DIFFERENTIAL ALGEBRA-BASED ORBIT DETERMINATION WITH THE SEMI-ANALYTICAL PROPAGATOR SADA

M. Losacco[1], **G. Principe**[2], **L. Pirovano**[3], **D. Gondelach**[4], **R. Armellin**[5], **J.F. San Juan**[6], **M. Lara**[6], **R. Dominguez**[7], **F. Pina**[7], **and I. Urdampilleta**[8]

[1] *Scientific Computing Group (GRUCACI), University of La Rioja, 26006 Logroño (La Rioja), Spain*
[2] *Deimos Space SLU, Ronda de Poniente 19, Tres Cantos, Madrid (Spain)*
[3] *Te Pūnaha Ātea - Auckland Space Institute - University of Auckland, 20 Symonds street, 1010 Auckland,*
[4] *Vyoma GmbH, Eckhardtstraße 28, D-64289 Darmstadt, Germany*
[5] *Te Pūnaha Ātea - Auckland Space Institute - University of Auckland, 20 Symonds street, 1010 Auckland,*
[6] *Scientific Computing Group (GRUCACI), University of La Rioja, 26006 Logroño (La Rioja), Spain*
[7] *Deimos Space SLU, Ronda de Poniente 19, Tres Cantos, Madrid (Spain)*
[8] *CDTI, Calle del Cid 4, Madrid, Madrid (Spain)*

ABSTRACT

The ever-increasing amount of uncontrolled space objects, known as space debris, poses a risk to space activities due to possible collisions with operational spacecraft. An accurate prediction of their orbit is needed to prevent damages to infrastructure in orbit and on the ground. In the frame of the Spanish Space Surveillance and Tracking (SST) (S3T) activities, the S3TOC (Spanish Operational Centre for SST) is investigating new techniques and methods that are expected to improve current procedures to determine and predict the orbit.

Several estimation techniques have been developed over the years, aiming to accurately determine the orbit of space objects from observations. Such techniques can be classified depending on the propagation method they implement. In a numerical propagation, the equations of motion are numerically integrated, leading to accurate results but long computational time. In contrast, analytical methods integrate the equations of motion analytically using approximated closed-form solutions for the orbital motion to speed up the process. This is, however, not always possible and limits the accuracy. Consequently, semi-analytical (SA) methods can be considered. SA propagation methods combine analytical methods and numerical integration techniques, trying to get the best mix of accuracy and efficiency. To this aim, short-periodic motion is filtered out and larger time steps can therefore be used for the integration of the equations of motion.

We present a first operative tool for the orbit determination (OD) of space objects that exploits differential algebra (DA) techniques to implement a least squares (LS) method compatible with both numerical and SA propagators. In the abovementioned context, DA allows us to obtain a polynomial representation of the residuals function on which efficient optimization techniques can be run to determine the optimum. Additionally, DA removes the need to both integrate the variational equations for the state transition matrix and to linearise the measurement functions thus enabling the use of arbitrary dynamics and observation types. The effort of this work is mainly focused on the coupling of the resulting Differential Algebra based Least Squares (DALS) solver with the SA propagator SADA. The resulting algorithm (sDALS) benefits from the high efficiency of both the DALS solver and the SADA propagator, while partially sacrificing orbit accuracy. The accuracy and efficiency of sDALS are assessed against a high-fidelity version of the same tool, aDALS, that exploits the numerical AIDA (Accurate Integrator for Debris Analysis) propagator. Both versions are compared against conventional OD techniques. The comparison is done in terms of OD result accuracy and computational cost, and a detailed analysis on the sensitivity of the tool under operative conditions (object parameters availability and OD first guess accuracy) is offered. The whole analysis is performed with real optical observations for objects in GEO and HEO regimes.

Keywords: Orbit Determination; Semi-analytical.

Proc. 8th European Conference on Space Debris (virtual), Darmstadt, Germany, 20–23 April 2021, published by the ESA Space Debris Office
Ed. T. Flohrer, S. Lemmens & F. Schmitz, (http://conference.sdo.esoc.esa.int, May 2021)

1. INTRODUCTION

Orbit determination (OD) can be referred to as the process of estimating the position and velocity of a resident space object (RSO) from tracking measurements [8, 21, 28]. The process requires both the capability of accurately predicting an orbit given an initial condition and applying an estimation algorithm. The propagation of an RSO consists in solving a set of three second-order or six first-order differential equations, so as to determine the position and velocity of an orbiter at a given final time, t_f , from its initial conditions at epoch t_i [25]. As these equations are not directly integrable, there are three well-established techniques aimed at providing a solution to the problem, namely General Perturbation (GP) theories, Special Perturbation (SP) theories, and Semi-Analytical (SA) techniques. Each of these methods can be characterized in terms of the formulation of the equations of motion, the integration method used to obtain the solution to these equations and the selected perturbation technique. GP theories apply perturbation methods to the determination of an approximate analytical solution [1, 3, 5, 11, 13, 20]. Such solution, which is given as an explicit function of time and some physical parameters, allows for a fast determination of the coordinates at t_f. Nevertheless, the analytical solution is usually a low-order approximation in which only the most relevant forces are considered. SP theories, in contrast, perform a numerical integration of the equations of motion [12, 19]. They have the advantage of allowing for the consideration of any effect into the model, but they require small integration steps. SA theories offer a trade-off between the speed of analytical methods and the accuracy of SP by computing the orbital state as the sum of two different parts: one has very slow dynamics while the other corresponds to fast variations. The slowly-varying part includes secular and long-period terms and can be numerically integrated with very long steps. The exact formulas for the slowly-varying part (the mean element equations of motion or averaged equations) and for the fast-varying part (the short-periodic motion) are derived from assumed asymptotic expansions for the mean element equations of motion and the short-periodic motion, respectively. The short-period terms are represented as Fourier series. Computation is fast because numerical integration can be carried out with long steps, and the short-period terms are closed-form evaluations. Furthermore, high accuracy is obtained because the mean-element rate and short-period term computations can be performed for a wide range of perturbation types.

Once defined the propagation method, an estimation technique is required. Estimation algorithms can be divided into three categories: Initial Orbit Determination (IOD), Batch Least-Squares (BLSQ) differential correction and Sequential Filters (SFs). IOD estimates the orbit using the information from a few observations. The method is intended to provide the initial guess of an RSO orbit. A detailed description of several IOD methods can be found in [8] and [28]. BLSQ and SF need more observations than IOD. BLSQ takes all available observations over an arc and determines the best fit orbit at the epoch for a given system model, while SF processes sequentially the observations.

This paper presents a differential algebra-based least squares solved to perform accurate OD of resident space objects considering both numerical or SA propagators. The work is organized as follows. Section 2 offers an overview of the propagators. The differential algebra least squares (DALS) tool is then described in Section 3. Finally, the performance of the software is described in Section 5, where the results of its application on data collected while observing GEO and GTO objects are presented.

2. PROPAGATORS

Sections 2.1 and 2.2 describe the numerical and SA propagators used by the DALS solver. Regardless of the selected propagator, the state estimates and uncertainty propagation process is performed with a Differential Algebra (DA) implementation of the 8^{th}-order variable-stepsize Runge-Kutta integrator (RK78) by [27], with an 8^{th}-order solution for propagation and 7^{th}-order solution used for the stepsize control. A detailed description of the DA tools and implementation can instead be found in [7].

2.1. Numerical propagator AIDA

The Accurate Integrator for Debris Analysis (AIDA) is the numerical propagator used by the tool, tailored for space debris analysis within a DA framework [23]. The perturbations included in AIDA are the geopotential acceleration, atmospheric drag, solar radiation pressure (SRP), and third body gravity.

The acceleration due to the Earth's gravity potential is expressed as

$$\boldsymbol{a}_{grav} = \boldsymbol{\nabla}\frac{\mu_\oplus}{r}\sum_{n=0}^{\infty}\sum_{m=0}^{n}\frac{R_\oplus^n}{r^n}\bar{P}_{nm}(\sin\phi)\cdot \\ (\bar{C}_{nm}\cos(m\lambda)+\bar{S}_{nm}\sin(m\lambda)) \quad (1)$$

where $\mu_\oplus$ is the Earth's gravitational parameter, $R_\oplus$ is the Earth's radius, $\bar{C}_{nm}$ and $\bar{S}_{nm}$ the normalized geopotential coefficients, $\bar{P}_{nm}$ the normalized associated Legendre functions, r is the object distance from the centre of the Earth, and ϕ and λ are the geocentric latitude and longitude. Two possible gravitational models can be selected, i.e. the EGM96 and the EGM2008[24].

The perturbing acceleration due to the atmospheric drag is modelled as

$$\boldsymbol{a}_{drag} = -\frac{1}{2}C_D\frac{A}{m}\rho v_r\boldsymbol{v}_r = -\frac{1}{2}\text{Bfactor}\rho v_r\boldsymbol{v}_r \quad (2)$$

where C_D is the drag coefficient, A is the object cross-sectional area, m the object mass, whereas $\boldsymbol{v}_r$ is the object relative velocity with respect to the Earth atmosphere, and ρ is the atmospheric density at the object's position. The atmospheric density is computed using the Naval Research Laboratory's Mass Spectrometer and Incoherent Scatter Radar of year 2000 (NRLMSISE-00) model [26]. The model requires as inputs the solar and geomagnetic activity, geodetic altitude and latitude, longitude, year, day, and time of day in UT. AIDA allows for the definition of two different atmospheric models, namely with non rotating and rotating atmosphere.

The AIDA propagator allows the user to include Sun and Moon third body perturbations. The perturbing acceleration on the orbiting object is modelled as

$$\boldsymbol{a}_{3^{rd}} = \mu_{3^{rd}}\left(\frac{\boldsymbol{s}}{s^3}-\frac{\boldsymbol{r}_{3^{rd}}}{r_{3^{rd}}^3}\right) \quad (3)$$

where

$$\boldsymbol{s} = \boldsymbol{r}_{3^{rd}} - \boldsymbol{r}_{obj} \quad (4)$$

is the relative position of the third body with respect to the object. The position of the third body is computed using NASA's SPICE[1] toolkit. Three different scenarios can be considered: no perturbation, Moon-only perturbation or Moon and Sun perturbations.

Assuming that the detected object can be treated as a sphere, the SRP is modelled as

$$\boldsymbol{a}_{SRP} = \frac{L_S}{4\pi c}\frac{A}{m}(1+\epsilon)\frac{\boldsymbol{r}_{obj}-\boldsymbol{r}_S}{||\boldsymbol{r}_{obj}-\boldsymbol{r}_S||^3}\nu \\ = \frac{L_S}{4\pi c}\text{SRPC}\frac{\boldsymbol{r}_{obj}-\boldsymbol{r}_S}{||\boldsymbol{r}_{obj}-\boldsymbol{r}_S||^3}\nu \quad (5)$$

where c is the speed of light, L_S the solar luminosity, ϵ the body reflectivity, whereas ν is the shadow function. The tool allows the user to consider both Sun and Moon shadow, using either a cylindrical or biconical model.

2.2. SADA

The Semi-Analytical Propagation with Differential Algebra (SADA) propagator used in the DALS tool is a DA reformulation of HEOSAT, a SA propagator developed by [18] to study the long-term evolution of objects in Highly Elliptical Orbit (HEO) [9]. The perturbation model, which is propagated in time in the True of Date reference system, takes into account the gravitational effects due to zonal terms and lunisolar perturbations, SRP and atmospheric drag. The gravitational terms are expressed in Hamiltonian form to obtain the mean elements equations of motion using Deprit's perturbation algorithm [4] based on Lie transformations. The equations of motion due to SRP and drag perturbations are averaged over the mean anomaly via Gauss equations. The averaging techniques applied for developing HEOSAT are described in detail by [18].

The zonal-term Hamiltonians are simplified by removing parallactic terms (via elimination of the parallax, see [6] and [17]) and short-periodic terms are eliminated by Delaunay normalization. This is carried out up to second order of the second zonal harmonic J_2 and to first order for J_3-J_{10}. Mean element rates due to atmospheric drag are computed by numerically averaging Gauss equations over the mean anomaly assuming a spherical object and a rotating atmosphere. The atmospheric density is taken from the Harris-Priester atmospheric density model [10].

The disturbing potentials of the Sun and Moon (point-mass approximation) are expanded using Legendre series to obtain the Hamiltonians for averaging out the short-periodic terms [16]. Second and sixth-order Legendre polynomials are taken for the Sun and Moon potentials, respectively.

For averaging the equations of motion due to SRP, a spherical object and constant solar flux along the orbit (i.e. no shadow) are assumed. Kozai's analytical expressions for perturbations due to SRP [14] are then used to analytically average Gauss equations over the mean anomaly.

The averaged equations of motion due to tesseral resonance are obtained by averaging the tesseral Hamiltonian terms over the mean anomaly in the rotating frame to preserve the resonant terms

[1] https://naif.jpl.nasa.gov/naif/

[15]. For this, Kaula's expansion of the geopotential in orbital elements is used. In the current version of the code, only the 1:1 and 2:1 tesseral resonances, relevant for 24-hour and 12-hour orbits, respectively, are considered. Then, at each integration step, the SADA propagator computes the resonance period of each resonance, and the perturbation is considered only if this period is larger than 5 days (i.e. if the rate of change of the resonant angle is smaller than 2π/5days) [22].

3. DALS

The DALS software allows the user to estimate the state and associated covariance matrix of a transiting object detected by a radar or optical sensor starting from a set of measurements and an apriori estimate of the state of the object at a desired time epoch. The estimation is done with a least squares process formulated in the DA framework. The outputs of the program are an orbit parameter message (OPM) file, which contains the obtained state estimate, the associated covariance matrix and, if required, an estimate of two object parameters, namely the "Bfactor" parameter and the SRP coefficient, and two text files with measurement residuals. The process is performed in seven steps, illustrated hereafter.

1. Inputs elaboration The DALS toolbox requires three different inputs:

- A JSON input file, summarizing all the variables and data required by the process;
- A set of Tracking Data Message (TDM) files, which contain all epochs and measurements collected by the set of sensors considered;
- A .txt file including the epoch, the available a priori state estimate for the observed object and, if any, the associated state covariance matrix, the "Bfactor" coefficient and the SRP coefficient ("SRPC").

An example of JSON input file is offered in the appendix.

2. Observer kernel generation Once defined the DALS variables, the program generates the required SPICE kernels for all the involved ground stations of all the considered sensors. Optical and monostatic radar sensors require a single kernel per sensor. Two different kernels per sensor are instead generated in case of bistatic radars. For each ground station, the starting point of the kernel generation process is represented by the geodetic latitude, longitude and altitude provided by the user in the JSON input file. Then, according to the Earth model defined, these coordinates are either maintained in the WGS84 reference frame or converted in the ITRF93 system. A so-called "definitions file" is then generated, and used as input for the SPICE pinpoint [2] utility. The function generates a binary file, providing the position of the ground station in any reference frame, and the associated frame kernel, which defines the topocentric reference system associated with the observer.

3. TDM reading Starting from the data written in the JSON input file, the program then stores all the available observation epochs and associated sensor measurements. The parsing of all TDM files is done on the basis of the sensor specifications, i.e. the sensor type and the availability of the single measurements.

4. State estimates reading Once defined the set of available measurements, the program considers the state estimates file defined in the JSON input file and retrieves the apriori state estimate $\boldsymbol{x}_{data}$ and the associated time epoch t_{data}. Then, depending on their availability, the program stores the apriori covariance matrix $\boldsymbol{C}_{data}$ and the Bfactor and SRPC estimates. While the availability of the covariance is not essential for the process, a first guess for the two parameters is required, thus they are arbitrarily set to 1e-3 in case no estimate is provided. This value is later refined during the DALS process.

5. State estimates propagation The available state estimate is then propagated to the estimation epoch t_{est} defined by the user in the JSON input file, thus obtaining the apriori state estimate $\boldsymbol{x}_{est,apr}$. If t_{est} coincides with t_{data}, no propagation is required. Otherwise, two different scenarios are possible. If no apriori covariance is available, then $\boldsymbol{x}_{data}$ is propagated from t_{data} to t_{est}, thus obtaining $\boldsymbol{x}_{est,apr}$. Otherwise, a DA state vector of arbitrary domain is built on $\boldsymbol{x}_{data}$, i.e.

$$[\boldsymbol{x}_{data}] = \boldsymbol{x}_{data} + \delta\boldsymbol{x} \tag{6}$$

The DA vector is then propagated to t_{est} using a DA-based RK78 propagator, and the expansion of the state vector at t_{est}, $[\boldsymbol{x}_{est}]$, is thus obtained.

[2] https://naif.jpl.nasa.gov/pub/naif/utilities/PC_Linux_64bit/pinpoint.ug

This DA map is then used to compute the derivatives of the final state with respect to the initial conditions, i.e. the high order expansion of the state transition matrix

$$[\Phi]_{ij} = \frac{\partial [x_{est}]_i}{\partial \delta x_j} \tag{7}$$

with $i, j = 1, \dots, 6$. By extracting the constant part of this matrix, the state transition matrix is obtained, and it used to propagate the covariance matrix from t_{data} to t_{est}

$$\boldsymbol{C}_{est,apr} = \boldsymbol{\Phi} \boldsymbol{C}_{data} \boldsymbol{\Phi}^{\mathrm{T}} \tag{8}$$

The state vector $\boldsymbol{x}_{est,apr}$ is instead obtained by extracting the constant part of $[\boldsymbol{x}_{est}]$.

6. DA guess initialization Once computed $\boldsymbol{x}_{est,apr}$, a first guess for the DALS solver is generated. Once again, the process changes according to the availability of $\boldsymbol{C}_{est,apr}$. If the covariance matrix is available, this matrix is used to infer the search interval of the DALS, and thus initialize the DA state vector. Starting from $\boldsymbol{C}_{est,apr}$, the first step consists in computing the eigenvalues and eigenvectors of the matrix, namely $\boldsymbol{\lambda}$ and $\boldsymbol{V}$. At this point, a DA state error vector is built in the eigenspace as

$$[\boldsymbol{\varepsilon}]_i^{eig} = k \sqrt{\lambda_i} \delta x_i \tag{9}$$

with $i = 1, \dots, 6$ and k an amplification factor defined in the JSON input file. The vector is then rotated in the ECI reference frame

$$[\boldsymbol{\varepsilon}] = \boldsymbol{V} [\boldsymbol{\varepsilon}]^{eig} \tag{10}$$

and used to initialize a DA state at t_{est}, that is

$$[\boldsymbol{x}_{est}] = \boldsymbol{x}_{est,apr} + [\boldsymbol{\varepsilon}] \tag{11}$$

In case no covariance is available the DA state vector is directly initialized in the ECIJ2000 reference frame by setting an arbitrary scaling factor for each single DA variable, that is

$$[x_{est}]_i = (x_{est,apr})_i + k_{pos} \delta x_i \tag{12}$$

with $i = 1, \dots, 3$ and

$$[x_{est}]_i = (x_{est,apr})_i + k_{vel} \delta x_i \tag{13}$$

with $i = 4, \dots, 6$. The two coefficients k_{pos} and k_{vel} define the dimension of the search space around the current estimate, and are set in the JSON input file.

The user can decide to estimate also the Bfactor or the SRPC parameter or both. This is done by setting to "true" the corresponding flags in the JSON input file, or if no a priori estimate for one parameter is available. By defining with β an arbitrary parameter, the initialization is done by selecting a scaling factor equal to ten percent of the value of the parameter, i.e.

$$[\beta] = \beta + 0.1 \beta \delta x_k \tag{14}$$

with k equal to 7 or 8, depending on how many parameters must be initialized (1 or 2) and which parameter is considered.

At the end of the initialization process, an augmented DA state vector $[\boldsymbol{x}_{est}^a]$ is built. This vector coincides with $[\boldsymbol{x}_{est}]$ in case neither Bfactor nor SRPC need no be estimated, otherwise $[\boldsymbol{x}_{est}^a]$ can be

$$[\boldsymbol{x}_{est}^a] = \{[\boldsymbol{x}_{est}], [\text{Bfactor}]\} \tag{15}$$

or

$$[\boldsymbol{x}_{est}^a] = \{[\boldsymbol{x}_{est}], [\text{SRPC}]\} \tag{16}$$

or

$$[\boldsymbol{x}_{est}^a] = \{[\boldsymbol{x}_{est}], [\text{Bfactor}], [\text{SRPC}]\} \tag{17}$$

The length of the augmented state vector defines the number of required DA variables and is here defined as n_{DA}.

7. The DALS solver The DALS solver starts from the DA first guess initialized in the previous step and iteratively changes its constant part so that the value of the measurements residual is minimized. The process is run in the DA framework and makes use of the find_min_box_constrained function of the DLib C++ optimization library. The function requires a search and stop strategy, a function to minimize and its gradient, a starting point and the search domain. The DALS solver adopts the DLib bfgs_search_strategy search strategy and the objective_delta_stop_strategy with a tolerance of 1e-16 as stopping criterion, whereas the DA expansions of the residual and of its gradient are used. The centre of the DA domain is used as starting point, whereas the scaling strategy adopted to build the DA variables allows the solver to always use $[-1, 1]^{n_{DA}}$ as search domain. During the iteration process, the DALS solver constantly controls the accuracy of the resulting residual map. A dedicated performance index is computed, and if it is lower than the ε_{DA}^{lim} value set in the JSON file, then the map is considered accurate. In parallel, two possible different convergence criteria can be considered: a check on how much the residual changes and how far the new point is with respect to the previous one, or a check on the norm of the gradient function as computed in the optimal solution. In the first case, the threshold parameters are defined as ε_r^{lim} and

$\varepsilon^{lim}_{\delta \boldsymbol{x}}$, and are defined in the JSON input file. In case the gradient is controlled, the $\varepsilon^{\mathrm{lim}}_{\nabla r}$ is the only parameter to be defined in the JSON file.

The process starts by setting $[\boldsymbol{x}^a]_{(k)} = [\boldsymbol{x}^a_{est}]$, $\delta \boldsymbol{x}^a_{opt,(k-1)} = \boldsymbol{0}$ and $k = 1$, where the subscript (k) refers to the iteration index. Then, the process runs as follows

1. Propagate the available state estimate $\boldsymbol{x}_{est}$ from t_{est} to the set of observation instants $T = \{t_1, \ldots, t_N\}$.
2. Compute, for each time instant, the required set of measurements. The measurement estimation process changes according to the sensor considered. In case of optical sensors, the steps are as follows

   2.1. Compute the relative position object-observer

   $$[\boldsymbol{\rho}(t_k)] = [\boldsymbol{rr}(t_k)] - \boldsymbol{rr}_{GS}(t_k) \qquad (18)$$

   where $[\boldsymbol{rr}(t_k)]$ is the DA expansion of the object position vector at time t_k, whereas $\boldsymbol{rr}_{GS}(t_k)$ is the ground station position at time t_k. This position is computed using a dedicated SPICE kernel generated by the tool.

   2.2. Compute the topocentric right ascension

   $$[\alpha(t_k)] = \text{atan2}\left(\frac{[\rho(t_k)]_2}{[\rho(t_k)]_1}\right) \qquad (19)$$

   where the subscripts "1" and "2" refer to the components of the vector.

   2.3 Compute the topocentric declination

   $$[\delta(t_k)] = \text{asin}\left(\frac{[\rho(t_k)]_3}{||[\boldsymbol{\rho}(t_k)]||}\right) \qquad (20)$$

   The tool allows the user to include light-time and stellar aberration (LTS) corrections. The corrections are made with a DA-based version of SPICE's LTS correction routines. The set of measurements provided by a radar sensor is instead potentially larger. Both monostatic and bistatic radars theoretically can provide azimuth and elevation of the object as measured by the receiver, along with Doppler shift and range (or better slant range) measurements. Similarly to optical sensors, the availability and accuracy of these measurements are governed by the associated entries in the JSON input file object. Starting from the available set of DA state vectors $\{[\boldsymbol{x}]\}$, the steps for computing, for a generic time instant t_k, the DA expansions of azimuth, elevation, Doppler shift and slant range are the following:

   2.1. Compute the relative position object-receiver

   $$[\boldsymbol{\rho}_{RX}(t_k)] = [\boldsymbol{rr}(t_k)] - \boldsymbol{rr}_{RX}(t_k) \qquad (21)$$

   where $\boldsymbol{rr}_{RX}(t_k)$ is the receiver (RX) position at t_k.

   2.2. Compute the rotation matrix $A_{eci2RX}(t_k)$ from the ECIJ2000 to the RX topocentric reference frame and rotate $[\boldsymbol{\rho}_{RX}(t_k)]$

   $$[\boldsymbol{\rho}_{RX}(t_k)]^{RX} = A_{eci2RX}(t_k)[\boldsymbol{\rho}_{RX}(t_k)] \qquad (22)$$

   2.3. Compute the azimuth of the object at the RX

   $$[Az_{RX}(t_k))] = -\text{atan2}\left(\frac{[\boldsymbol{\rho}_{RX}(t_k)]_2^{RX}}{[\boldsymbol{\rho}_{RX}(t_k)]_1^{RX}}\right) \qquad (23)$$

   2.4. Compute the elevation of the object at the RX

   $$[El_{RX}(t_k)] = \text{asin}\left(\frac{[\boldsymbol{\rho}_{RX}(t_k)]_3^{RX}}{||[\boldsymbol{\rho}_{RX}(t_k)]||}\right) \qquad (24)$$

   2.5. Compute the slant range DA expansion $[SR(t_k)]$. If the radar is monostatic, then

   $$[SR(t_k)] = 2||[\boldsymbol{\rho}_{RX}(t_k)]|| \qquad (25)$$

   If the radar is bistatic, then compute the relative position object-transmitter

   $$[\boldsymbol{\rho}_{TX}(t_k)] = [\boldsymbol{rr}(t_k)] - \boldsymbol{rr}_{TX}(t_k) \qquad (26)$$

   where $\boldsymbol{rr}_{TX}(t_k)$ is the transmitter (TX) position at t_k. Then the slant range becomes

   $$[SR(t_k)] = ||[\boldsymbol{\rho}_{RX}(t_k)]|| + ||[\boldsymbol{\rho}_{TX}(t_k)]|| \qquad (27)$$

   2.6. Compute the rotation matrix $A_{eci2ecef}(t_k)$ from the ECIJ2000 to the Earth-fixed reference frame

   2.7. Compute the Doppler shift expansion $[DS(t_k)]$. In case of monostatic radar

   $$[DS(t_k)] = -\frac{2f}{c}A_{eci2ecef}\cdot \cdot\left([\boldsymbol{vv}(t_k))]\cdot\frac{[\boldsymbol{\rho}_{RX}]}{||[\boldsymbol{\rho}_{RX}]||}\right) \qquad (28)$$

   where f is the radar frequency and c is the speed of light, whereas for a bistatic radar

   $$[DS(t_k)] = -\frac{f}{c}A_{eci2ecef}\left([\boldsymbol{vv}(t_k))]\cdot \cdot\frac{[\boldsymbol{\rho}_{RX}]}{||[\boldsymbol{\rho}_{RX}]||} + [\boldsymbol{vv}(t_k))]\cdot\frac{[\boldsymbol{\rho}_{TX}]}{||[\boldsymbol{\rho}_{TX}]||}\right) \qquad (29)$$

3. Compute the DA expansion of the residuals. The residuals are computed by comparing estimated and real measurements and considering the possible availability of an apriori estimate for state and covariance. The residual expansion can be essentially divided in two parts

$$[r] = [r_{meas}] + [r_{apr}] \tag{30}$$

that is, a portion due to the measurement difference and a portion due to a priori estimates. The computation of the first part depends on the sensor type. For optical sensors, the residual expansion computed at time epoch t_k can be expressed as

$$[r_{meas}(t_k)] = \frac{f_\alpha}{\sigma_\alpha^2}\left([\alpha(t_k)] - \alpha_{real}(t_k)\right)^2 + \frac{f_\delta}{\sigma_\delta^2}\left([\delta(t_k)] - \delta_{real}(t_k)\right)^2 \tag{31}$$

where the subscript "real" refers to the real available measurements, σ_α and σ_δ define the accuracy of the sensor in measuring right ascension and declination and are set in the JSON input file, whereas f_α and f_δ govern the availability of each type of measurement, and can be either 0 or 1. In case of radar measurements, the expression becomes

$$[r_{meas}(t_k)] = \frac{f_{Az}}{\sigma_{Az}^2}\left([Az_{RX}(t_k)] - Az_{RX,real}(t_k)\right)^2 + \frac{f_{El}}{\sigma_{El}^2}\left([El_{RX}(t_k)] - El_{real}(t_k)\right)^2 + \frac{f_{SR}}{\sigma_{SR}^2}\left([SR(t_k)] - SR_{real}(t_k)\right)^2 + \frac{f_{DS}}{\sigma_{DS}^2}\left([DS(t_k)] - DS_{real}(t_k)\right)^2 \tag{32}$$

where σ_{Az}, σ_{El}, σ_{SR} and σ_{DS} define the accuracy of the sensor in measuring azimuth, elevation, slant range and Doppler shift and are set in the JSON input file, whereas f_{Az}, f_{El}, f_{SR} and f_{DS} govern the availability of each type of measurement. The complete residual expansion can be then expressed as

$$[r_{meas}] = \sum_{k=1}^{N} [r_{meas}(t_k)] \tag{33}$$

where, at each time instant t_k, the optical or radar expression for $[r_{meas}(t_k)]$ are used depending on the sensor used at time t_k. The computation of $[r_{apr}]$ is instead insensitive to the sensor type. By defining with t_{est} the estimation epoch of the DALS process, $[\boldsymbol{x}_{est}] = [\boldsymbol{x}](t_{est})$ the estimate at t_{est}, $\boldsymbol{x}_{est,apr}$ the apriori estimate and $\boldsymbol{C}_{est,apr}$ the associated covariance, one obtains

$$[r_{apr}] = \left([\boldsymbol{x}_{est}] - \boldsymbol{x}_{est,apr}\right)^{\mathrm{T}} \boldsymbol{C}_{est,apr}^{-1} \left([\boldsymbol{x}_{est}] - \boldsymbol{x}_{est,apr}\right) \tag{34}$$

While $\boldsymbol{x}_{est,apr}$ is always available, the presence of the apriori covariance is not granted and it is governed by a dedicated input flag: if it is set to "true", then it means that the state estimates file contains an apriori estimate for the covariance, thus $[r_{apr}]$ can be computed, otherwise this step is skipped and $[r] = [r_{meas}]$.

4. Compute the gradient of the residuals. The operation is done in the DA framework, that is

$$[\nabla r]_{(k)} = \frac{\partial [r]_{(k)}}{\partial \delta \boldsymbol{x}} \tag{35}$$

where $\delta \boldsymbol{x} \in \mathbb{R}^{n_{DA}}$

5. Run the minimization process using the find_min_box_constrained function. The function provides the optimal deviation $\delta \boldsymbol{x}^a_{opt,k}$.

6. Compute the value of the residual in the optimal point

$$r_{opt,(k)} = [r]_{(k)}(\delta \boldsymbol{x}^a_{opt,(k)}) \tag{36}$$

7. Update the DA map of the state vector

$$[\boldsymbol{x}^a]_{(k+1)} = [\boldsymbol{x}^a]_{(k)} + [\boldsymbol{x}^a]_{(k)}(\delta \boldsymbol{x}^a_{opt,(k)}) - [\boldsymbol{x}^a]_{(k)}(\mathbf{0}) \tag{37}$$

8. Compute the new DA expansion of the residuals

$$[r]_{(k+1)} = f([\boldsymbol{x}^a]_{(k+1)}) \tag{38}$$

9. Estimate the residuals map accuracy

$$\varepsilon_{DA,(k)} = \frac{|[r]_{(k+1)}(\mathbf{0}) - r_{opt,(k)}|}{1 + r_{opt,(k)}} \tag{39}$$

10. If $\varepsilon_{DA,(k)} < \varepsilon_{DA}^{lim}$, continue, otherwise set $k = k + 1$ and go back to step 1.

11. If the convergence criterion is set to "relative", compute residual and step relative variations

$$\varepsilon_{r,(k)} = \frac{|[r]_{(k+1)}(\mathbf{0}) - [r]_{(k)}(\mathbf{0})|}{1 + [r]_{(k)}(\mathbf{0})}$$
$$\varepsilon_{\delta \boldsymbol{x},(k)} = \frac{||\delta \boldsymbol{x}^a_{opt,(k)} - \delta \boldsymbol{x}^a_{opt,(k-1)}||}{(1 + ||\delta \boldsymbol{x}^a_{opt,(k-1)}||)} \tag{40}$$

If $\varepsilon_{r,(k)} < \varepsilon_{r}^{lim}$ or $\varepsilon_{\delta \boldsymbol{x},(k)} < \varepsilon_{\delta \boldsymbol{x}}^{lim}$ stop, otherwise set $k = k+1$ and go back to step 1. If instead the convergence criterion is set to "gradient", then evaluate the new gradient map in the optimal point and compute its norm

$$\varepsilon_{\nabla r,(k)} = ||[\nabla r]_{(k+1)}(\mathbf{0})|| \tag{41}$$

If $\varepsilon_{\nabla r,(k)} < \varepsilon_{\nabla r}^{lim}$ stop, otherwise set $k = k+1$ and go back to step 1.

At the end of the process, a refined estimate of the augmented state vector $\boldsymbol{x}^{a}_{est,opt} = [\boldsymbol{x}^{a}_{est,opt}](\mathbf{0})$ is obtained. In addition, the availability of the residual map allows the solver to provide an estimate of the covariance matrix associated with the estimated state vector and object parameters, if any. Let us define with $[r]$ the residuals DA map associated with the optimal solution of the DALS process. By relying on differential algebra, one can compute the DA expansion of the normal matrix as

$$[N]_{i,j} = \frac{1}{2}\frac{\partial^2 [r]}{\partial \delta x_i \partial \delta x_j} \tag{42}$$

where $i, j = 1, \ldots, n_{DA}$ The covariance matrix can be then computed as

$$\boldsymbol{C}^{a}_{est,opt} = \frac{[r](\mathbf{0})}{n_{meas} - n_{DA}} \boldsymbol{N}^{-1} \tag{43}$$

where n_{meas} is the overall number of measurements. The so-obtained matrix is an “augmented” covariance matrix, that is it is a (n_{DA} x n_{DA}) matrix and includes the contribution of both the estimated state and object parameters, if any.

Once estimated the state and associated covariance matrix, the results are organized in an OPM file. The epoch of the estimate and the associated state vector are always included, and are expressed in the UTC time system and ECIJ2000 reference frame, respectively. The covariance matrix can be included or not, depending on the flag set in the JSON input file. If the Bfactor or the SRPC parameters are estimated, they are included as user defined parameters. In addition, the associated standard deviations can be provided.

4. TRADE

TRADE is a tool developed by Deimos, which performs offline OD with different algorithms. It implements a Square Root Information Filter (SRIF) [2], a BLSQ and a Bayes filter [29]. In this context, we focused on the BLSQ to compare its results against the DALS tool. TRADE supports optical telescopes and monostatic radars. It is capable of processing tracks from different sensors (and of different types) simultaneously. It includes a numerical propagator with a detailed set of perturbations: non-spheric Earth, third bodies, SRP, atmoshpheric drag, albedo effect and solid tides. These perturbations are toggleable. It allows determining the orbital elements, the drag and SRP coefficients, as well as the transponder delay at the satellite (part of the time budget in the two-way ranging measurements). Finally, either impulsive or continuous manoeuvres can be included in the dynamical model, as long as enough data about them are available. Given epoch, duration and direction of the manoeuvre, TRADE is able to estimate its magnitude provided a good initial guess is available: such a scenario is likely to happen when satellite operators want to obtain useful information about the efficiency of planned manoeuvres and performance of the engine.
TRADE computes the adjusted orbit at the epoch of the first measurement in the batch or at an user-defined epoch. It allows providing an initial estimation of the solution by means of Two-Line Elements (TLE), by interpolating over a user-provided Orbit Ephemeris Message (OEM) file or by manually inserting the value. It also implements IOD algorithms for telescope and monostatic radars, so it is possible to obtain solutions without initial estimations (as long as the input data allow a good IOD). As outputs, it provides per-sensor plots of the estimated residuals, as well as plots of the optical measurements in Right Ascension/Declination and Azimuth/Elevation. It also provides standard-compliant OEM orbits with or without covariances, and it has the possibility of computing pointing opportunities for the determined orbits and the configured sensors network.

5. RESULTS

This section illustrates the performance of the DALS tool when analysing real measurements obtained with optical sensors. Two different targets, a GEO and a GTO object, were observed and analysed, and the performance of the software is here compared against results obtained with TRADE. The aim of the analyses is to assess the accuracy of the DALS solver and check the validity of the propagaton models according to the selected orbital regimes. The results are illustrated hereafter.

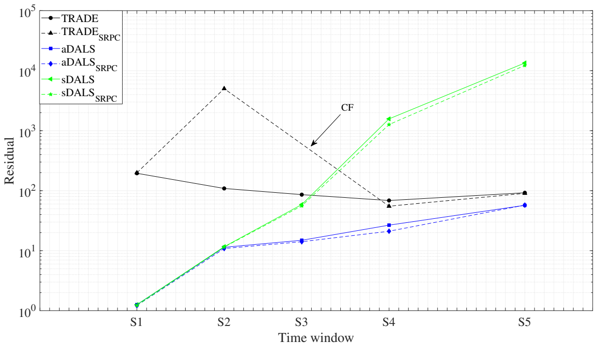


Figure 1: *DALS and TRADE residuals as a function on the time slot. Six different solvers are shown: aDALS (blue lines), sDALS (green lines) and TRADE, with an without estimation of SRPC (dashed and solid lines, respectively). The term “CF” indicates a converge failure.*

5.1. The GEO regime

The first analysed object is a GEO object, object SL-12 R/B(2), NORAD ID 22115. Optical measurements coming from three different sensors were collected during a time frame of around one month and a half, from the beginning of May 2020 to the end of June 2020. The locations of the optical sensors are shown in Tab. 1. The obtained TDM files are organized in five different time slots of increasing time length, and for each slot the available TDM files are processed by both DALS and TRADE to obtain an estimate of the state of the transiting object at the epoch of the first available measurement. Details about the time slots and the number of available TDMs per optical sensor are given in Table 2. The first guess for the orbital state of the object at the epoch of the first available measurement is obtained by relying on the latest available TLE. Starting from this value, having no covariance matrix available, the DALS software is run assuming four different configurations: numerical (aDALS) or SA (sDALS) propagation, and, for each configuration, SRP coefficient constant (9.5e-2) or estimated (“SRPC” pedex). The Bfactor coefficient is not relevant here, given the considered orbital regime. This section illustrates the results obtained by processing each slot of TDMs separately.

A first evidence of the performance of the tool can be obtained by looking at the trend of the residuals. Fig. 1 shows how the least-squares residuals evolve with increasing observation windows considering six different solvers: aDALS (blue lines), sDALS (green lines) and TRADE, with an without estimation of SRPC (dashed and solid lines, respectively). A logarithmic scale for the residuals (y-axis) is used to help visualization, whereas the x-axis simply reports the acronym of each single time slot. The spacing is proportional to the elapsed time since the first measurement.

Table 1: Optical sensors location.

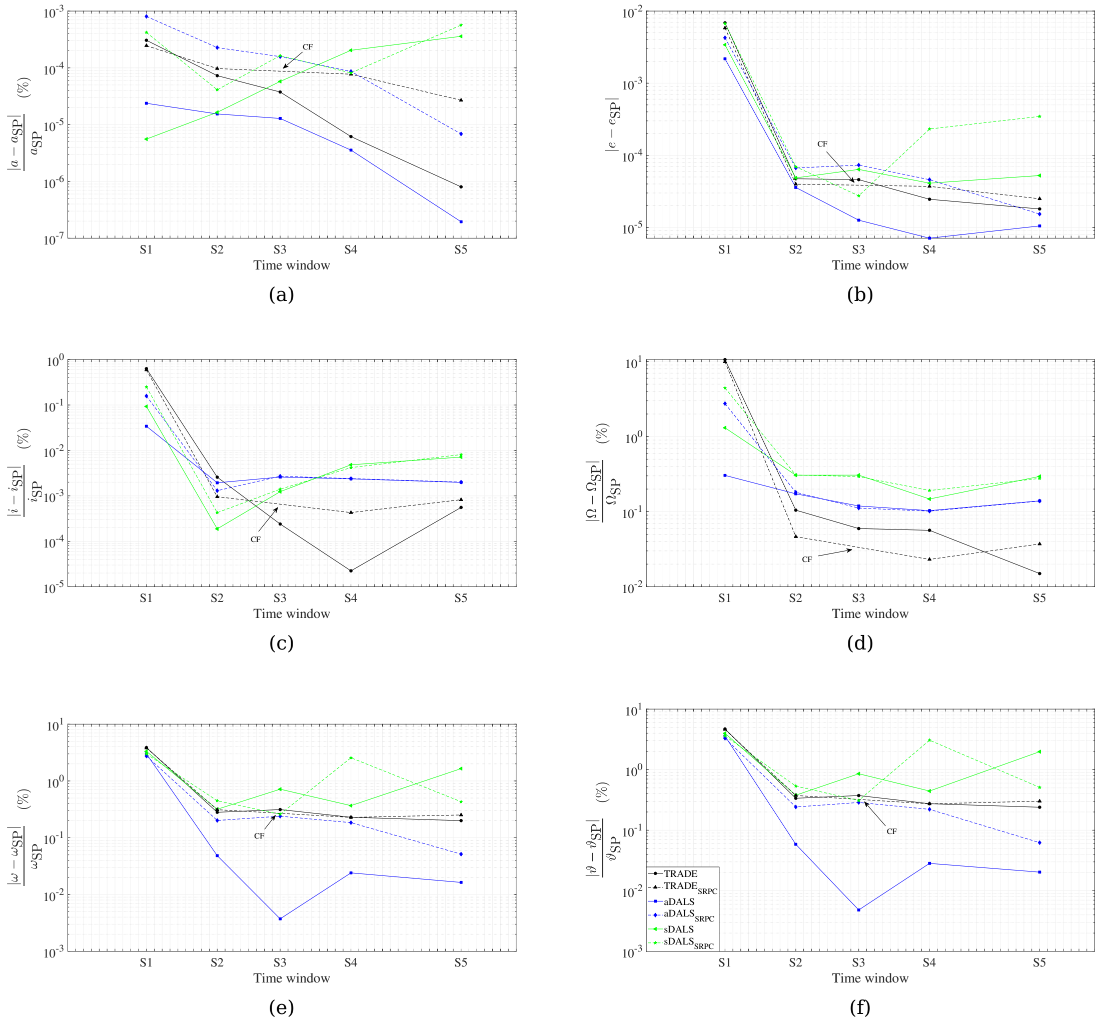
Name	Latitude (deg)	Longitude (deg)	Altitude (m)
TFRM	42.052 N	0.729 E	1622
CENTU1	38.544 N	-4.408 W	1120
PIGGY	42.052 N	0.729 E	1622
TRE	21.199 S	55.410 E	982

Table 2: Time slots (start and end date, 2020) and number of available TDM per time slot for each considered optical sensor.

Slot	TFRM	CENTU1	PIGGY	TRE
S1: 08 May–13 May	6	0	0	0
S2: 08 May–22 May	11	6	0	0
S3: 08 May–30 May	15	7	0	0
S4: 08 May–08 June	23	10	1	0
S5: 08 May–22 June	29	10	7	0

A first consideration on the effectiveness of the DA-based least squares solver can be made by comparing the behaviour of TRADE’s and aDALS’s residuals. Both methods, indeed, rely on numerical propagators, so, assuming no major differences between the underlying orbital dynamics exist, a comparison between their performance allows us to assess the validity of the DA-based least squares solver. The aDALS systematically has lower residuals than TRADE, both in case of SRPC estimated and fixed. The difference between the two methods is significant when a shorter observation window is considered, while it progressively reduces as the number of considered TDMs and the covered time window increase. Specifically, while for the S1 case (see Tab. 2) aDALS’s residuals are two orders of magnitude lower than TRADE’s, the difference between the residuals collapses to a factor 1.5 when the whole set of available TDMs is processed. In addition, the performance of aDALS is less affected by the estimation of SRPC, while TRADE shows high residuals for the second time slot, and cannot reach convergence in S3 (“CF”,convergence failure text box in the plot).

Once assessed the quality of the DALS solver, the comparison between aDALS and sDALS provides a better picture of the impact of the selected or-



(a)

(b)

(c)

(d)

(e)

(f)

Figure 2: *Keplerian parameters as a function of the observation window: (a) semi-major axis; (b) eccentricity; (c) inclination; (d) right ascension of the ascending node; (e) argument of perigee; (f) true anomaly (object NORAD ID 22115). The term "CF" indicates a converge failure.*

bital dynamics on the DALS results. sDALS seems to provide good residuals up to 2-3 weeks of observations, whereas its performance drastically decreases as the considered time frame increases. This can be easily inferred by analysing the results for slots S3, S4, and S5. While the difference between aDALS and sDALS residuals is just a factor three for a three-week observation window, it rapidly grows to one and two orders of magnitude when one month and one and a half months of observations are processed, respectively. Therefore, the application of the SADA propagator to the GEO regime seems valid only for relatively short time frames.

The knowledge of the observed object and the availability of high accuracy ephemerides allows us to make further considerations on the performance of the three methods. Fig. 2 shows the trend of the obtained state estimate expressed in terms of Keplerian parameters, namely semi-major axis (a), eccentricity (e), inclination (i), right ascension of the ascending node (Ω) and true anomaly (ϑ). Each parameter is plotted as a normalized difference with respect to a reference value provided by the special perturbations ("SP" pedex) catalogue. The legend for this plot is the same used for Fig. 1, and it is shown in Fig. 2f.

The trend of the semi-major axis estimates and the result obtained without estimating the SRP coefficient are presented in Fig. 2a, solid lines. The considerations previously made for the residuals are still valid. Both the aDALS and the sDALS methods can accurately estimate the value of a since the very beginning, when the observations cover less than one week. On the contrary, TRADE's result is less accurate, with an error of around 50 m.

Table 3: *Time slots (start and end date, 2020) and number of available TDM per time slot for each considered optical sensor (object NORAD ID 25503).*

Slot	TFRM	CENTU1	PIGGY	TRE
S1: 21 Feb–27 Feb	5	5	5	2
S2: 21 Feb–19 Mar	7	8	6	2
S3: 21 Mar–29 Mar	4	1	6	0
S4: 21 Mar–10 Apr	10	2	9	0
S5: 13 May–27 May	8	3	0	0
S6: 13 May–19 June	9	3	1	0

As the number of collected TDMs and the covered observation window increase, the difference between TRADE and aDALS progressively reduces, whereas sDALS starts diverging, losing accuracy after three weeks of observations. In case the estimation of the SRPC coefficient is included in the least-squares process, the balance of power between the methods slightly changes. The most accurate results with short observation windows are now granted by TRADE, whereas aDALS is initially relatively inaccurate. The difference between the two progressively reduces for increasing observation windows, and aDALS eventually outperforms TRADE when more than one month of observations is considered. On the contrary, the results granted by sDALS are initially reliable, and match or even outperform aDALS's till one months of observations. When longer observation windows are considered, sDALS loses its validity, thus resulting into a divergent trend.

The trend observed for the eccentricity is similar, with aDALS showing always the best performance and TRADE progressively reducing the gap as the number of processed TDMs increases. Again, the performance of sDALS results acceptable only for limited observation windows. The results are different for the estimation of the orbital plane. Here an increase in the number of processed TDMs leads TRADE to outperform both DALS methods. The opposite trend can be instead observed for the estimation of the in-plane position of the object: in this case, aDALS is always the best choice, regardless of the estimation approach for the SRPC.

Overall, if no estimation of the SRPC is included and the longest time frame is considered, we can see that aDALS provides the best estimates for a, e, ω and ϑ, whereas TRADE better estimates the orientation of the orbital plane. Similar considerations can be made in case of estimated SRPC, with aDALS outperforming TRADE in 4 cases out of 6.

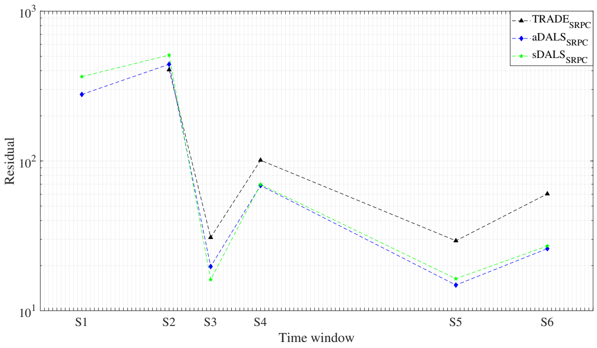
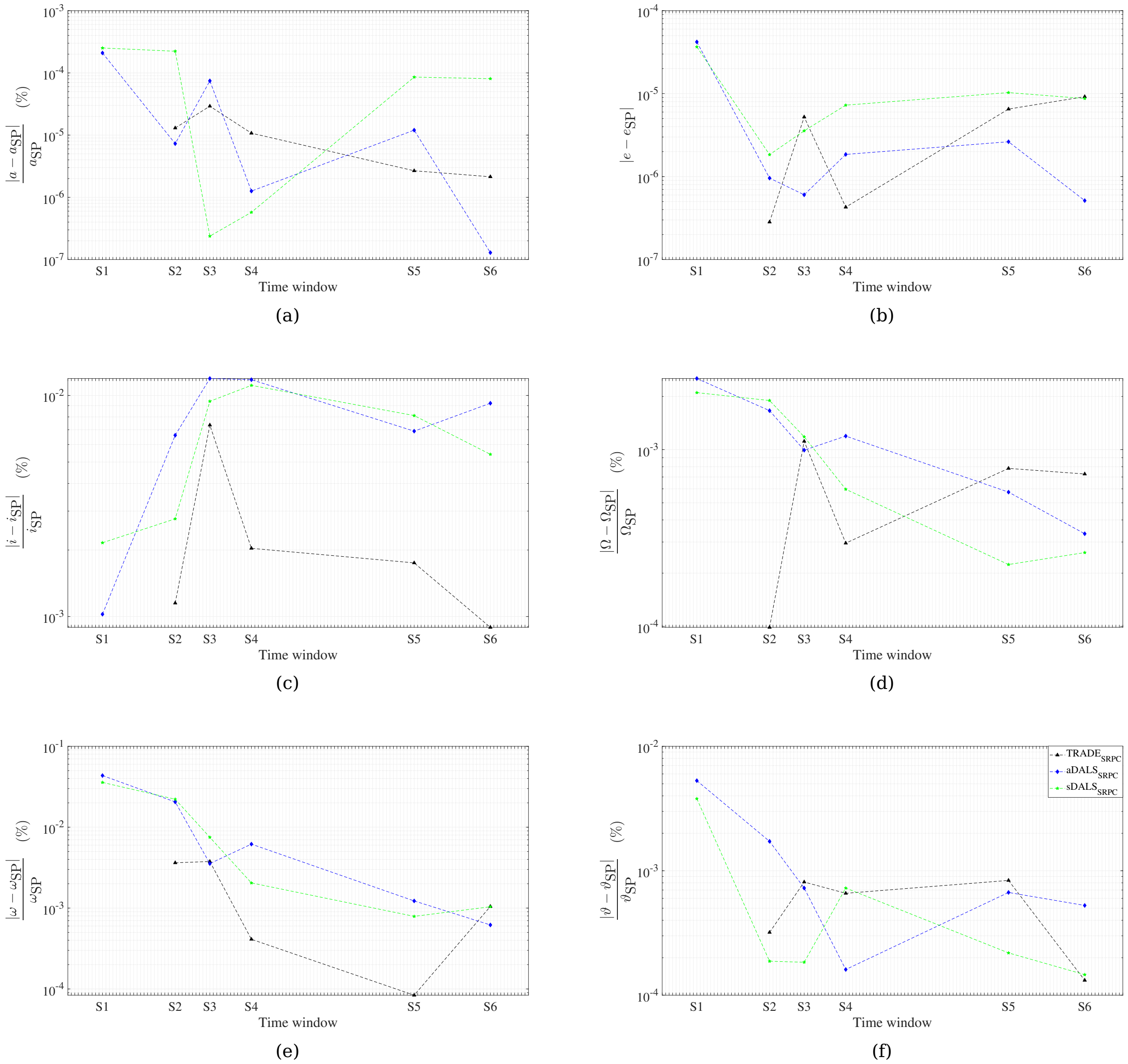


Figure 3: *DALS and TRADE residuals trend as a function on the time slot considered (object NORAD ID 25503).*

5.2. The GTO regime

The second object here presented is a GTO object, MAQSAT/ARIANE 5 (NORAD ID 25503). The object was observed several times by the four optical sensors listed in Tab. 1, between the end of February and the mid of June 2020. In order to make computations feasible, the overall bulk of TDMs was divided in three groups, and two subslots per group were considered. Details about the considered time frames and associated numbers of TDMs per sensor are given in Tab. 2. Then, the three solvers (aDALS, sDALS and TRADE) were executed, and their performance compared. In this case, given the higher impact of the SRPC on the orbital motion, only the case of estimated SRPC is considered. The trend of the obtained residuals in shown in Fig. 3. As previously explained, unlike the previous case now TDMs are divided into three groups, so moving from the left to the right does not necessarily implies that the number of processed TDMs increases. The following plots refer to the time slots displayed in Tab. 3.

Similarly to the GEO case, TRADE is not always capable of converging, and this typically occurs when the number of processed TDMs is low (S1, see Tab. 3). Moreover, aDALS almost systematically outperforms it in terms of obtained residuals. This occurs for slots S1, S3, S4, S5 and S6, i.e. 5 slots out of 6. The biggest difference with respect to the trend shown in Fig. 1 can be noticed in the behaviour of sDALS. In 4 cases out of 5 sDALS shows a lower residual than TRADE, and in one case (S3, window 21 March–29 March) it outperforms aDALS as well. This result is significant because, while 3 of these 4 slots include observations spread on less than 3 weeks, in one case (S6) the TDMs cover more than one month of observations, and here sDALS almost matches aDALS and outperforms TRADE by more than a



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4: *Keplerian parameters as a function of the observation window: (a) semi-major axis; (b) eccentricity; (c) inclination; (d) right ascension of the ascending node; (e) argument of perigee; (f) true anomaly (object NORAD ID 25503).*

factor 2. If one recalls the results obtained with the GEO object, the difference is quite evident. In GTO regime the SA propagator better captures the orbital motion of the system, and so it provides more accurate results.

As explained before, a lower residual is an evidence of the quality of the least squares solver but it does not necessarily implies a more accurate solution, as the underlying orbital dynamics is not the same. A better illustration of the performance of the three approaches in estimating the GTO object state can be obtained by looking at the relative errors in the orbital parameters, as normalized with respect to the SP reference. The results are shown in Fig. 4. The comparison between TRADE and aDALS shows that the trend is now less uniform, while in the case of the GEO object aDALS nearly always outmatched TRADE. Regardless of the considered time window, TRADE almost always outperforms aDALS in the estimation of the orbital inclination and argument of perigee. For the other orbital parameters the trends are less evident, and none of the three methods shows better performance than the others.

Moreover, it is interesting to study the behaviour

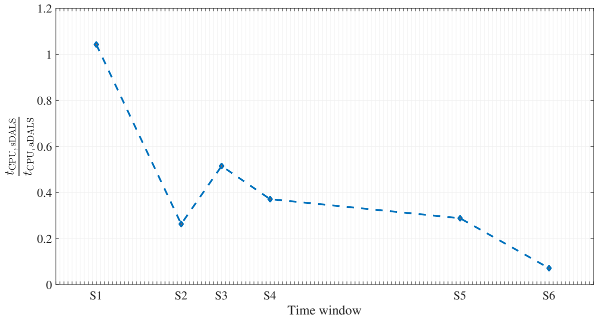
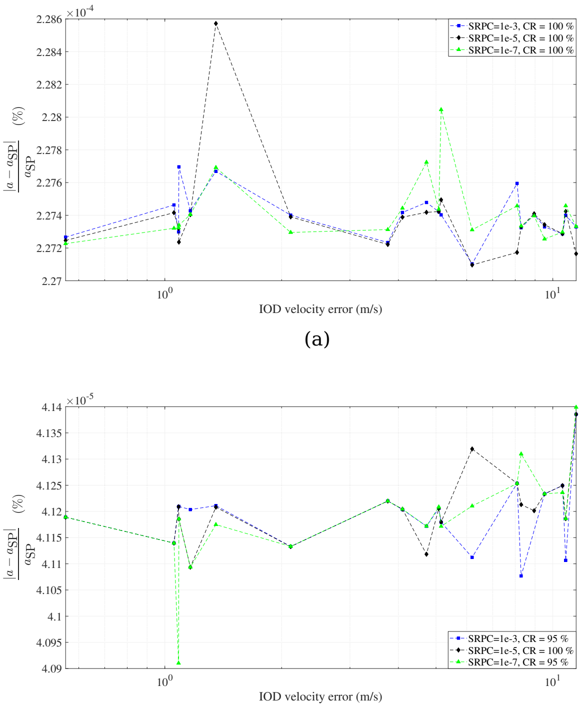


Figure 5: *DALS computational time (object NORAD ID 25503).*

of sDALS. Divergent behaviours can no longer be identified, and the performance of the SA tool can be labelled as valuable in all the considered time frames. In particular, it is worth mentioning the good behaviour of the tool in the last time slot (S6, 13 May–19 June), where, despite a time frame of more than one month, the method can grant very accurate estimates that for some parameters result as the best ones (Ω) or nearly the best ones (ϑ). Overall, the application of a SA propagator to the estimation of the state of a GTO orbiting objects seems a valuable solution. These results are actually not unexpected. As mentioned in Section 2.2, the SADA propagator, indeed, is specifically tailored to the HEO regime, so the best performance can be obtained for these objects. The appeal of such a solution becomes evident if the required computational time is inspected. Figure 5 shows a comparison between aDALS and sDALS in terms of computational effort. As can be seen, the sDALS method almost systematically outperforms aDALS, with a gain that goes up to a factor 7 in case of longer observation windows. That is, in case of GTO objects, the sDALS allows us to save computational time while still granting the same level of accuracy.

5.3. Sensitivity on IOD accuracy

The results previously illustrated were obtained using TLEs to compute an initial guess. The latest available TLE at the epoch of the first measurement is generally quite accurate, and convergence failures are therefore seldom encountered. In order to assess the robustness of the algorithms against the accuracy level of the state estimate first guess, a sensitivity analysis is carried out. Four different ranges of increasing errors in position and velocity are considered: (1 km, 1m/s), (5 km, 5 m/s), (10 km, 10 m/s) and (100 km, 10 m/s). For each range, and for each position and velocity component, 5 samples are drawn, and the corresponding error vectors in the radial-along track-cross track reference frame built. These vectors are then rotated in the ECI reference frame and added to the nominal TLE state, thus generating a perturbed initial guess. In addition, for each generated perturbed state, three different initial guesses for the SRPC (1e-3, 1e-5 and 1e-7) are assumed. The DALS process is then carried out allowing the estimation of the SRPC. The goal of the analysis is to check how the DALS performance changes as a function of the initial errors. A single object and a single time frame is assumed: the GEO object 22115 and the time frame S2 (08 May–22 May). The results are shown in Fig. 6. The two plots display the relative error in semi-major axis as a function of the error in velocity of the initial guess. Three different colours are used for the three different initial values for the SRPC. In addition, the convergence rate (CR) is reported. Figure 6a shows the results for aDALS, whereas the performance of sDALS is illustrated in Fig. 6b. As the IOD estimate error increases, no divergence trend can be noticed, for none of the two methods. That is, both aDALS and sDALS are robust enough and can converge to the correct solution also when the initial errors in position and velocity are of the order of hundred of meters and tens of meters per second, respectively.



(a)

(b)

Figure 6: *Relative error on semi-major axis as a function of the IOD velocity error for three different SRPC values: (a) aDALS; (b) sDALS (object NORAD ID 22115, slot S2).*

6. CONCLUSIONS

This paper introduced a differential-algebra based least squares solver, referred to as DALS, for the accurate estimation of the orbital state of RSOs. Two different versions of the algorithm were presented, one relying on a numerical propagator and a second one based on a SA propagation model. The performance of the tool was assessed against two different objects, a GEO and a GTO object. The results show that the SA version of the tool seems really promising for the GTO regime. Here the algorithm can offer a good accuracy level while granting significant savings in the required computational burden. In contrast, this approach is not optimized for GEO regime, where the reduced computational cost is associated with a lower accuracy, especially for longer observation windows. The presented results represent the first stage of our analysis, as they are limited to two orbital regimes and optical sensors. The tool, indeed, is capable of handling any sensor configuration for a large variety of orbiting objects. Future studies will be dedicated to the analysis of the DALS performance when observing LEO objects, with both radar and optical sensors.

APPENDIX A: JSON INPUT FILE

An example of JSON input file is shown below.

```
{
  "DALS": {
    "date_est": "29 MAR 2020 16:20:42.934672 (UTC)",
    "flag_Bfactor_est":true,
    "flag_SRPC_est":true,
    "fun_tol":1e-5,
    "map_tol":,0.001,
    "max_iter":10,
    "order":2,
    "output": {
      "flag_Cov":true,
      "flag_KeplPar":true,
      "name":"Sensor_results.opm",
      "obj_ID":41223,
      "obj_name":"unknown"
      "path":"./Results/"
      },
    "step_tol":1e-6,
    "stop_crit":"relative"
  },

  "Dynamics":{
    "AIDAparam":{
      "flag_SRP":3,
      "flag_drag":2,
      "flag_thirdbody":2,
      "gravmodel":{
        "name":"egm2008",
        "order":15,
        "path":"./data/gravmodels/"
      }
    },
    "method":"AIDA",
    "tolerance":1e-11
  },

  "Earth":{
    "Earth_model":"ITRF93",
    "kernels":{
      "IAU_assoc":"earth_assoc_IAU.tf",
      "ITRF93":"earth_000101_200629_200407.bpc",
      "ITRF93_assoc":"earth_assoc_itrf93.tf",
      "path":"./data/kernels/",
      "pck":"pck00010.tpc"
    }
  },

  "OD":{
    "flag_Bfactor_av":true,
    "Covariance":{
      "flag_Cov_av":true,
      "sigma":5
    },
    "flag_SRPC_av":true,
    "name":"Sensor_OD.txt".
    "path":"./OD/"
  },

  "Observer":{
    "n_sensors":1,
    "sensor_1":{
      "participant_1":{
        "altitude":1.622,
        "latitude":42.0516,
        "longitude":0.7293,
        "name":"TFRM"
        "path":"./Observatory"
      },
      "sensor_setup":{
        "declination":{
          "TDM_name":"ANGLE_1",
          "accuracy":0.3333333,
          "availability":true
        },
        "right_ascension":{
          "TDM_name":"ANGLE_2",
          "accuracy":0.3333333,
          "availability":true
        }
      },
      "type":"optical",
      "flag_LTS":true,
      "n_TDM":1,
      "TDM":{
        "TDM_1":"Sensor1.tdm",
        "path":"./Measurements/"
      }
    }
  }
}
```

The file is organized in five portions. The first portion ("DALS") includes all the variables required by the DALS solver, such as the estimation date, flags for the estimations of the object parameters, and settings for the least squares process. The second portion ("Dynamics") is dedicated to the definition of the propagator (AIDA or SADA) and associated flags for orbital perturbations, whereas the third part ("Earth") is used to declare the selected Earth model and associated SPICE kernel names. The fourth slot ("OD") includes information on the IOD estimate, whereas the last slot ("Observer") is used to define all the involved sensors, their location and measurements accuracy and the associated TDMs. In the example, a single optical sensor with a single TDM is used, but the software can handle whatever configuration: multiple TDMs, homogeneous sensors, heterogeneous sensors (e.g. optical and radar, both monostatic and bistatic).

ACKNOWLEDGMENTS

The EU SST activities have received funding from the European Union programmes, notably from the Horizon 2020 research and innovation programme under grant agreements No 760459 and No 952852.

Disclaimer: The content of this paper reflects only the view of the SST Cooperation and the European Commission and the Research Executive Agency are not responsible for any use that may be made of the information it contains.

REFERENCES

1. Aksnes K., (1970). *The Astronomical Journal*, 75:1066-1076.
2. G.J. Bierman, (1977). Factorization methods for discrete sequential estimation.
3. Brouwer D., (1959). *The Astronomical Journal*, 64:378–397.
4. Deprit A., (1969). Celestial mechanics, 1:12–30.
5. Deprit A. and Rom A., (1970). *Celestial mechanics*, 2:166-206.
6. Deprit A., (1981). *Celestial mechanics*, 24:111–153.
7. Di Lizia P., (2008). Robust space trajectory and space system design using differential algebra. PhD Thesis.
8. Escobal P., (1965). Methods of orbit determination. New York:Wiley.
9. Gondelach, D. J., (2019). PhD Thesis.
10. Harris I. and Priester W., (1962). *Journal of Atmospheric Sciences*, 19(4):286–301.
11. Kinoshita H., (1977). Third-order solution of an artificial-satellite theory. SAO special report no. 379, Smithsonian Astrophysical Observatory, Cambridge, MA, USA.
12. Kinoshita H. and Nakai H., (1988). *Celestial mechanics*, 45:231–244.
13. Kozai, Y., (1962). *The Astronomical Journal*, 67:446–461.
14. Kozai Y., (1963). *Smithsonian Contributions to Astrophysics*, 6:109–112.
15. Lara M., San-Juan J.F., Folcik Z.J. and Cefola P., (2011). *The Journal of the Astronautical Sciences*, 58:661–676.
16. Lara M., San-Juan J.F., López L.M. and Cefola P.J., (2012). *Celestial Mechanics and Dynamical Astronomy*, 113:435–452.
17. Lara M., San-Juan J.F. and López-Ochoa L.M., (2014). *Celestial Mechanics and Dynamical Astronomy*, 120:39–56.
18. Lara M., San-Juan J.F. and Hautesserres D., (2018). *CEAS Space Journal*, 10:3-23.
19. Long A.C., Cappellari J.O., Velez C.E. and Fuchs A.J., (1989). Goddard trajectory determination system (GTDS) mathematical theory (revision 1). Technical Report CSC/TR-89/6001, Computer Sciences Corporation.
20. Lyddane R.H., (1963). *The Astronomical Journal*, 68:555–558.
21. Montenbruck O. and Gill E., (2000). Satellite Orbits: Models, Methods, and Applications. Springer
22. Morand V., Caubet A., Deleflie F., Daquin J. and Fraysse H., (2013). Semi analytical implementation of tesseral harmonics perturbations for high eccentricity orbits. Proceedings of the AAS/AIAA Astrodynamics Specialist Conference.
23. Morselli A., (2014). High order methods for Space Situational Awareness. PhD Thesis
24. Pavlis N.K., Holmes S.A., Kenyon S.C. and Factor J.K., (2012). *Journal of geophysical research: solid earth*, 117(B4):1–38.
25. Pérez, I., San-Martín, M., López, R., Vergara, E.P., Wittig, A., San-Juan, J.F., (2017). *Lecture Notes in Computer Science*, 10334:650–661.
26. Picone J.M., Hedin A.E., Drob D. Pj and Aikin A.C., (2002). *Journal of Geophysical Research: Space Physics*, 107(A12):1–16.
27. Prince P.J. and Dormand J.R., (1981). *Journal of computational and applied mathematics*, 7(1):67–75.
28. Vallado D.A., (2013). Fundamentals of Astrodynamics and Applications. Space Technology Library
29. Wiesel W.E., (2003). Modern orbit determination.