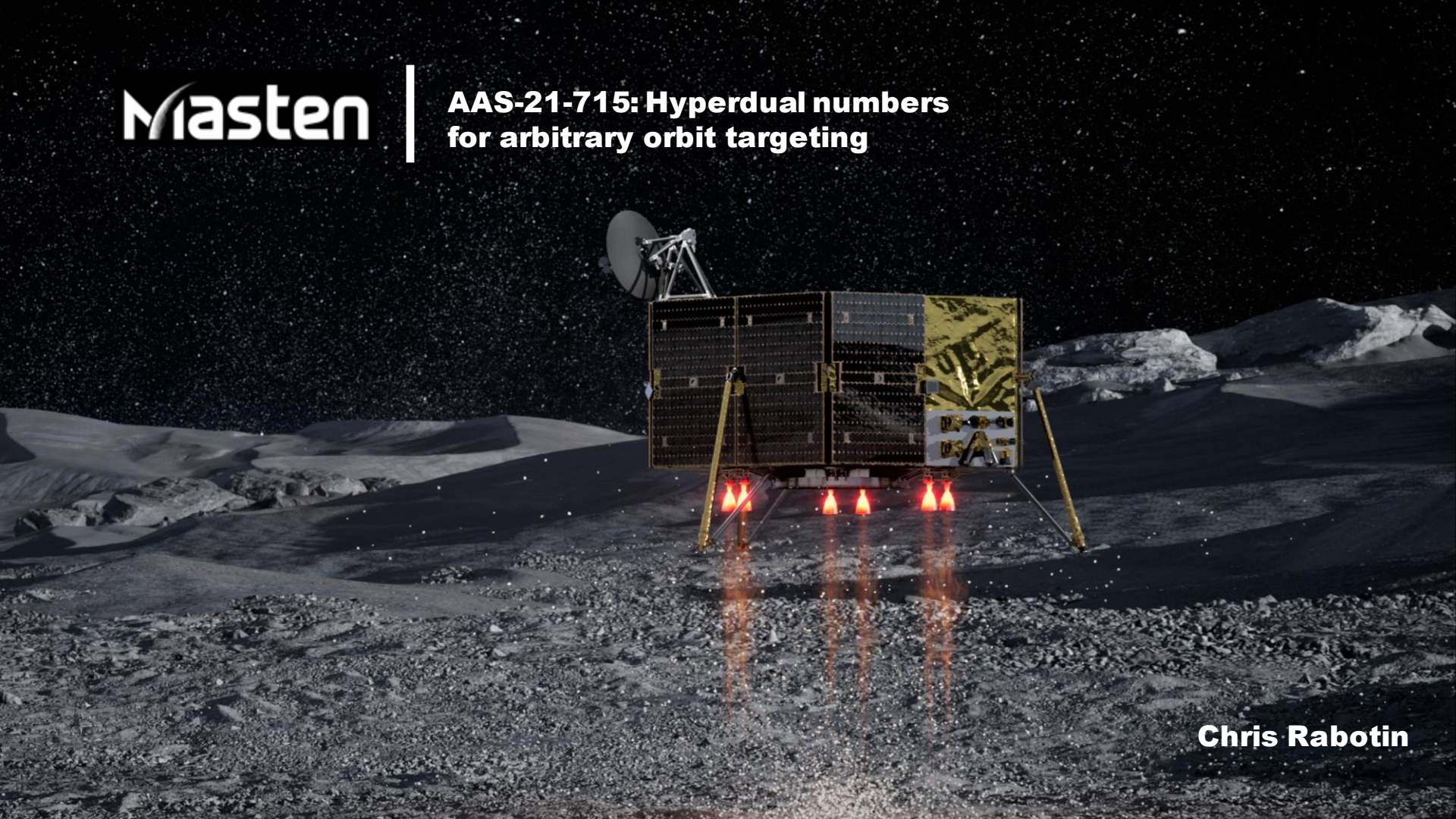
Masten

AAS-21-715: Hyperdual numbers for arbitrary orbit targeting



Chris Rabotin

Brief review of orbital targeting

Achieve an orbit with specific characteristics, objective vector (Gamma)

Requires the computation of the partial derivatives of the objectives with respect to the control vector

A control vector (U) is modified iteratively with a Newton Raphson process

The Newton Raphson differential correction rederived in paper, section V.B.

$$\frac{\partial \boldsymbol{\Gamma}_f}{\partial \mathbf{U}_i}$$

Masten

Computing partial derivatives

Analytically
Impractical for arbitrary objectives
Error prone and labor intensive

Finite differencing
Common method (used by NASA GMAT and STK Astrogator)
Precision of computation limited to half of machine precision

Automatic differentiation
Machine precision computation
Negligible computational overhead
Uses dual number theory

Dual number theory 1/2

Definition

Flavor of complex numbers with a nilpotent element, epsilon

$$\mathbb{D} = \mathbb{R}[\epsilon] = \{z = a + b\epsilon \mid (a, b) \in \mathbb{R}^2, \epsilon^2 = 0 \text{ and } \epsilon \neq 0\}$$

Moreover, for $z = a + b\epsilon$ where $z \in \mathbb{D}$, $(a, b) \in \mathbb{R}$, let us define the *real* and *dual* parts of a dual number such as

$$\begin{cases} real(z) = a \\ dual(z) = b \end{cases}$$

Dual number theory 2/2

Why it works

The nilpotent element enables automatic differentiation, as can be seen from a Taylor series expansion (recall that epsilon squared is zero)

$$f : \mathbb{D} \to \mathbb{D}\ , (a, b) \in \mathbb{R}^2$$

$$f(a + b\varepsilon) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a) b^n \varepsilon^n}{n!}$$

$$= f(a) + b\frac{df(a)}{da}\varepsilon$$

$$\begin{cases} real(f(a + b\epsilon)) = f(a) \\ dual(f(a + b\epsilon)) = b\frac{df(a)}{da} \end{cases}$$

Hyperdual numbers 1/3

Definition

Defining multiple orthogonal nilpotent elements enables multiple derivatives

$$\mathbb{D}^2 = \mathbb{R}[\epsilon_x, \epsilon_y] = \{z = a + b\epsilon_x + c\epsilon_y + d\epsilon_x\epsilon_y \mid (a,b,c,d) \in \mathbb{R}^4, \epsilon_\gamma^2 = 0,\ \epsilon_\gamma \neq 0,\ \gamma \in \{x,y\},\ \epsilon_x\epsilon_y \neq 0\} \quad (6)$$

This mathematical tool enables auto-differentiation of multi-variate functions as follows, where $dual_\gamma$ corresponds to the γ-th dual number, i.e. the number associated with ϵ_γ.

$$f : \mathbb{D}^2 \to \mathbb{D}^2,\ (x,y) \in \mathbb{R}^2$$

$$\begin{cases} real(f(x+\epsilon_x,\ y+\epsilon_y)) = f(x,y) \\ dual_x(f(x+\epsilon_x,\ y+\epsilon_y)) = \frac{\partial}{\partial x} f(x,y) \\ dual_y(f(x+\epsilon_x,\ y+\epsilon_y)) = \frac{\partial}{\partial y} f(x,y) \end{cases}$$

Hyperdual numbers 2/3

Example

Assume the following function and its hyperdual space

$$f : \mathbb{R} \to \mathbb{R},\ (x, y) \in \mathbb{R}^2$$

$$f(x, y) = 2x^3 - 0.2y^2 + x$$

$$\frac{\partial}{\partial x} f(x, y) = 6x^2 + 1$$

$$\frac{\partial}{\partial y} f(x, y) = -0.4y$$

Let us extend the definition of this function to $\mathbb{D}^2$.

$$g : \mathbb{D} \to \mathbb{D},\ (x, y) \in \mathbb{R}^2$$

$$g(x + \epsilon_x,\ y + \epsilon_y) = 2(x + \epsilon_x)^3 - 0.2(y + \epsilon_y)^2 + (x + \epsilon_x)$$

Hyperdual numbers 3/3

Example

Each nilpotent element ends up being a factor of its respective partial derivative

$$
\begin{aligned}
g(x+\epsilon_x,\ y+\epsilon_y) &= 2(x+\epsilon_x)^3 - 0.2(y+\epsilon_y)^2 + (x+\epsilon_x) \\
&= 2(x^3 + 3x^2\epsilon_x) - 0.2(y^2 + 2y\epsilon_y) + (x+\epsilon_x) \\
&= (2x^3 - 0.2y^2 + x) + (6x^2 + 1)\epsilon_x - 0.4y\epsilon_y
\end{aligned}
$$

Arbitrary orbital targeting 1/2

Compute partial derivative of orbital element

Cartesian representation of an orbit

Create a dual space for each Cartesian component

Compute any orbital parameter (semi-major axis, eccentricity, etc.) in the hyperdual space

Partial derivative of that parameter with respect to each component appears

Many programming libraries compute hyperdual numbers, such as *hyperdual* in Rust, HyperDualNumbers.jl in Julia

$$o_{\mathbb{D}} = \begin{bmatrix} -2436.45 + \epsilon_x + 0 + 0 + 0 + 0 + 0 & \text{km} \\ -2436.45 + 0 + \epsilon_y + 0 + 0 + 0 + 0 & \text{km} \\ 6891.037 + 0 + 0 + \epsilon_z + 0 + 0 + 0 & \text{km} \\ 5.0886 + 0 + 0 + 0 + \epsilon_{v_x} + 0 + 0 & \text{km/s} \\ -5.0886 + 0 + 0 + 0 + 0 + \epsilon_{v_y} + 0 & \text{km/s} \\ 1.0 + 0 + 0 + 0 + 0 + 0 + \epsilon_{v_z} & \text{km/s} \end{bmatrix}$$

$$\xi = -25.3422 \ \text{km}^2/\text{s}^2$$

$$\frac{\partial \xi}{\partial x} = -0.0021 \ \text{km/s}^2 \quad \frac{\partial \xi}{\partial y} = -0.0021 \ \text{km/s}^2 \quad \frac{\partial \xi}{\partial z} = +0.0060 \ \text{km/s}^2$$

$$\frac{\partial \xi}{\partial v_x} = 5.0886 \ \text{km/s}^3 \quad \frac{\partial \xi}{\partial v_y} = -5.0886 \ \text{km/s}^3 \quad \frac{\partial \xi}{\partial v_z} = 1.0000 \ \text{km/s}^3$$

Arbitrary orbital targeting 2/2

Compute partial derivative matrix of objectives to control

Hyperdual numbers can be used to compute the state transition matrix (Phi)

A component wise replacement can be used to compute the partial derivative matrix without finite differencing

Method only valid if the state transition matrix is an acceptable approximation of the dynamics

$$\mathbf{J} = \frac{\partial \mathbf{\Gamma}_f}{\partial \mathbf{U}_i} = \begin{bmatrix} \frac{\partial \Gamma_{f_0}}{\partial U_{i_0}} & \cdots & \frac{\partial \Gamma_{f_0}}{\partial U_{i_n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \Gamma_{fm}}{\partial U_{i_0}} & \cdots & \frac{\partial \Gamma_{fm}}{\partial U_{i_n}} \end{bmatrix}$$

$$\frac{\partial \mathbf{\Gamma}_f}{\partial \mathbf{U}_i} = \frac{\partial \mathbf{\Gamma}_f}{\partial \mathbf{X}_i} \cdot \frac{\partial \mathbf{X}_i}{\partial \mathbf{U}_i} = \frac{\partial \mathbf{\Gamma}_f}{\partial \mathbf{X}_f} \cdot \frac{\partial \mathbf{X}_f}{\partial \mathbf{X}_i} \cdot \frac{\partial \mathbf{X}_i}{\partial \mathbf{U}_i} = \frac{\partial \mathbf{\Gamma}_f}{\partial \mathbf{X}_f} \cdot \mathbf{\Phi}(t_i, t_f) \cdot \frac{\partial \mathbf{X}_i}{\partial \mathbf{U}_i}$$

Masten

Conclusion

Hyperdual numbers simplify the targeting method

Results match GMAT but up to 3 times faster

Limitation of state transition matrix implies that this method is extremely well suited for multiple shooting and finite burn optimization

Implemented in Nyx Space, a free toolkit for high fidelity mission design and orbit determination, validated against GMAT: https://nyxspace.com

Objective	Finite differencing			Hyperdual targeting		
	Δv (m/s)	Iterations	Time (s)	Δv (m/s)	Iterations	Time (s)
SMA = 8100 km	35.504	3	0.326	35.505	3	0.099
SMA = 8100 km and ECC = 0.4	3116.1	8	0.226	3094.0	8	0.104