GTOC13: Solution File Format Specification

Etienne Pellegrini, Gregory Lantoine, Anastassios Petropoulos, Damon Landau, Gregory Whiffen, Mark Wallace, Sungmoon Choi, Brian Anderson, Zubin Olikara, Jon Sims

Jet Propulsion Laboratory, California Institute of Technology [*]

Version 1.0 — October 10, 2025

1 Purpose and Scope

This document defines the file format for trajectory solution submissions to the 13th Global Trajectory Optimisation Competition (GTOC13).

This document describes only the file format and structural requirements: field definitions, allowed values and units, record ordering, trajectory structure, and certain limited tolerances that are intrinsic to the solution format.

Dynamical validation constraints, including propagation accuracy, flyby feasibility, control constraints, etc, are described separately in the GTOC13 Problem Statement document.

2 Solution File Overview

A solution file contains the data for an entire GTOC13 trajectory. If a team wishes to alter part or all of their solution, a new file for the entire trajectory must be submitted. The submission epoch used in scoring the new file is the epoch at which that new file was submitted. If the file passes the validation and scoring steps, the team's current Leaderboard entry is updated if the new submission improves upon it.

GTOC13 trajectories consist of sequences of three fundamental arc types:

1. **Flyby arcs**: Gravity assist maneuvers at planetary bodies
2. **Conic arcs**: Heliocentric coast phases, computed analytically
3. **Propagated arcs**: Numerically integrated heliocentric phase with solar sail

Section 3 describes the structure of each arc type, and the structural requirements that apply to each arc and to the transitions between arcs.

[*]Copyright © 2025 California Institute of Technology. US Government sponsorship acknowledged.

2.1 File Format

Solution files shall be ASCII text files. Any line beginning with `#` or `!` is treated as comments and ignored. Empty lines are ignored. All other lines are data rows, and only contain numerical data. Each row represents a single state point along the trajectory. Each row must contain exactly 12 fields, which may be separated by commas (',') and/or ASCII whitespace (spaces or tabs). Multiple adjacent spaces, or a comma followed by spaces, are treated as one separator.

2.2 Data fields

Each row contains the following 12 fields:

Column	Name	Type	Description
1	`body_id`	integer	Central body identifier
2	`flag`	integer	Flag (arc-dependent)
3	`epoch`	float	Time (seconds past $t = 0$)
4–6	`position`	float	Heliocentric position vector $[x, y, z]$
7–9	`velocity`	float	Heliocentric velocity vector $[v_x, v_y, v_z]$
10–12	`control`	float	Control vector (arc-dependent)

The first two columns are flags used to categorize the arc and determine its scoring. If written to file as floating point numbers, they are cast to integers upon parsing the file.

Body Identifier (`body_id`)

- `body_id` $= 0$: Heliocentric (two-body motion relative to the star)
- `body_id` > 0: Body flyby (body identifier as defined in ephemeris)

Flag (`flag`)

For heliocentric arcs (`body_id` $= 0$):

- `flag` $= 0$: Conic arc (ballistic coast, no propulsion, analytical propagation)
- `flag` $= 1$: Numerically propagated arc (solar sail active or coast with propagator)

For flyby arcs (`body_id` > 0):

- `flag` $= 0$: Not counted as a science flyby
- `flag` $= 1$: Counted as a science flyby

2.3 Units and Coordinate Frame

All trajectory data shall use the following units:

- **Epoch**: seconds past reference epoch $t = 0$
- **Position**: kilometers (km)
- **Velocity**: kilometers per second (km/s)
- **Control vectors**: dimensionless unit vectors normal to the sail for heliocentric propagated arcs (see sec. 3.3) ; for conic arcs, they are $[0, 0, 0]$; for flyby arcs, they represent v_∞ in km/s (see sec. 3.1).

The coordinate frame is defined in Section 4 of the GTOC13 Problem Statement.

3 Arc Type Specifications

This section describes the structural requirements for each arc type. Each record in the solution file is categorized as one of the three fundamental arc types, according to the values of `body_id` and `flag`:

$$\begin{cases} \texttt{body\_id} > 0: & \text{flyby arc} \\ \texttt{body\_id} = 0: & \begin{cases} \texttt{flag} = 0: & \text{conic arc} \\ \texttt{flag} = 1: & \text{propagated arc} \end{cases} \end{cases}$$

3.1 Flyby Arcs

3.1.1 Structure

A flyby arc consists of exactly **2 consecutive rows** with `body_id` > 0. The structure is shown in Table 1.

Table 1: Flyby Arc Structure

Description	`body_id`	`flag`	`epoch`	`position`	`velocity`	`control`
Incoming state	k	f	t_{fb}	$\mathbf{r}_{\text{fb}}$	$\mathbf{v}^-$	$\mathbf{v}_\infty^-$
Outgoing state	k	f	t_{fb}	$\mathbf{r}_{\text{fb}}$	$\mathbf{v}^+$	$\mathbf{v}_\infty^+$

where:

- $\mathbf{r}_{\text{fb}}$: Spacecraft heliocentric position at flyby
- $\mathbf{v}^-$: Incoming spacecraft heliocentric velocity
- $\mathbf{v}^+$: Outgoing spacecraft heliocentric velocity
- $\mathbf{v}_\infty^-$: Incoming v_∞ vector
- $\mathbf{v}_\infty^+$: Outgoing v_∞ vector

3.1.2 Structural Requirements

- Both rows have identical `body_id` > 0
- Both rows have identical `flag`
- Both rows have identical epochs: $|t_{i+1} - t_i| < \epsilon_{\text{time}}$
- Both rows have identical positions: $\|\mathbf{r}_{i+1} - \mathbf{r}_i\| < \epsilon_{\text{pos}}$

3.1.3 Transitions

Flybys must be preceded by a heliocentric row ending at the flyby epoch and followed by a heliocentric row starting at the flyby epoch.

3.2 Conic Arcs

3.2.1 Structure

A conic arc represents pure two-body Keplerian motion and consists of exactly **2 consecutive rows**, representing its start and end states, as shown in Table 2.

Table 2: Conic Arc Structure

Description	`body_id`	`flag`	`epoch`	`position`	`velocity`	`control`
Conic start	0	0	t_{start}	$\mathbf{r}_{\text{start}}$	$\mathbf{v}_{\text{start}}$	$[0, 0, 0]$
Conic end	0	0	t_{end}	$\mathbf{r}_{\text{end}}$	$\mathbf{v}_{\text{end}}$	$[0, 0, 0]$

3.2.2 Structural Requirements

- Both rows must have `body_id` $= 0$ (heliocentric)
- Both rows must have `flag` $= 0$ (conic propagation)
- Control vectors must be $[0, 0, 0]$ for both rows
- Time must increase: $t_{\text{end}} > t_{\text{start}} + \epsilon_{\text{time}}$

3.2.3 Transitions

Conic arcs can be preceded or followed by flybys or propagated arcs. The previous arc must end at the conic's starting epoch, and the next arc must start at the conic's ending epoch.

3.3 Propagated Arcs

3.3.1 Structure

A propagated arc represents solar sail propulsion or numerical integration and consists of a variable number of rows (≥ 2). The basic structure is shown in Table 3.

Table 3: Propagated Arc Structure with continuous controls

Description	`body_id`	`flag`	`epoch`	`position`	`velocity`	`control`
Row 1	0	1	t_1	$\mathbf{r}_1$	$\mathbf{v}_1$	$\mathbf{u}_1$
Row 2	0	1	t_2	$\mathbf{r}_2$	$\mathbf{v}_2$	$\mathbf{u}_2$
Row 3	0	1	t_3	$\mathbf{r}_3$	$\mathbf{v}_3$	$\mathbf{u}_3$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Row n	0	1	t_n	$\mathbf{r}_n$	$\mathbf{v}_n$	$\mathbf{u}_n$

3.3.2 Structural Requirements

- All rows must have `body_id` $= 0$ (heliocentric)
- All rows must have `flag` $= 1$ (numerically propagated)
- Time must be non-decreasing
- The minimum time step is 60 seconds: $t_{i+1} - t_i >= 60s$
- Minimum of 2 rows required per arc

3.3.3 Control Specification

Control vectors $\mathbf{u}_i$ are unit vectors normal to the sail, specifically the inward-pointing vector, $\hat{\mathbf{u}}_{\mathbf{n}}$, as defined in Section 5 of the GTOC13 Problem Statement. Each segment from t_i to t_{i+1} is defined by the control vectors $\mathbf{u}_i$ and $\mathbf{u}_{i+1}$ at its endpoints. The control profile along the segment interior is computed during validation such that integration error is minimized.

Control discontinuities are represented by two consecutive rows with the same epoch and states (within tolerances), but different control vectors. This allows piecewise-constant control profiles, as well as abrupt changes in continuous controls. The structure for control discontinuities is shown in Table 4.

Table 4: Propagated Arc Structure with discontinuous controls

Description	`body_id`	`flag`	`epoch`	`position`	`velocity`	`control`
Seg. 1 start	0	1	t_1	$\mathbf{r}_1$	$\mathbf{v}_1$	$\mathbf{u}_1$
Seg. 1 end	0	1	t_2	$\mathbf{r}_2$	$\mathbf{v}_2$	$\mathbf{u}_1$
Seg. 2 start	0	1	t_2	$\mathbf{r}_2$	$\mathbf{v}_2$	$\mathbf{u}_2$
Seg. 2 end	0	1	t_3	$\mathbf{r}_3$	$\mathbf{v}_3$	$\mathbf{u}_2$
Seg. 3 start	0	1	t_3	$\mathbf{r}_3$	$\mathbf{v}_3$	$\mathbf{u}_3$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Requirements for control discontinuities:

- $|t_{i+} - t_{i-}| < \epsilon_{\text{time}}$
- $\|\mathbf{r}_{i+} - \mathbf{r}_{i-}\| < \epsilon_{\text{pos}}$
- $\|\mathbf{v}_{i+} - \mathbf{v}_{i-}\| < \epsilon_{\text{vel}}$

For piecewise-constant controls, set the control at segment end equal to the control at segment start, as shown in the example where segment 1 uses constant control $\mathbf{u}_1$ throughout, then segment 2 switches to constant $\mathbf{u}_2$, etc.

3.3.4 Transitions

Propagated arcs can be preceded and followed by conic or flyby arcs starting or ending at the arc's end or start epochs respectively.

4 Validation Overview

4.1 Structural Validation

The following structural requirements are enforced during file parsing:

1. **Time matching**: Rows representing the same epoch must satisfy $|t_i - t_j| < \epsilon_{\text{time}}$

2. **State continuity**: Rows representing the same epoch must satisfy:

   - $\|\mathbf{r}_i - \mathbf{r}_j\| < \epsilon_{\text{pos}}$
   - $\|\mathbf{v}_i - \mathbf{v}_j\| < \epsilon_{\text{vel}}$

   except for flybys, which can have velocity discontinuities.

3. **Arc structure**: Each arc type must follow the structural requirements specified in Section 3

4. **Arc transition**: Transition between arcs requires two rows at the same epoch: one ending the incoming arc, and one starting the outgoing one

5. **Time monotonicity**: Epochs must be non-decreasing (within tolerance)

6. **Arc duration**: For heliocentric arcs, time must advance: $t_{\text{end}} > t_{\text{start}} + \epsilon_{\text{time}}$
7. **Initial condition**: First arc must be heliocentric (`body_id` $= 0$)
8. **Final condition**: We do not prescribe a specific arc type for the solution to end at. However:
   - Solutions ending with a flyby can end on a single row with `body_id` > 0 (representing the incoming state only). Such single-row flybys can count for science scoring. If the outgoing row is provided, the arc will undergo dynamical validation.
   - Solutions ending with a heliocentric arc will undergo dynamical validation
   - Note that since scoring is only affected by flybys, heliocentric arcs after the last science flyby will not impact the score

4.2 Dynamical Validation

Dynamical validation (propagation accuracy, flyby modeling and constraints, control constraints) is described in the GTOC13 Problem Statement and is not part of this format specification.

5 Generic File Example

Table 5 shows the generic structure of a solution file containing multiple arc types. Note that this is an illustrative example; actual trajectories may have different arc sequences.

Table 5: Solution File format

Line number	Data					
Header (optional)	`body_id`	`flag`	`epoch`	`pos`	`vel`	`control`
1 (initial conic)	0,	0,	t_0,	$\mathbf{r}_0$,	$\mathbf{v}_0$,	$\mathbf{0}$
2	0,	0,	t_1,	$\mathbf{r}_1$,	$\mathbf{v}_1^-$,	$\mathbf{0}$
3 (science flyby of body k_1)	k_1,	1,	t_1,	$\mathbf{r}_1$,	$\mathbf{v}_1^-$,	$\mathbf{v}_{\infty,1}^-$
4	k_1,	1,	t_1,	$\mathbf{r}_1$,	$\mathbf{v}_1^+$,	$\mathbf{v}_{\infty,1}^+$
5 (numerical prop.)	0,	1,	t_1,	$\mathbf{r}_1$,	$\mathbf{v}_1^+$,	$\mathbf{u}_1$
6	0,	1,	t_2,	$\mathbf{r}_2$,	$\mathbf{v}_2$,	$\mathbf{u}_2$
7	0,	1,	t_3,	$\mathbf{r}_3$,	$\mathbf{v}_3$,	$\mathbf{u}_3$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	0,	1,	t_n,	$\mathbf{r}_n$,	$\mathbf{v}_n$,	$\mathbf{u}_n^-$
$n+1$ *(control disc.)*	0,	1,	t_n,	$\mathbf{r}_n$,	$\mathbf{v}_n$,	$\mathbf{u}_n^+$
$n+2$	0,	1,	t_{n+1},	$\mathbf{r}_{n+1}$,	$\mathbf{v}_{n+1}$,	$\mathbf{u}_{n+1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
m	0,	1,	t_m,	$\mathbf{r}_m$,	$\mathbf{v}_m$,	$\mathbf{u}_m$
$m+1$ *(conic)*	0,	0,	t_m,	$\mathbf{r}_m$,	$\mathbf{v}_m$,	$\mathbf{0}$
$m+2$	0,	0,	t_{m+1},	$\mathbf{r}_{m+1}$,	$\mathbf{v}_{m+1}$,	$\mathbf{0}$
$m+3$ *(numerical prop.)*	0,	1,	t_{m+1},	$\mathbf{r}_{m+1}$,	$\mathbf{v}_{m+1}$,	$\mathbf{u}_{m+1}$
$m+4$ *(continuous controls)*	0,	1,	t_{m+2},	$\mathbf{r}_{m+2}$,	$\mathbf{v}_{m+2}$,	$\mathbf{u}_{m+2}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n_f *- 1*	0,	1,	t_f,	$\mathbf{r}_f$,	$\mathbf{v}_f$,	$\mathbf{u}_{n_f-1}$
n_f *(final flyby)*	k_f,	f_f,	t_f,	$\mathbf{r}_f$,	$\mathbf{v}_f$,	$\mathbf{v}_{\infty,f}^-$

where:

- k_i = body identifier at t_i
- t_i = epoch (seconds past $t = 0$)
- $\mathbf{r}_i = [x, y, z]$ = position vector (km)
- $\mathbf{v}_i = [v_x, v_y, v_z]$ = velocity vector (km/s)
- $\mathbf{v}_\infty^\pm$ = incoming/outgoing hyperbolic excess velocity relative to body
- $\mathbf{u}_i$ = propagated arc control vector (unit vector normal to the sail)
- Superscripts $-/+$ denote pre/post-flyby states and controls, and pre/post-control discontinuity controls

Notes

- The sequences and examples shown in this document are illustrative only.
- The numerical values of the tolerances ϵ_{time}, ϵ_{pos}, ϵ_{vel} are the continuity tolerances specified in the Problem Statement.
- Body IDs are provided in the different ephemeris files.