ENSTA Bretagne

EASIBEX-MATLAB : a simple tool to start with interval contractors

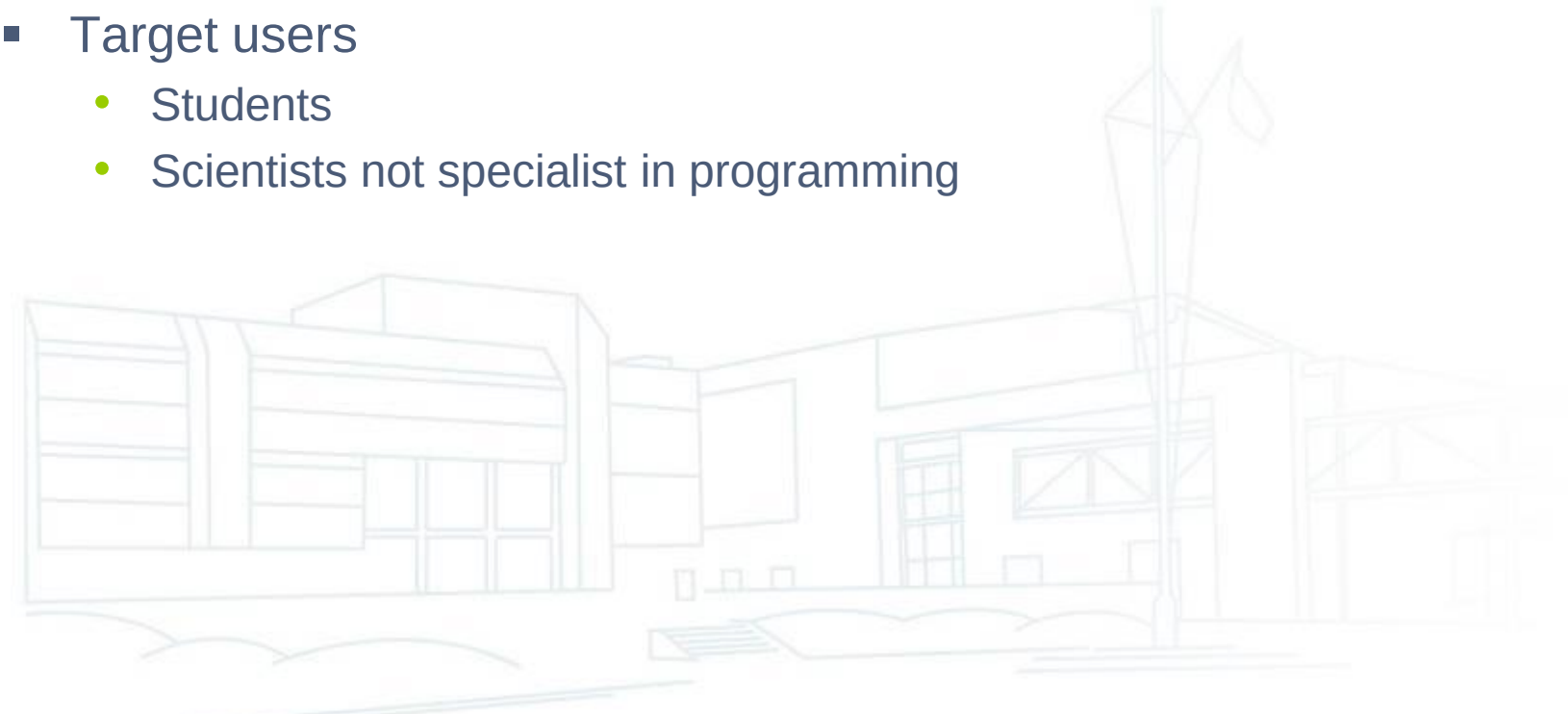
Fabrice LE BARS

Description

Description

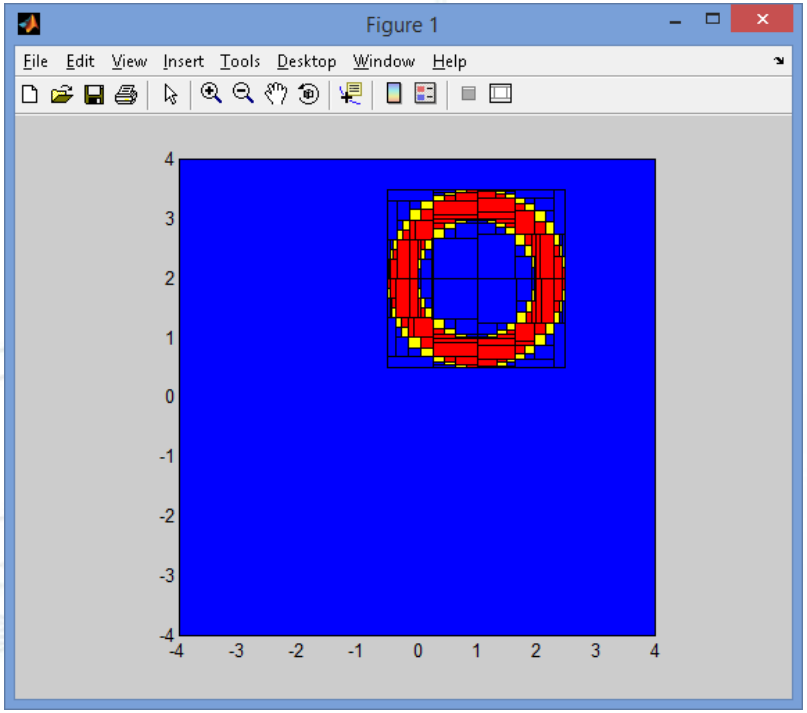
- Main goals
  - Start using interval arithmetic and contractors
  - Quickly prototype and test new algorithms

- Target users
  - Students
  - Scientists not specialist in programming



Description

- Base and philosophy
  - Simple MATLAB layer of IBEX
  - Non object-oriented naming conventions
  - Easily draw results thanks to MATLAB or VIBes



Description

- Limitations
  - To keep it simple, not all IBEX features are available
  - No guaranties w.r.t. rounding…
  - Simplified function names to avoid object-oriented paradigms

- Other options
  - EASIBEX-CPP is available if C++ is needed : see www.ensta-bretagne.fr/jaulin/easibex.html
  - Once you are comfortable with intervals, contractors, C++ and object-oriented programming, use directly IBEX

Demo

Demo

- Download and test
  - https://github.com/ENSTABretagneRobotics/EASIBEX-MATLAB/

Demo

- To define an EASIBEX-MATLAB interval
  - x=[-2,2]
  - x(1,2) would be 2
- An empty interval would be
  - x=[NaN,NaN]
- Infinity
  - x=[-Inf,Inf]

Demo

- To define a box
  - x=[[-2,2];[2,4];[-4,1]]
  - x(2,:) would be [2,4]
- 2 intervals can be added using
  - Z=i_Add([0,2],[-1,2])
- And 2 boxes
  - Z=i_Add([[0,1];[0,10];[0,10]],[[-1,0];[2,5];[-1,0]])

Demo

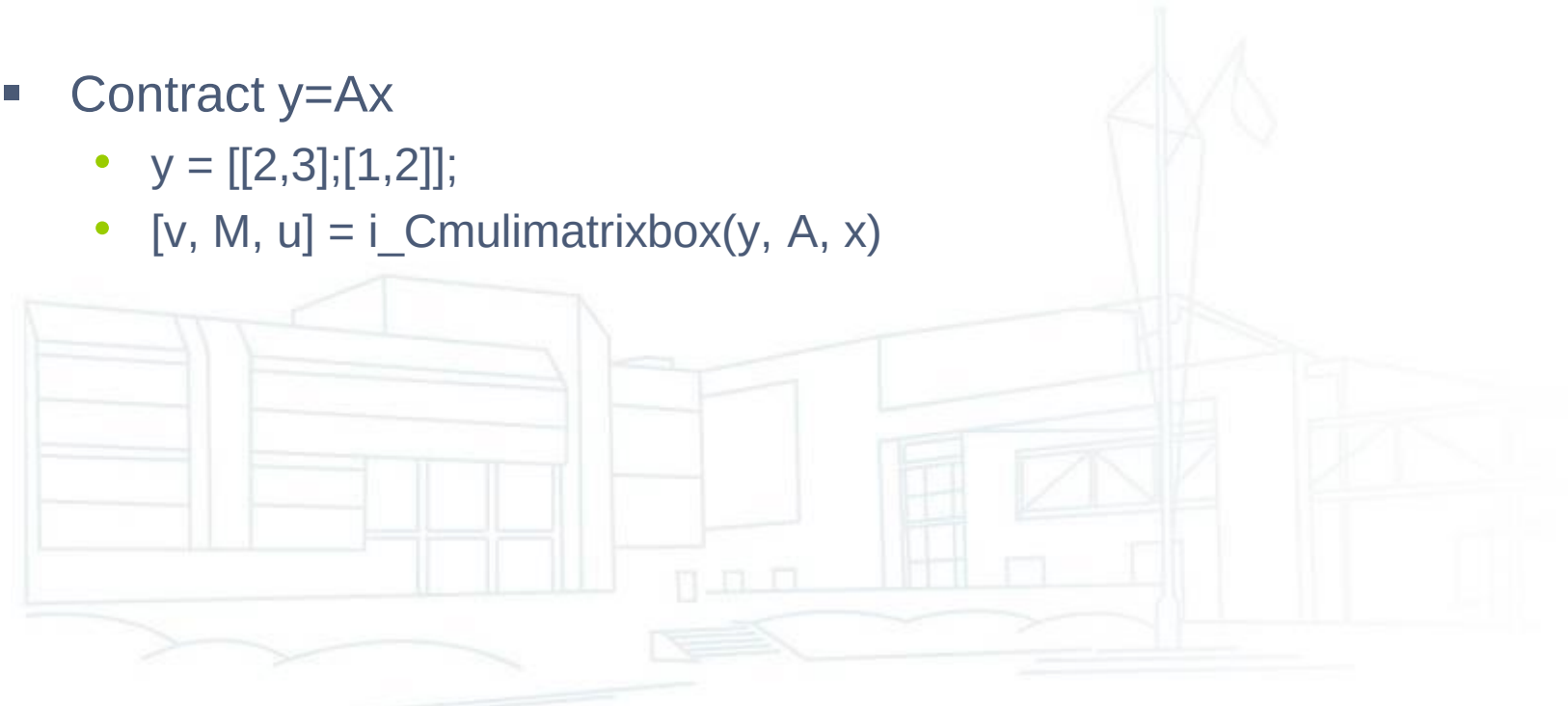
- To contract 3 intervals Z=[-10,1], X=[0,2], Y=[-1,2] knowing the constraint Z=X+Y :
  - [Z,X,Y]=i_Cadd([-10,1],[0,2],[-1,2])

- To contract the vector x=[-10,10]*[0,10]*[-10,0] w.r.t. the q-relaxed intersection of the 4 vectors [-2,2]*[2,4]*[-4,1], [-1,5]*[-5,8]*[-7,2], [-1,1]*[0,2]*[1,2], [-2,2]*[2,8]*[-1,2], with q=2 :
  - x = [[-10,10];[0,10];[-10,0]]
  - y_j = {[[-2,2];[2,4];[-4,1]];[[-1,5];[-5,8];[-7,2]];[[-1,1];[0,2];[1,2]];[[-2,2];[2,8];[-1,2]]}
  - x = i_C_q_in(x, 2, y_j)

Demo

- Interval matrix imatrix
  - x = [[1,2];[0,1]];
  - A = i_imatrix(2, 2, [[[0,1];[0,0]];[[0,0];[0,1]]]);
  - i_Mulimatrixbox(A, x)

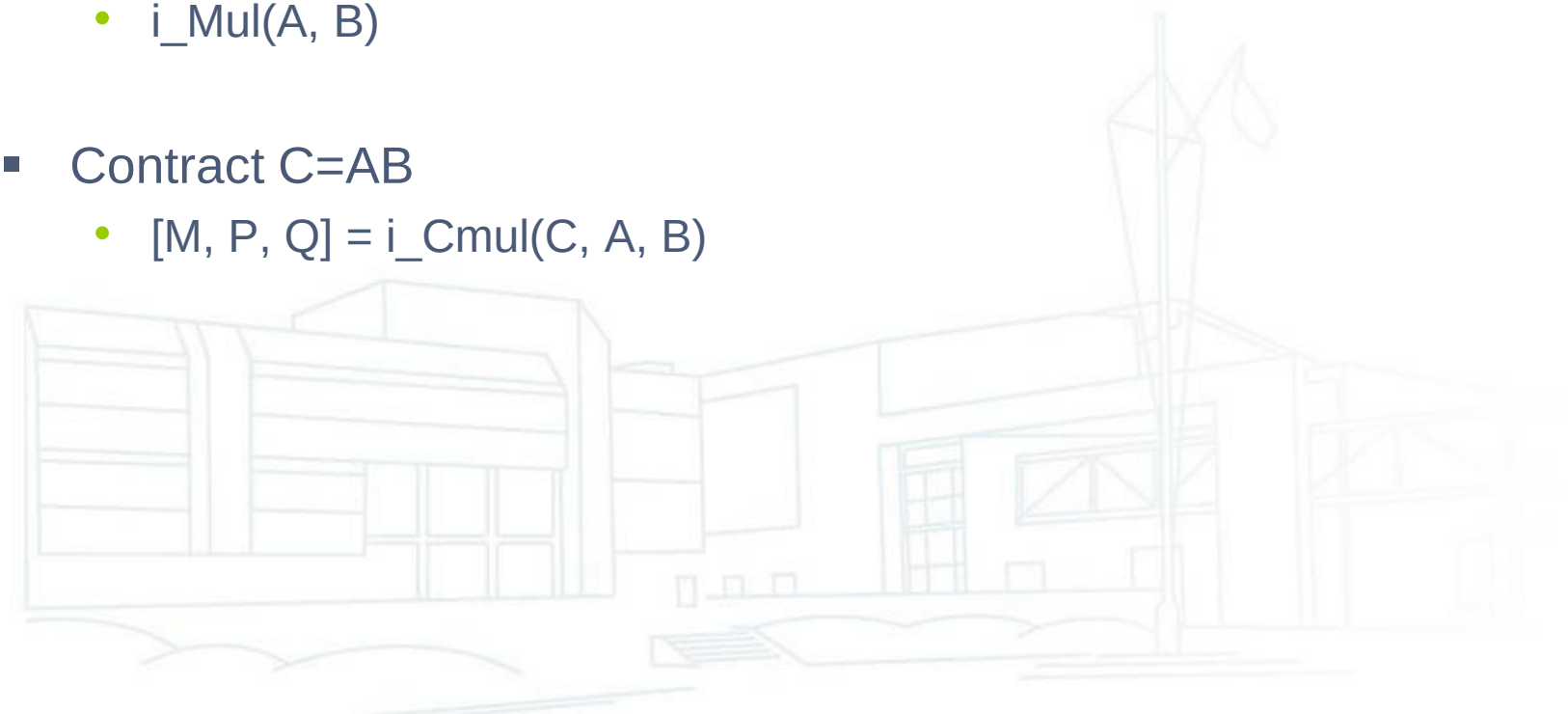
- Contract y=Ax
  - y = [[2,3];[1,2]];
  - [v, M, u] = i_Cmulimatrixbox(y, A, x)



Demo

- C=AB
  - A = i_imatrix(2, 2, [[[0,1];[0,0]];[[0,0];[0,1]]]);
  - B = i_imatrix(2, 2, [[[1,2];[0,0]];[[0,0];[1,2]]]);
  - C = i_imatrix(2, 2, [[[2,3];[0,0]];[[0,0];[2,3]]]);
  - i_Mul(A, B)

- Contract C=AB
  - [M, P, Q] = i_Cmul(C, A, B)



Questions?