Quick Trajectory Optimization Method for Mars Airplane Based on Adaptive Gauss Pseudospectral Method

Z Y Sun, Z Y Liu and P Zhang

1 Beijing Institute of Control Engineering, Beijing, China, 100194

*Email: biceszy@sina.com

Abstract. This paper gives a new method based on adaptive Gauss pseudospectral method for quick trajectory optimization problem of Mars airplane. The method converts the trajectory optimization problem to a non-linear programming problem, taking into account for the dynamic constrains, the boundary constrains, the path constrains and the control constrains during the cruise phase of the Mars airplane. And the sequential quadratic programming is adapted to solve the non-linear programming problem. In combination with the adaptive Gauss pseudospectral method and sequential quadratic programming in Matlab, numerical results of the optimal trajectory with 267 collocations could be obtained in 800s with an approximation accuracy of $10^{-6}$.

1. Introduction

Recently, NASA is carrying out a research on a new Mars exploration vehicle named Daedalon, which is a kind of unmanned airplane and can enlarge its platform area in low speed to gain enough lift [1]. This kind of designed Mars exploration vehicle is not only satisfied with the requirement of Mars entry, cruise and landing phases, but also could realize large-range glide to gain more scientific data than traditional Mars exploration vehicles.

The Daedalon flight profile consists of three phases after Mars entry, high-speed flight descent phase, low-speed cruise phase and landing phase. The high-speed flight descent phase begins from Mars atmosphere entry, and then the vehicle discards the heat shield TPS. In this phase, the vehicle’s speed decreases from 5km/s to about 0.5km/s and the height from 120km to about 3km. And in low-speed cruise phase, the vehicle performs a continuous morphing of the wing at about 500m altitude, such that the final low-sweep, high-aspect-ratio configuration to realize a large range cruise. In this phase, the vehicle also surveys the ground in search of the ideal landing site. In landing phase, the vehicle initiates a deep stall at 10m altitude and the thrusters allow it to land gently on its wheels.

Different from traditional Mars exploration vehicles, this kind of vehicle such as Daedalon has the property of nonlinear, parameter-varying and model-varying during the Mars entry, cruise and landing phases. In addition, the dynamic pressure constrains and the control output constrain must be taken into account in the trajectory design process. All of these constrains result in difficulties for the trajectory design. However, the trajectory programming theory, which could give an optimal trajectory of some specified mission under the consideration of the control constrains, the path constrains and boundary constrains, is a good choice to solve the problem.

The trajectory programming problem is an optimization problem essentially, and many fruits about optimization method could be borrowed. The method to solve these optimization problem consists of direct method and indirect method [2]. The indirect method solve the problem to get analytic result.
using maximum principle, but some problem with complicate constrains couldn’t get the analytic result usually. In contrast, direct method is of fast convergence velocity and broad convergence region, which converts the optimization problem to a nonlinear programming problem and then solves the problem with numerical method. Recently, as a kind of direct method, pseudospectral method gained considerable attention in solving complicated optimization problem. In addition, as a kind of pseudospectral method, Gauss pseudospectral method (GPM) has been applied widely in ascent and reentry trajectory programming and lunar exact-landing trajectory optimization with its advantages of high accuracy and fast convergence velocity [3][4][5]. The GPM, like all other pseudospectral methods, approximates the state using a basis of global interpolating polynomials [6]. These global polynomials are based on a set of discrete points across the interval. Adding discrete points is the only method to increase the approximate accuracy, which increases the optimization variable number in nonlinear programming and thus the optimization time. In order to increase the optimization accuracy and reduce the optimization time, Darby introduced an adaptive GPM [7]. In the low accuracy region, the adaptive GPM could increase the segment or colocation points, and in the high accuracy region, it will not make any changes, which could raise the computation efficiency enormously and has been applied on robot trajectory programming and ascent Phase trajectory optimization [8][9].

This paper gives a quick trajectory programming method of Mars airplane as Daedalon in cruise phase adapting the adaptive GPM.

2. Mathematical model
Under in no consideration of the Mars rotation and wind, the Mars airplane dynamic equation can be described as [10],
\begin{align}
\dot{r} &= v \cdot \sin \theta \\
\dot{x} &= v \cdot \cos \theta \cdot \cos \psi \\
\dot{y} &= -v \cdot \cos \theta \cdot \sin \psi \\
\dot{v} &= -D - g \cdot \sin \theta \\
\dot{\theta} &= L \cdot \frac{\cos \sigma}{v} - g \cdot \frac{\cos \theta}{v} \\
\dot{\psi} &= L \cdot \frac{\sin \sigma}{v} \cdot \frac{\cos \theta}{v} 
\end{align}

where \( r \) is the distance between the airplane and the Mars centre, \( v \) the airplane’s velocity, \( \theta \) the flight path angle, \( x \)、\( y \) the distance to the destination, \( \sigma \) the bank angle, \( \psi \) trajectory deflection angle, \( g \) the gravitational acceleration at the Mars surface. And \( L \) and \( D \) are the lift and drag respectively, which can be expressed as,
\begin{align}
D &= \frac{1}{2} \rho v^2 C_D \frac{S}{m} \\
L &= \frac{1}{2} \rho v^2 C_L \frac{S}{m}
\end{align}

where \( C_D \), \( C_L \) are the drag and lift coefficient related to the Mach number and the attack angle [11], \( S \) the vehicle’s area, \( m \) the mass of the vehicle. And \( \rho \) is the density of the Mars atmosphere, which can be expressed as [12],
\[ \rho = \exp \left( \sum_{i=0}^{4} \beta_i h^i \right) \]

where \( h \) is the altitude, \( \beta_i \) is the constant coefficient.

According to the attitude dynamic equation, the velocity and acceleration of the bank angle \( \sigma \) and the attack angle \( \alpha \) can be expressed as,
\[ \dot{\sigma} = \omega_1, \quad \dot{\alpha} = u_1, \quad \dot{\omega}_2 = \alpha = u_2 \]  \hspace{1cm} (10)

where \( \omega_1, \omega_2 \) are the angular velocity of the bank angle \( \sigma \) and the attack angle \( \alpha \) and \( u_1, u_2 \) the acceleration of the bank angle \( \sigma \) and the attack angle \( \alpha \) produced by the control torque from the attitude control system on board.

### 3. Principle of adaptive Gauss pseudospectral method

#### 3.1. Continuous Bolza problem

The trajectory programming problem for Mars airplane could be considered to be a general Bolza optimal control problem that is to find the control variable \( u(t) \in \mathbb{R}^m \), minimizing the Bolza index, under all constrains in practice. The Bolza index can be expressed as,

\[
J = \Phi(x(t_0), t_0, x(t_f), t_f) + \int_{t_0}^{t_f} g(x(t), u(t), t) dt,
\]

where \( x(t) \in \mathbb{R}^n \), the initial time \( t_0 \) and the final time \( t_f \) are satisfied with the following constrains,

1. **Dynamic constrains**

\[
\dot{x}(t) = f(x(t), u(t), t), \quad t \in [t_0, t_f].
\]

The dynamic constrains of the Mars plane consist of (1) ~ (6) and (10).

2. **Path constrain**

\[
C(x(t), u(t), t) \leq 0, \quad t \in [t_0, t_f].
\]

The path constrain is just the dynamic pressure constrain, which can be expressed as,

\[
q = \frac{1}{2} \rho v^2 \leq q_{\text{max}}
\]

where \( q_{\text{max}} \) is the maximum dynamic pressure in the duration.

3. **Boundary conditions**

\[
\phi(x(t_0), t_0, x(t_f), t_f) = 0.
\]

The boundary constrain consists of the initial state constrain and the final state constrain, and can be expressed as,

\[
r(t_0) = r_0, \quad v(t_0) = v_0, \quad \theta(t_0) = \theta_0, \quad x(t_0) = x_0, \quad y(t_0) = y_0, \quad \psi(t_0) = \psi_0
\]

\[
r(t_f) = r_f, \quad v(t_f) = v_f, \quad \theta(t_f) = \theta_f, \quad x(t_f) = x_f, \quad y(t_f) = y_f, \quad \psi(t_f) = \psi_f
\]

The optimal trajectory of the state variable and control variable could be gained by solving the equation set inclusive of Bolza index, dynamic constrains, path constrain and boundary constrains.

#### 3.2. Adaptive Gauss pseudospectral method

GPM converts the above continuous Bolza problem to a discrete optimal problem by approximating the state variable and control variable using Lagrange interpolation in a series of separate points. The time derivative of the state variable can be expressed by the time derivative of global interpolation polynomial, the integration item in Bolza index by Gauss integration equation and the final state constrain by integration during the total process from initial state. Under the above numerical approximation, the continual optimal control problem could be converted to be a discrete and nonlinear programming problem with some constrains.

In GPM, the Lagrange polynomial is constructed at the Legendre-Gauss points, initial point and final point. And the state variable and control variable could be approximated as [7],

\[
\sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\partial \Phi}{\partial x_{ij}} = 0
\]
\[
x(\tau) \approx X(\tau) = \sum_{i=0}^{K} X_i \ell_i(\tau), \tag{18}
\]
\[
u(\tau) \approx U(\tau) = \sum_{i=1}^{K} U_i \tilde{\ell}_i(\tau), \tag{19}
\]

where, \( X_i = X(\tau_i) \), \( U_i = U(\tau_i) \), and \( \ell_i(\tau) \) is the Lagrange basis function.

In combination with equation (18), the dynamic constrains in equation (12) can be expressed as,

\[
DX - \frac{t_f - t_0}{2} F(X^{LG}, U^{LG}, \tau^{LG}, t_0, t_f) = 0, \tag{20}
\]

where the differential matrix \( \mathbf{D} \in \mathbb{R}^{k \times (k+1)} \).

The final state \( X_f \) should meet the dynamic constrains,

\[
x(\tau_f) = x(\tau_0) + \int_{\tau_0}^{\tau_f} f(x(\tau), u(\tau), \tau) d\tau, \tag{21}
\]

According to the Gauss-Lobatto formula, equation (21) could be expressed as,

\[
X_{N+1} = X(\tau_{N+1}) = X_0 + \frac{t_f - t_0}{2} \mathbf{w}^T \mathbf{F}, \tag{22}
\]

where \( \tau_i \) are LG points, \( \mathbf{w}^T \) is the Gauss integration weight.

And thus, the Bolza index can be expressed as,

\[
J = \Phi(X_0, X_f, t_0, t_f) + \frac{t_f - t_0}{2} \sum_{i=1}^{K} \omega_i g(X_i, U_i, \tau_i, t_0, t_f), \tag{23}
\]

The boundary constrain and path constrain can be described at LG points as,

\[
\phi(X_0, \tau_0, X_{N+1}, \tau_f) = 0, \tag{24}
\]

\[
C(X_k, U_k, \tau_k, t_0, t_f) \leq 0. \tag{25}
\]

After the above numerical approximation, the continuous Bolza problem can be converted to a discrete and nonlinear programming problem, to find the discrete state variables \( (X_1, X_2, \ldots, X_K) \) and control variables \( (U_1, U_2, \ldots, U_K) \) at LG points, which both are under the constrains of (20), (24) and (25) and minimize the index expressed in (23).

The above described global GPM can’t heighten precision except for adding more LG points, and this in turn prolong the computation time extensively. In order to taking into account both the precision and the computation time, a new adaptive GPM is introduced in [7], which could decide whether to add LG points or segments according to the approximating precision. In contrast to the above global GPM, the adaptive GPM divides the NLP into some segments, and applies the GPM in each segment. Assuming there are \( S \) segments and \( N \) LG points.

And in this condition, the dynamics in (20) can be expressed as,

\[
\begin{bmatrix}
D_1 & 0 & L & 0 \\
0 & D_2 & L & 0 \\
M & O & M & \\
0 & \ldots & 0 & D_s
\end{bmatrix}
\begin{bmatrix}
t_1 - t_0 \\
t_2 - t_1 \\
0 \\
\ldots \\
0
\end{bmatrix} = \frac{1}{2}
\begin{bmatrix}
t_1 - t_0 & 0 & L & 0 \\
0 & t_2 - t_1 & L & 0 \\
M & O & M \\
0 & \ldots & 0 & t_s - t_{s-1}
\end{bmatrix}
\begin{bmatrix}
X \\
F
\end{bmatrix}, \tag{26}
\]
Obviously, the derivative matrix in (27) is sparser than the one in (24), and the sparse matrix helps to heighten computation efficiency and decrease computation time. The adaptive GPM adds segments or LG points in low precision area and does nothing in the area where the precision is enough. And hence, this method can heighten the computational efficiency at the same time assuring the precision.

At the low precision area, the adaptive GPM decides whether to add segments or LG points according to the midpoint residuals matrix. Each middle time it can be expressed as,

\[ \tilde{t}_i = \frac{t_i + t_{i+1}}{2}, \quad (i = 1, \ldots, N_s - 1), \]  
(27)

Define \( \tilde{X} \in \mathbb{R}^{(N_s-1)\times n} \), \( \tilde{U} \in \mathbb{R}^{(N_s-1)\times m} \) as the state variable and control variable at \( \tilde{t}_i \), and then the midpoint residuals matrix \( \mathbf{R} \) can be expressed as,

\[ \mathbf{R} = \frac{\partial \tilde{X}}{\partial \tilde{t}} \left[ \begin{array}{c} 0_{(N_s-1)\times n} \\ \mathbf{F} \left( \tilde{X}, \tilde{U}, \tilde{t}, t_{s}, t_{s-1} \right) \end{array} \right] \in \mathbb{R}^{(N_s-1)\times n}, \]  
(28)

where \( \partial \tilde{X} \) is the derivative matrix at \( \tilde{t} = (\tilde{t}_1, \ldots, \tilde{t}_{N_s-1}) \).

Define the scaled midpoint residual vector \( \mathbf{r} \) as,

\[ \mathbf{r} = \frac{\mathbf{r}}{\mathbf{r}} = \begin{bmatrix} \beta(\tilde{t}_1), \ldots, \beta(\tilde{t}_{N_s-1}) \end{bmatrix} = \mathbf{r} / \mathbf{r}, \]  
(29)

where \( \mathbf{r} \in \mathbb{R}^{(N_s-1)\times 1} \) is the column matrix with biggest residuals in matrix \( \mathbf{R} \) and \( \mathbf{r} \) can be expressed as,

\[ \mathbf{r} = \sum_{i=1}^{N_s-1} r(t_i), \]  
(30)

Define \( \varepsilon \) as the tolerance and consider the condition where the maximum residual is bigger than \( \varepsilon \). When the element of uniform-type errors in \( \mathbf{r} \) is bigger than the threshold \( \rho (\rho > 1) \), we add LG points to heighten the approximated precision, that is, \( N_s' = N_s + L \), where \( L \) is the LG points to be added.

The adaptive GPM adds LG points or segments at low precision region to heighten the precision. And it performs well in both approximation precision and computation efficiency for non-smooth optimization problems.

4. Numerical results
Taking into consideration the limited fuel of the Mars airplane, the Bolza index of the trajectory programming is selected to be the fuel consumed during the cruise phase, which can be expressed as followed,

\[ J = \int_{t_0}^{t_f} uu^T dt, \]  
(31)

where \( u \) is the control variable, satisfied with the equation \( u = [\dot{\sigma}, \dot{\alpha}] \).

The control constrain consists mainly of the maneuver capability of the vehicle, inclusive of the bank acceleration constrain and the attack angle constrain, which can be expressed as,

\[ |\dot{\sigma}| \leq \dot{\sigma}_{\max}, \quad |\dot{\alpha}| \leq \dot{\alpha}_{\max} \]  
(32)

The state variable is selected to be \( x = [r, \theta, x, y, v, \psi, \sigma, \dot{\sigma}, \alpha, \dot{\alpha}] \).

The dynamic equation is listed in (1) ~ (6) and (8). The path constrains and the boundary constrains are listed in (14), (16) ~ (17) respectively.

The parameters in the trajectory programming are listed in Table 1~3.
Table 1 Physical parameters in simulation

| Parameter                  | Value          |
|----------------------------|----------------|
| Radius of Mars (km)        | 3397           |
| Gravitational constant     | $4.283 \times 10^{13}$ |
| Mass of the vehicle (kg)   | 600            |
| Heat flux coefficient      | $8.49 \times 10^{-8}$ |

Table 2 Boundary parameters in simulation

| Parameter | Initial Value | Final Value |
|-----------|---------------|-------------|
| $r$ (km)  | 3504.1        | 3400.08     |
| $v$ (m)   | 500           | 160.5       |
| $x$ (m)   | -20000        | -4282.35    |
| $y$ (m)   | 70000         | 98.23       |
| $\theta$ (°) | 0.192        | 0.03        |
| $\psi$ (°) | 90           | 0.006       |
| $\sigma$ (°) | 0            | 0           |
| $\dot{\sigma}$ (/s) | 0          | 0           |
| $\alpha$ (°) | 0            | 0           |
| $\dot{\alpha}$ (/s) | 0          | 0           |

Table 3 Path and control constrain parameters in simulation

| Parameter                        | Value          |
|----------------------------------|----------------|
| Maximum bank angle $\sigma_{\text{max}}$ (°) | 90             |
| Maximum angular velocity of the bank angle $\dot{\sigma}_{\text{max}}$ (/s) | 5              |
| Maximum angular acceleration of the bank angle $\ddot{\sigma}_{\text{max}}$ (/s²) | 3.36           |
| Maximum attack angle $\alpha_{\text{max}}$ (°) | 15             |
| Maximum angular velocity of the attack angle $\dot{\alpha}_{\text{max}}$ (/s) | 5              |
| Maximum angular acceleration of the attack angle $\ddot{\alpha}_{\text{max}}$ (/s²) | 3.36           |
| Maximum dynamic pressure $q_{\text{max}}$ (Pa) | 1000           |

Figure 1. Optimal trajectory of the distance

Figure 2. Optimal trajectory of the velocity
The trajectory programming problem is run on a computer with Matlab R2017b and GPOPS II (V1.0) software [13]. The minimum number of the collocation point in each segment is 4 and the maximum error tolerance is $10^{-6}$. The GPOPS software adopted sequential quadratic programming to
solve the nonlinear programming problem, and gave the optimal reference trajectory with 257 collocation points in about 800 seconds after 21 iterations. And the trajectory programming results are shown in Figure 1–Figure 10. Figure 1–Figure 5 are the optimal trajectories of some state variables, and suggest that the Mars airplane could fly from the initial point to the destination accurately by tuning the bank angle and the attack angle as shown in Figure 6–figure 7 in spite of all the constrains. And the cruise phase lasts about 240s. Fig. 8 is the dynamic pressure trajectory, which is satisfied with the path constrains. Figure 9–figure 10 are the angular acceleration of the bank and attack angle, which is also under the control constrains. In conclusion, the adaptive Gauss pseudospectral method solves the trajectory programming problem with complicated constrains for Mars airplane perfectly.

5. Conclusions
In combination with the adaptive Gauss pseudospectral method and sequential quadratic programming, this paper gives a new method to solve the trajectory programming problem of the Mars airplane quickly. This method takes into account all constrains inclusive of path constrains, boundary constrains and control constrains, and gives the optimal trajectory of the Mars airplane in cruise phase. Numerical results suggest that the high accuracy and satisfying all constrains trajectory could be produced in Matlab quickly, and thus the method introduced in this paper is of great practical application.

Acknowledgement
This work is supported by National Nature Science Foundation under Grant 61803029.

References
[1] Jarret M L, John R O, Robert D B. Daedalon: A Revolutionary Morphing Spacecraft Design for Planetary Exploration [C]. 1st Space Exploration Conference: Continuing the Voyage of Discovery 30 January -1 February, 2005, Orlando, Florida.
[2] Yong En-mi, Chen Lei, Tang Guo-jin. A survey of numerical methods for trajectory optimization of spacecraft [J]. Journal of Astronautics, 2008, 29(2): 397-406.
[3] Yang Xi-xiang, Zhang Wie-hua. Rapid optimization of ascent trajectory for solid launch vehicles based on GPM [J]. Journal of Astronautics, 2011, 32(1): 15-21.
[4] Zhou Wenya, Yang Di, Li Shunli. Using Gauss Pseudospectral Method to Solve Optimal Reentry Trajectory for Lifting Vehicle [J]. Journal of Nanjing University of Science and Technology, 2010, 34(1): 85-90.
[5] Peng Qi-bo, Li Hai-yang, Shen Hong-xin. Rapid lunar exact-landing trajectory optimization via GPM [J]. Journal of Astronautics, 2010, 31(4): 1012-1016.
[6] Benson D A, Huntington G T. Direct trajectory optimization and costate estimation via an orthogonal collocation method [J]. Journal of Guidance Control, and Dynamics, 2006, 29(6): 1435-1440.
[7] Darby C L, Hager W W, Rao A V. An hp-adaptive pseudospectral method for solving optimal control problems [J], Optimal Control Applications and Methods, 2011, 32(4): 476-502.
[8] Zeng Xiangxin, Cui Naigang, Guo Jifeng, Path planning of space robot based on hp-adaptive pseudospectral method [J], Robot, 2018, 40(3): 385-342.
[9] Liu Ruifan, Yu Yunfeng, Yan Binbin, Ascent Phase Trajectory Optimization for Hypersonic Vehicle Based on hp-Adaptive Pseudospectral Method [J], Journal of Northwestern Polytechnic University, 2016, 10(5): 790-797.
[10] Vinh N X. Optimal trajectories in atmospheric flight [M], New York: Elsevier Scientific Publishing Co., 1981.
[11] Ming Yin, Coordinate control of deformation and flight for morphing aircraft [D], Nanjing University of Aeronautics and Astronautics, 2015.
[12] Benito Manrique J. Advances in spacecraft atmospheric entry guidance [D]. University of California, Irvine, 2010.
[13] www.gpops2.com