计算机科学
COMPUTER SCIENCE

基于OPENMP的再入飞行器轨迹多重打靶法并行计算

李思瑶, 李尚林, 罗井知

引用本文

李思瑶, 李尚林, 罗井知. 基于OPENMP的再入飞行器轨迹多重打靶法并行计算[J]. 计算机科学, 2024, 51(11A): 231000019-6.

LI Siyao, LI Shanglin, LUO Jingzhi. Parallel Computing of Reentry Vehicle Trajectory by Multiple Shooting Method Based onOPENMP [J]. Computer Science, 2024, 51(11A): 231000019-6.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.231000019

基于 OPENMP 的再入飞行器轨迹多重打靶法并行计算

李思瑶[1,2,3]　李尚林[1,2]　罗井知[4]
1 湘南学院计算机与人工智能学院　湖南 郴州 423000
2 湘南学院先进嵌入式计算技术与智能医疗系统湖南省工程研究中心　湖南 郴州 423000
3 中山大学航空航天学院　广东 深圳 518107
4 中南林业科技大学涉外学院信息与工程学院　长沙 410211
(307144779@qq.com)

摘　要　实现基于多重打靶法将助推滑翔飞行器的轨迹优化问题变成非线性规划问题,用多重打靶法对三自由度再入轨迹进行优化,使用序列二次规划优化器进行优化,同时使用 openmp 进行并行计算,可以对等式约束中的积分进行并行计算。使用多重打靶法的再入轨迹优化算法,对模型进行并行计算。在 MATLAB 版本上使用的优化方法是内点法,而在 C 上面使用的优化方法是序列二次规划算法,C 上面的程序是根据 MATLAB 转换的。仿真实验选取 CAV-H 模型进行计算,并行计算利用 openmp 获得了 8.398 倍加速比。多重打靶法与直接打靶法的结果基本一致,最小吸热量的多重打靶法的吸热量与以最小吸热量为目标函数的直接打靶法相差不多。通过仿真得出结论,线程数目在 13 时加速比最大,平均相对效率在 93%以上。

关键词:多重打靶法;内点法;序列二次规划;并行计算
中图分类号　V448.2

Parallel Computing of Reentry Vehicle Trajectory by Multiple Shooting Method Based on OPENMP

LI Siyao[1,2,3],LI Shanglin[1,2] and LUO Jingzhi[4]
1 School of Computer Science and Artificial Intelligence,Xiangnan University,Chenzhou,Hunan 423000,China
2 School of Information and Engineering,Swan College,Central South University of Forestry and Technology,Changsha,Hunan 423000,China
3 School of Astronautics and Aeronautics,Sun Yat-sen University,Shenzhen,Guangdong 518107,China
4 School of Information and Engineering,Swan College,Central South University of Forestry and Technologe,Changsha 410211,China

Abstract　The implementation is based on the multiple target method. By turning the trajectory optimization problem of a booster glider into a nonlinear programming problem,optimizing a three degree of freedom reentry trajectory,using openmp with sequence quadratic programming optimizer,it is possible to perform parallel computing on the integrals in equation constraints. Using multiple target methods constructs a reentry trajectory optimization algorithm,and performs parallel computation on the model. The optimization method used on the MATLAB version is the interior point method,while the optimization method used on C is the sequential quadratic programming algorithm. The above program is converted based on MATLAB. The CAV-H model is selected for computing in the simulation experiment. Parallel computing achieves 8.398 times the acceleration ratio using openmp. The results of the multiple target method and the direct target method are basically consistent. The heat absorption capacity of the minimum heat absorption multiple target method is not much different from that of the direct target method with the minimum heat absorption as the objective function. Through simulation,the acceleration ratio is the highest when the number of threads is 13,and the average relative efficiency is more than 93%.

Keywords　Multiple target method,Interior point method,Sequential quadratic programming,Parallel computing

多重打靶法对模型进行转化,将无数维的 question 转变为有数维的参数 question,使用 openmp 进行并行计算,优化出合适且可行的控制变量值,使得目标函数指标是最好的,并且得到加速效果。仿真的部位使用 CAV-H 助推滑翔式再入导弹飞行器[1],运行多组的仿真验证,证明多重打靶法的有效性和效率。Wang[2]对多重打靶法进行了实现,浙江大学刘兴高教授团队在多重打靶法方向进行了大量研究,华中科技大学卓琳人利用多重打靶法进行了飞行器再入研究[3]。

本文使用了多重打靶法对 CAV-H[4]飞行器进行了优化,并且利用优化算法可以使用并行的特点使用了 openmp 并行实现,CAV-H 是一个美国的比较经典的助推滑翔飞行器,多重打靶法在 matlab 实现上使用了内点法[5]进行优化求解,在 C 的实现上使用了序列二次规划算法。本文将 muitishooting

通信作者:李尚林(lsl@xnu.edu.cn)

与 direct-shooting 进行了对比，以验证 multishooting 的可行性。过程约束满足 heat flow，overload，dynamic presure 算法。动力学模型使用了三自由度再入动力学模型[6]，但积分对象是能量而不是时间，因为时间是不确定的，而最后状态点的能量是固定的。

1 多重打靶法

多重打靶法是本文使用的优化算法，是一种能够并行的算法。与 direct-shooting 算法进行对比，multi-shooting 算法具有可以并行的优点。

1.1 基本原理

与直接打靶法进行对比，多重打靶法具有以下优点：

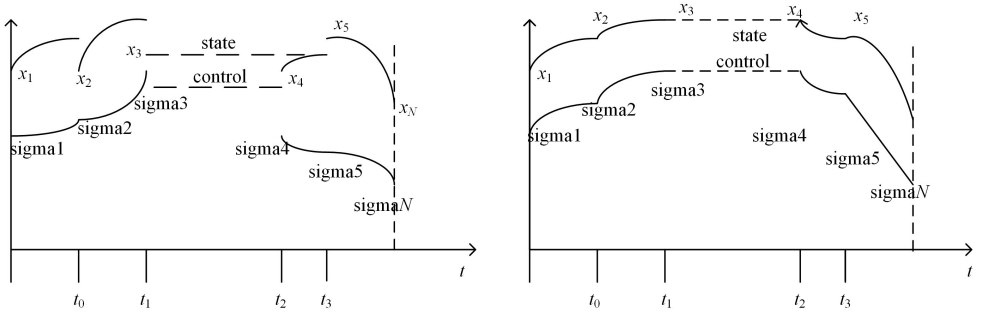


图 1 多重打靶法数值

Fig. 1 Numerical value of multiple shooting method

1.2 龙格库塔数值积分

龙格库塔算法是一种常用的解微分[7]方程的方法，通过利用多重打靶法计算出决策变量以后，通过采用数值积分方法来对状态量进行数值积分，本文采取的数值积分方法是经典的四级四阶龙格库塔数值积分方法。

$$\begin{aligned}
&y_{n+1}=y_n+\frac{h}{6}(k_1+2k_2+2k_3+k_4)\\
&k_1=f(x_n,y_n)\\
&k_2=f\left(x_n+\frac{h}{2},y_n+\frac{h}{2}k_1\right)\\
&k_3=f\left(x_n+\frac{h}{2},y_n+\frac{h}{2}k_2\right)\\
&k_4=f(x_n+h,y_n+hk_1)
\end{aligned}\tag{1}$$

其中，h 是积分步长，x_n 是自变量，y_n 是上一步的状态变量，y_{n+1}是下一步的状态变量，$f(x,y)=y'$是微分方程。

1.3 滑翔式再入飞行器的运动学模型

因为 CAV-H 飞行器正常的再入飞行过程[8]非常复杂，所以要确切地描述飞行器的三自由度运动模型非常困难。在合理的精度区域内，为了精简方程以方便计算，作出以下设定：

1）地球是一个不断自转的均匀椭球体，它的自转角速度 ω_e 为常数。

2）地球大气与地球比较是静止的，而且在同一高度上的大气密度以及大气压力等因素呈现分布均匀。

3）CAV-H 飞行器是无动力返回的理想刚体，它的质量不会缩小。

4）CAV-H 飞行器的侧滑角 β 为 0。

5）CAV-H 飞行器时时刻刻处在力矩瞬时的平衡情况。

1）多重打靶法在不一样的区间内的积分流程是解耦的，能够确保同步以及实施并行运算，增加计算的效率。

2）多重打靶法在整个计算中的状态量是通过插值计算来获得的，而不是依靠积分得出，缩短了大量的计算消耗。

3）面对找不到可行解空间的问题时，多重打靶法确保了除初始端间隔的其他区域内能够寻得可行解。但是直接打靶法在这种情况下在整个区域内都不能寻得可行解。

4）在参数问题求取的迭代初始时，通过积分求得的状态量有可能会出现越过边界的情况，而多重打靶法的状态量被限制在决策向量区域内，一直保持在状态量边界区域内。

归一化又称数据的无量纲化或者标准化以及规格化，是一个使用简单的数学转换来缩小各个状态量之间不同量纲产生的差异的步骤。无量纲化物理模型，常常使得方程的物理参数为 1，解出结果以后，再使用相关的关系式进行往回代入，以求得原量纲的状态量。而回代后的方程结果和原来方程的解是相同的，还可避免求解过程中出现系数不同的问题。

状态的变量在规模上差许多数量级别，为了防止产生奇异的情况，化简后续问题的理论计算，以下是无量纲的形式：

$$\begin{aligned}
&R=\frac{r}{r_0},\\
&V=\frac{v}{\sqrt{g_0 r_0}},\\
&\tau=\frac{t}{\sqrt{r_0/g_0}},\\
&\omega=\frac{\omega_e}{\sqrt{g_0/r_0}},\\
&L=\frac{F_L}{mg_0}=\frac{r_0\rho(R)V^2S_{ref}C_L}{2m},\\
&D=\frac{F_D}{mg_0}=\frac{r_0\rho(R)V^2S_{ref}C_D}{2m},\\
&\rho(R)=\rho_0 e^{\frac{r_0(1-R)}{h_s}}
\end{aligned}\tag{2}$$

最优化问题在积分计算的过程中需要确定积分时间。然而，某些轨迹最优问题无法确定终端的时间，即便已知初始时间，相关的积分区域也无法得知。为了确保不出现积分区域对时间的不确定性的情况，通常引入能量 e 来代替时间 t 作为自变量。e 是 CAV-H 飞行器的单位质量所含有的负值机械能，它的零势能在无尽处。

$$\begin{cases}\dfrac{\mathrm{d}R}{\mathrm{d}e}=V\sin\gamma\,(VD)^{-1}\\ \dfrac{\mathrm{d}\theta}{\mathrm{d}e}=\dfrac{V\cos\gamma\sin\psi}{R\cos\phi}(VD)^{-1}\\ \dfrac{\mathrm{d}\varphi}{\mathrm{d}e}=\dfrac{V\cos\gamma\cos\psi}{R}(VD)^{-1}\\ \dfrac{\mathrm{d}V}{\mathrm{d}e}=\left[-D-\left(\dfrac{\sin\gamma}{R^2}\right)+\varphi_{V3}\right](VD)^{-1}\\ \dfrac{\mathrm{d}\gamma}{\mathrm{d}e}=\dfrac{1}{V}\left[L\cos\sigma+\left(V^2-\dfrac{1}{R}\right)\left(\dfrac{\cos\gamma}{R}\right)+\varphi_{\gamma3}+\varphi_{\gamma4}\right](VD)^{-1}\\ \dfrac{\mathrm{d}\psi}{\mathrm{d}e}=\dfrac{1}{V}\left[\dfrac{L\sin\sigma}{\cos\gamma}+\dfrac{V^2\cos\gamma\sin\psi\tan\varphi}{R}-\varphi_{\gamma3}+\varphi_{\gamma4}\right](VD)^{-1}\end{cases}\tag{3}$$

其中，

$$\begin{cases}\varphi_{V3}=\omega^2R\cos\phi(\sin\gamma\cos\phi-\cos\gamma\sin\phi\cos\psi)\\ \varphi_{\gamma3}=2\omega V\cos\phi\sin\psi\\ \varphi_{\gamma4}=\omega^2R\cos\phi(\cos\gamma\cos\phi+\sin\gamma\cos\psi\sin\phi)\\ \varphi_{\psi3}=2\omega V(\tan\gamma\cos\psi\cos\phi-\sin\phi)\\ \varphi_{\psi4}=\dfrac{\omega^2R\sin\psi\sin\phi\cos\phi}{\cos\gamma}\end{cases}\tag{4}$$

其中，r 表示 CAV-H 飞行器的质心到地球中心的距离；θ 表示 CAV-H 飞行器所处的经度值；ϕ 表示 CAV-H 飞行器所处点的地理纬度值；v 表示 CAV-H 飞行器的飞行速度值；γ 表示 CAV-H 飞行器的航迹倾角（又称速度倾角），是速度的方向虚线与地平面的夹角角度；ψ 表示 CAV-H 飞行器的航向角度，是当地经度的方向线与速度的延伸线在水平面投影的夹角。σ 代表倾侧角，α 代表攻角，隐含在气动力方程中，这两个物理变量是控制的变量。ω_e 为地球的自转的角速度值，m 是 CAV-H 飞行器的质量。

控制向量：二自由度控制量 $u=[\sigma,\alpha]^{\mathrm{T}}$，其中 σ 为侧倾角，α 为攻角，与气动力（升力和阻力）相关。

CAV-H 飞行器单位的质量拥有的负值机械能为：

$$e=\frac{1}{R}-\frac{V^2}{2}$$

约束条件的介绍如下。

过程约束（刚性约束）：热流、动压和过载。

$$\begin{cases}\dot{Q}\leqslant k\rho^{0.5}V^{3.15}\leqslant\dot{Q}_{\max}\\ q=\rho V^2/2\leqslant q_{\max}\\ n=\sqrt{L^2+D^2}/g_0\leqslant n_{\max}\end{cases}\tag{5}$$

1.4 内点法的介绍

内点法（Interior Point Method）是一个解 linear 规划或者非线性的基于凸优化问题的算法。在 MATLAB 中由于 fmincon 使用比较简单，因此使用内点法进行优化。

MATLAB 中的算子 fmincon 可以用来求多元的函数的极值，它的约束包括 5 种：1）线性的不等式约束；2）线性的等式约束；3）变量的约束；4）非线性的不等式约束；5）非线性的等式约束。其形式如下：

$x=f$ min $con(fun,x0,A,b,Aeq,beq,lb,ub,noncon)$

$\min f(x)$

s. t. $Ax\leqslant b$（线性不等式约束）

$Aeqx=beq$（线性等式约束）

$G(x)\leqslant 0$（非线性不等式约束）

$Ceq(X)=0$（非线性等式约束）

$Ib\leqslant X\leqslant ub$（变量约束）

1.5 序列二次规划算法

在 MATLAB 实现多重打靶法之后由于在 C 语言上运行速度更快，因此需要将 MATLAB 程序转换为 C，而在 C 中不是用 fmincon 中的内点法，而是使用 SQP 算法。这里应该描述 SQP 算法的全称，以及实现细节。

求解非线性规划问题的参数的优化方法有多种，这里序列二次规划方法是一种比较有效的高精度的数值方法，在很多较好的优化软件中都被使用。这里应该描述序列二次规划算法的细节。

2 仿真结果

2.1 多重打靶法的仿真计算

本文的参数设置如下：质量 $m=907$ kg，参考面积 $S_{ref}=0.4839\ \mathrm{m}^2$，海平面重力加速度 $g_0=9.81\ \mathrm{m/s}^2$，地球平均半径 $r_0=6378$ km，热模型相关的常系数 $k_h=7.9686\times10^{-5}$，热模型相关的常指数 $k_c=3.25$，海平面大气密度 $\rho_0=1.225\ \mathrm{kg/m}^3$，标量高度系数 $h_s=7200.1$ m。仿真计算平台配置如表 1 所列。

表 1 仿真平台配置

Table 1 Simulation platform configuration

Cpu	显卡	系统	内存	使用软件
银牌 4214	Gtx2080ti	Win10	64 GB	MATLAB2020b

控制量约束如下：侧倾角及其变化率有界，攻角范围及其变化率有界。

状态量约束如下：末端时刻高度和速度在误差允许范围内，终端纵程严格等于设定值。

优化目标为最小化吸热量。

吸热量的表达式如下：

$$\begin{aligned}J&=\int_{t_0}^{t_f}\dot{Q}\mathrm{d}t\\&=\int_{\tau_0}^{\tau_f}\dot{Q}\cdot\sqrt{r_0/g_0}\,\mathrm{d}\tau\\&=\int_{e_0}^{e_f}\dot{Q}\cdot\frac{\sqrt{r_0/g_0}}{VD}\mathrm{d}e\\&=\int_{e_0}^{e_f}k_Q\sqrt{g_0R_0}^{3.15}\sqrt{\rho}\cdot V^{3.15}\frac{\sqrt{r_0/g_0}}{VD}\mathrm{d}e\end{aligned}\tag{7}$$

状态的向量：$\boldsymbol{x}=[R,\theta,\varphi,V,\gamma,\psi]^{\mathrm{T}}$，归一化的地心距离、经度、纬度、归一化的速度、速度倾角。

初始时刻条件：

$$\boldsymbol{x}(t_0)=\left[\frac{r(t_0)}{r_0},\theta(t_0),\varphi(t_0),\frac{v(t_0)}{\sqrt{g_0r_0}},\gamma(t_0),\psi(t_0)\right]^{\mathrm{T}},$$

$r(t_0)=60$ km，$\theta(t_0)=0$ rad，$\phi(t_0)=0$ rad，$v(t_0)=5\,000$ m/s，$\gamma(t_0)=0$ rad，$\psi(t_0)=0$ rad，$e(t_0)=\dfrac{1}{R(t_0)}-\dfrac{V(t_0)^2}{2}$。

末端时刻条件如下：

$$\boldsymbol{x}(t_f)=\left[\frac{r(t_f)}{r_0},\theta(t_f),\varphi(t_f),\frac{v(t_f)}{\sqrt{g_0 r_0}},\gamma(t_f),\psi(t_f)\right]^{\mathrm{T}},$$

$r(t_f)=30\,\mathrm{km}$，$v(t_f)=1\,500\,\mathrm{m/s}$，$e(t_f)=\dfrac{1}{R(t_f)}-\dfrac{V(t_f)^2}{2}$。

过程约束参数：最大热流 $Q_{s\max}=5\,000\,kW/m^2$，最大动压 $q_{\max}=30\,000\,\mathrm{Pa}$，最大过载 $n_{\max}=2g_0$。

时间积分计算末端时刻 t_f：$\dfrac{\mathrm{d}e}{\mathrm{d}\tau}=VD$，$\dfrac{\mathrm{d}\tau}{\mathrm{d}e}=\dfrac{1}{VD}$，初始时刻 $\tau(t_0)=0\,\mathrm{s}$。

选取能量为自变量，能量是等分，时间是对能量取积分计算得到的，不一定是等间隔。

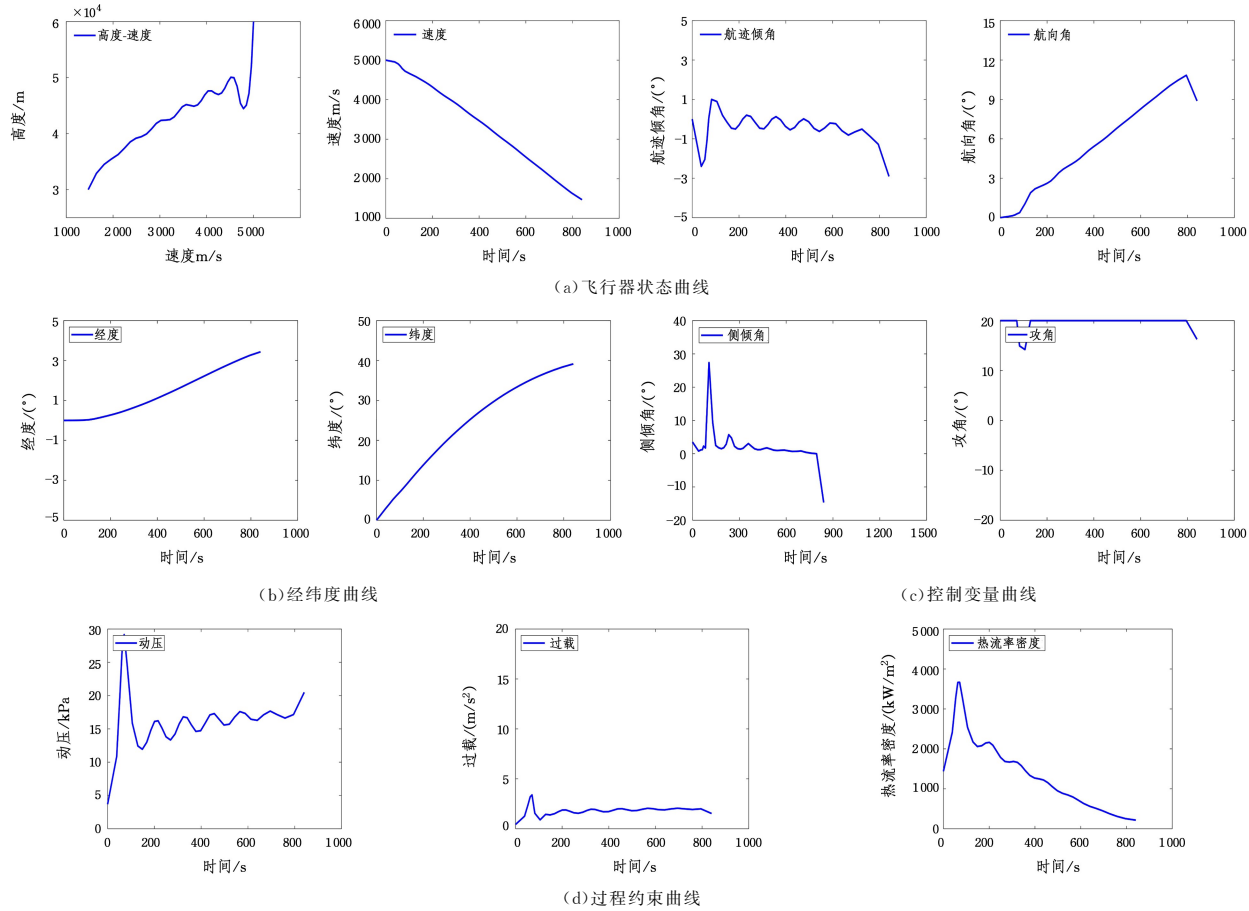
气动升力系数与气动的阻力系数的拟合计算式如式(8)所示，其中，系数值如表2所列，仿真结果如图2所示。图2中，动压、过载、热流满足条件，倾侧角在给定范围内。可见终端约束符合要求。

$$\begin{cases}C_L=a_1+b_1\cdot Ma+c_1\cdot\alpha+d_1\cdot Ma\cdot\alpha+\\ \qquad e_1\cdot Ma^2+f_1\cdot\alpha^2\\ C_D=a_2+b_2\cdot Ma+c_2\cdot\alpha+d_2\cdot Ma\cdot\alpha+\\ \qquad e_2\cdot Ma^2+f_2\cdot\alpha^2\end{cases}\tag{8}$$

表2 升阻力系数的公式参数表

Table 2 Formula parameters for lift resistance coefficient

$a1$	$b1$	$c1$	$d1$	$e1$	$f1$
−0.056 10	−0.004 43	0.050 00	−0.000 83	0.000 32	0.000 37
$a2$	$b2$	$c2$	$d2$	$e2$	$f2$
0.127 21	−0.015 42	0.004 86	−0.000 30	0.000 57	0.000 67



(a)飞行器状态曲线

(b)经纬度曲线

(c)控制变量曲线

(d)过程约束曲线

图2 仿真结果

Fig. 2 Simulation results

表3 终端误差

Table 3 Terminal error

末端状态约束高度误差	末端状态速度约束误差	实际结果(高度)	实际结果(速度)
10	100	9.999 7	30.068 2

2.2 MATLAB平台上多重打靶法与直接打靶法比较

仿真的条件1：初始的高度为60km，初始的速度为5 000 m/s，其他的状态变量为零；末端时刻高度为30 km，末端速度为1 500 m/s。

过程约束：最大吸热率5 000 kW/m²；最大动压30 kPa；最大过载2GB。如表4所列，多重打靶法以最小吸热量为目标函数比直接打靶法小，符合最小吸热量目标。如图3所示，多重打靶法在条件一样时非常接近。

表4 multi-shooting算法与direct-shooting算法的吸热量比较

Table 4 Comparison of heat absorption between multi-shooting algorithm and direct-shooting algorithm

方法	性能指标			过程约束		
	迭代次数	吸热量/KJ	运行时间/s	终端高度误差/m	终端速度误差/s	
多重打靶法	111	288.90	76.06	10.00	30.07	满足
直接打靶法	120	288.90	48.35	10.00	30.07	满足

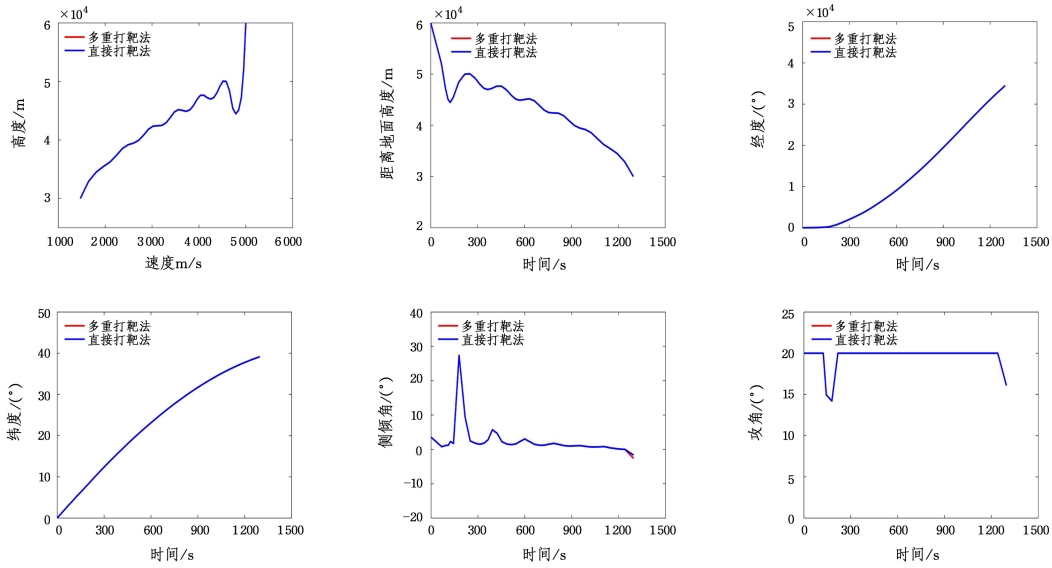


图 3　multi-shooting 算法与 direct-shooting 算法比较

Fig. 3　Comparison between multi-shooting algorithm and direct-shooting algorithm

飞行条件 2：初始高度为 65 km，初始速度为 6 000 m/s，其余状态量为零；终端时的高度为 30 km，速度为 1 500 m/s。

飞行的过程中的约束：最大吸热率为 5 000 kW/m^2；最大动压为 30 kPa；最大过载为 2 GB。

2.3　多重打靶法的并行计算

2.3.1　openmp 并行计算

由于多重打靶法中间存在一个直接打靶法得到，因此可以对每个区间实现并行计算，当区间打靶计算量匹配条件，即每个区间终端状态与下一个区间初始状态相等，而每个区间通过仿真常打靶时，加速比增大。

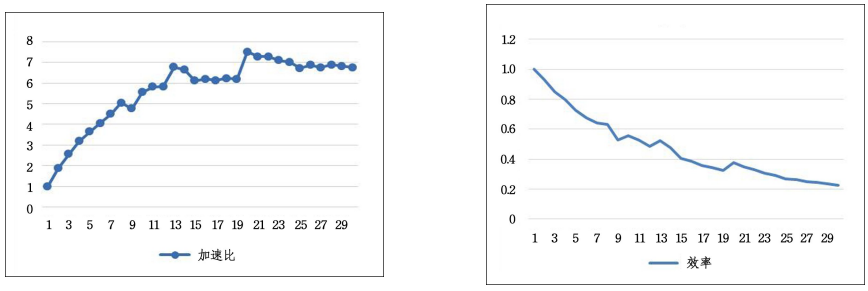
本文使用了 Intel(R) Xeon(R) Gold 5218，横坐标是线程数量，纵坐标是程序运行的加速比，打靶点个数为 178。不同的线程数的加速效果如表 5 所列。

当线程数目取 22 时加速比最大，因为线程开销太大，所以并不是线程越多加速比越大。加速比曲线与效率曲线如图 4 所示。

表 5　加速效果

Table 5　Acceleration effect

线程总数	时间/s	加速比	效率	线程总数	时间/s	加速比	效率
1	89.139 7	1	1	16	14.383 9	6.197 185 742	0.387 324 109
2	47.996 5	1.857 212 505	0.928 606 253	17	14.553 8	6.124 840 248	0.360 284 720
3	34.947 7	2.550 659 986	0.850 219 995	18	14.305 5	6.231 148 859	0.346 174 937
4	27.995 1	3.184 117 935	0.796 029 484	19	14.379 6	6.199 038 916	0.326 265 206
5	24.484 3	3.640 688 114	0.728 137 623	20	11.861 1	7.515 297 907	0.375 764 895
6	21.985 4	4.054 495 256	0.675 749 209	21	12.243 8	7.280 394 975	0.346 685 475
7	19.800 9	4.501 800 423	0.643 114 346	22	12.235 8	7.285 155 037	0.331 143 411
8	17.673 2	5.043 778 150	0.630 4722 690	23	12.555 3	7.099 766 632	0.308 685 506
9	18.717 5	4.762 372 112	0.529 152 457	24	12.703 2	7.017 105 926	0.292 379 414
10	16.059 7	5.550 520 869	0.555 052 087	25	13.246 4	6.729 352 881	0.269 174 115
11	15.302 7	5.825 096 225	0.529 554 202	26	12.958 9	6.878 647 107	0.264 563 35
12	15.259 8	5.841 472 365	0.486 789 364	27	13.179 8	6.763 357 562	0.250 494 725
13	13.141 8	6.782 914 060	0.521 762 620	28	12.937 4	6.890 078 377	0.246 074 228
14	13.399 7	6.652 365 351	0.475 168 954	29	13.049 6	6.830 837 727	0.235 546 129
15	14.590 7	6.109 350 477	0.407 290 032	30	13.184 9	6.760 741 454	0.225 358 048



(a)加速比曲线　　(b)效率曲线

图 4　加速比曲线与效率曲线

Fig. 4　Acceleration ratio curve and efficiency curve

程序在首次使用 openmp 时，两个核的核加速比（每个核产生的效率）在 openmp 版本中相对最高，单核是89.1397 s，双核是 47.9965 s。在 20 个核以后时间有增加趋势，其原因为随着线程数量增多，线程开销会增大，当单线程任务量下降到一定数量时，线程开销将会使加速比下降。并行结果如表 6 所列，使用打靶点数目 178 测得时间曲线，如图 5 所列。

表 6 误差表格

Table 6 Error table

打靶点个数	4	166	178
地心距误差	110.234353	83.869900673894 7	80.342889314601
经度误差	2.33×10^{-5}	$1.926014621836 39\times10^{-5}$	$2.317808090098 60\times10^{-5}$
纬度误差	0.000249898	0.000254017479209945	0.000252920955216229
速度误差	−0.555868898	−0.578833785313108	−0.555513875563673
速度倾角误差	−0.00355821	−0.003623823817804 80	−0.003604728898018 15
航迹角误差	0.000177864	0.000150261015123665	0.000173208545114212

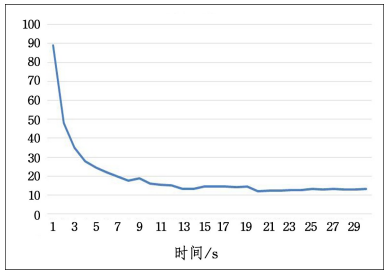


图 5 并行时间分布

Fig. 5 Parallel time distribution

2.3.2 openmp 的效率分析

当打靶点个数为 178 时，精度比较高，因此使用 178 打靶点数目进行并行计算。并行相对效率的计算式为：

$$\mu=\frac{t}{t_{\max}} \tag{9}$$

其中，t 为当前线程时间，$t_{\max}$ 为所有线程的最长时间。

表 7 9 个线程时的时间计算

Table 7 Time calculation for 9 threads

线程号	时间/s	线程号	时间/s	线程号	时间/s
0	3.62692	13	4.18537	26	4.02867
1	3.77417	14	3.74314	27	3.57643
2	3.40376	15	4.01822	28	3.82908
3	3.48291	16	3.84351	29	3.85941
4	3.60541	17	3.77117	30	3.84506
5	4.28976	18	3.75921	31	3.32056
6	3.93385	19	3.64012	32	3.56956
7	3.93661	20	3.49819	33	3.34444
8	3.63366	21	3.29957	34	3.27926
9	3.85655	22	4.07101	35	3.95723
10	3.43067	23	4.34699	36	3.97563
11	3.59724	24	3.83197	37	3.24880
12	4.11160	25	3.84857	38	3.45700

因为测试所用的处理器为两块 Gold 5218，共 32 实核，所以在线程号 31 以后的线程均需要等待前面的线程结束后再执行。由此可知，虽然每个线程的执行时间变短，但是总时间却变长，因为实际线程时间约为两倍的最小线程时间。20 线程时，线程时间约为 5 s，39 线程时，线程时间约为 3 s，则 3 * 2−5=1 s，正好是 20 线程总时间和 39 线程总时间的差值。

当任务数目分布越均衡，相对效率越高，即当最大任务数最小时加速比最大，当有最大任务数最多的线程时，相对效率最大，本文中，如果一个处理器有 40 个核，每个核计算一个区间，这样分配理论上时间最短，加速比最大，相对效率最高，如果没有 40 个核，同时考虑时间和效率，20 个核是最佳分配。

结束语 本文实现了基于多重打靶法的 CAV-H 的再入轨迹优化算法。仿真结果表示，本文算法在求解 CAV-H 再入飞行器滑翔段的优化轨迹问题时，能够计算出满足各项约束而且精度较高的再入轨迹，说明了算法的可靠性。使用多重打靶法时，每个区间打靶点个数越多，多重打靶法的精度就越高，并且可以实现并行计算。使用 OpenMp 的多重打靶法的弹道优化能够大大缩短优化时间。

参 考 文 献

[1] AN P T, HAI N N, HOAI T V. Direct multiple shooting method for solving approximate shortest path problems[J]. Journal of Computational and Applied Mathematics, 2013, 244: 67-76.

[2] 王凯. 最优控制的直接打靶法实现与改进研究[D]. 武汉：华中科技大学，2019.

[3] SUBCHAN S. A direct multiple shooting method for missile trajectory optimization with the terminal bunt manoeuvre[J]. Journal for Technology and Science, 2011, 22(3): 147-151.

[4] BETTS J T. Survey of Numerical Methods for Trajectory Optimization. Journal of Guidance[J]. ontrol, and Dynamics, 1998, 21(2): 193-207.

[5] HULL D G, SPEYER J L. Optimal Reentry and Plane-Change Trajectories[J]. Journal of the Astronautical Sciences, 1982, 30(2): 117-130.

[6] BOCK H G, PLITT K J. A Multiple Shooting Algorithm for Direct Solution of Optimal Control Problem[C]// Proceeding of the 9th IFAC Conference. Budapest: 1984: 1603-1608.

[7] BOCK H G. Handling Path Constraints in a Direct Multiple Shooting Method for Optimal Control Problems[M]. Heidelberg, 2006: 42-45.

[8] LIU X, SHEN Z, LU P. Solving the maximum-crossrange problem via successive second-order cone programming with a line search [J]. Aerospace Science and Technology, 2015, 47: 10-20.

LI Siyao, born in 1988, Ph.D. His main research interests include flight mechanics and control, parallel computing.

LI Shanglin, born in 1987, Ph.D, associate professor. His main research interests include computer graphics and machine learning.