IMPORTANT thing is to explain WHY? we do things
(context of research agenda)

- Slide 0: Explain title enthusiastically, don't read.
- Slide 1: Verification of Hybrid Systems
  - VERY SIMPLE example
  - "moving in a straight line at a constant speed v_0 to its docking station"
  - we model the behaviour of the spaceship as a hybrid program
  - we "execute" the differential equation after the control of the spaceship
  - we can place this hybrid program inside a PARTIAL correctness specification
- Slide 2: Previous work
  - in previous work we created FIRST verification components inside Isabelle/HOL for verification of hybrid programs
  - the rules for verification condition generation tackle the program structure recursively until we are left with a proof about properties of real numbers
  - because there is lack for proof support in Isabelle, to verify ODEs, users of our components need to follow this procedure
  - procedure is very IMPORTANT
- Slide 3: Affine systems of ODEs
  - going back to our example this procedure requires us to...
  - observe that we can rephrase the dynamics as a matrix
  - in fact, many systems in nature behave as affine systems
- Slide 4: Main contributions
  - NOT ONLY serves verification but it is a formalisation of undergraduate mathematics on its own

  - GENERIC proof of existence and uniqueness
  - MEDIUM sized contribution to the AFP (40 pages of proofs)

- Slide 5: Hybrid Programs: …to understand the need for this procedure, I want to deviate a bit and explain the behaviour of our verification components

- Slide 6: What about ODEs?…however, in verification of hybrid programs, people like to add boundary conditions that restrict the evolution of the system

- Slide 7: Semantics for ODEs
  - guarded orbits are just traditional orbits whose initial segments are guarded by the boundary condition
  - we can derive in Isabelle one rule for verification condition generation per each program construct as in Hoare logic

- Slide 8: Verification of Affine Systems

- Slide 9: Formalising existence and uniqueness
  - Isabelle works like a functional programming language
  - 10 pages of proofs equiv. to 600+ lines of code

- Slide 10: Formalising solutions of affine systems (16 pages of proofs equiv. to 900+ lines of code)

- Slide 11: verification example
  - Very simple verification example, but more advanced examples can be found in the AFP and in the ARCH 2020 competition.

- Slide 12: handling exponentials

- Slide 13: ceritfying a diagonalisation

- Slide 14: conclusions

Hello everyone, I am Jonathan Julian Huerta y Munive and I am finishing my PhD studies at the university of Sheffield. I came to talk to you about a formalisation of affine systems of ordinary differential equations that we did in Isabelle with the purpose of verifying hybrid programs.

To exemplify what I mean by hybrid program verification, I brought a small example. So imagine there is a spaceship moving at a constant speed v_0 in a straight line towards its docking station. The ship needs to stop exactly at a distance d of its centre of mass and the current position of the tip of its nose is at x_0. The ship's computer determines that it needs to decelerate at this rate in order to stop just at the moment where its tip reaches the docking station. To avoid any catastrophic scenarios, we can model the ship's control with an assignment setting its current acceleration to the one determined by the computer. Then we can execute the differential equation of the dynamics of the ship after