteration solutions or reference trajectories is defined by global integral, which is given as follows

$$\Delta \mathbf{x}_f^{(k)} = \bar{\mathbf{x}}_f^{(k)} - \mathbf{x}_f^{(k)} \quad (41)$$

where the subscript f represents the final mesh point. $\Delta_f^{(k)}$ represents the approximation error of iteration solutions, so it is also called as global error. $\mathbf{x}_f^{(k)}$ is the final state vector from the iteration solution. The global integral state vector $\bar{\mathbf{x}}_f^{(k)}$ is defined by

$$\bar{\mathbf{x}}_f^{(k)} = \int_0^1 \mathbf{F}(\mathbf{x}, \mathbf{u}, \tau) d\tau \quad (42)$$

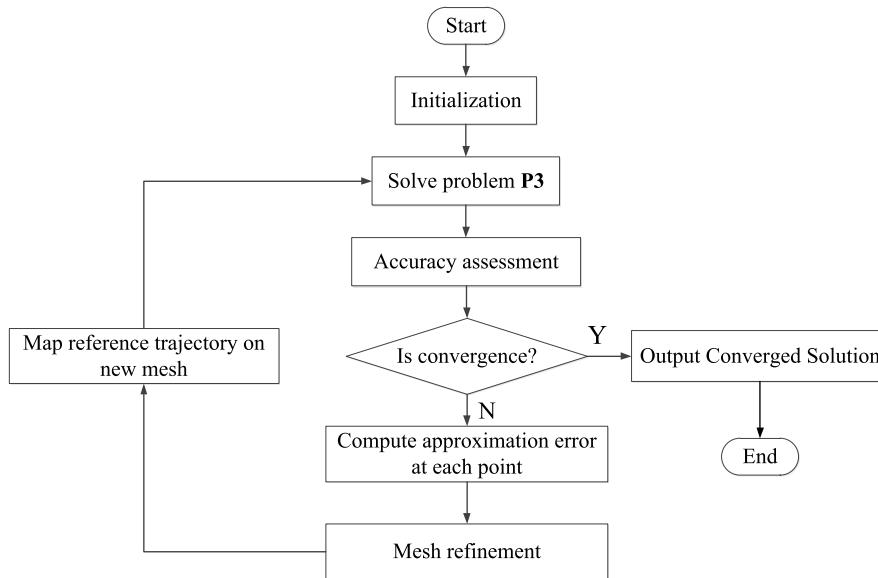
where \mathbf{u} is obtained by linear interpolation on the interval $[u_{\tau=0}^{(k)}, u_{\tau=1}^{(k)}]$. $\bar{\mathbf{x}}_f^{(k)}$ represents the accurate final state vector without linearization error. As shown in Fig. 3, the deviation between $\mathbf{x}_f^{(k)}$ and $\bar{\mathbf{x}}_f^{(k)}$ can be referred as a global assessment of the linearization error of the k -th iteration solution or reference trajectory.

According to the global accuracy assessment of the iteration solution, the convergence deviation is defined by the deviation between the global errors of two adjacent iteration solutions. Thus, the convergence condition is given as follows

$$|\Delta \mathbf{x}_f^{(k+1)} - \Delta \mathbf{x}_f^{(k)}| \leq \varepsilon \quad (43)$$

where the “ \leq ” is a component-wise inequality, the threshold ε is a user-defined sufficiently small constant vector, and $\Delta \mathbf{x}_f^{(k)} = [\Delta r_f^{(k)} \Delta \lambda_f^{(k)} \Delta \phi_f^{(k)} \Delta V_f^{(k)} \Delta \theta_f^{(k)} \Delta \sigma_f^{(k)}]^T$.

Due to the cumulative effect of numerical integration, when the global error is very small, the state deviation at all mesh points along the converged trajectory can be controlled within a certain range. Therefore, the convergence condition given in Eq. (43) can comprehensively evaluate the violation degree of nonlinear dynamics constraints at all mesh points along the converged trajectory. The feasibility of the converged trajectory for the original problem can be measured directly by the corresponding global error.

**Fig. 4.** Flow diagram of the proposed SCP method.

4. SCP method with adaptive mesh refinement

4.1. Initialization

The entry flight of reusable launch vehicle is divided into initial descent phase and glide phase. Because the glide phase has a longer flight time, the trajectory planning problem in glide phase is only considered in this paper. The initial state in glide phase is determined by integrating the nonlinear dynamics with zero bank angle. When the flight path angle is equal to zero, the corresponding state is taken as the initial state in glide phase, which is also the starting state of trajectory planning in this paper.

The initial glide trajectory for starting the SCP method is obtained by the linear assumption, where the state variables change linearly with increasing time [39]. The initial radial distance profile as a straight line connects r_0 and r_f^* . The longitude/latitude profile as a straight line connects the starting location (λ_0, ϕ_0) and the target location (λ_f^*, ϕ_f^*) . The determination method of the initial flight path angle profile and heading angle is consistent with that of the initial radial distance. The initial control variable is set as a constant zero. The initial flight time is set as a constant. In order to reduce the optimization space of flight time, an optimization range of flight time should be given, i.e. $t \in [t_{\min}, t_{\max}]$. In theory, the initial flight time should be within the time optimization range, and the selection of the time lower bound t_{\min} and upper bound t_{\max} needs to be determined based on experience. The selection principle is to ensure that the feasible flight time of the original problem exists within the time optimization range, and it is better that most of the feasible flight time are within the range. If the interval of the time optimization range is too small, the optimization space may be too small to find the appropriate solution. But if the interval is too large, it is helpless to rapid convergence. Based on the flight properties of reusable launch vehicle, the two bounds are $t_{\min} = 800$ s and $t_{\max} = 2500$ s.

The proposed mesh refinement method has not the unique demand for the initial mesh, so the initial mesh uses the simplest uniform distribution in this paper. After determining the number of initial mesh points, the initial trajectory with discrete form is determined, which includes the state variables, the control variables and the initial flight time.

4.2. Solution procedure

Combining the proposed mesh refinement method, a novel SCP method using adaptive mesh refinement is developed, and its solution procedure is summarized in the following steps:

Step 1: Initialize reference trajectory. Supply an initial mesh that consists of $N^{(0)}$ mesh points and two fixed points, which is denoted as $S^{(0)} = \{\tau_j^{(0)} | j = 0, \dots, N^{(0)}, N^{(0)} + 1\}$. In the mesh, there are $N^{(0)} + 1$ intervals $D^{(0)} = \{\Delta\tau_j^{(0)} | j = 1, \dots, N^{(0)} + 1\}$. The initial mesh is denoted as $\{S^{(0)}, D^{(0)}\}$. The initial trajectory with discrete form is denoted as $(\mathbf{x}^{(0)}, \mathbf{u}^{(0)}, \xi^{(0)})$. Set $k = 0$.

Step 2: Solve the problem **P3** based on the reference trajectory $(\mathbf{x}^{(k)}, \mathbf{u}^{(k)}, \xi^{(k)})$, and the iteration solution $(\mathbf{x}^{(k+1)}, \mathbf{u}^{(k+1)}, \xi^{(k+1)})$ on the mesh $\{S^{(k)}, D^{(k)}\}$ is obtained, which is denoted as $\{(\mathbf{x}^{(k+1)}, \mathbf{u}^{(k+1)}, \xi^{(k+1)}) | (S^{(k)}, D^{(k)})\}$.

Step 3: The global error of the iteration solution $(\mathbf{x}^{(k+1)}, \mathbf{u}^{(k+1)}, \xi^{(k+1)})$ is obtained by Eq. (41). If the convergence condition defined in Eq. (43) is satisfied, then quit; Otherwise, proceed to Step 4;

Step 4: Compute the local error at each mesh point defined in Eq. (39);

Step 5: The proposed strategy of inserting and removing mesh points shown in Section 3.2 is performed. For each mesh point on the mesh $\{S^{(k)}, D^{(k)}\}$,

- If the local error $\Delta_j^{(k)}$ is larger than prescribed threshold ε_+ , insert a mesh point using the strategy shown in Fig. 2(a);
- If the local error $\Delta_j^{(k)}$ is smaller than prescribed threshold ε_- , remove a mesh point using the strategy shown in Fig. 2(b);
- If the above conditions are not satisfied, retain the mesh point.

Arrange the redefined mesh, denoted as $\{S^{(k+1)}, D^{(k+1)}\}$;

Step 6: Map the iteration solution $\{(\mathbf{x}^{(k+1)}, \mathbf{u}^{(k+1)}, \xi^{(k+1)}) | (S^{(k)}, D^{(k)})\}$ to $\{(\mathbf{x}^{(k+1)}, \mathbf{u}^{(k+1)}, \xi^{(k+1)}) | (S^{(k+1)}, D^{(k+1)})\}$ by linear interpolation. Set $k = k + 1$. Return to **Step 2**;

A flow diagram of the proposed SCP method is shown in Fig. 4.

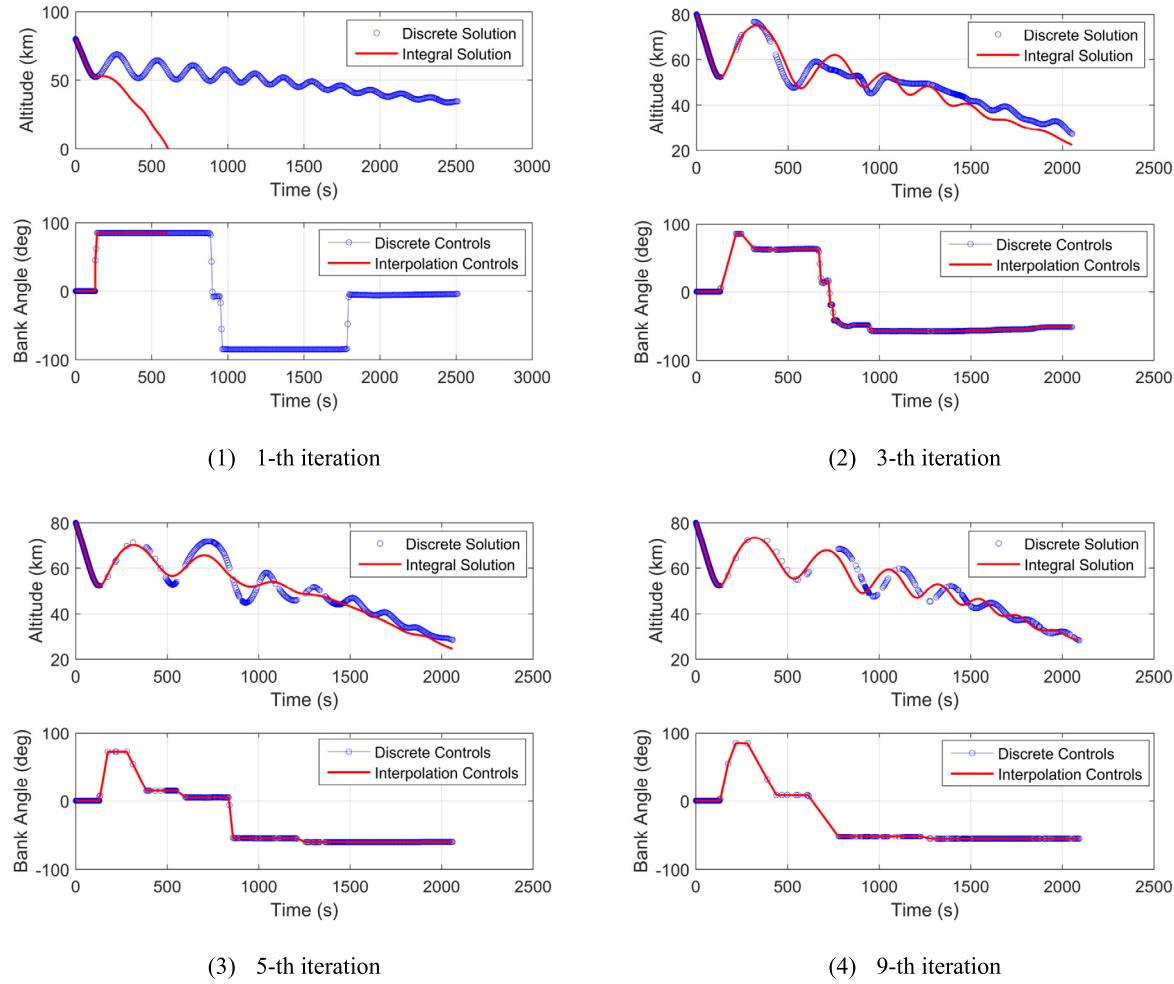


Fig. 5. Iteration solutions with adaptive mesh.

5. Numerical simulations

The numerical simulation is performed to demonstrate the effectiveness of the proposed SCP method. The CAV-H model is adopted, and its parameters are given in Ref. [40]. The initial entry conditions are $H_0 = 80$ km, $\lambda_0 = 60^\circ$, $\phi_0 = -20^\circ$, $V_0 = 7000$ m/s, $\theta_0 = -2^\circ$, and $\sigma_0 = 90^\circ$. The terminal state constraints are $H_f^* = 27$ km, $\lambda_f^* = 140^\circ$, $\phi_f^* = 20^\circ$, $V_f^* = 2500$ m/s, $\theta_f^* = -5^\circ$, and $\sigma_f^* = 30^\circ$. For the three path constraints, the maximum allowable values are $\dot{Q}_{\max} = 3800$ kw/m², $q_{\max} = 90$ kPa and $n_{\max} = 3.5g_0$. The bank angle limits are $\nu_{\max} = 85^\circ$ and $\nu_{\min} = -85^\circ$. The change rate limits of bank angle are $\dot{\nu}_{\max} = 20^\circ/\text{s}$, and $\dot{\nu}_{\min} = -20^\circ/\text{s}$.

The number of initial mesh points is 400. The trust-region constraint is $\delta = \left[\frac{20000}{R_0} \quad \frac{20\pi}{180} \quad \frac{20\pi}{180} \quad \frac{500}{\sqrt{g_0 R_0}} \quad \frac{20\pi}{180} \quad \frac{30\pi}{180} \quad \frac{90\pi}{180} \right]$. The convergence threshold $\epsilon = \left[\frac{100}{R_0} \quad \frac{0.5\pi}{180} \quad \frac{0.5\pi}{180} \quad \frac{50}{\sqrt{g_0 R_0}} \quad \frac{0.5\pi}{180} \quad \frac{5\pi}{180} \right]$. The maximum iteration number is 40. The weight coefficients in the combined objective function are $C_1 = 1 \times 10^6$, $C_2 = 1 \times 10^4$, $C_3 = 100$, and $C_4 = 1$. The two tolerances for adjusting mesh points are $\varepsilon_+ = 1 \times 10^{-2}$, $\varepsilon_- = 1 \times 10^{-3}$.

The CVX is used to call a state-of-art primal-dual interior-point algorithm in the solver Mosek to solve the convex subproblem [41]. All the results are obtained by running the method on a desktop with Intel Core i5-7200 2.70 GHz, 8G RAM, and Windows 10 operating system.

In Section 5.1, the solution accuracy is verified for the proposed SCP method. In Section 5.2, the property of the proposed adaptive

mesh refinement method is analyzed. In Section 5.3, the convergence of the proposed SCP method is analyzed.

5.1. Verification of solution accuracy

In this section, the original problem **P1** is solved by two methods respectively, which are the SCP method using adaptive mesh refinement proposed in this paper and the traditional SCP method using fixed and uniform mesh. In this paper, except for the distribution of mesh points, the traditional SCP method are mainly consistent with the proposed SCP method, including the convex subproblem to be solved iteratively, the convergence condition and so on. The proposed SCP method uses the adaptive mesh refinement technique given in Section 3.2 to adjust the distribution and number of mesh points at each iteration. However, the traditional SCP method uses the fixed and uniform to discretize the convex subproblem, so the distribution and number of mesh points are unchanged at each iteration.

The middle solutions of problem **P3** via the proposed method are shown in Fig. 5. “Discrete Solution” represents the direct optimized trajectory at current iteration, and “Integral Solution” represents the accurate trajectory, which is obtained by integrating the nonlinear dynamics from t_0 to t_f using optimized controls. “Discrete Controls” represents the direct optimized controls with discrete form, and “Interpolation Controls” represents the continuous controls, which is obtained by linear interpolation in mesh

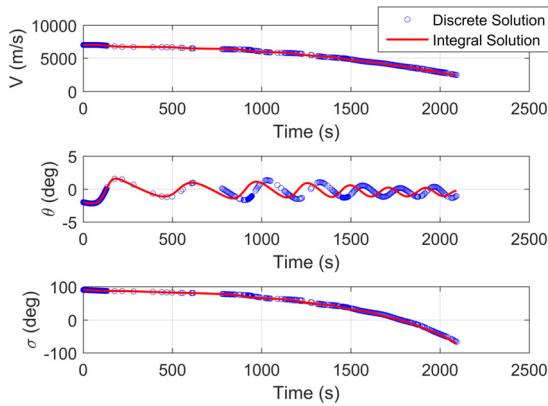


Fig. 6. Histories of velocity and two angles from converged solution.

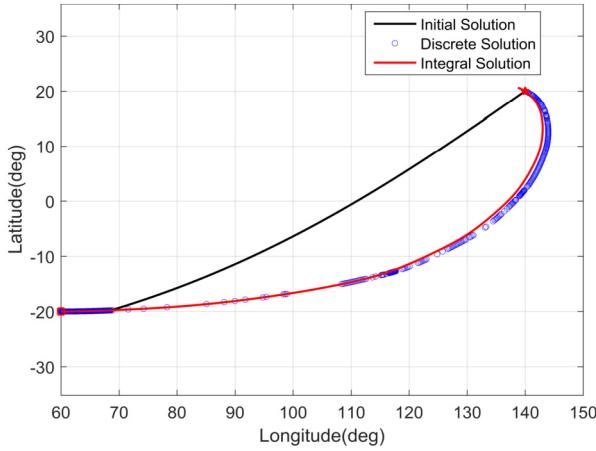


Fig. 7. Footprints from converged solution.

interval. Note that the definition of legends is universal in this paper.

The solution shown in Fig. 5(1) is obtained at the first iteration, which uses the initial uniform mesh with 400 mesh points. In Fig. 5(1), there is a large deviation between the altitude histories of Integral Solution and Discrete Solution. It means that the violation of nonlinear dynamics constraint is very high, so the solution accuracy at first iteration is very low. With the iteration number increasing, the deviation between two altitude histories is gradually decreasing. After 9 iterations, the convergence condition defined in Eq. (43) is satisfied, and a solution with higher accuracy is obtained, which is shown in Fig. 5(4). The discrete solution at 9-th iteration is known as the converged solution of the problem **P1** via the proposed method.

The other state variables of the above converged solution are shown in Fig. 6 and Fig. 7, including velocity, flight path angle (FPA), heading angle (HA) and longitude/latitude. The corresponding path constraints are shown in Fig. 8. The initial solution in Fig. 7 is obtained by the method introduced in Section 4.1. It is shown that the deviation of state variables between Discrete Solution and Integral Solution are small. Combining Fig. 5(4), if the converged solution is obtained, the deviations of all state variables between Discrete Solution and Integral Solution are small, which shows that the violation of nonlinear dynamics constraint is low. Thus, the solution accuracy of the proposed method is high. The same conclusion can also be obtained by Fig. 8.

With the same simulation conditions, the Problem **P1** is solved by traditional SCP method with fixed and uniform mesh. After 11 iterations, the convergence condition defined in Eq. (43) is satisfied. The converged discrete trajectory and integral trajectory are

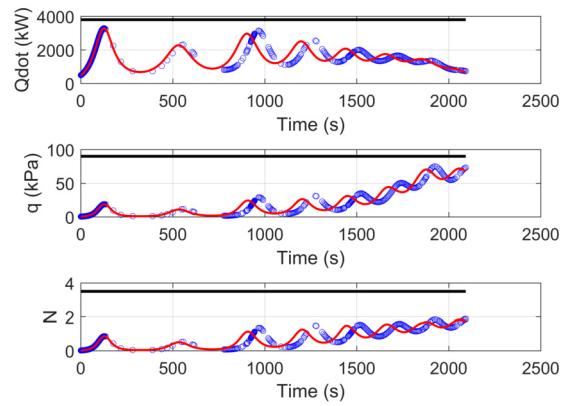


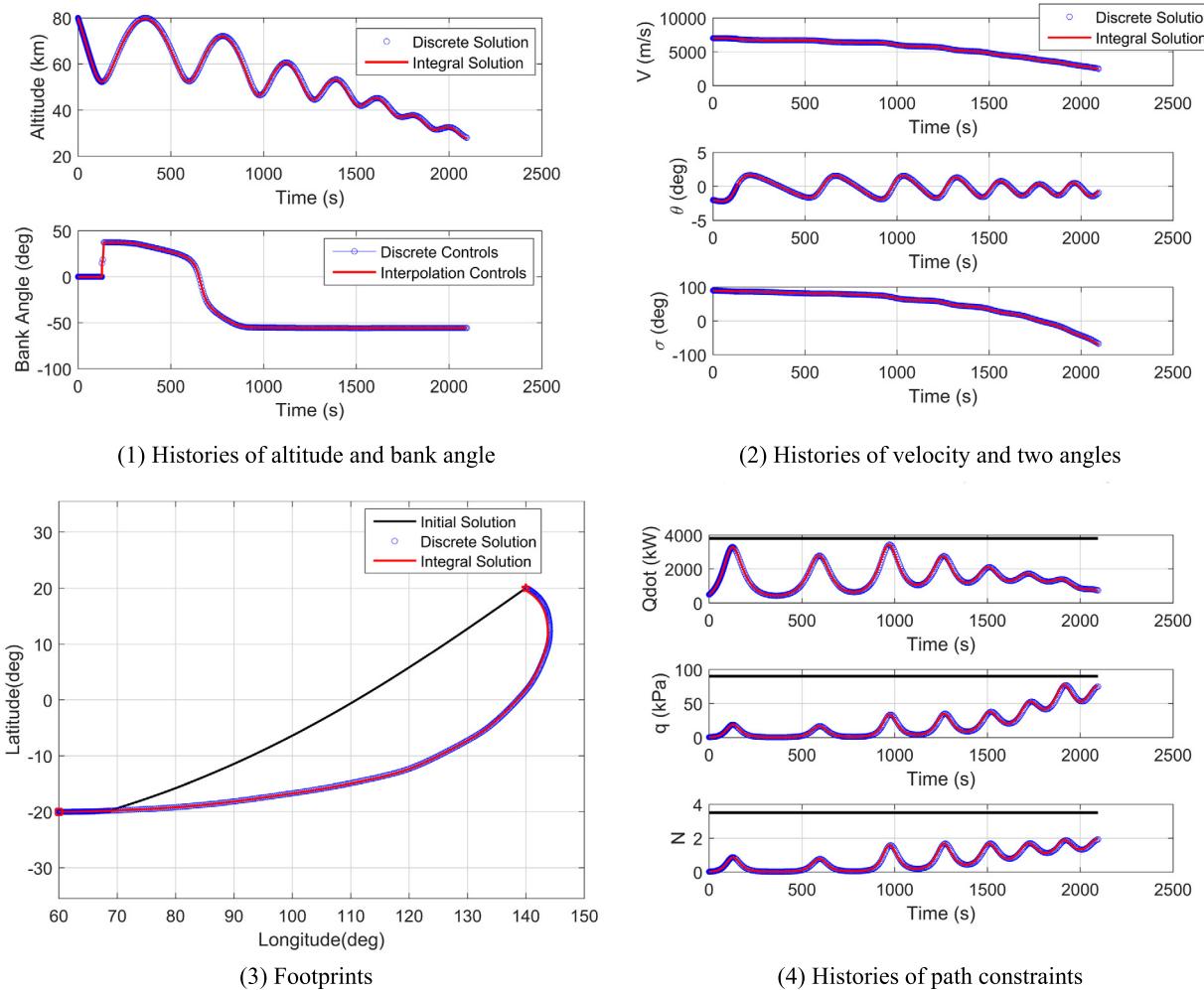
Fig. 8. Histories of path constraints from converged solution.

shown in Fig. 9. It is shown that the deviation between Discrete Solution and Integral Solution is very low.

Comparing the two converged trajectories with different mesh in Fig. 5(4) and Fig. 9(1), it is shown the deviation between Discrete Solution and Integral Solution is higher for the SCP method with adaptive mesh. Because the adaptive mesh is sparse in most regions when the convergence condition is satisfied. Due to the mesh sparsity, the linear interpolation error of controls is higher in the larger mesh interval, thus the integral trajectory has a larger deviation with the discrete trajectory when using the optimized controls obtained by interpolation. As for the traditional SCP method, because the mesh is dense in overall region, the linear interpolation error of controls will be very low, thus the deviation is very small between Discrete Solution and Integral Solution.

In this paper, “Adaptive mesh” and “Fixed mesh” are used to distinguish the two SCP methods, i.e. the proposed SCP method using adaptive mesh and the traditional SCP method with fixed and uniform mesh. The two converged discrete trajectories are compared in Fig. 10. It is shown that in the former region, the deviation between two histories is higher, including state variables and control variables, but in the latter region, the deviation is lower. A few peak values in Fig. 10(4) represent that the amplitude of bank angle is changed at corresponding moment. The deviation is related to the density of the mesh points. The converged solution of the SCP method is discrete, so the density of mesh points determines the approximate error of the continuous controls and the states. It can be seen from Fig. 10(4) that the density of the fixed mesh in most regions is higher than that of the adaptive mesh, so the approximation error is lower for the fixed mesh. As can be seen from the bank angle history in Fig. 10(3), the curve of the fixed mesh is smoother, and no broken line appears at the key points. However, in order to reduce the computation cost for the proposed SCP method, the mesh points in the region with low nonlinearity are relatively sparse, resulting in rough approximation of the controls. The deviation reason of state variables is the same as that of controls. Therefore, there is a deviation between the two converged solutions.

When the convergence condition defined in Eq. (43) is satisfied, the two SCP methods terminate. The convergence deviation $|\Delta \mathbf{x}_f^{(k+1)} - \Delta \mathbf{x}_f^{(k)}|$ is compared in Fig. 11, including six state variables, altitude, velocity, longitude, latitude, flight path angle, and heading angle. The normalized global error $\|\Delta \mathbf{x}_f\|_2$ is compared in Fig. 12, where the horizontal axis represents the number of sequential iteration (SI). It is shown that the convergence deviation decreases gradually, and thus the proposed method can converge with finite iterations. Meanwhile, the global error decreases gradually. When the method converges, the global error $\|\Delta \mathbf{x}_f\|_2$ will converge to a small and stable value. Thus, the converged solution has a low violation of nonlinear dynamics constraint, which

**Fig. 9.** Converged solution of traditional SCP method.
Table 1
Comparison of global error.

	Adaptive mesh			Fixed mesh		
	$\mathbf{x}_f^{(k)}$	$\bar{\mathbf{x}}_f^{(k)}$	$\Delta \mathbf{x}_f^{(k)}$	$\mathbf{x}_f^{(k)}$	$\bar{\mathbf{x}}_f^{(k)}$	$\Delta \mathbf{x}_f^{(k)}$
Altitude(m)	28133.55	28284.83	151.2827	27961.24	27633.82	-327.4255
Longitude(deg)	140	138.72	-0.9552	140	139.49	-0.5807
Latitude(deg)	20	20.62	1.0205	20	19.74	-0.0615
Velocity(m/s)	2500	2431.96	-68.0315	2500	2447.69	-52.3068
FPA(deg)	-1.05	-0.25	0.7915	-0.93	-0.68	0.2519
HA(deg)	-66.64	-71.61	-4.8987	67.59	-70.60	-3.0594
2-norm (Normalized)			0.0904			0.0549

shows that the converged solution can be considered as a feasible solution for the original problem **P1**.

The flight time is free in the convex subproblem **P3**. In Fig. 13(1), the iteration process of flight times is compared. It is shown that the converged flight time of two methods are consistent mainly, which are 2092.2707 s for the proposed method and 2095.6861 s for the tradition method.

In Fig. 13(2), the relaxation coefficient of trust-region constraint is compared. It is shown that the relaxation coefficients of the two methods all converge to a small value, which is 0.1766 for the proposed method and 0.1140 for the traditional method. A small relaxation coefficient shows that the optimization space of states is small in final iteration process, which can guarantee the solution accuracy.

In Fig. 13(3), the performance index in the problem **P3** is compared. In Fig. 13(4), the value of accumulation term about controls

is compared, which is $C_4 \sum_{j=0}^{N+1} |\mathbf{u}_j| \Delta \bar{\tau}$ in problem **P3**. It is shown that the deviation between two converged performance indexes is 306.6, and the deviation between two accumulation terms is 311.03. The two deviations are roughly consistent, so the deviation of performance index results from the deviation of accumulation terms, instead of the deviation of terminal positions. Because the value of accumulation term depends on mainly the number of mesh points N , the converged number of mesh points for the proposed SCP method is markedly smaller than the fixed number of mesh points for the traditional SCP method.

The global error represents the violation of nonlinear dynamics constraint for a discrete solution, which is used to evaluate the solution accuracy. The global errors of converged solutions from the two SCP methods are given in Table 1, where $\mathbf{x}_f^{(k)}$ is the terminal

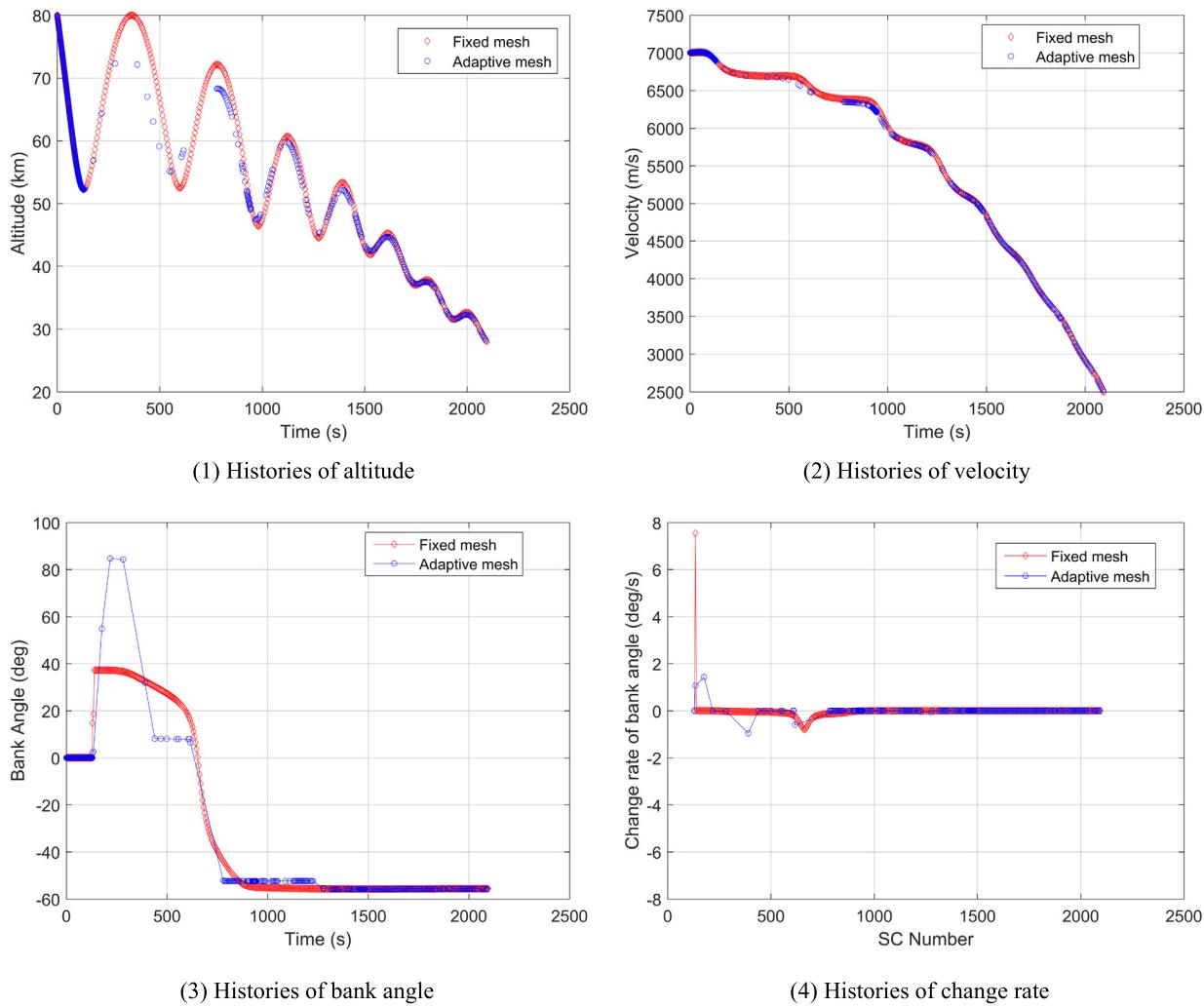


Fig. 10. Converged solution comparison of two SCP methods.

states of a discrete solution, $\bar{\mathbf{x}}_f^{(k)}$ is the terminal states of an integral solution, and the global error $\Delta\mathbf{x}_f^{(k)}$ is obtained by Eq. (41).

From Table 1, the desired terminal longitude and latitude are reachable in two SCP methods. The terminal velocity constraint is satisfied. But there is a little deviation between optimized value and desired value about altitude, which shows that the desired terminal altitude is not reachable. When the two SCP methods converge, due to the satisfactory of terminal position, the terms of position in performance index are very small numerically. The deviation between two performance indexes in Fig. 13(3) results from the accumulation term about controls $C_4 \sum_{j=0}^{N+1} |\mathbf{u}_j| \Delta \bar{\tau}$.

In Table 1, the difference of global error between two SCP methods is small, and the global error of the traditional SCP method is slightly smaller than that of the proposed SCP method, including six state variables. It is obviously because under the fixed mesh, the density of mesh points is relatively high in most regions, which leads to the smaller interpolation error of the controls when integrating nonlinear dynamic equations. However, with the proposed SCP method, the global error is also very small, so the converged solutions of the two SCP methods can be considered as a feasible solution of the original problem. Due to the same convergence condition, the accuracies of converged solutions from two SCP methods are consistent basically. Therefore, the solution accuracy of the proposed method is verified.

The time consumptions of the two SCP methods are compared in Table 2. The total time is obtained by summing all the computa-

Table 2
Comparison of time consumption.

	Adaptive mesh	Fixed mesh
Iteration number	9	11
Converged number of mesh points	237	400
Average number of mesh points	316	400
Average time of single subproblem (s)	0.44	0.47
Total time (s)	4.02	5.24

tion time to solve the single convex subproblem, and the solution time of single subproblem is the output time of Mosek solver. It is shown that the converged number of mesh points for the proposed SCP method with adaptive mesh is lower than that for the tradition SCP method with fixed mesh. Thus, the average number of mesh points at each iteration is lower for the proposed SCP method, which means that the number of optimization variables and constraints is lower, and the computational burden of solving subproblems is lower. As a result, the average time of solving single convex subproblem are lower for the proposed SCP method.

The hp adaptive pseudospectral method is a popular direct method to solve the general optimal control problem, which uses hp adaptive mesh to make mesh refinement [33]. In Table 3, the converged results of the proposed SCP method and the hp adaptive pseudospectral method are compared. Note that the total time of hp adaptive pseudospectral method is the output time of SNOPT

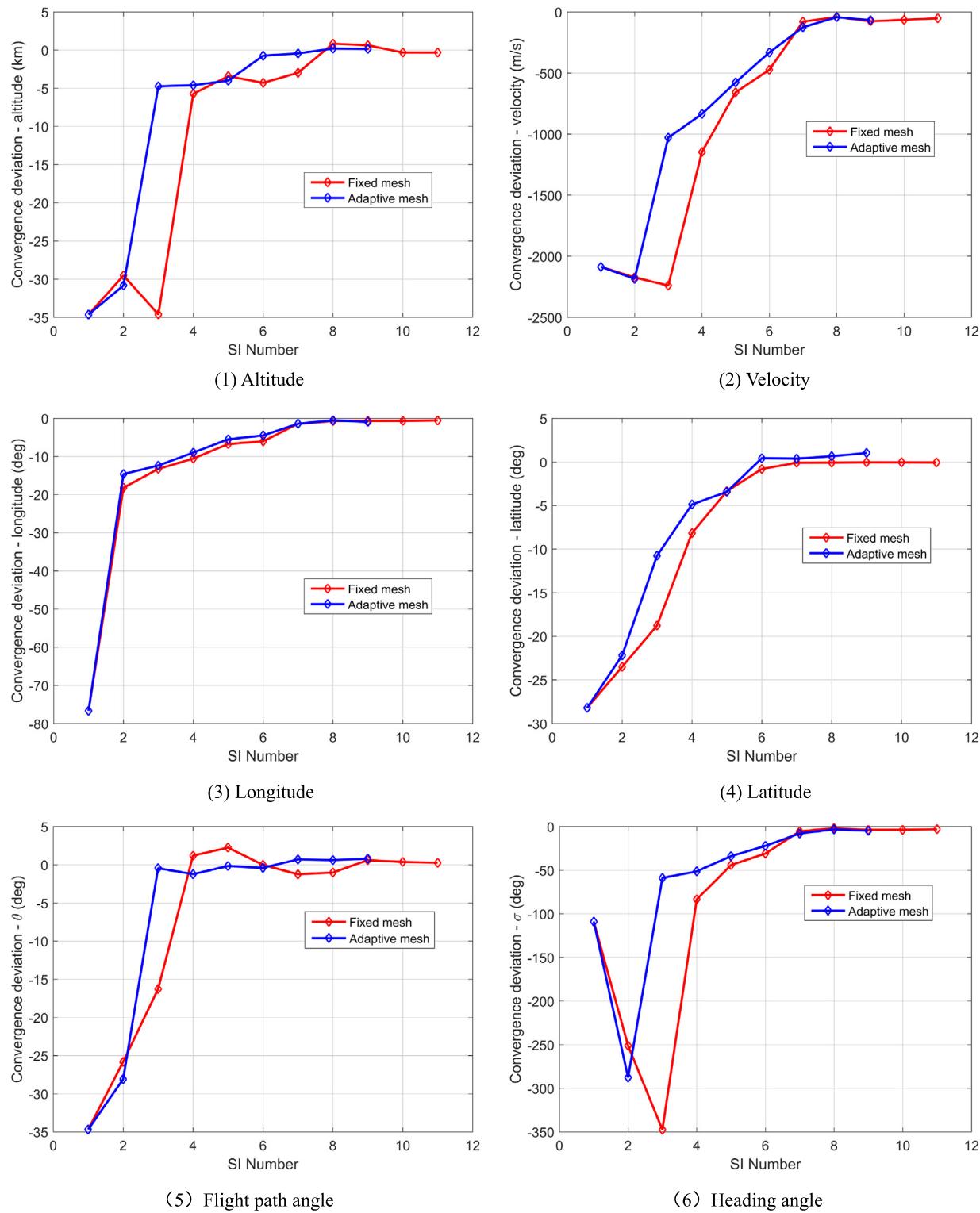


Fig. 11. Comparison of convergence deviation about six states.

solver. The global error of hp adaptive pseudospectral method is obtained by the same equation given in Eq. (41).

It is shown that the performance index and global error of the proposed SCP method are less than that of the hp adaptive pseudospectral method, which indicates the converged solution of the proposed SCP method has a higher precision compared with the hp adaptive pseudospectral method. It can be seen that the converged number of mesh points is lower for the proposed SCP

method, so the proposed SCP method can reduce effectively the number of mesh points in the iteration process, and the computational burden of single subproblem. Because the average solution time of single subproblem is shorter for the proposed SCP method. Meanwhile, the total computational time of solving the original problem **P1** is also shorter for the proposed SCP method, due to the lower number of mesh points and iterations. Therefore, the proposed SCP method has a higher computational effi-

Table 3
Comparison of time consumption.

	SCP method proposed in this paper	hp adaptive pseudospectral method
Performance index	86.8	122.6
Global error	0.0904	0.3678
Iteration number	9	25
Converged number of mesh points	237	1260
Average time of single subproblem (s)	0.44	11.38
Total time (s)	4.02	284.49

Table 4
Comparison of convergence deviation.

	Condition 1 (22 iterations)		Condition 2 (9 iterations)	
	$\max_i \mathbf{x}_i^{k+1} - \mathbf{x}_i^k $	$ \Delta \mathbf{x}_f^{(k+1)} - \Delta \mathbf{x}_f^{(k)} $	$\max_i \mathbf{x}_i^{k+1} - \mathbf{x}_i^k $	$ \Delta \mathbf{x}_f^{(k+1)} - \Delta \mathbf{x}_f^{(k)} $
Altitude (m)	80.5323	25.2462	2514.7478	48.8069
Longitude (deg)	0.0250	0.0061	0.1567	0.4148
Latitude (deg)	0.0166	0.0082	0.1742	0.3679
Velocity (m/s)	4.7719	1.1324	47.8585	24.4585
FPA (deg)	0.0326	0.0076	0.5902	0.1833
HA (deg)	0.2359	0.0439	1.2032	1.6281

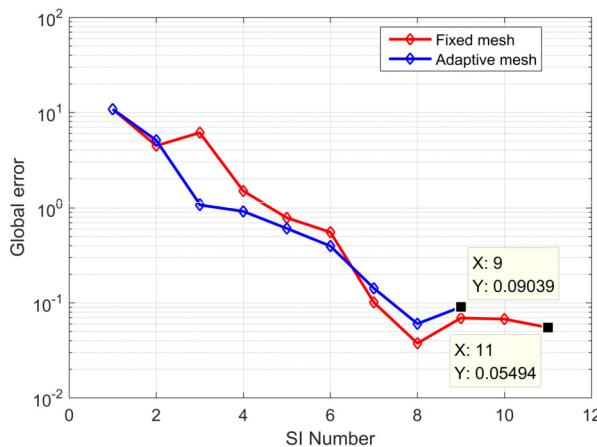


Fig. 12. Comparison of global error.

ciency to solve the original problem while guaranteeing the solution accuracy, compared with the hp adaptive pseudospectral method.

5.2. Adaptive mesh analysis

The adaptive mesh refinement based on local error at each mesh point is the most contribution in this paper. In the process of mesh refinement, the distribution and number of mesh points are adjusted simultaneously. For the simulation case of the proposed SCP method with adaptive mesh refinement in Section 5.1, the properties of adaptive mesh is analyzed in this section.

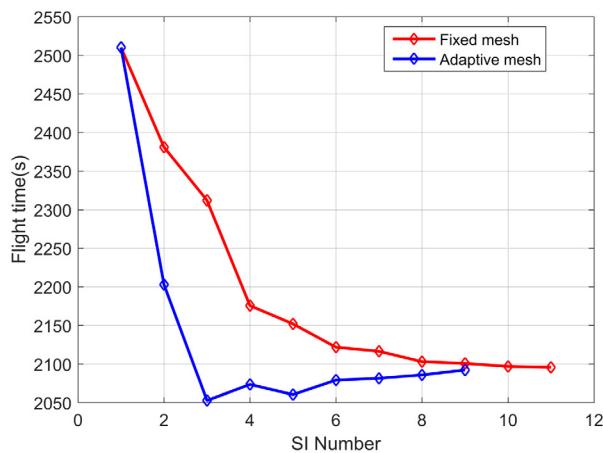
The distribution of mesh points at each iteration is shown in Fig. 14, which is defined in the region from 0 to 1. It is shown that the sparse mesh region is mainly located in the initial entry period, and the dense mesh domain is mainly located in the later period. The reason of resulting in the distribution is that the nonlinearity of original dynamics defined in Eq. (1) is non-uniform in entry. According to the proposed adaptive mesh refinement method, in the region with high nonlinearity, the distribution of mesh points will be dense, but in the region with low nonlinearity, the distribution will be sparse. The nonlinearity results from many factors, including the atmosphere density with exponential form, the complex aerodynamic model, the trigonometric function defined in Eq. (1) and so on. Obviously, the nonlinearity is low in the initial entry pe-

riod, because the higher altitude results in the lower aerodynamic effect.

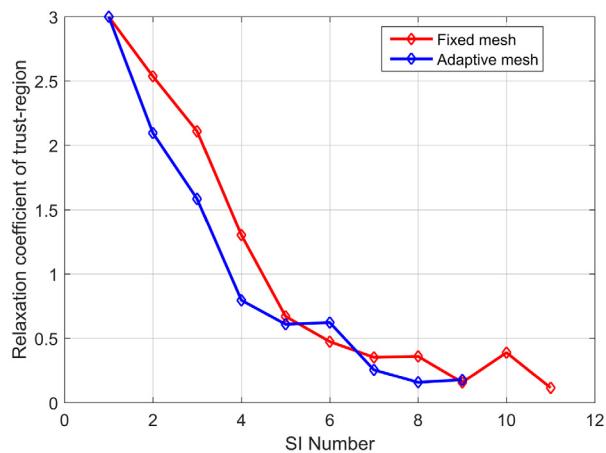
The local error at each mesh point is obtained by Eq. (39) to determine how to adjust the point. After 9 iterations, the convergence condition given in Eq. (43) is satisfied. The distribution of adjusting points is shown in Fig. 15, including inserting points and removing points, where the red rhombus represents the removing points, and the blue circle represents the inserting points. In first iteration, there are only some inserting points, which shows that the local errors at all mesh points are greater than the prescribed threshold ε_- of removing points. In middle iterations, there are not only inserting points, but also removing points. In last iteration, there are only some removing points, which shows that the local errors at all mesh points are less than the prescribed threshold ε_+ of inserting points. Combining with Fig. 14, it is shown that the distribution of mesh points is denser in the region with inserting points, and the distribution of mesh points is sparser in the region with removing points. Furthermore, the aim of adjusting mesh points is to decrease the local error at each mesh point in the former iterations, but in the latter iterations, the aim is to decrease the number of mesh points for solving single convex subproblem.

The mesh points number in iteration process is shown in Fig. 16, where the reference number represents the reference value of mesh points number at each iteration, which is obtained by Eq. (36), and the real number is the mesh points number of each subproblem to be solved in iteration process. It is shown that the real number is decreasing after second iteration, but the reference number is increasing. At the last iteration, the real value is 237, but the reference value is 1596. The increasing reference number shows that the maximum density of mesh points is increasing, and the number of mesh points will be increased theoretically if the uniform mesh is applied. However, when using the proposed SCP method with adaptive mesh, the number of mesh points can be decreased markedly, which means that the computational efficiency of single convex subproblem P3 in latter iterations is higher. Therefore, the proposed method is beneficial to improve the computational efficiency for solving the original problem P1.

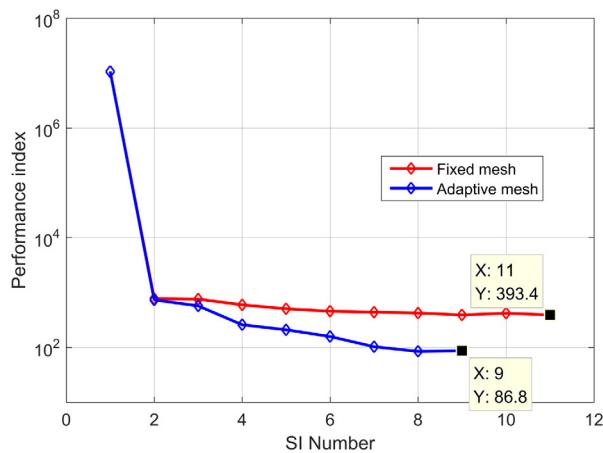
The number of adjusting mesh points is shown in Fig. 17, including the number of inserting points, the number of removing points, and the number of adjusting points in each iteration. If the number of adjusting points is positive, the number of inserting points is larger than that of removing points in the iteration. If the number of adjusting points is negative, the number of inserting points is smaller than that of removing points in the iteration. The



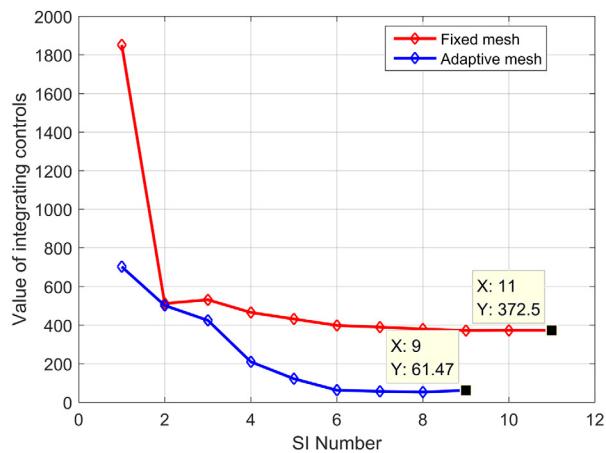
(1) Flight time (s)



(2) Relaxation coefficient of trust-region



(3) Performance index



(4) Value of integrating controls

Fig. 13. Comparison of optimized results.

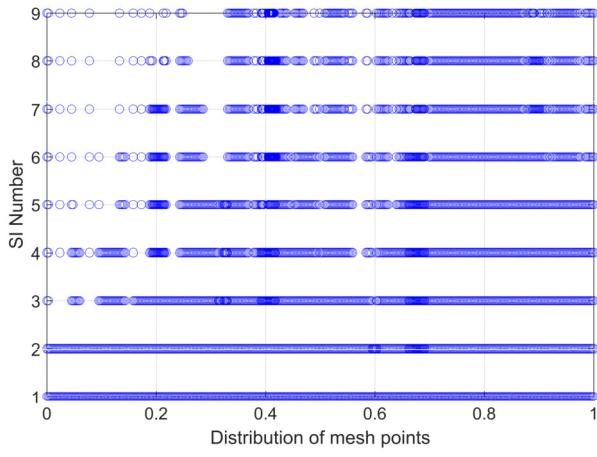


Fig. 14. Distribution of mesh points.

numbers of inserting points and the number of removing points all decrease with the iteration number increasing. In the last two iterations, the number of inserting points is zero, but the number of removing points is not zero. It is shown that the local errors at all mesh points are very low, and thus the global error is converged to a small value.

In this paper, the local error defined in Eq. (39) is used to evaluate the violation of nonlinear dynamics constraint at each mesh

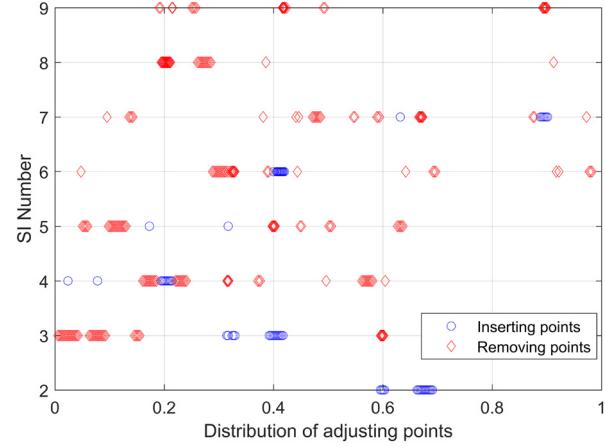


Fig. 15. Distribution of adjusting points. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

point, which is also a criterion to adjust mesh points. The local errors of two converged solutions from Section 5.1 are compared in Fig. 18. It is shown that the deviation between the two histories of local error is small. Thus, the proposed SCP method with adaptive mesh can obtain the mainly consistent local error with fewer iterations, compared with the traditional SCP method with fixed mesh.

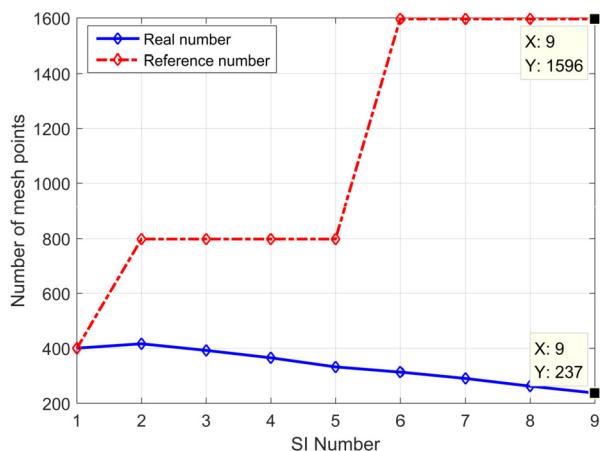


Fig. 16. History of mesh points number.

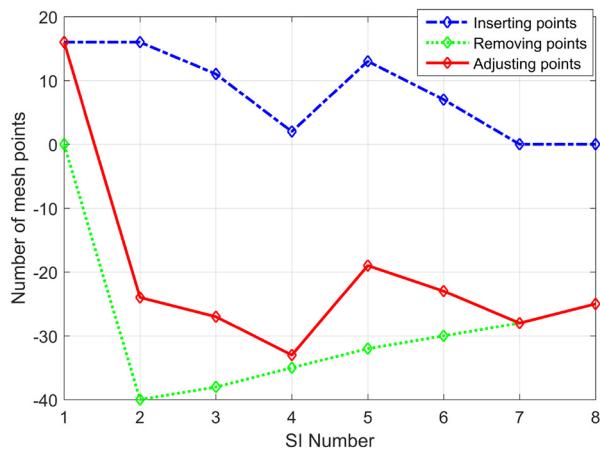


Fig. 17. History of adjusting points number.

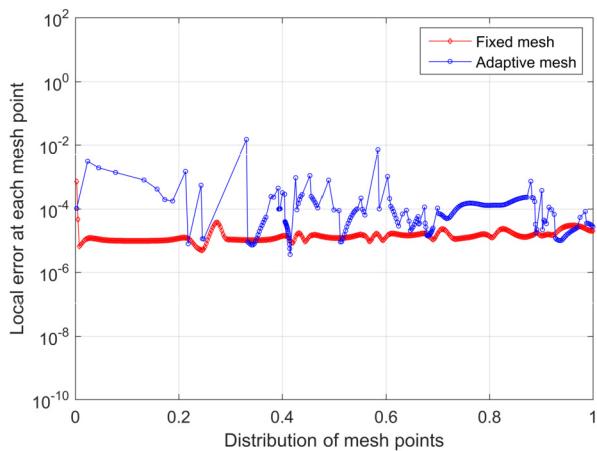


Fig. 18. Comparison of local error at each mesh point.

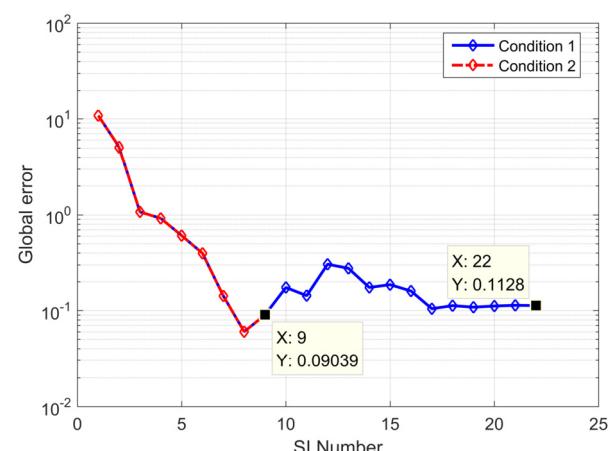


Fig. 19. Iteration process of global error.

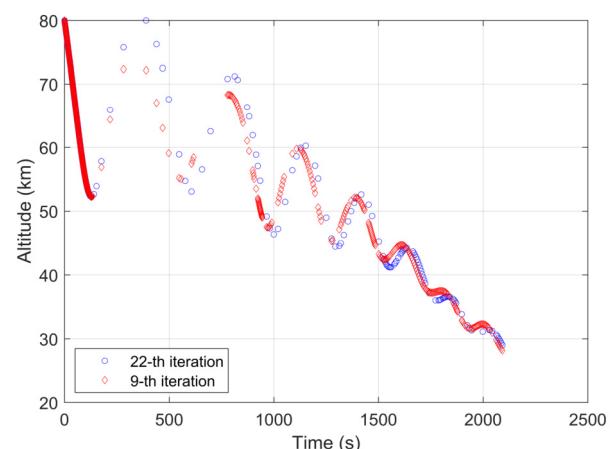


Fig. 20. Comparison of altitude.

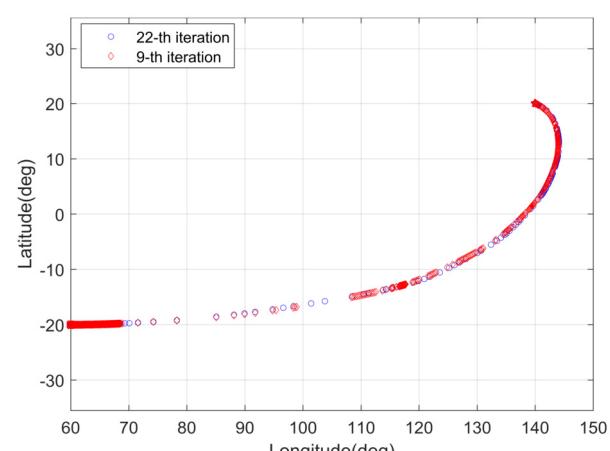


Fig. 21. Comparison of footprints.

Table 5
Comparison of global error.

	Condition (22 iterations)	Condition 2 (9 iterations)
Altitude (m)	-1570.0869	151.2827
Longitude (deg)	-1.1565	-0.9552
Latitude (deg)	0.6076	1.0205
Velocity (m/s)	91.0698	-68.0315
FPA (deg)	0.9378	0.7915
HA (deg)	-6.225	-4.8987
2-norm (Normalized)	0.1128	0.0904

5.3. Convergence analysis

There are two issues to be considered about the convergence of SCP method. Whether it converges or not is the first, and the choice of convergence condition is the second. In Section 5.1, it is shown that the proposed SCP method with adaptive mesh can converge to a feasible solution for the original problem **P1**. As for the second issue, in general, when the deviation between the two adjacent iteration solutions or the performance indexes is smaller than

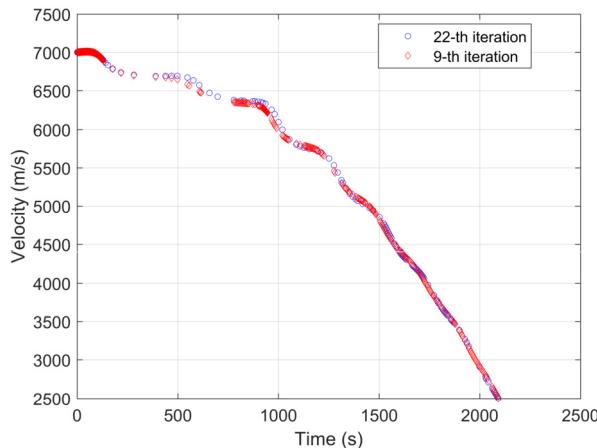


Fig. 22. Comparison of velocity.

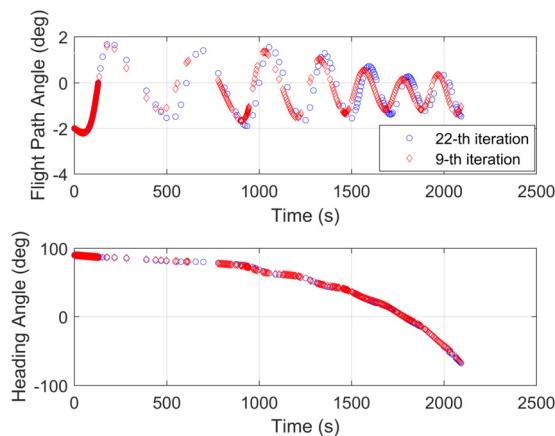


Fig. 23. Comparison of FPA and HA.

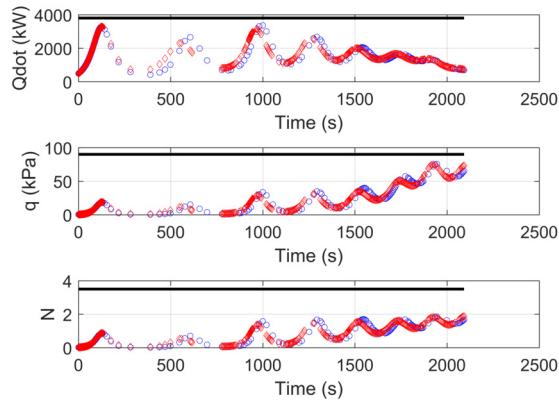


Fig. 24. Comparison of three constraints.

a prescribed threshold, the SCP method converges. In this case, the corresponding convergence condition is given by

$$\max_i |\mathbf{x}_i^{(k+1)} - \mathbf{x}_i^{(k)}| \leq \epsilon \quad (44)$$

where $\mathbf{x}_i^{(k)}$ is the state vector at the i -th mesh point of the k -th iteration solution. The left term of “ \leq ” represents the maximum state deviation vector between two adjacent iteration solutions. The convergence condition given in Eq. (44) is represented by condition 1. The convergence condition given in Eq. (43) is applied

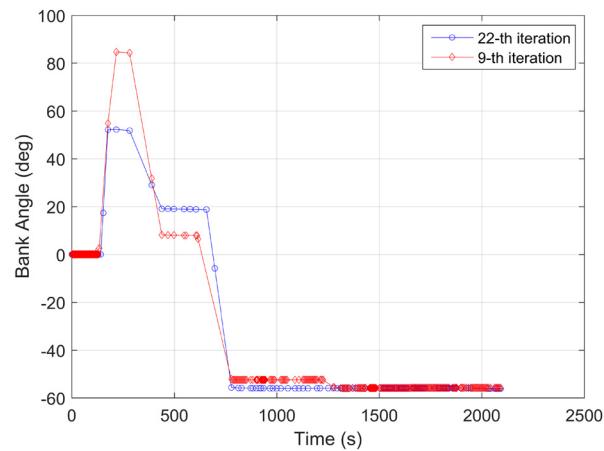


Fig. 25. Comparison of bank angle.

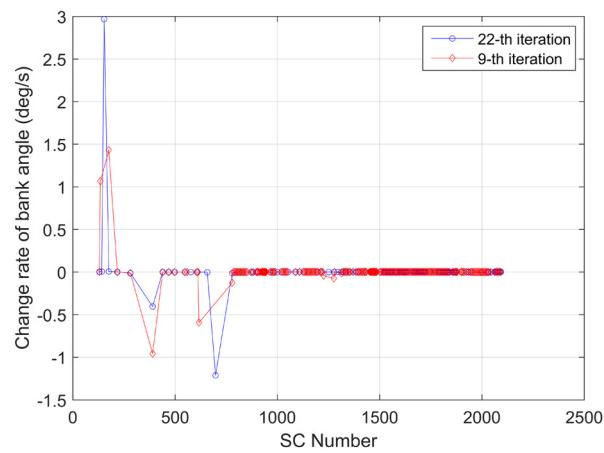


Fig. 26. Comparison of bank angle change rate.

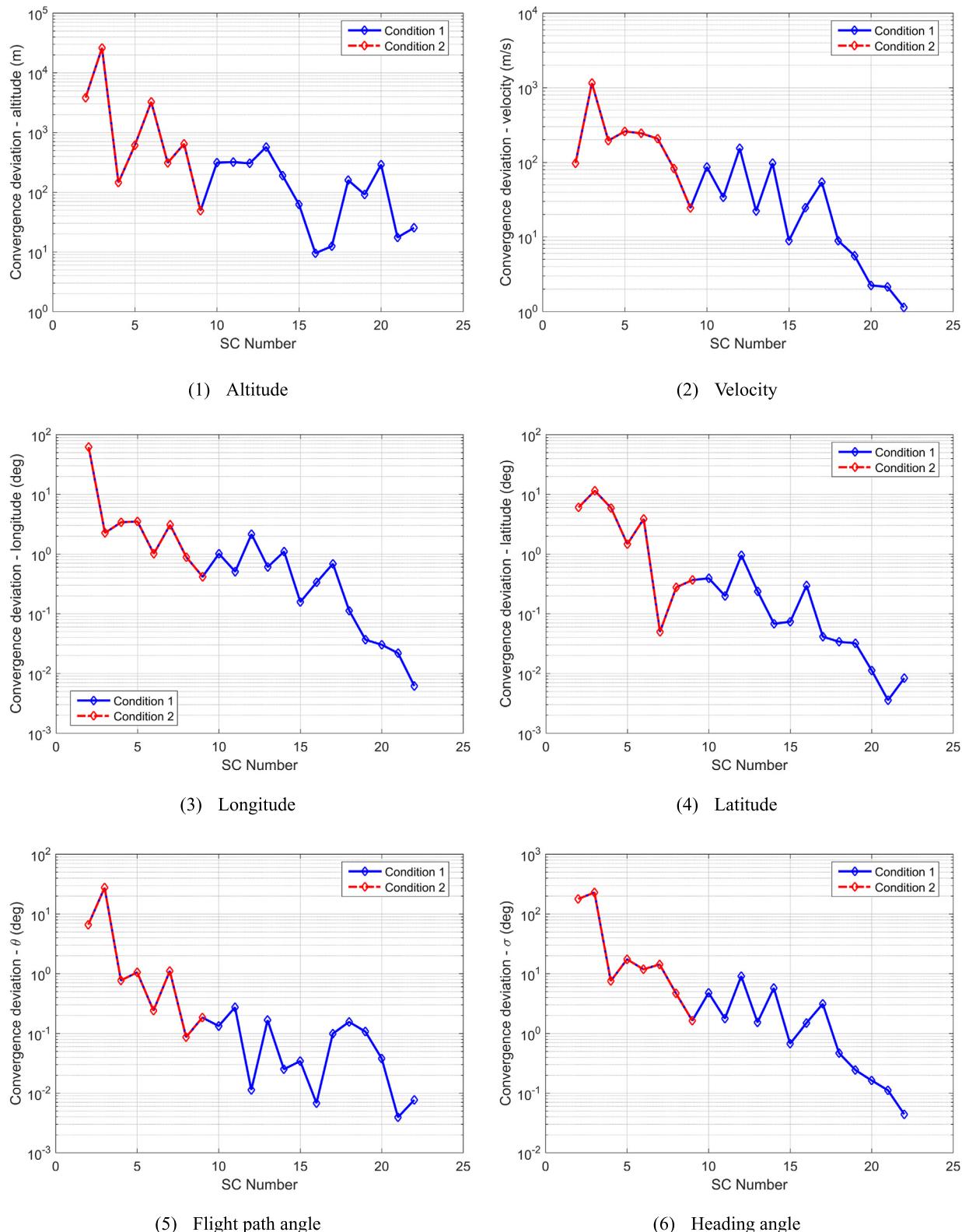
for the proposed SCP method with adaptive mesh, which is represented by condition 2.

In this section, the property of the proposed convergence condition is analyzed, which is compared with the convergence condition given in Eq. (44). The thresholds of two convergence conditions are all set as $\epsilon = \left[\frac{100}{R_0} \quad \frac{0.5\pi}{180} \quad \frac{0.5\pi}{180} \quad \frac{50}{\sqrt{g_0 R_0}} \quad \frac{0.5\pi}{180} \quad \frac{5\pi}{180} \right]$. Other simulation conditions are the same as that in Section 5.1.

The proposed SCP method with adaptive mesh is used to solve the original problem **P1**, and two converged solutions are obtained by the two convergence conditions. The convergence deviations of two conditions are compared in Table 4. It is shown that at 9-th iteration, the convergence condition 2 is satisfied, but the condition 1 is not. At 22-th iteration, the two convergence conditions are satisfied. Thus, the convergence condition 2 can be satisfied with fewer iterations compared with condition 1.

The global errors of the two converged solutions are compared in Table 5. It is shown that the global error of converged solution obtained by condition 2 is mainly consistent with that obtained by condition 1. The two global errors are low, which shows that the violations of nonlinear dynamics constraint are low for the two converged solutions. Thus, the two converged solutions can be considered as feasible solutions for the original problem **P1**.

The iteration process of global error 2-norm $\|\Delta \mathbf{x}_f\|$ is given in Fig. 19. Because the convergence condition only determines whether the SCP method terminates or not, the iteration process obtained by the two convergence conditions are consistent. At 9-th iteration, the convergence condition 2 is satisfied. Before 9-th iteration, the global error has reduced a lot, which indicates that

**Fig. 27.** Comparison of convergence deviation.

the iteration solution moves into the feasible solution space from the infeasible solution space of the original problem **P1**. After 9-th iteration, the global error is still very low, which indicates that the corresponding iteration solution moves in the feasible space.

The two converged discrete solutions obtained by the two convergence conditions are compared in Fig. 20–26, where the 22-th

iteration represents the converged solution obtained by the condition 1, and the 9-th iteration represents the converged solution obtained by the condition 2. It is shown that the path constraints and controls constraints are satisfied. Due to the difference between two convergence conditions, there is a small deviation between the

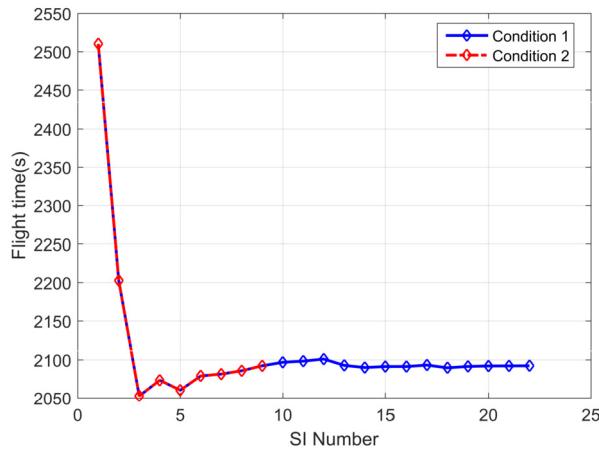


Fig. 28. Comparison of flight time.

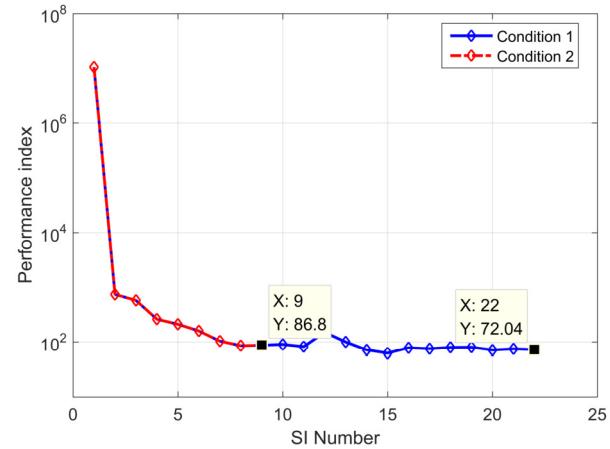


Fig. 30. Comparison of performance index.

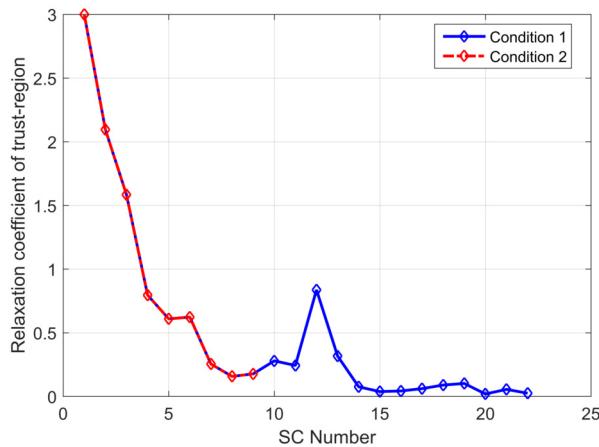


Fig. 29. Comparison of relaxation coefficient.

two converged trajectories. However, the two converged solutions are feasible to the original problem **P1**.

The convergence deviation $|\Delta\mathbf{x}_f^{(k+1)} - \Delta\mathbf{x}_f^{(k)}|$ about six states is given in Fig. 27, including altitude, velocity, longitude, latitude, flight path angle (FPA), heading angle (HA). After 9-th iteration, the convergence deviations continue decreasing, which shows that the global error $\Delta\mathbf{x}_f$ converges to a stable value. The change law about the global error from Fig. 27 is consistent with that obtained by Fig. 19.

The iteration process of flight time t_f is given in Fig. 28. At 9-th iteration, the flight time is 2092.2707 s. At 22-th iteration, the flight time is 2092.2835 s. It is shown that the deviation between the two converged flight time is very low. The iteration process of relaxation coefficient of trust-region η is given in Fig. 29. At 9-th iteration, the coefficient is 0.1766. At 22-th iteration, the coefficient is 0.246. After 9-th iteration, the relaxation coefficient is always small, which shows that the iteration solution has moved into a limited feasible solution space. The iteration process of performance index is given in Fig. 30. At 9-th iteration, the performance index is 86.8. At 22-th iteration, the performance index is 72.04. Therefore, from the perspective of performance index, the converged solution obtained by condition 1 is much closer to the optimal solution for the original problem **P1**. But from the view of feasibility, the two converged solutions are feasible.

The time consumption of solving the original problem **P1** is determined by two factors, including the number of iterations and the time consumption of solving single convex subproblem **P3**. From Table 3, the customized adaptive mesh can make that the

Table 6
Comparison of time consumption.

	Condition 1	Condition 2
Iteration number	22	9
Converged number of mesh points	114	237
Average number of mesh points	220	316
Average time of single subproblem (s)	0.32	0.44
Total time (s)	7.08	4.02

solution time of single convex subproblem **P3** is decreased. The convergence condition has a main influence on the number of iterations. The time consumptions of the converged solutions obtained by the two conditions are compared in Table 6. It is shown that when the convergence condition 2 is satisfied, the number of iterations is small, so the total time consumption is lower. When the condition 1 is satisfied, because the number of iterations is larger, the converged number of mesh points is lower, and the average solution time of single subproblem is lower. Thus, the convergence condition given in Eq. (43) is the main reason of decreasing the total solution time for the original problem **P1**.

6. Conclusions

Sequential convex programming method is a popular direct method to solve the entry trajectory planning problem in recent years. There is a contradiction between the solution accuracy and the computational efficiency for the method. For handling the problem, a novel sequential convex programming method based on customized adaptive mesh refinement is proposed in this paper, which can achieve a high computational efficiency while ensuring the feasibility of converged solution. For ensuring the solution accuracy, the density of mesh points is increased to decrease the local linearization error in highly nonlinear region. For improving the computational efficiency, there are two approaches. Firstly, the density of mesh points is decreased to improve the efficiency of solving single subproblem. Secondly, the new convergence condition is used to terminate the iteration process with fewer iterations. Simulation results show that with the iteration number increasing, the number of mesh points is decreased gradually, and the iteration solution converges to a feasible solution of the original problem. If the iteration process continues, the iteration solution will be closer to the optimal solution. Compared with the decreasing time of solving single subproblem, the decrease of iteration number is the main reason of improving the computational efficiency for the original problem.

Declaration of competing interest

The authors declared that they have no conflicts of interest to this work.

Acknowledgement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

- [1] L. Zang, D. Lin, S. Chen, H. Wang, Y. Ji, An on-line guidance algorithm for high L/D hypersonic reentry vehicles, *Aerosp. Sci. Technol.* 89 (2019) 150–162.
- [2] Z. Li, C. Hu, C. Ding, G. Liu, B. He, Stochastic gradient particle swarm optimization-based entry trajectory rapid planning for hypersonic glide vehicles, *Aerosp. Sci. Technol.* 76 (2018) 176–186.
- [3] P. Lu, Introducing computational guidance and control, *J. Guid. Control Dyn.* 40 (2017) 193.
- [4] Y. Li, B. Pang, C. Wei, N. Cui, Y. Liu, Online trajectory optimization for power system fault of launch vehicles via convex programming, *Aerosp. Sci. Technol.* 98 (2020) 105682, <https://doi.org/10.1016/j.ast.2020.105682>.
- [5] D. Malyuta, T.P. Reynolds, M. Szmuk, M. Mesbah, B. Acikmese, Discretization performance and accuracy analysis for the powered descent guidance problem, in: AIAA Guidance, Navigation, and Control Conference, 2019.
- [6] M.E. Dennis, W.W. Hager, A.V. Rao, Computational method for optimal guidance and control using adaptive Gaussian quadrature collocation, *J. Guid. Control Dyn.* 42 (2019) 2026–2041.
- [7] Y.L. Zhang, Y. Xie, S.C. Peng, G.J. Tang, W.M. Bao, Entry trajectory generation with complex constraints based on three-dimensional acceleration profile, *Aerosp. Sci. Technol.* 91 (2019) 231–240.
- [8] Z. Liang, G. Duan, Z. Ren, Mars entry guidance based on an adaptive reference drag profile, *Adv. Space Res.* 60 (2017) 692–701.
- [9] J. Zhao, R. Zhou, Reentry trajectory optimization for hypersonic vehicle satisfying complex constraints, *Chin. J. Aeronaut.* 26 (2013) 1544–1553.
- [10] J. Wang, H. Liang, Z. Qi, D. Ye, Mapped Chebyshev pseudospectral methods for optimal trajectory planning of differentially flat hypersonic vehicle, *Aerosp. Sci. Technol.* 89 (2019) 420–430.
- [11] S. Boyd, L. Vandenberghe, *Convex Optimization*, Volume 1: Introduction, Cambridge University Press, 2004.
- [12] H. Yang, X. Bai, H. Baoyin, Rapid generation of time-optimal trajectories for asteroid landing via convex optimization, *J. Guid. Control Dyn.* 40 (2017) 628–641.
- [13] H. Jiang, Z. An, Y. Yu, S. Chen, F. Xiong, Cooperative guidance with multiple constraints using convex optimization, *Aerosp. Sci. Technol.* 79 (2018) 426–440.
- [14] F. Wang, S. Yang, F. Xiong, Q. Lin, J. Song, Robust trajectory optimization using polynomial chaos and convex optimization, *Aerosp. Sci. Technol.* 92 (2019) 314–325.
- [15] Z. Wang, Optimal trajectories and normal load analysis of hypersonic glide vehicles via convex optimization, *Aerosp. Sci. Technol.* 87 (2019) 357–368.
- [16] Z. Wang, M.J. Grant, Constrained trajectory optimization for planetary entry via sequential convex programming, *J. Guid. Control Dyn.* 40 (2017) 2603–2615.
- [17] Z. Wang, M.J. Grant, Minimum-fuel low-thrust transfers for spacecraft: a convex approach, *IEEE Trans. Aerosp. Electron. Syst.* 54 (2018) 2274–2290.
- [18] H. Han, D. Qiao, H. Chen, X. Li, Rapid planning for aerocapture trajectory via convex optimization, *Aerosp. Sci. Technol.* 84 (2019) 763–775.
- [19] X.F. Liu, Fuel-optimal rocket landing with aerodynamic controls, in: AIAA Guidance, Navigation, and Control Conference, 2017.
- [20] Z. Wang, S.T. McDonald, Convex relaxation for optimal rendezvous of unmanned aerial and ground vehicles, *Aerosp. Sci. Technol.* 99 (2020) 105756, <https://doi.org/10.1016/j.ast.2020.105756>.
- [21] Y. Mao, M. Szmuk, B. Acikmese, Successive convexification: a superlinearly convergent algorithm for non-convex optimal control problems, *Automatica* (2018), <https://arxiv.org/abs/1804.06539v1>.
- [22] Y. Mao, M. Szmuk, B. Acikmese, Successive convexification of non-convex optimal control problems and its convergence properties, in: 2016 IEEE 55th Conference on Decision and Control (CDC), 2016.
- [23] M. Szmuk, U. Eren, B. Acikmese, Successive convexification for Mars 6-dof powered descent landing guidance, in: AIAA Guidance, Navigation, and Control Conference, 2017.
- [24] Z. Wang, M.J. Grant, Improved sequential convex programming algorithms for entry trajectory optimization, in: AIAA Guidance, Navigation, and Control Conference, 2019.
- [25] Z. Wang, M.J. Grant, Autonomous entry guidance for hypersonic vehicles by convex optimization, *J. Spacecr. Rockets* 55 (2018) 993–1006.
- [26] P. Simplicio, A. Marcos, S. Bennani, Guidance of reusable launchers: improving descent and landing performance, *J. Guid. Control Dyn.* 42 (2019) 2206–2219.
- [27] C. Darby, W. Hager, A. Rao, An hp-adaptive pseudospectral method for solving optimal control problems, *Optim. Control Appl. Methods* 32 (2011) 476–502.
- [28] C. Yu, D. Zhao, Y. Yang, Efficient convex optimization of reentry trajectory via the Chebyshev pseudospectral method, *Int. J. Aerosp. Eng.* 2019 (2019) 1414279, <https://doi.org/10.1155/2019/1414279>.
- [29] M. Patterson, W. Hager, A. Rao, A ph mesh refinement method for optimal control, *Optim. Control Appl. Methods* 36 (2015) 398–421.
- [30] F. Liu, W.W. Hager, A.V. Rao, Adaptive mesh refinement method for optimal control using decay rates of Legendre polynomial coefficients, *IEEE Trans. Control Syst. Technol.* 26 (2017) 1475–1483.
- [31] F. Liu, W.W. Hager, A.V. Rao, Adaptive mesh refinement method for optimal control using nonsmoothness detection and mesh size reduction, *J. Franklin Inst.* 352 (2015) 4081–4106.
- [32] C.L. Darby, W.W. Hager, A.V. Rao, Direct trajectory optimization using a variable low-order adaptive pseudospectral method, *J. Spacecr. Rockets* 48 (2011) 433–445.
- [33] M.A. Patterson, A.V. Rao, GPOPS – II: a MATLAB software for solving multiphase optimal control problems using hp-adaptive Gaussian quadrature collocation methods and sparse nonlinear programming, *ACM Trans. Math. Softw.* 41 (2014) 1–37.
- [34] M. Sagliano, Pseudospectral convex optimization for powered descent and landing, *J. Guid. Control Dyn.* 41 (2018) 320–334.
- [35] M. Sagliano, Generalized hp pseudospectral-convex programming for powered descent and landing, *J. Guid. Control Dyn.* 42 (2019) 1562–1570.
- [36] J. Wang, N. Cui, A pseudospectral-convex optimization algorithm for rocket landing guidance, in: AIAA Guidance, Navigation, and Control Conference, 2018.
- [37] M. Sagliano, E. Mooij, Optimal drag-energy entry guidance via pseudospectral convex optimization, in: AIAA Guidance, Navigation, and Control Conference, 2018.
- [38] J.S. Zhao, S. Li, Mars atmospheric entry trajectory optimization with maximum parachute deployment altitude using adaptive mesh refinement, *Acta Astronaut.* 160 (2019) 401–413.
- [39] X. Zhou, H.B. Zhang, R.Z. He, L. Xie, G.J. Tang, W.M. Bao, An improved solution method via the pole-transformation process for the maximum-crossrange problem, Part G, *J. Aerosp. Eng.* 234 (2020) 1491–1506.
- [40] T.H. Phillips, A common aero vehicle model, description, and employment guide, [www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&docname=GetTRDoc.pdf](http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&docname=GetTRDoc.pdf&ADOCID=GetTRDoc&GetTRDoc.pdf&Location=U2&docname=GetTRDoc.pdf), 2013.
- [41] M.J. Grant, S. Boyd, CVX: Matlab software for disciplined convex programming, version 2.1, <http://cvxr.com/cvx>, 2018.