SGN – Assignment #1

Matteo Luciardello Lecardi, 977701

June 15, 2023

1 Periodic orbit

Exercise 1

Consider the 3D Sun–Earth Circular Restricted Three-Body Problem with $\mu = 3.0359 \times 10^{-6}$.

1) Find the x-coordinate of the Lagrange point L_2 in the rotating, adimensional reference frame with at least 10-digit accuracy.

Solutions to the 3D CRTBP satisfy the symmetry

$$\mathcal{S} : (x, y, z, \dot{x}, \dot{y}, \dot{z}, t) \rightarrow (x, -y, z, -\dot{x}, \dot{y}, -\dot{z}, -t).$$

Thus, a trajectory that crosses perpendicularly the $y = 0$ plane twice is a periodic orbit.

2) Given the initial guess $\mathbf{x}_0 = (x_0, y_0, z_0, v_{x0}, v_{y0}, v_{z0})$, with

$$\begin{aligned} x_0 &= 1.008296144180133 \\ y_0 &= 0 \\ z_0 &= 0.001214294450297 \\ v_{x0} &= 0 \\ v_{y0} &= 0.010020975499502 \\ v_{z0} &= 0 \end{aligned}$$

Find the periodic halo orbit that passes through z_0; that is, develop the theoretical framework and implement a differential correction scheme that uses the STM either approximated through finite differences or achieved by integrating the variational equation.

The periodic orbits in the CRTBP exist in families. These can be computed by continuing the orbits along one coordinate, e.g., z_0. This is an iterative process in which one component of the state is varied, while the other components are taken from the solution of the previous iteration.

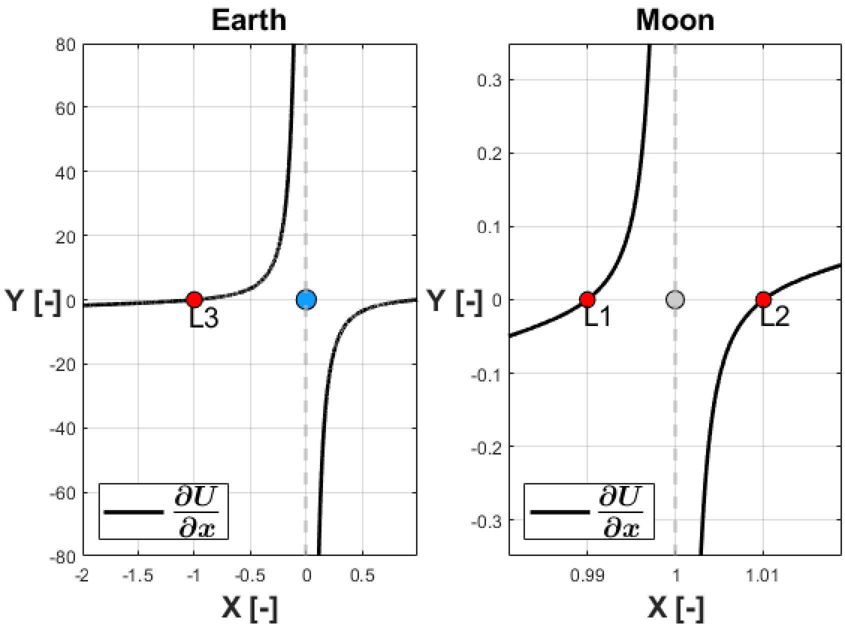
3) By gradually increasing z_0 and using numerical continuation, compute the families of halo orbits until $z_0 = 0.0046$.

(8 points)

---

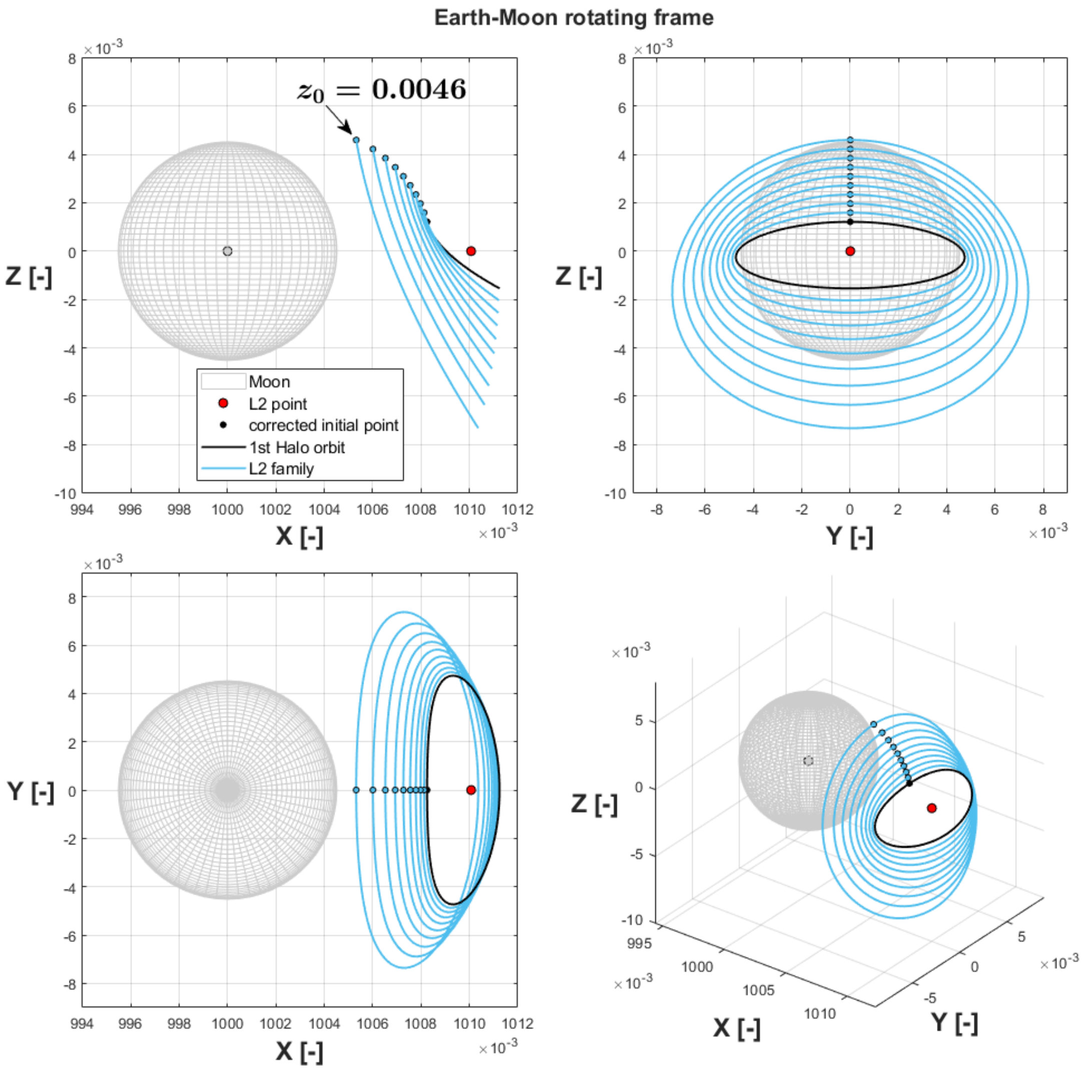
1) Libration/Langrange points (Earth-Moon Rotating frame):

$$\begin{aligned} x_{L1} &= 0.9899909372 \\ x_{L2} &= 1.0100701876 \\ x_{L3} &= -1.0000012650 \end{aligned}$$



2) See A.1 for STM derivatives

3) Family of 10 CRTBP periodic orbits in Earth-Moon rotating reference frame :



2 Impulsive guidance

Exercise 2

Consider the two-impulse transfer problem stated in Section 3.1 (Topputo, 2013)*.

1) Using the procedure in Section 3.2, produce a first guess solution using $\alpha = 1.5\pi$, $\beta = 1.41$, $\delta = 7$, and $t_i = 0$. Plot the solution in both the rotating frame and Earth-centered inertial frame (see Appendix 1 in (Topputo, 2013)).

2) Considering the first guess in 1) and using $\{\mathbf{x}_i, t_i, t_f\}$ as variables, solve the problem in Section 3.1 with simple shooting in the following cases

   a) without providing any derivative to the solver, and

   b) by providing the derivatives and by estimating the state transition matrix with variational equations.

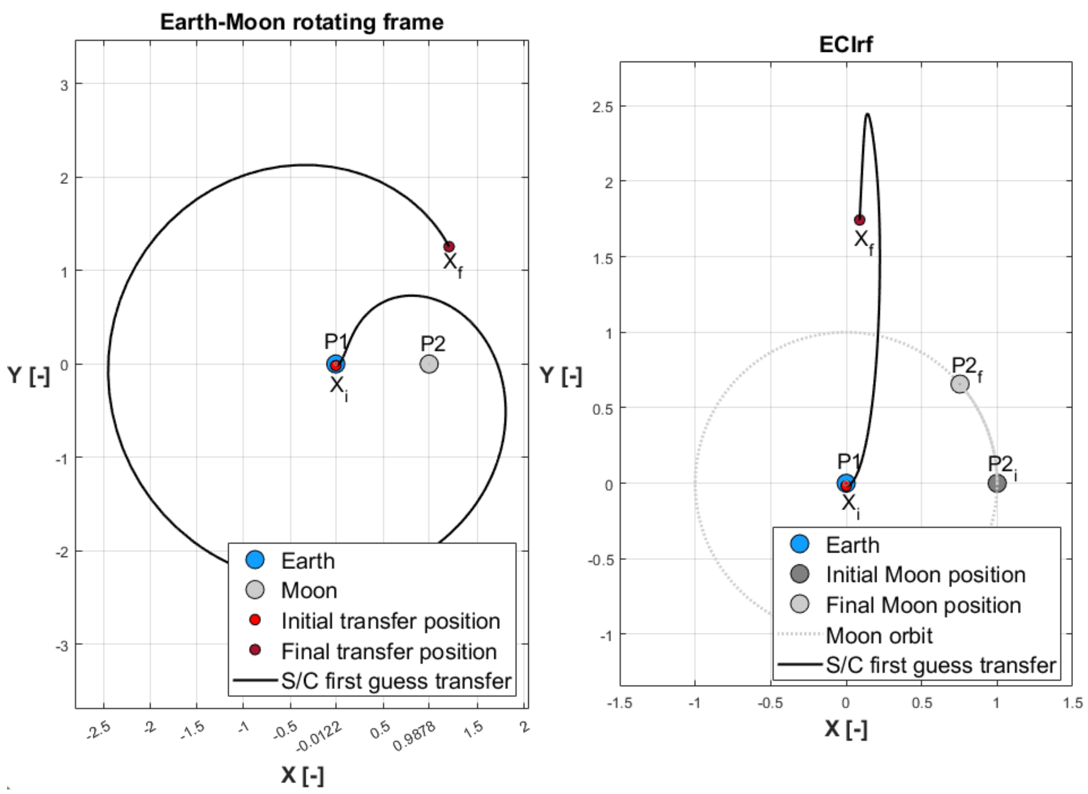
3) Considering the first guess solution in 1) and the procedure in Section 3.3, solve the problem with multiple shooting taking $N = 4$ and using the variational equation to compute the Jacobian of the nonlinear equality constraints.

(11 points)

---

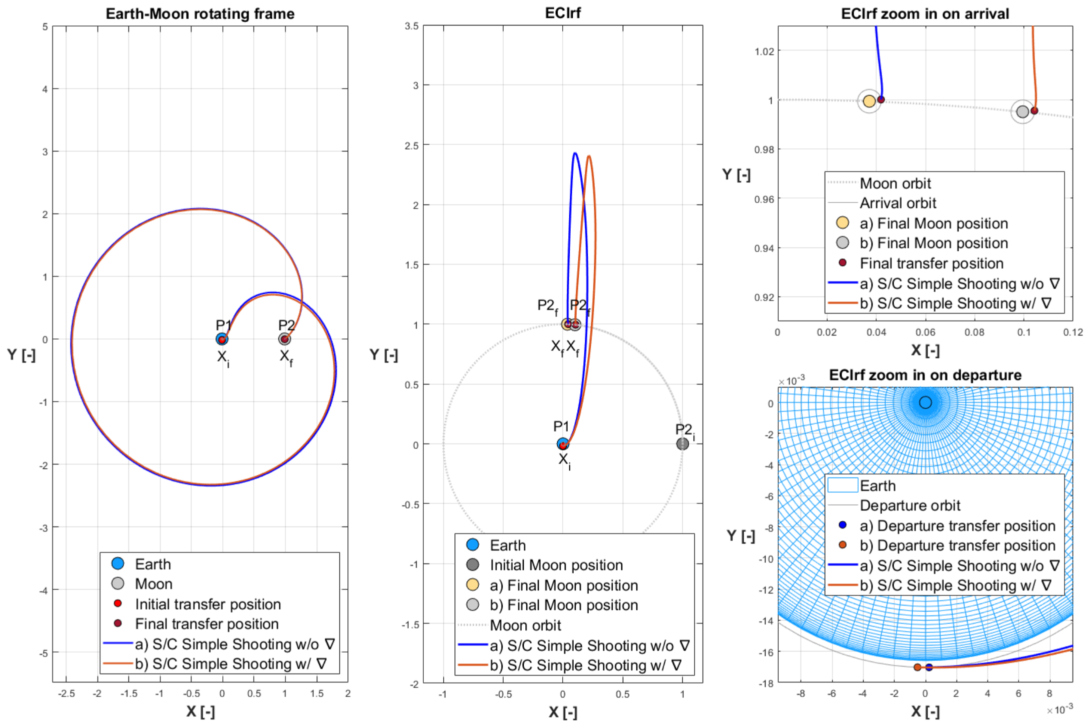
1) First guess solution (Earth-Moon rotating frame):

$$x_0 = -0.0121506683 \; ; \quad y_0 = -0.0170263134 \; ; \quad u_0 = 10.7229705957 \; ; \quad v_0 = 0$$

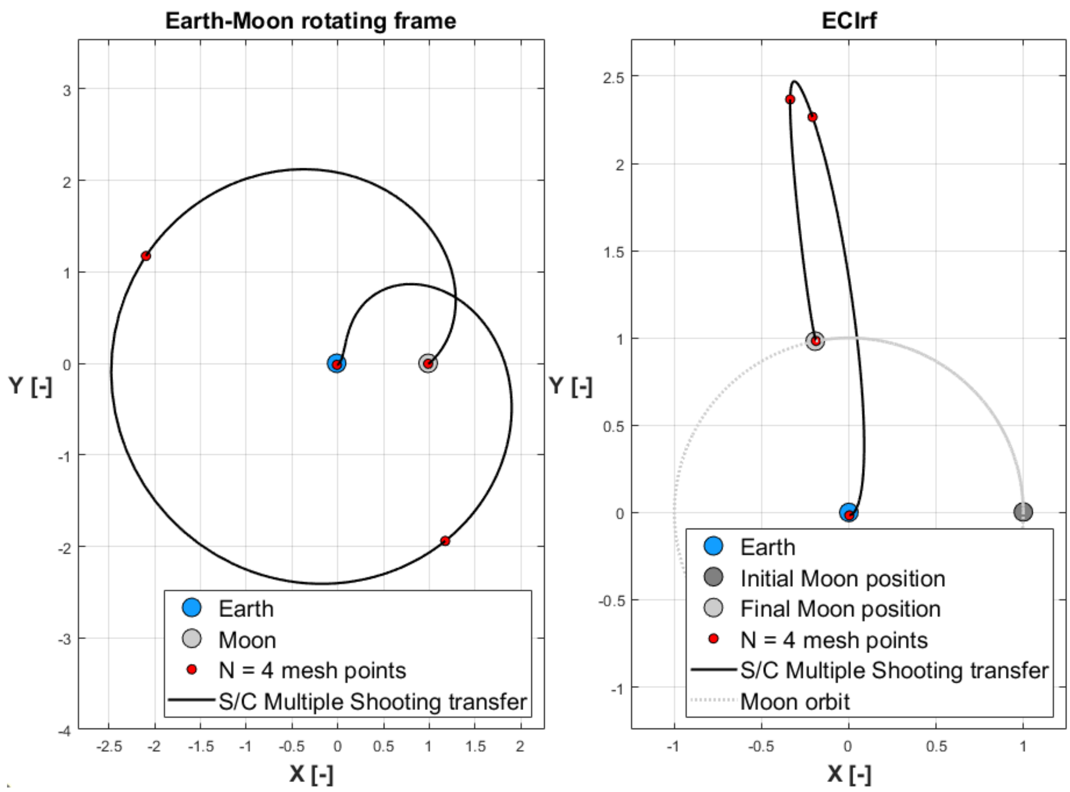


*F. Topputo, "On optimal two-impulse Earth–Moon transfers in a four-body model", Celestial Mechanics and Dynamical Astronomy, Vol. 117, pp. 279–313, 2013, DOI: 10.1007/s10569-013-9513-8.

2) Simple shooting with and without gradients for objective function and constraints function, computed with `fmincon`; derivatives, results and methods for computational efficiency and high accuracy reported in A.2:



3) Multiple Shooting, from first guess solution of ex. 2.1, gradient of non linear equality and inequality constraints referenced inA.2f



3 Continuous guidance

Exercise 3

A low-thrust option is being considered for an Earth-Mars transfer†. Provide a *time-optimal* solution under the following assumptions: the spacecraft moves in the heliocentric two-body problem, Mars instantaneous acceleration is determined only by the Sun's gravitational attraction, the departure date is fixed, and the spacecraft initial and final states are coincident with those of the Earth and Mars, respectively.

1) Write down the spacecraft equations of motion, the costate dynamics, and the zero-finding problem for the unknowns $\{\boldsymbol{\lambda}_0, t_f\}$ with the appropriate transversality condition.

2) Solve the problem considering the following data:

   - Launch date: `2022-08-03-12:45:20.000 UTC`
   - Spacecraft mass: $m_0 = 1500$ kg
   - Electric properties: $T_{\max} = 150$ mN, $I_{sp} = 3000$ s
   - Number of thrusters: 4

   Report the obtained solution in terms of $\{\boldsymbol{\lambda}_0, t_f\}$ and the error with respect to the target. Validate your results exploiting the properties of time-optimal solutions.

3) Solve the problem for a degraded configuration with only 3 thrusters available, assuming that the failure occurs immediately after launch. Plot the thrust angles and compare them to the nominal case in 2).

(11 points)

---

1) Two Point Boundary Value problem:

$$\text{states (EOM)}: \begin{cases} \dot{\underline{\boldsymbol{r}}} &= \underline{\boldsymbol{v}} \\ \dot{\underline{\boldsymbol{v}}} &= -\dfrac{\mu}{r^3}\underline{\boldsymbol{r}} - u^*(\lambda_v, \lambda_m)\dfrac{T_{max}}{m}\dfrac{\underline{\boldsymbol{\lambda_v}}}{\lambda_v} \\ \dot{m} &= -u^*(\lambda_v, \lambda_m)\dfrac{T_{max}}{I_{sp}\, g_0} \end{cases} \tag{1}$$

$$\text{costates}: \begin{cases} \dot{\underline{\boldsymbol{\lambda}}}_{\boldsymbol{r}} &= -3\dfrac{\mu}{r^5}(\underline{\boldsymbol{r}} \cdot \underline{\boldsymbol{\lambda}}_v)\underline{\boldsymbol{r}} + \dfrac{\mu}{r^3}\underline{\boldsymbol{\lambda}}_v \\ \dot{\underline{\boldsymbol{\lambda}}}_{\boldsymbol{v}} &= -\underline{\boldsymbol{\lambda}}_r \\ \dot{\lambda}_m &= -u^*(\lambda_v, \lambda_m)\dfrac{\lambda_v T_{max}}{m^2} \end{cases} \tag{2}$$

$$\text{Initial Conditions}: \begin{cases} \underline{\boldsymbol{r}}(t_0) &= \underline{\boldsymbol{r_0}} \\ \underline{\boldsymbol{v}}(t_0) &= \underline{\boldsymbol{v_0}} \\ m(t_0) &= m_0 \end{cases} \tag{3}$$

$$F(\mathbf{x}) = 0: \begin{cases} \underline{\boldsymbol{r}}(t_f) - \underline{\boldsymbol{r_f}} &= 0 \\ \underline{\boldsymbol{v}}(t_f) - \underline{\boldsymbol{v_f}} &= 0 \\ \lambda_m(t_f) = 0 & \text{transversality condition} \end{cases} \tag{4}$$

$$\text{unknowns } \mathbf{x}: \begin{cases} \underline{\boldsymbol{\lambda_0}} = \begin{bmatrix} \underline{\boldsymbol{\lambda_r}}(t_0) \\ \underline{\boldsymbol{\lambda_v}}(t_0) \\ \lambda_m(t_0) \end{bmatrix} \\ t_f \end{cases} \tag{5}$$

†Read the necessary gravitational constants and planets positions from SPICE. Use the kernels provided on WeBeep for this assignment.

2) Adopted units triplet for developing the first guess solution (Zero-Finding problem): [AU,YEARS,KG]
For details A.3

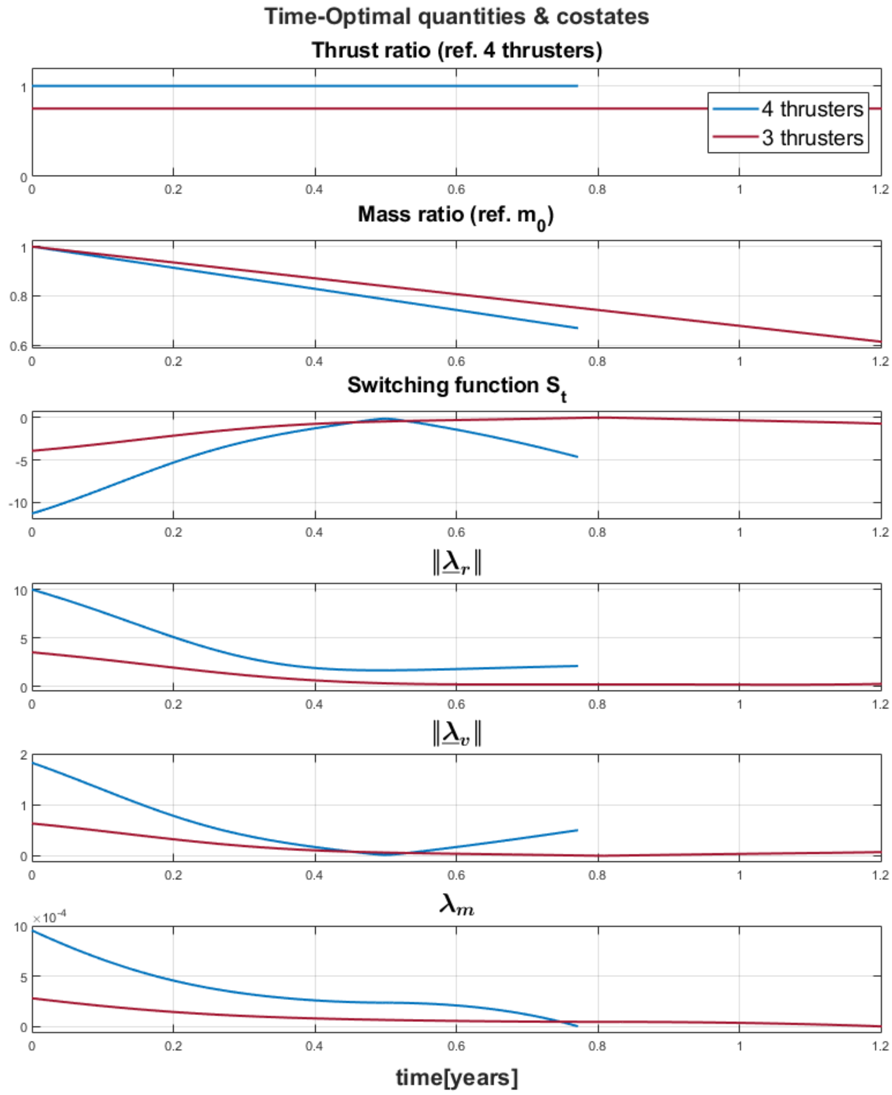
$\boldsymbol{\lambda}_{0,r}$	-8.0814966	$+5.8228483$	$+0.40115421$
$\boldsymbol{\lambda}_{0,v}$	-1.6532641	-0.77030439	-0.021376942
$\lambda_{0,m}$	$9.5403056e-4$		
t_f	`2023-05-12-08:05:14.725 UTC`		
TOF [days]	281.83119		

Table 1: Time-optimal Earth-Mars transfer solution.

$\lvert\lvert\mathbf{r}_f(t_f) - \mathbf{r}_M(t_f)\rvert\rvert$ [km]	$2.6818e-5$
$\lvert\lvert\mathbf{v}_f(t_f) - \mathbf{v}_M(t_f)\rvert\rvert$ [m/s]	$3.5144e-9$

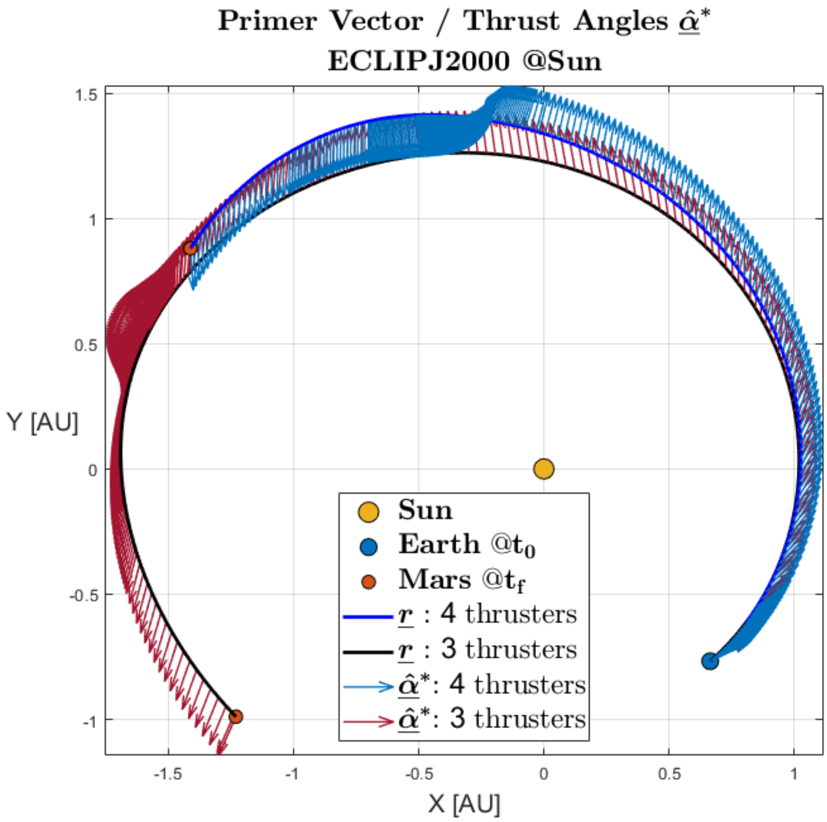
Table 2: Final state error with respect to Mars' center.

Validation:

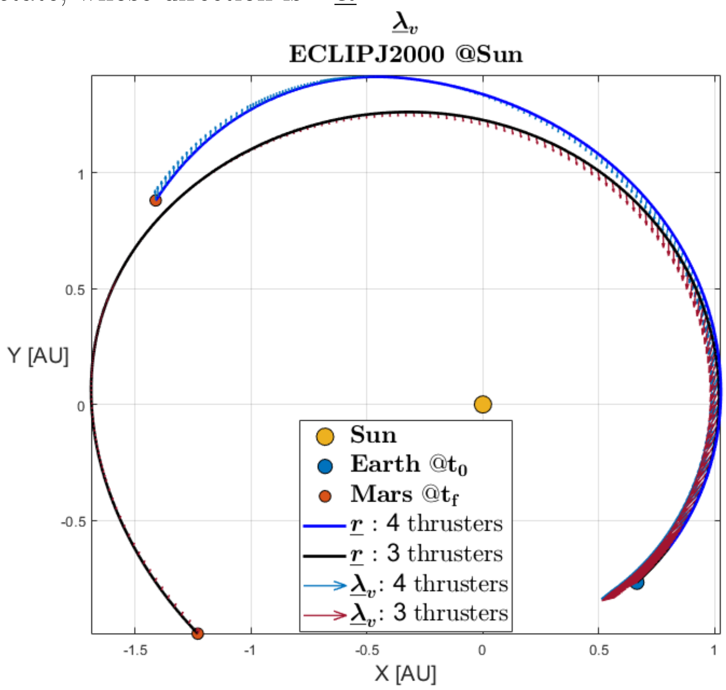


As expected for time-optimal solutions, the "switching function" is negative throughout the maneuver; λ_m is descending monotone till arrival epoch, where it coincides with 0.

3) Thrust angles have been plotted directly along the trajectory, so it's possible to perceive immediately the physical meaning of the costates, and how the solution changes for a different number of thrusters.



The $\underline{\boldsymbol{\lambda}}_v$ costate, whose direction is $-\underline{\hat{\alpha}}^*$



4 Appendix

A.1: STM variational approach derivatives for ex. 1.2

$$
\begin{aligned}
r_1 &= \sqrt{(x+\mu)^2+y^2+z^2} \\
r_2 &= \sqrt{(x+\mu-1)^2+y^2+z^2} \\
\omega_{xx} &= 1-(1-\mu)/r_1^3+3(1-\mu)(x+\mu)^2/r_1^5-\mu/r_2^3+3\mu(x-1+\mu)^2/r_2^5 && (6)\\
\omega_{yy} &= 1-(1-\mu)/r_1^3+3(1-\mu)y^2/r_1^5-\mu/r_2^3+3\mu y^2/r_2^5 && (7)\\
\omega_{zz} &= -(1-\mu)/r_1^3-\mu/r_2^3+3z^2(1-\mu)/r_1^5+3\mu z^2/r_2^5 && (8)\\
\omega_{xy} &= 3(1-\mu)(x+\mu)y/r_1^5+3\mu(x+\mu-1)y/r_2^5 && (9)\\
\omega_{zx} &= 3\mu z(\mu+x-1)/r_2^5+3z(1-\mu)(\mu+x)/r_1^5 && (10)\\
\omega_{yz} &= 3\mu yz/r_2^5+3yz(1-\mu)/r_1^5 && (11)
\end{aligned}
$$

$$
\mathbf{A}=\begin{bmatrix}0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\\\omega_{xx}&\omega_{xy}&\omega_{xz}&0&2&0\\\omega_{yx}&\omega_{yy}&\omega_{yz}&-2&0&0\\\omega_{zx}&\omega_{zy}&\omega_{zz}&0&0&0\end{bmatrix} \tag{12}
$$

A.2a: PCR4BP dynamics derivatives to compute propagation for ex. 2

$$
\begin{aligned}
\frac{\partial\Omega_4}{\partial x} &= x-m_s(x-\rho\cos(\omega_s t))/((x-\rho\cos(\omega_s t))^2+(y-\rho\sin(\omega_s t))^2)^{3/2}+\\
&\quad-\mu(\mu+x-1)/((\mu+x-1)^2+y^2)^{3/2}-(m_s\cos(\omega_s t))/\rho^2+\\
&\quad+(\mu+x)(\mu-1)/((\mu+x)^2+y^2)^{3/2} && (13)\\
\frac{\partial\Omega_4}{\partial y} &= y-m_s(y-\rho\sin(\omega_s t))/((x-\rho\cos(\omega_s t))^2+(y-\rho\sin(\omega_s t))^2)^{3/2}+\\
&\quad-\mu y/((\mu+x-1)^2+y^2)^{3/2}+\\
&\quad-m_s\sin(\omega_s t)/\rho^2+y(\mu-1)/((\mu+x)^2+y^2)^{3/2} && (14)
\end{aligned}
$$

A.2b: STM derivatives to compute gradients for ex. 2.2.b

$$
\begin{aligned}
\frac{\partial\Omega_4}{\partial x\partial x} &= 1-m_s/((x-\rho\cos(\omega_s t))^2+(y-\rho\sin(\omega_s t))^2)^{3/2}+\\
&\quad+3m_s(x-\rho\cos(\omega_s t))^2/((x-\rho\cos(\omega_s t))^2+(y-\rho\sin(\omega_s t))^2)^{5/2}+\\
&\quad-(1-\mu)/((x+\mu)^2+y^2)^{3/2}+3(1-\mu)(x+\mu)^2/((x+\mu)^2+y^2)^{5/2}+\\
&\quad-\mu/((x+\mu-1)^2+y^2)^{3/2}+3\mu(x+\mu-1)^2/((x+\mu-1)^2+y^2)^{5/2} && (15)\\
\frac{\partial\Omega_4}{\partial x\partial y} &= 3m_s(x-\rho\cos(\omega_s t))(y-\rho\sin(\omega_s t))/((y-\rho\sin(\omega_s t))^2+(x-\rho\cos(\omega_s t))^2)^{5/2}+\\
&\quad+3y(1-\mu)(x+\mu)/(y^2+(x+\mu)^2)^{5/2}+\\
&\quad+3y\mu(x+\mu-1)/(y^2+(x+\mu-1)^2)^{5/2} && (16)
\end{aligned}
$$

$$\begin{aligned}\frac{\partial \Omega_4}{\partial y \partial y} \;=\;& 1 - m_s/((y-\rho\sin(\omega_s t))^2 + (x-\rho\cos(\omega_s t))^2)^{3/2} + \\ & + 3m_s(y-\rho\sin(\omega_s t))^2/((y-\rho\sin(\omega_s t))^2 + (x-\rho\cos(\omega_s t))^2)^{5/2} + \\ & - (1-\mu)/(y^2+(x+\mu)^2)^{3/2} + 3y^2(1-\mu)/(y^2+(x+\mu)^2)^{5/2} + \\ & - \mu/(y^2+(x+\mu-1)^2)^{3/2} + 3\mu y^2/(y^2+(x+\mu-1)^2)^{5/2} \qquad (17)\end{aligned}$$

A.2c: gradient of objective function for simple shooting, ex. 2.2.b

$$\boldsymbol{P_1} := \frac{\partial J}{\partial \mathbf{x_1}} = \frac{[\dot{y}_1 + x_1 + \mu, \;\; y_1 - \dot{x}_1, \;\; \dot{x}_1 - y_1, \;\; \dot{y}_1 + x_1 + \mu]}{\sqrt{(\dot{x}_1 - y_1)^2 + (\dot{y}_1 + x_1 + \mu)^2}} \qquad (18)$$

$$\boldsymbol{P_2} := \frac{\partial J}{\partial \mathbf{x_2}} = \frac{[\dot{y}_2 + x_2 + \mu - 1, \;\; y_2 - \dot{x}_2, \;\; \dot{x}_2 - y_2, \;\; \dot{y}_2 + x_2 + \mu - 1]}{\sqrt{(\dot{x}_2 - y_2)^2 + (\dot{y}_2 + x_2 + \mu - 1)^2}} \qquad (19)$$

$$\frac{\partial J}{\partial t_1} = \frac{\partial J}{\partial \mathbf{x_2}}\frac{\partial \mathbf{x_2}}{\partial t_1} = \boldsymbol{P_2}\left[-\boldsymbol{\Phi}\begin{pmatrix}\dot{x}_1\\ \dot{y}_1\\ \ddot{x}_1\\ \ddot{y}_1\end{pmatrix}\right] \qquad (20)$$

$$\frac{\partial J}{\partial t_2} = \frac{\partial J}{\partial \mathbf{x_2}}\frac{\partial \mathbf{x_2}}{\partial t_2} = \boldsymbol{P_2}\begin{pmatrix}\dot{x}_2\\ \dot{y}_2\\ \ddot{x}_2\\ \ddot{y}_2\end{pmatrix} \qquad (21)$$

$$\nabla J = \begin{bmatrix}\boldsymbol{P_1} & \frac{\partial J}{\partial t_1} & \frac{\partial J}{\partial t_2}\end{bmatrix}^T \qquad (22)$$

A.2d: gradient of constraints function for simple shooting, ex. 2.2.b

$$\boldsymbol{R_1} = \frac{\partial \Psi_1}{\partial \mathbf{x_1}}^T = \begin{bmatrix}2(x_1+\mu) & 2y_1 & 0 & 0\\ \dot{x}_1 & \dot{y}_1 & x_1+\mu & y_1\end{bmatrix} \qquad (23)$$

$$\boldsymbol{R_2} = \frac{\partial \Psi_2}{\partial \mathbf{x_2}}^T = \begin{bmatrix}2(x_2+\mu-1) & 2y_2 & 0 & 0\\ \dot{x}_2 & \dot{y}_2 & x_2+\mu-1 & y_2\end{bmatrix} \qquad (24)$$

$$\frac{\partial \Psi_2}{\partial \mathbf{x_1}} = \frac{\partial \Psi_2}{\partial \mathbf{x_2}}\frac{\partial \mathbf{x_2}}{\partial \mathbf{x_1}} = (\boldsymbol{R_2}\,\boldsymbol{\Phi})^T \qquad (25)$$

$$\frac{\partial \Psi_1}{\partial t_1} = \begin{bmatrix}2(x_1+\mu)\dot{x}_1 + 2y_1\dot{y}_1, & \dot{x}_1^2 + \dot{y}_1^2 + \ddot{x}_1(x_1+\mu) + \ddot{y}_1 y_1\end{bmatrix} \qquad (26)$$

$$\frac{\partial \Psi_2}{\partial t_1} = \frac{\partial \Psi_2}{\partial \mathbf{x_2}}\frac{\partial \mathbf{x_2}}{\partial t_1} = \left\{\boldsymbol{R_2}\left[-\boldsymbol{\Phi}\begin{pmatrix}\dot{x}_1\\ \dot{y}_1\\ \ddot{x}_1\\ \ddot{y}_1\end{pmatrix}\right]\right\}^T \qquad (27)$$

$$\frac{\partial \Psi_1}{\partial t_2} = [0 \;\; 0] \qquad (28)$$

$$\frac{\partial \Psi_2}{\partial t_2} = \begin{bmatrix}2(x_2+\mu-1)\dot{x}_2 + 2y_2\dot{y}_2, & \dot{x}_2^2 + \dot{y}_2^2 + \ddot{x}_2(x_2+\mu-1) + \ddot{y}_2 y_2\end{bmatrix} \qquad (29)$$

$$\nabla \boldsymbol{c_{eq}} = \begin{bmatrix}\frac{\partial \Psi_1}{\partial \mathbf{x_1}} & \frac{\partial \Psi_2}{\partial \mathbf{x_1}}\\ \frac{\partial \Psi_1}{\partial t_1} & \frac{\partial \Psi_2}{\partial t_1}\\ \frac{\partial \Psi_1}{\partial t_2} & \frac{\partial \Psi_2}{\partial t_2}\end{bmatrix}; \qquad \nabla \boldsymbol{c_{ineq}} = \begin{bmatrix}0 & 0 & 0 & 0 & 1 & -1\end{bmatrix}^T \qquad (30)$$

A.2e: Methods for performance and accuracy for simple shooting

- It has been tested that faster convergence wrt. the standard method could be achieved substituting the matrix multiplication in (25) with the following equivalent system of equations (this due to `floating-point` precision):

$$\frac{\partial \Psi_2}{\partial \mathbf{x_1}}(i,1) = \begin{Bmatrix} 2(x_2+\mu-1)\mathbf{\Phi}(1,i)+2y_2\mathbf{\Phi}(2,i) \\ \vdots \end{Bmatrix} \tag{31}$$

$$\frac{\partial \Psi_2}{\partial \mathbf{x_1}}(i,2) = \tag{32}$$

$$\begin{Bmatrix} \mathbf{\Phi}(1,i)(\dot{x}_2-y_2)+(x_2+\mu-1)(\mathbf{\Phi}(3,i)-\mathbf{\Phi}(2,i))+\mathbf{\Phi}(2,i)(\dot{y}_2+x_2+\mu-1)+y_2(\mathbf{\Phi}(4,i)-\mathbf{\Phi}(1,i)) \\ \vdots \end{Bmatrix}$$

for i = 1, ... ,4

This method will be called "best accuracy", it is performed in "`double` precision". Furthermore, the accuracy of the standard method has been tested implementing symbolic elements after every propagation, while computing the objective function, the constraints and their gradients. 6 methods have been evaluated: without gradients (ex.2.2.a) with double precision and `symbolic` 16 digits precision (plus guard digits); with gradients (ex.2.2.b): best accuracy described above, standard automatic matricial product with double precision, symbolic only within constraints gradient, and symbolic for every calculation after the `ode113` propagation. In this scenario the accuracy index considered for the solution is "Constraint Violation", a value $\sim 1e-10$ is targeted.

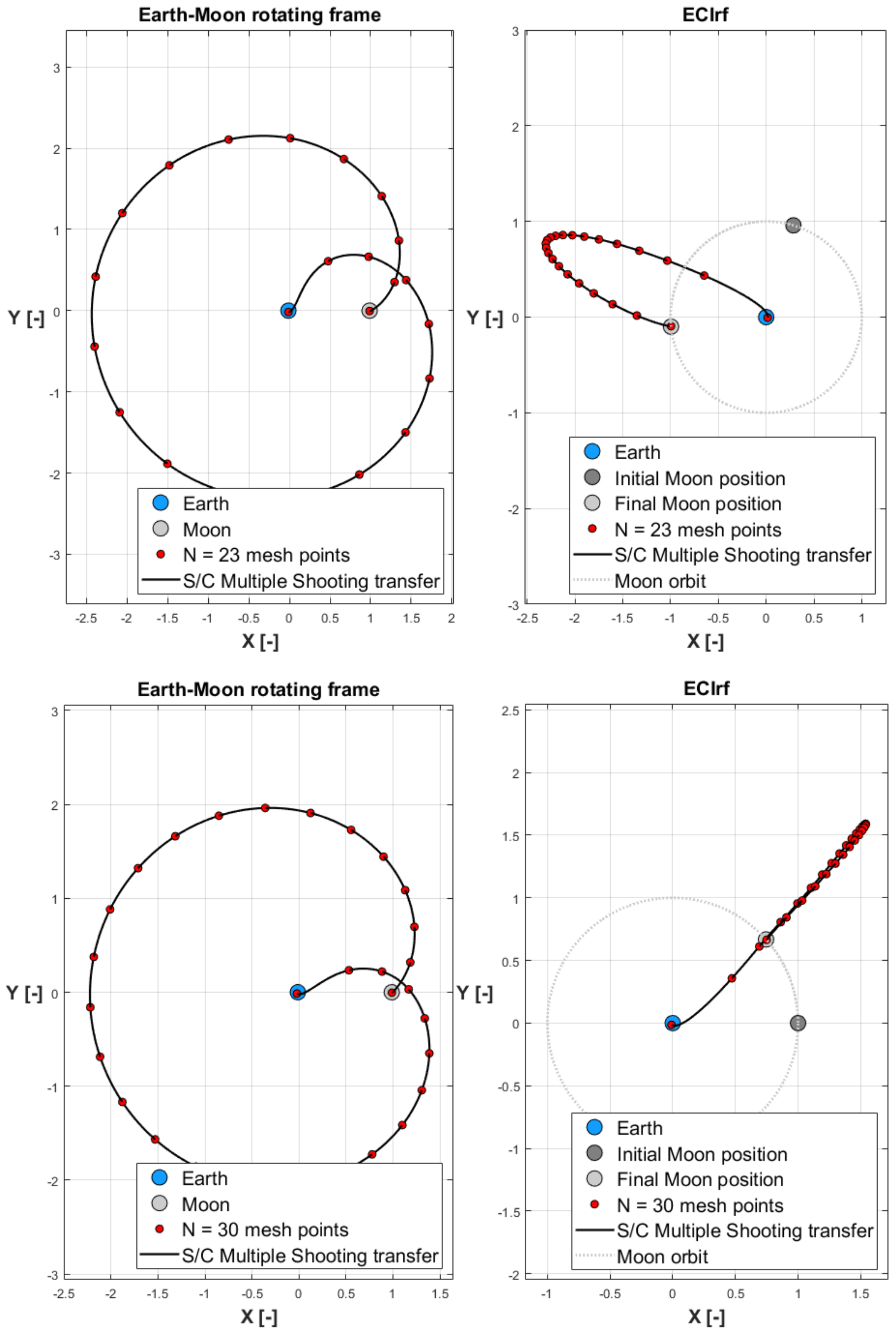
∇	Precision	Iterations	Exe. Time[s]	ΔV[-/s]	First Order Optimality	Constraints Violation
0	double	22	3.76	4.2167660597	$4.7680215230e-02$	$5.0782765146e-11$
0	16 digits	19	10.15	4.2178582995	$4.3677611412e-02$	$2.6265882698e-11$
1	best accuracy	5158	278.87	4.2207554901	$4.1464732189e-01$	$7.6710865091e-11$
1	double	880	52.60	4.2198826498	$4.0239922723e-01$	$6.1539786045e-05$
1	16 dig.Constr	78	7.71	4.2200984192	$4.0323922109e-01$	$9.9888712336e-05$
1	16 digits	409	80.40	4.2198886123	$4.0100014958e-01$	$9.9868547360e-05$

In general the target accuracy will not be reached as fast by gradient evaluation, e.g. the standard method tends to approach values of $\sim 1e-5$ to $\sim 1e-7$ in a few minutes, without moving any further within a reasonable run time. On the other hand the "best accuracy method" grants the optimal convergence within a few minutes.

- The main issue with the large computational request is the chance that `fmincon` passes the same inputs through both the objective and constraints function, thus requiring double the amount of propagations and STM computations. A tool which comes in handy is `memoize`, which is capable of storing the desirable amount of combinations `I/O` within its cache, so to avoid repeating the exact same calculations more than once. It is to be said that it's performance is not universally reliable for any class of inputs, but in general it helps computing simple shooting, saving ∼20/50% of the execution time.

A.2f: Multiple Shooting results and comparisons for ex 2.3

N	∇	Iterations	Exe. Time[s]	ΔV[-/s]	First Order Optimality	Constraints Violation
4	1	1155	118.25	4.0414400890	$6.0782239007e-03$	$3.8871794672e-11$
23	1	959	171.16	4.0130719922	$9.1313644510e-04$	$1.4742429499e-11$
30	1	24	3.21	4.1997249345	$3.2000518228e-02$	$3.8299585725e-11$



Multiple Shooting method has been achieved developing the derivatives of objective and constraints functions presented in: K. Oshima, F. Topputo,T. Yanao, CMDA,"Low-energy transfers to the Moon with long transfer time", 131:4, 2019.
In particular the Jacobian of equality constraints directly calls the State Transition matrix and the dynamics, whose equations have been presented above. The code has been implemented so to carry out the the transfer problem with a tunable number of N points. Enforcing low Constraints Tolerance in fmincon, it is possible to achieve better performances wrt to simple shooting, with the solver converging much faster, while obtaining the same or better target accuracy (final Constraint Violation $< 1e-10$).

A.3 method to perform fast convergence with fsolve, and costate $\underline{\boldsymbol{\lambda}}_r$, ex.3

It is required to compute an educated randomization of the initial guess imposing a few conditions, one being the λ_m costate to always be taken positive, after the first guess generation and within the boundary function, and the second being the time span in which to search for a guess of t_f. This trick makes it easy to tune the number of active thrusters, and simply changing the expected time window of arrival, the convergence will be computationally trivial.

$$\Delta t_{guess} = t_1 + (t_2 - t_1)\texttt{rand} \tag{33}$$

N.Thrusters	First order opt.	Δt [months]	$\|\|\underline{e}_r\|\|$ [km]	$\|\|\underline{e}_v\|\|$ [km/s]
4	$5.492e-11$	9.26	$2.6818918075e-05$	$3.5144737303e-12$
3	$1.302e-08$	14.41	$1.3962124171e-03$	$1.6467380286e-10$