[a]

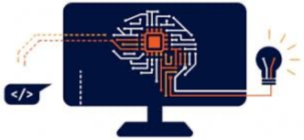
Titan Odyssey

PHASE 3

A project done by Alexander, Jack, Abdullah, Claudia, Ilian and Giacomo

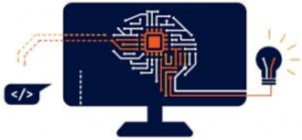
Group 27

Maastricht University



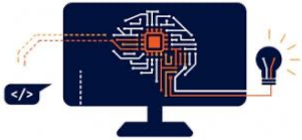
Presentation Layout:

1) Problem statement
2) Assumptions
3) Differential equations solver
4) Numerical integration
5) Trajectory
6) Landing
   a) Wind
   b) Controllers
7) Experiments
8) Discussion
9) Conclusion
10) References



Problem statement

- ❏ How do we accurately reproduce the trajectory of a launched spacecraft from earth to a given target in the solar system?
- ❏ How can we safely and precisely execute a spatial landing taking into consideration the external forces of Titan’s atmosphere?



Assumptions

- Only taking into consideration the dominant bodies of the solar system, such as planets and their corresponding forces
- The forces that act upon our probe are merely, the gravitational forces and the wind and drag force during the landing
- Landing on Titan is visualized and calculated in two dimensions, this applies also for the wind model and the controllers
- Rounding-off and truncation errors are assumed to be small enough to not be relevant while analysing the final results

Differential Equation

- Newton’s universal law of gravity
- Newton’s second law of motion
- State rate-of-change over time dy/dt = [v, a]

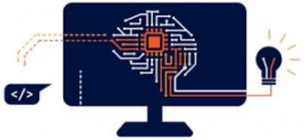
$$F_g = \sum G m_i m_j \frac{x_j - x_i}{||x_i - x_j||}$$

$$F = ma$$

Differential Equations solver

- Used to approximate the next state of solar system and thus the state (position, velocity) of our rocket or probe

- Two different different equation solvers were implemented for this purpose:
  - Euler’s method
  - Fourth order Runge-Kutta method



Euler’s method

- ❏ Next state y_{n+1} based on current state y_n and its respective slope dy/dt
- ❏ Easy to implement
- ❏ Used to bootstrap the starting point of the other solvers
- ❏ Global truncation error of O(h) and local error of O(h^2)

Runge-Kutta’s method

- Classical RK4 implemented
- Leads to an usage of four different derivatives to calculate the next state (backup slide)
- Global error of O(h^4) and local error of O(h^5)
- Time complexity in general was relatively high

Numerical Integration

We implemented two numerical methods for two different tasks:

- Verlet integration:
  Very good at simulating systems with energy conservation

- Newton’s method:
  find a trajectory to Titan from Earth and back

Verlet Integration

- This integration method is used to solve kinematic equations.
- In order to solve these equations the previous position must be known, so the use of the Euler’s method is necessary to compute the first time step
- Involved equations:

$$x_{n+1} = 2x_n - x_{n-1} + at^2$$

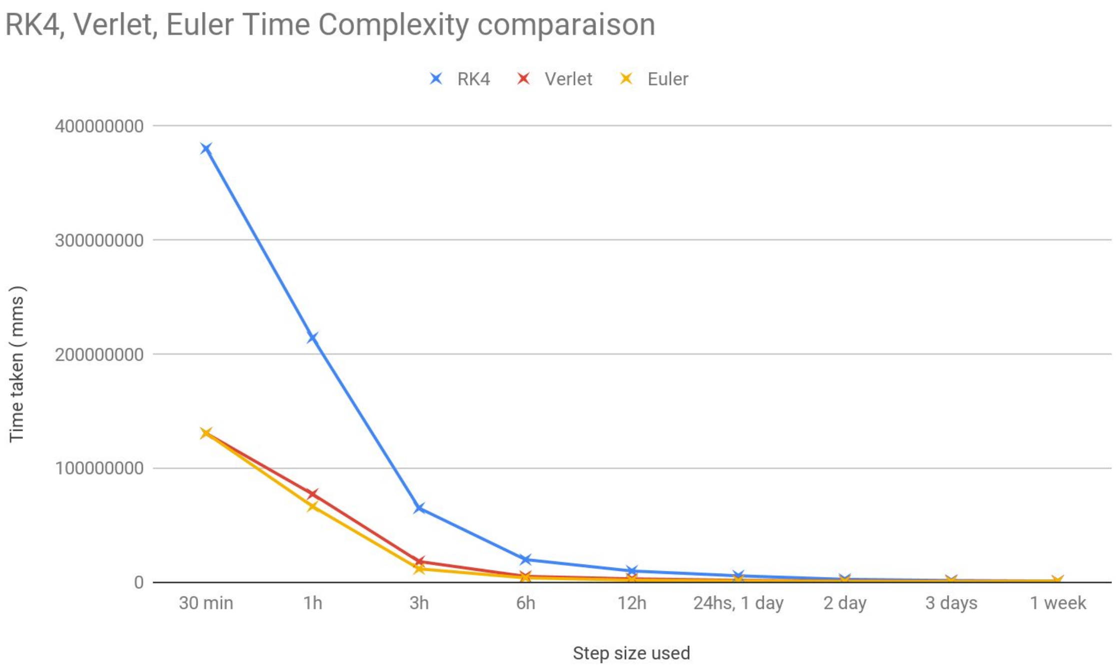
$$v_{n+1} = 2v_n + at$$

Newton’s method

- Finds a velocity such that d(rocket or probe, Target) = 0
- Leads to a complicated distance function : $g(x) = x_T - x_k$
- Uses an iterative step to determine 3-dimensional velocity
- Leads to a multivariable step: $v_{k+1} = v_k - D_{g'(x)} g(x)$

Time Complexity

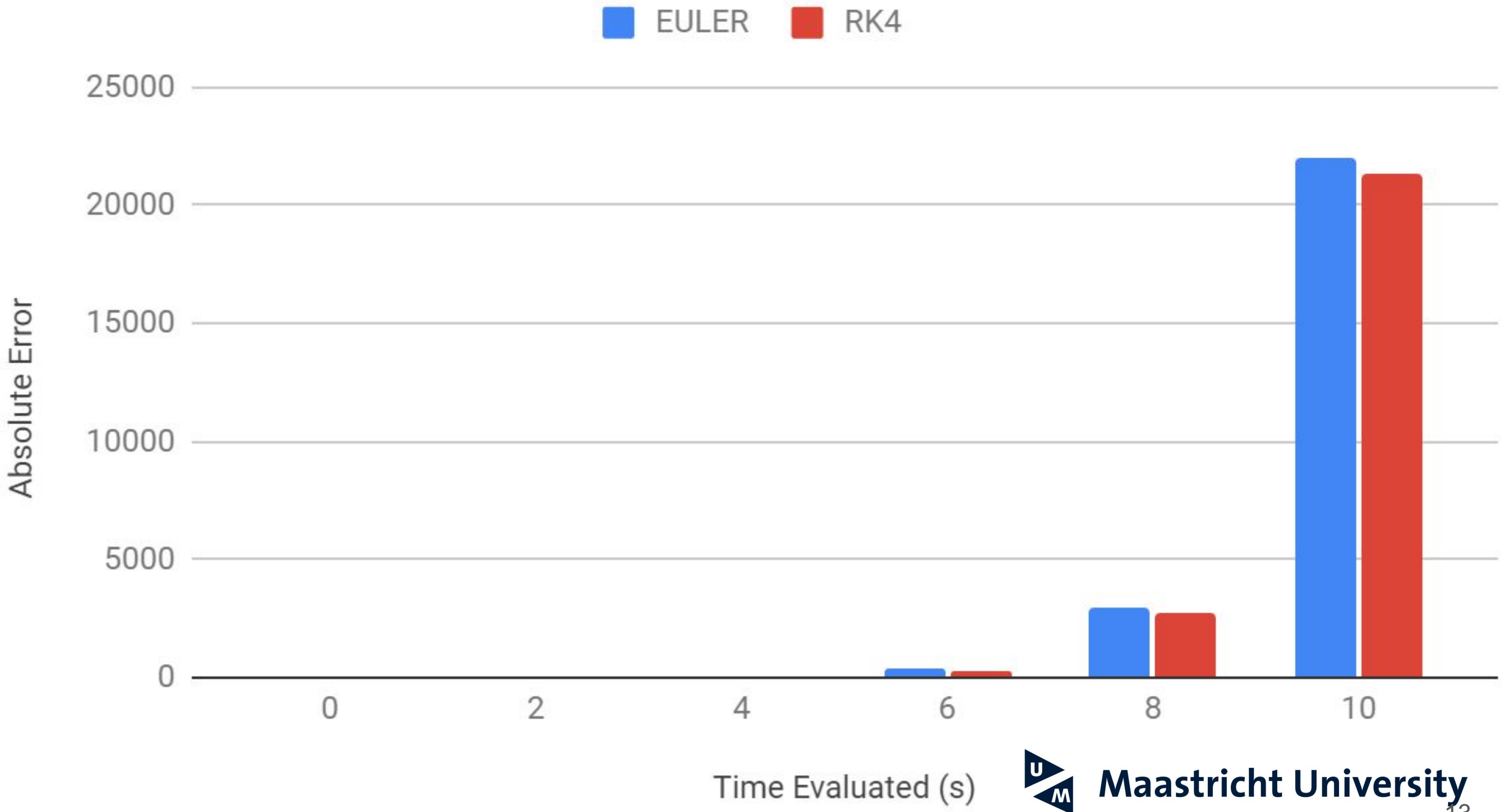
The time taken for each solver to compute one year of simulation was measured.



Group 27

Maastricht University

Absolute Error: Euler and RK4 h= 216000s



Maastricht University

Group 27

Accuracy

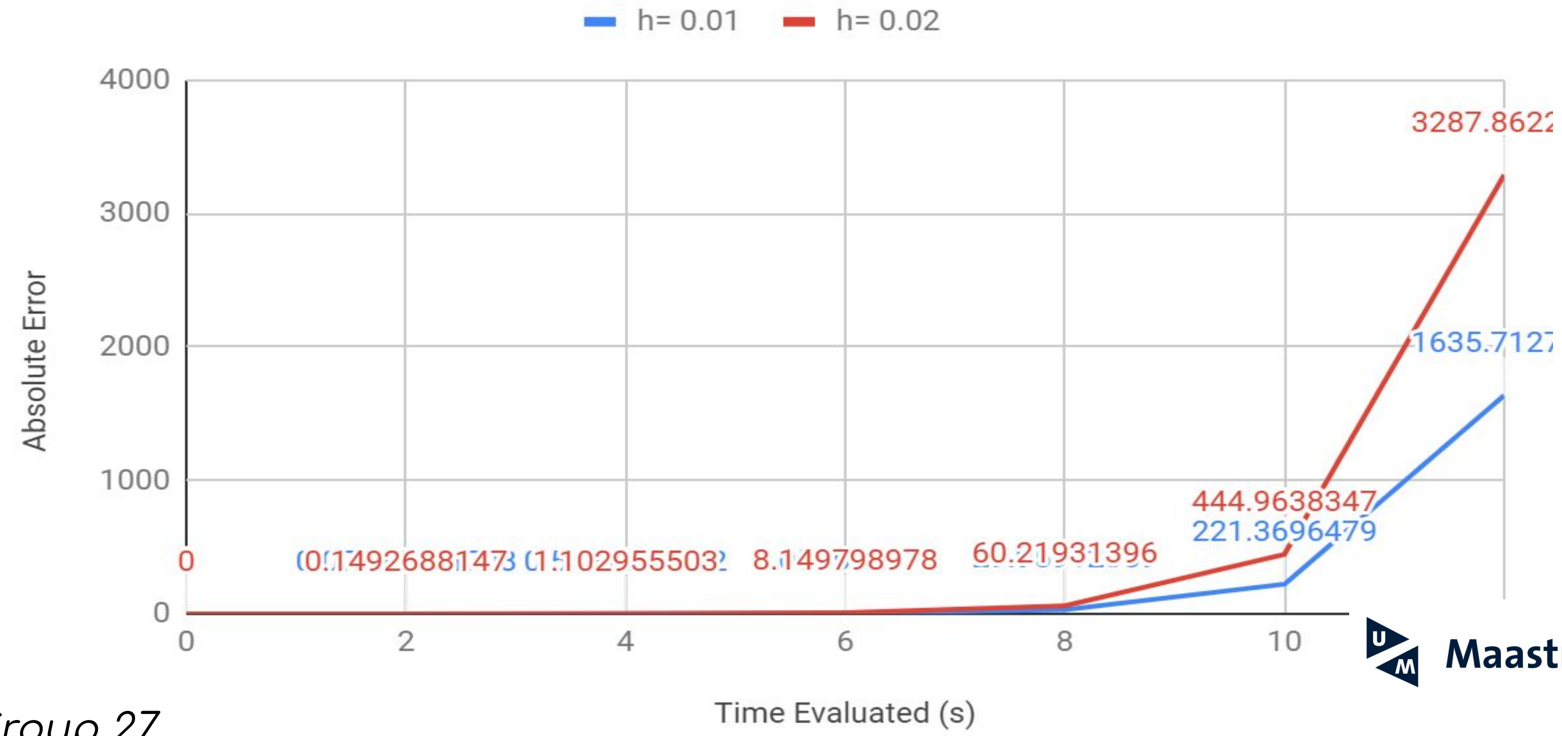
- The exact solution used for testing was the derivative of a given function, namely:

$$y=e^{-t}-y^{2}$$

- Only Euler and RK4 were examined.
- At several points the analytical and experimental were compared, calculating the corresponding absolute error.

$$max_{n\epsilon\{0,\ 1,\ \dots,\ T/h\}}\left|v^{n}-u^{n}\right| \quad \text{as} \quad h -\gg 0$$

Absolute Error: RK4 with 0.01s and 0.02s



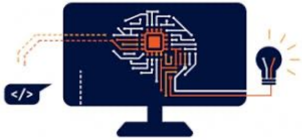
Trajectory

- Large parts of trajectory covered by solution found through Newton’s method

- Depending on how Newton’s method is applied, minor-corrections may be required due to mass loss of fuel

- Easiest solution is using Newton’s method to determine a velocity such that the final distance between Titan and the rocket or probe of the simulation is zero.

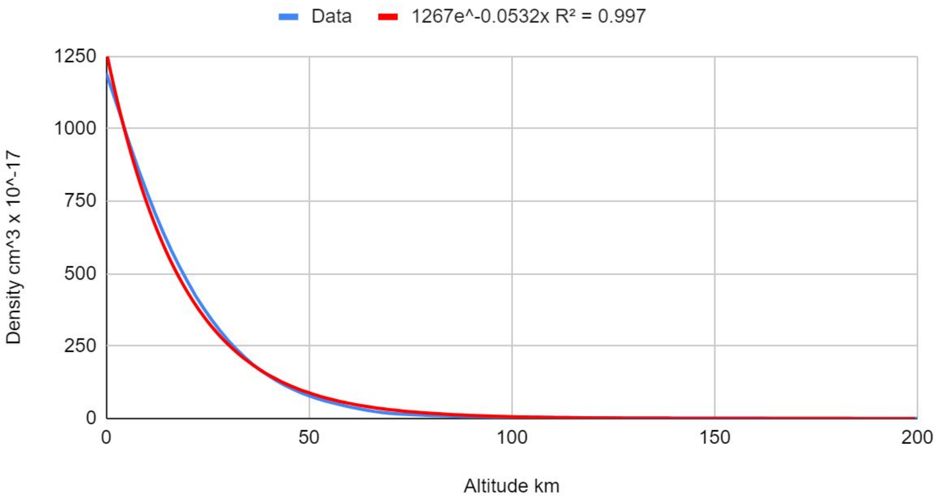
- This leads to only major loss of fuel mass at the beginning of the journey and not during.



Landing Simulation

- **Wind Model**
  - **Air Density**
  - **Wind speed**
- **Feedback Controller**

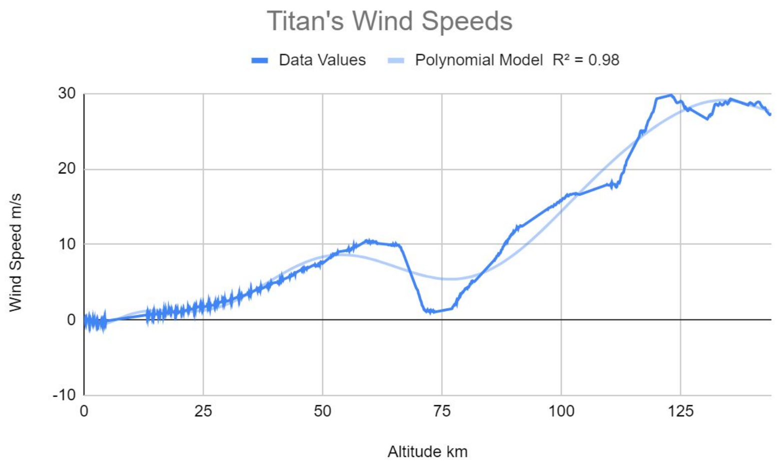
Titan’s Air Density



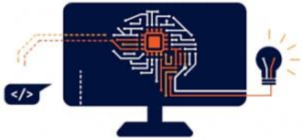
Lindal, G., Wood, G., Hotz, H., Sweetnam, D., Eshleman, V., & Tyler, G. (1983). The atmosphere of Titan: An analysis of the Voyager 1 radio occultation measurements. Icarus, 53(2), 348-363. doi: 10.1016/0019-1035(83)90155-0

Wind Speed & Angle

- Data taken from ESA Cassini Huygens mission.
- Fitted to a degree 10 polynomial function with a R^2 of 0.98.
- A random starting angle was taken for the wind.
- It randomly deviated by ±0-0.5 radians each time step. With a 1/100 chance of changing by **π**.



Lindal, G., Wood, G., Hotz, H., Sweetnam, D., Eshleman, V., & Tyler, G. (1983). The atmosphere of Titan: An analysis of the Voyager 1 radio occultation measurements. Icarus, 53(2), 348-363. doi: 10.1016/0019-1035(83)90155-0



Controllers

A) Closed loop controller

- ❑ Measure the current state of a system
- ❑ Error : Difference between the current state and the desired state

Main forces acting on the Lander :

- ❑ Acceleration due to gravity
- ❑ Wind force acting on the lander.
- ❑ The drag force as the lander travels through Titan’s atmosphere

Controllers

Drag Force & Wind Force:

At each time step the gravitational force needs to be calculated, then the drag force.

$$F_w = \frac{1}{2} \times \rho \times A \times V_w^2 \qquad F_D = C_d \times A \times \frac{\rho V^2}{2}$$

Fd = Drag force N
Fw = Wind Force N
Vw = Wind Speed m/s
Cd = Drag coefficient
A = Reference area m
ϱ = Density of the air kg/m^3
V = Velocity of the lander m/s

Controllers

At each time step the total wind, drag and gravitational, X & Y force is calculated. This used to find the acceleration in the X & Y direction.

$$\frac{F}{m} = a \qquad x_{n+1} = x_n + v_n t + \frac{1}{2} a t \qquad v_{n+1} = v_n + a t$$

Controllers

Gravitational Acceleration:

- Gravitational force is found using Newton's law of gravitation.

$$F_g = \frac{GM_1M_2}{R^2} \qquad F_r = \sqrt{F_x^2 + F_y^2}$$

- Angle of action and X & Y components are found using.

$$\theta = tan^{-1}\left(\frac{Y}{X}\right) \qquad \begin{aligned} F_x &= F_g \times cos\,(\theta) \\ F_y &= F_g \times sin\,(\theta) \end{aligned}$$

Controllers

Open Loop Controller Methodology:

- ❑ The angle for the corrective burn can be found using the equation:

$$\theta = tan^{-1}\left(\frac{F_y}{F_x}\right)$$

- ❑ The probe is then adjusted to the target angle using RCS thruster and the burn is performed.

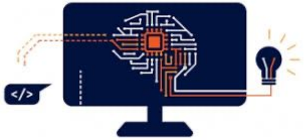
Controllers

Adjusting Inclination:

- Torque Equation: $\tau = Fr$
- Moment of Inertia: $I = \sum mr^2$
- Angular Acceleration: $\alpha = \frac{net\tau}{I}$
- Final Velocity: $v = \sqrt{u^2 + 2\alpha s}$
- Rotation Time $t = \frac{v - u}{a}$

Conclusion (1)

- Euler preferred over Verlet and Runge-Kutta due to the GUI simulation done through a constant step size of a day
- Furthermore, both the probe and rocket trajectory were obtained through Newton’s method in combination with Euler
- A stochastic was applied for the landing, using a closed loop controller. The lander was able to land but not a target location.



Conclusion (2)

- Multiple limitations:
  - Not everything was integrated to GUI
  - Errors visible in GUI due to the large step size of a day
  - Many more

References

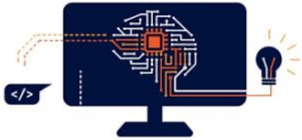
Striepe, S. A., Blanchard, R. C., Kirsch, M. F., & Fowler, W. T. (2007). Huygens Titan Probe Trajectory Reconstruction Using Traditional Methods And The Program To Optimize Simulated Trajectories II (aas 07-226). Advances in Astronautical Sciences, 127, 1853–1880.

Lindal, G., Wood, G., Hotz, H., Sweetnam, D., Eshleman, V., & Tyler, G. (1983). The atmosphere of Titan: An analysis of the Voyager 1 radio occultation measurements. Icarus, 53(2), 348-363. doi: 10.1016/0019-1035(83)90155-0

Scott, D. (2007). Lunar Module Descent Stage Rocket Engine Thrust Chamber. Retrieved 15 June 2021, from https://www.apolloartifacts.com/2007/09/tr-201-bipropel.html

Woolf, P. 11.1: Feedback Control. Retrieved 15 June 2021, from https://eng.libretexts.org/Bookshelves/Industrial_and_Systems_Engineering/Book%3A_Chemical_Process_Dynamics_and_Controls_(Woolf)/11%3A_Control_Architectures/11.01%3A_Feedback_control-_What_is_it%3F_When_useful%3F_When_not%3F_Common_usage

BACKUP SLIDES



RK4 formulae

$$w_{i+1} = w_i + 1/6\,(k_1 + 2k_2 + 2k_3 + k_4)$$

Where;

$h = t_{i+1} - t_i$; Step size

$k_1 = h\,f\,(t_i, w_i)$; Start point, it is the same as Euler's method

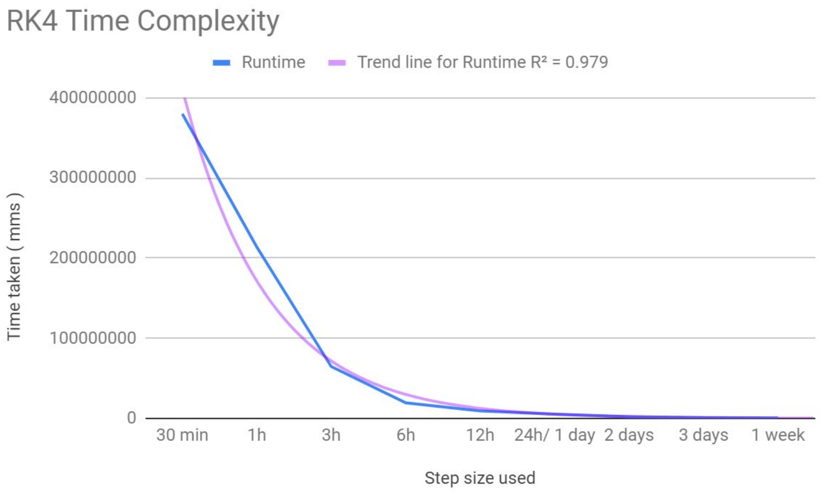
$k_2 = h\,f\,(t_i + h/2, w_i + k_1/2)$; Midpoint derivative

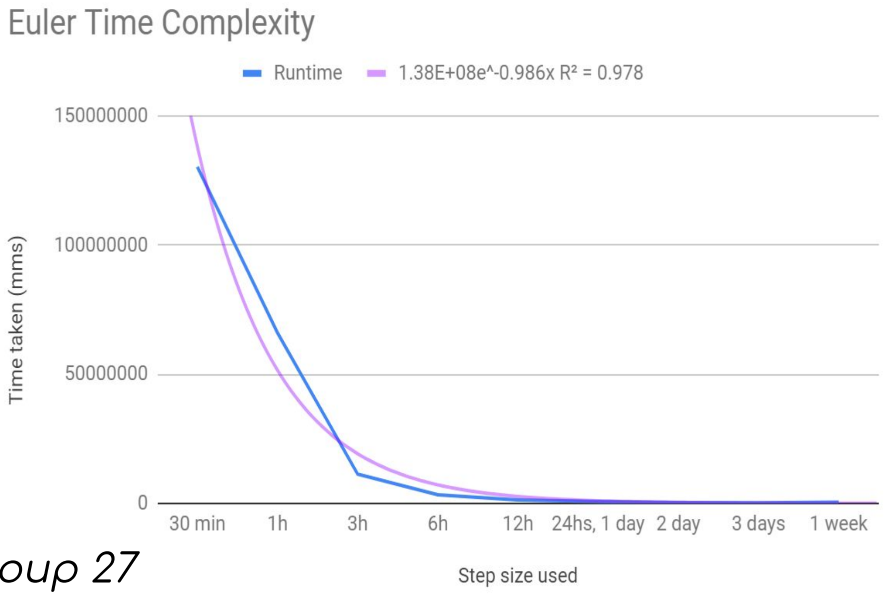
$k_3 = h\,f\,(t_i + h/2, w_i + k_2/2)$; Midpoint derivative

$k_4 = h\,f\,(t_i + h, w_i + k_3)$; Endpoint derivative

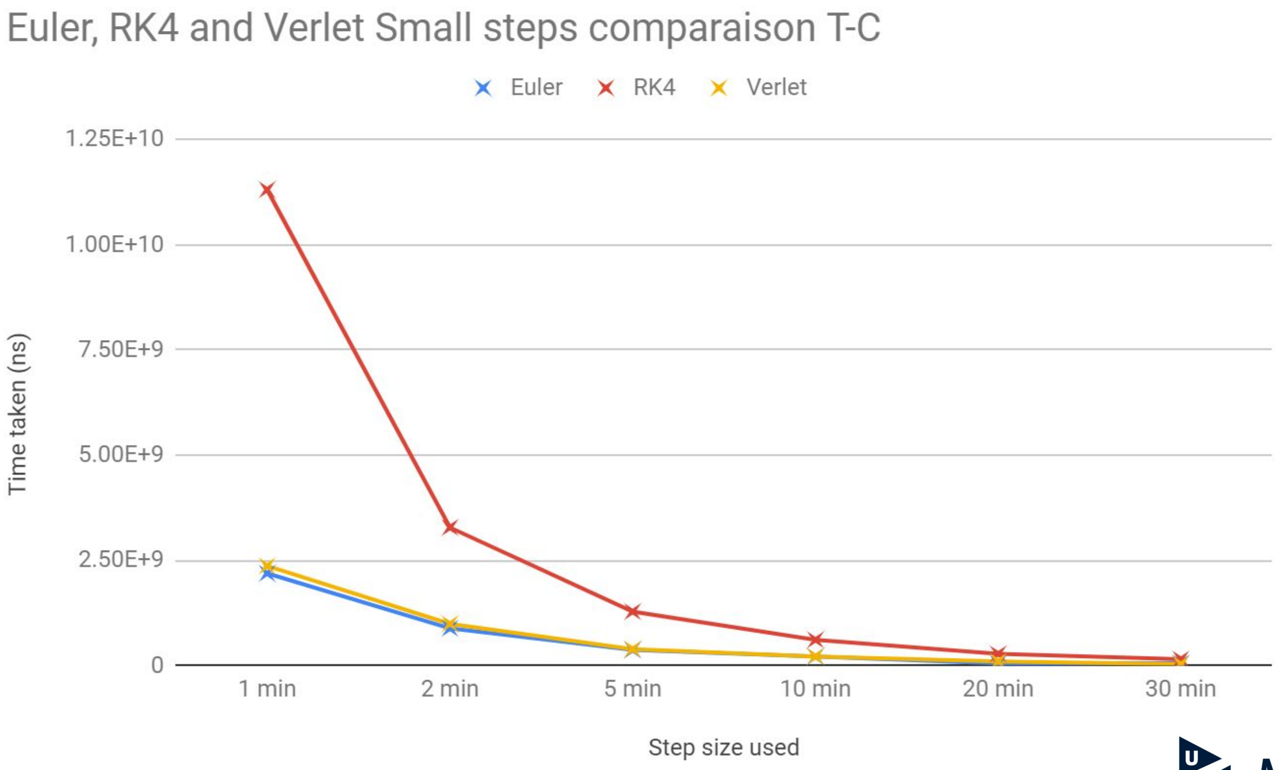
Time complexity graphs

Each solver alone, 1 year simulation using big step sizes

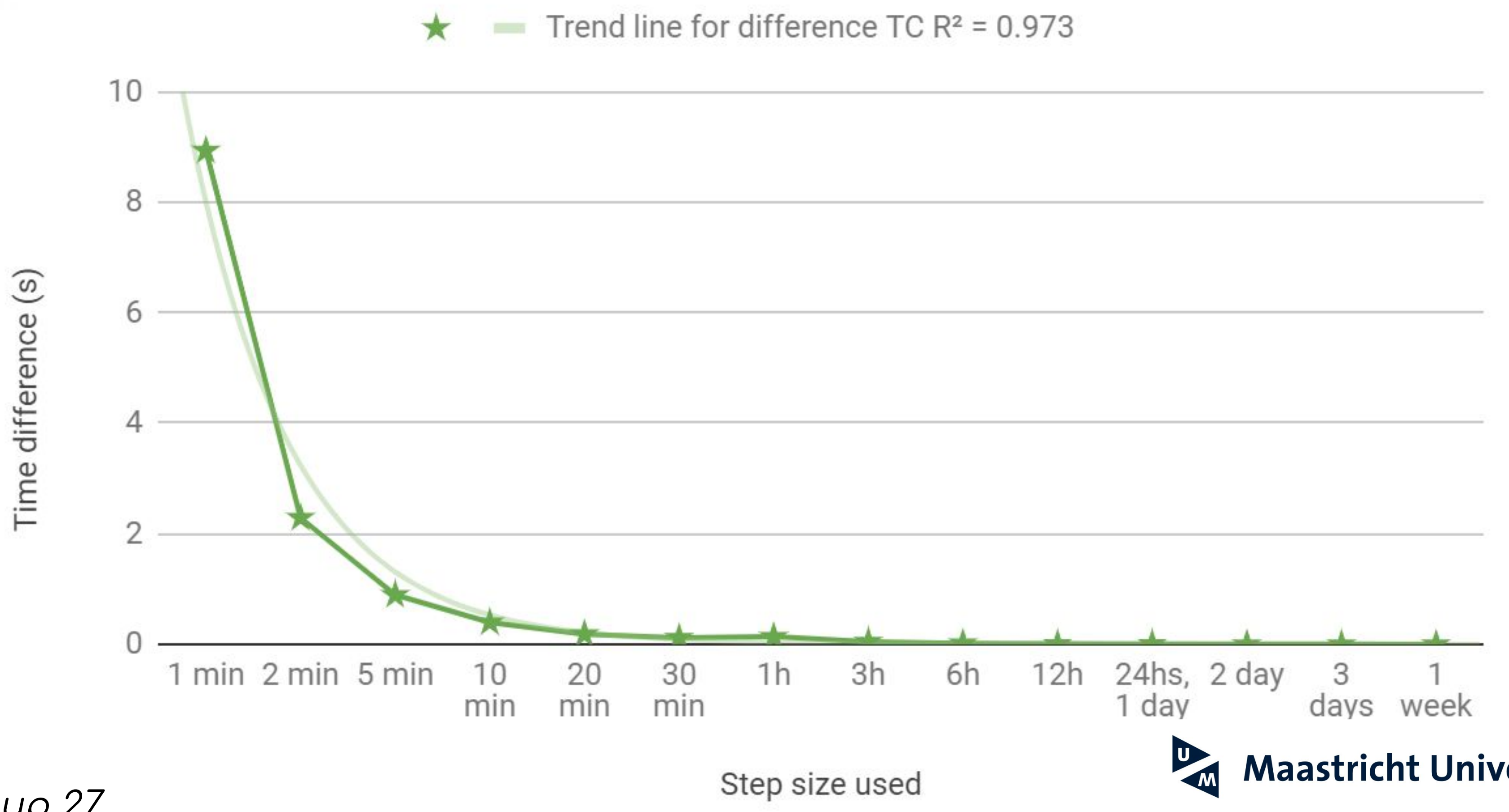




Time Complexity graph (small step sizes)



RK4, Verlet; Difference in computation time



Wind

1)

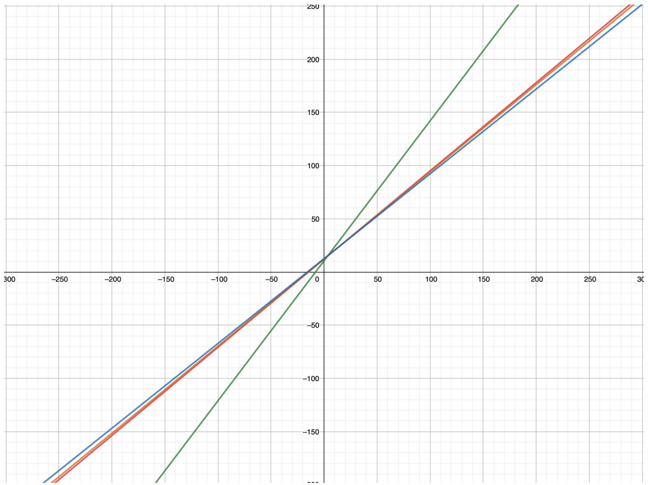
Modeling the wind on Titan :

- ❏ Extracted a data set from the Hugyen wind experiment indicating the relationship between the wind speed and altitude
- ❏ Linear regression
- ❏ Random factor
- ❏ Conversion from wind speed to a force that will act on the Lander’s X axis

Wind

Stochastic Model:

- ❑ Randomized the wind by deviating the previous model by 20%
- ❑ Change of directions every 1/100



Original wind model without any random factors

Figure 1： Linear functions representing the relationship between the wind speed and altitude with the added randomness factors