Simulation Verification Validation

We start At 08:45

Make notes by hand

Turn your camera off

Turn your mic off

Ask questions via chat

Session is being recorded

Introduction to V&V and software verification

Wouter van der Wal

Faculty of Aerospace Engineering, Astrodynamics and Space Missions

3:03:49

See lecture notes on Brightspace

TUDelft

Simulation, verification,

Wouter van der Wal

8-2-2021

Staff

Wouter van der Wal
(Course coordinator)

w.vanderwal@tudelft.nl

Julien van Campen
(Coordinator, structural analysis)

Alexander in’t Veld
(Flight Dynamics)

Hans Mulder
(Flight Test)

+ 12 TA’s, PhD students and staff from C&S and ASM

Imagine you work for a small airplane manufacturer and you have developed an autopilot.

Would you fly in the airplane and let the autopilot land the airplane?

Would you fly in the airplane and let the autopilot land the airplane when your colleague programmed the autopilot?

Airbus.com

What could go wrong?

ESA.com

ExoMars, October 2016

“an unexplained saturation of its inertial measurement unit, which delivered bad data to the lander’s computer and forced a premature release of its parachute.”

Spacenews.com

Contents for today

- 8:45-9:30

  Introduction to simulation, verification and validation

  Software verification

  (Wouter van der Wal)

- 9:45-10:30

  Structural Analysis Theory and Structural Analysis Assignment

  (Julien van Campen)

- 10:45-11:30

  Case study + Rules & Guidelines

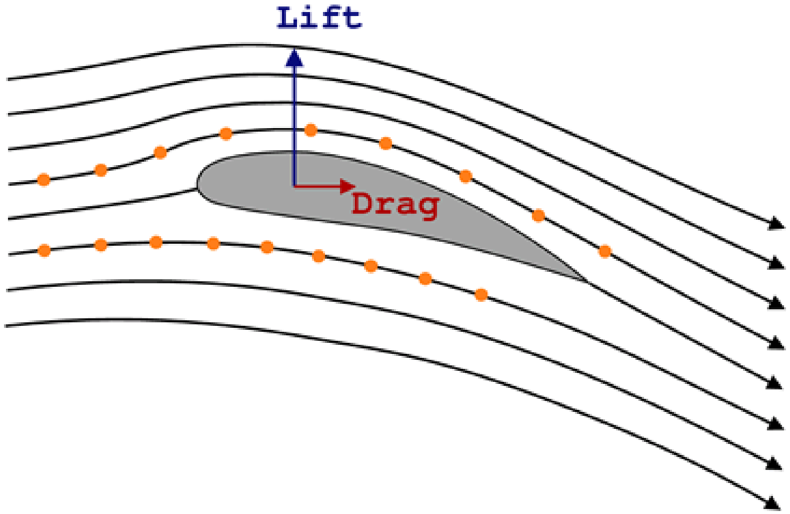
  (Wouter van der Wal)

What is simulation?

“develop a (mathematical / numerical) model of the physical model of a physical problem and generate results”

BSc program

- Statics
- Dynamics
- Thermodynamics
- Waves and electromagnetism
- Aerodynamics
- Structural analysis and design
- Vibrations
- Flight and orbital mechanics



Go to: www.menti.com and use the code 781049

Consider the generation of lift by a wing. What statement is closest to the truth?

a) Lift is generated because air particles on top of the wing have a longer travel path so they have a higher speed and lower pressure (Bernouilli)

b) Lift is generated because the wing deflects particles downwards, which generates an opposite force on the wing

c) There is no universally accepted explanation for the generation of lift by a wing

d) None of the above

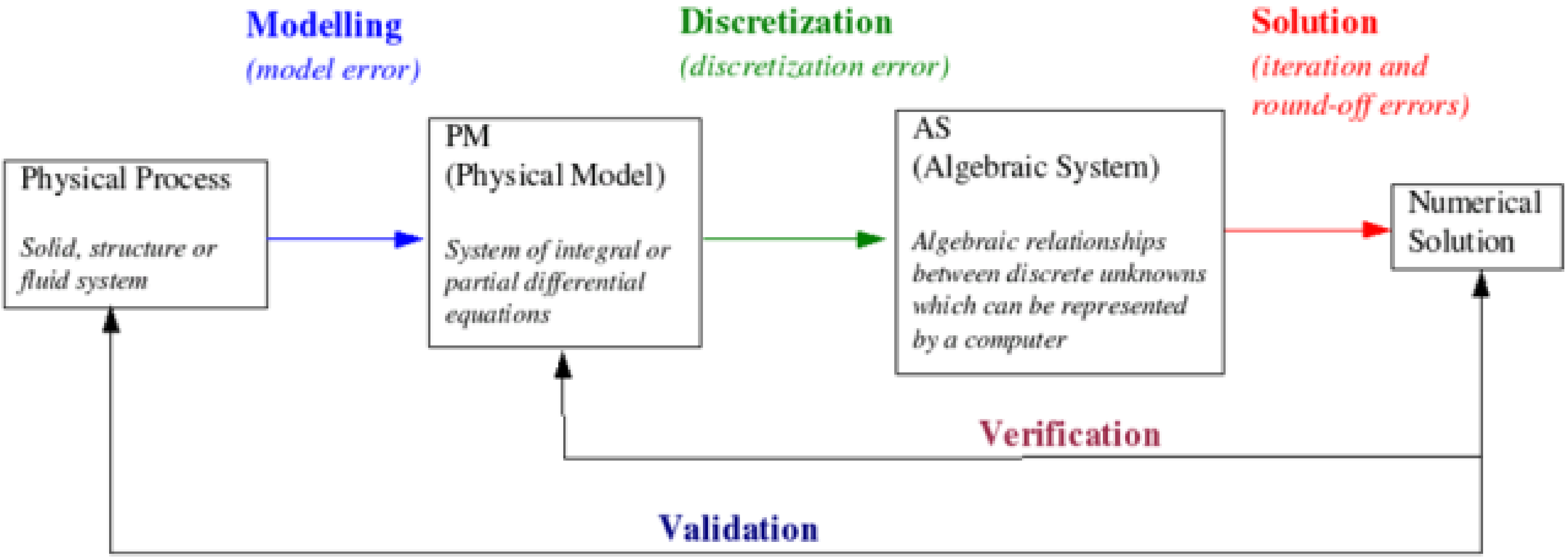
https://www.scientificamerican.com/article/no-one-can-explain-why-planes-stay-in-the-air/

Simulation

Requirements

1. What are the equations?
2. Which phenomena should be considered?
3. What assumptions are made?
4. What is the effect of assumptions on the results?
5. What should the software do?
6. What is the effect of discretization? (e.g. taking numerical derivatives, numerical integration)
7. Did I make an error in calculation?
8. Did I make an error in my computer program?
9. What is the effect of my assumptions? (e.g. linear behavior)
10. How reliable is my input data?
11. What is the accuracy of the model?
12. Is the model validated?

Framework


Modelling
(model error)
Discretization
(discretization error)
Solution
(iteration and round-off errors)
Physical Process
Solid, structure or fluid system
PM
(Physical Model)
System of integral or partial differential equations
AS
(Algebraic System)
Algebraic relationships between discrete unknowns which can be represented by a computer
Numerical Solution
Verification
Validation

BSc program

- **Programming (AE1205)** ← **Simulation**
- Statics
- Dynamics
- Thermodynamics
- **Probability and Statistics (WI2180-LR-II)** ← **Validation**
- Waves and electromagnetism
- Aerodynamics
- Structural analysis and design ← **Simulation**
- Vibrations
- **Experimental research and data analysis (AE2223-II)** ← **Validation**
- Flight Dynamics ← **Simulation**
- **Computational modelling (AE2220-II)**← **Verification**

What is Verification?

To determine if a simulation model accurately represents the chosen physical model

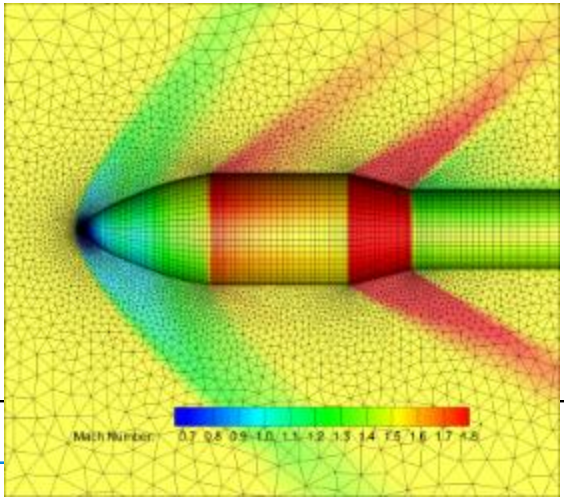
> “Verification proves that a realized product for any system model within the system structure conforms to the build-to requirements (for software elements) or realize-to specifications and design descriptive documents (for hardware elements, manual procedures…).”

NASA system engineering handbook

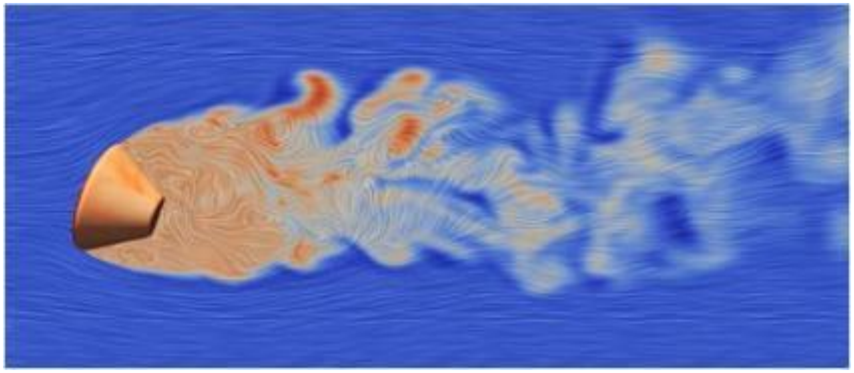
DID YOU ADD THE NEW FEATURE YET?
NO, I HAD TO FIX A CRITICAL BUG IN THE PLATFORM FIRST.
I HAVE NO WAY TO VERIFY THAT CLAIM.
THAT'S WHY IT'S A GOOD ONE.
Dilbert.com @ScottAdamsSays
4-15-17 © 2017 Scott Adams, Inc./Dist. by Andrews McMeel

Verification

1. What are the equations?
2. Which phenomena should be considered?
3. What assumptions are made?
4. What is the expected effect of assumptions on the results?
5. What should the software do?
6. What is the effect of discretization? (e.g. taking numerical derivatives, numerical integration)
7. Did I make an error in calculation?
8. Did I make an error in my computer program?
9. What is the effect of my assumptions? (e.g. linear behavior)
10. How reliable is my input data?
11. What is the accuracy of the model?
12. Is the model validated?



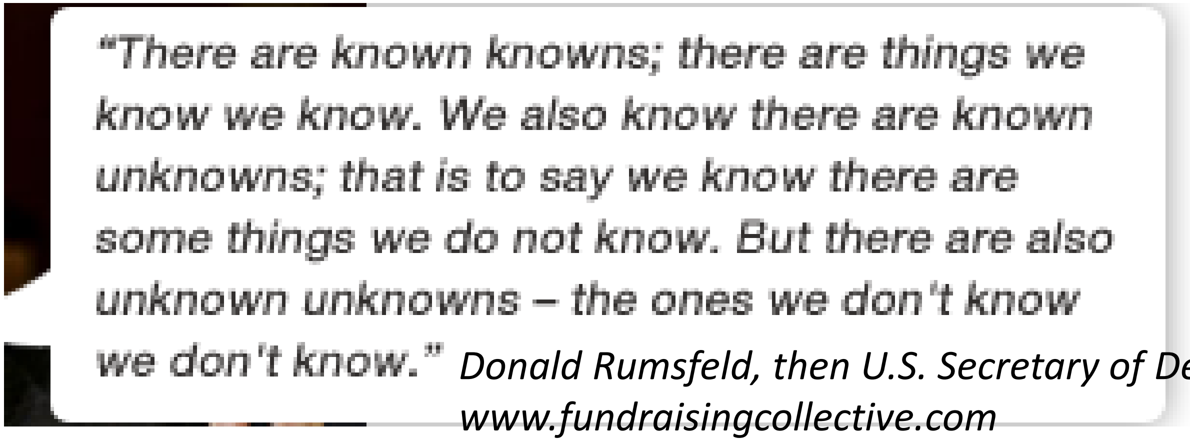
Errors and uncertainties



Computational modelling (AE2220-II):

Uncertainties: deficiencies due to lack of information about the system

Errors: deficiencies due to approximations used



"There are known knowns; there are things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know. But there are also unknown unknowns – the ones we don't know we don't know."

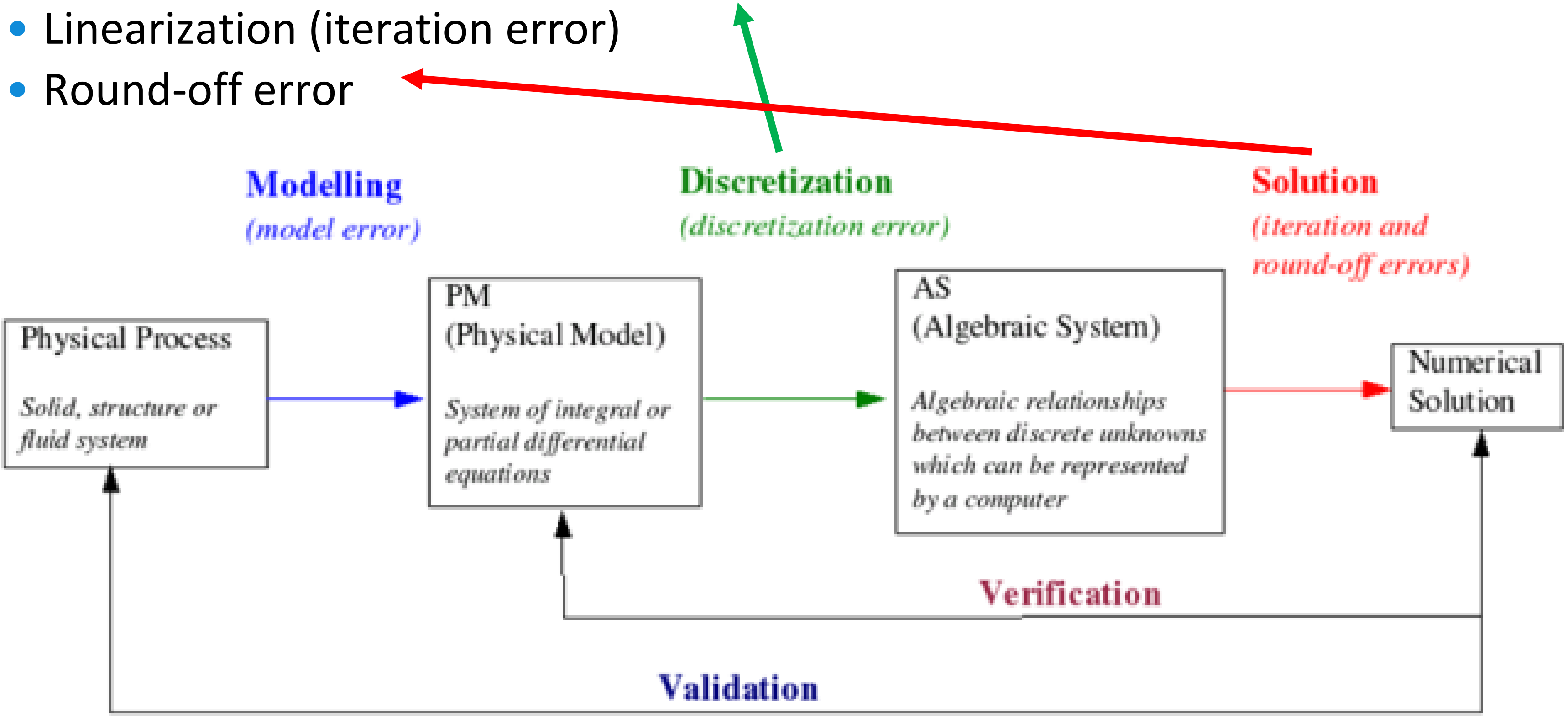
Donald Rumsfeld, then U.S. Secretary of Defense, www.fundraisingcollective.com

+ bugs, typos, …

Verification of numerical model

→ See Computational Modelling AE2220-II

- Continuous → discrete (discretization error)
- Linearization (iteration error)
- Round-off error



Modelling
(model error)

Discretization
(discretization error)

Solution
(iteration and round-off errors)

Physical Process
Solid, structure or fluid system

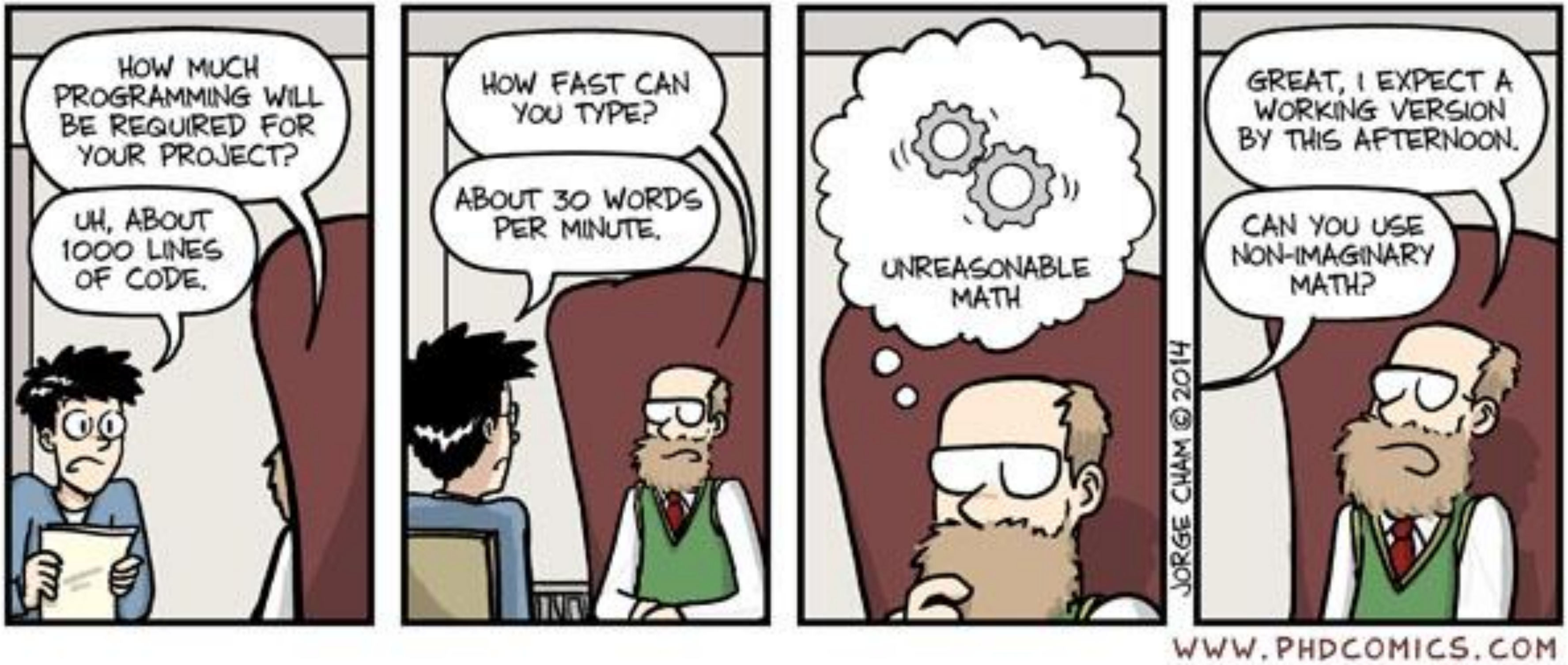
PM
(Physical Model)
System of integral or partial differential equations

AS
(Algebraic System)
Algebraic relationships between discrete unknowns which can be represented by a computer

Numerical Solution

Verification

Validation


HOW MUCH PROGRAMMING WILL BE REQUIRED FOR YOUR PROJECT?
UH, ABOUT 1000 LINES OF CODE.
HOW FAST CAN YOU TYPE?
ABOUT 30 WORDS PER MINUTE.
UNREASONABLE MATH
GREAT, I EXPECT A WORKING VERSION BY THIS AFTERNOON.
CAN YOU USE NON-IMAGINARY MATH?
JORGE CHAM © 2014
WWW.PHDCOMICS.COM

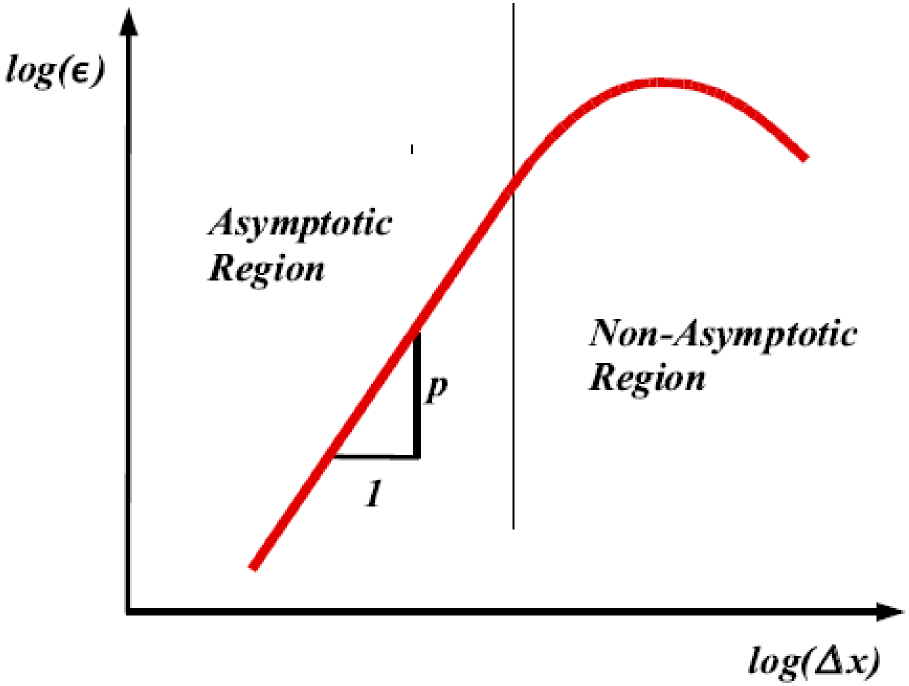
Industry Average: ″about 15 – 50 errors per 1000 lines of delivered code.″

Space Shuttle: “0 errors in 500,000 lines of code″

Source: Code Complete, Steve McConnell

Error estimation (see AE2220-II)

Order of accuracy test



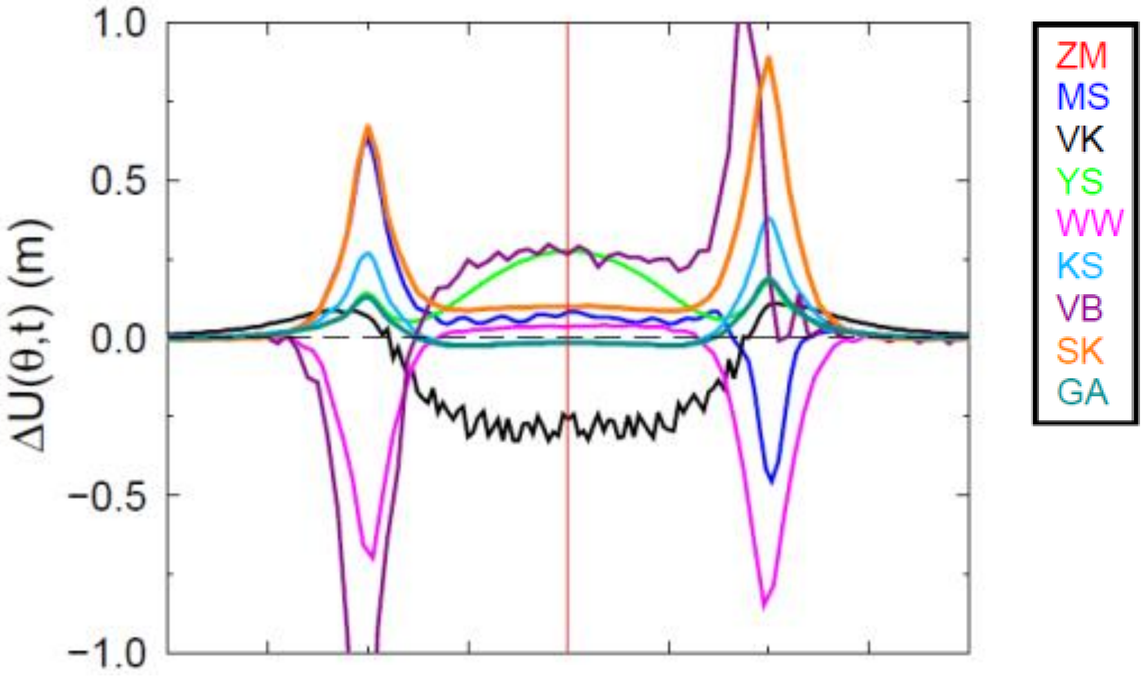
You can quantify discretization error if in asymptotic region

If errors are uncorrelated:

$$\sigma_{total} = \sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2}$$

Error in computer program?

Check against an independent model



Martinec, Klemann, van der Wal, et al. (2018)

Verification

1. What are the equations?
2. Which phenomena should be considered?
3. What assumptions are made?
4. What is the effect of assumptions on the results?
5. What should the software do?
6. What is the effect of discretization? (e.g. taking numerical derivatives, numerical integration)
7. What is the error in calculation? (round-off, iteration)
8. Where are the errors in my computer program?
9. What is the effect of my assumptions? (e.g. linear behavior)
10. How reliable is my input data?
11. What is the accuracy of the model prediction?
12. To what extent is the model validated?

See chapter 3 lecture notes

Validation

1. What are the equations?
2. Which phenomena should be considered?
3. What assumptions are made?
4. What is the effect of assumptions on the results?
5. What should the software do?
6. What is the effect of discretization? (e.g. taking numerical derivatives, numerical integration)
7. What is the error in calculation? (round-off, iteration)
8. Where are the errors in my computer program?
9. What is the effect of my assumptions? (e.g. linear behavior)
10. How reliable is my input data?
11. What is the accuracy of the model prediction?
12. To what extent is the model validated?

See chapter 4 lecture notes

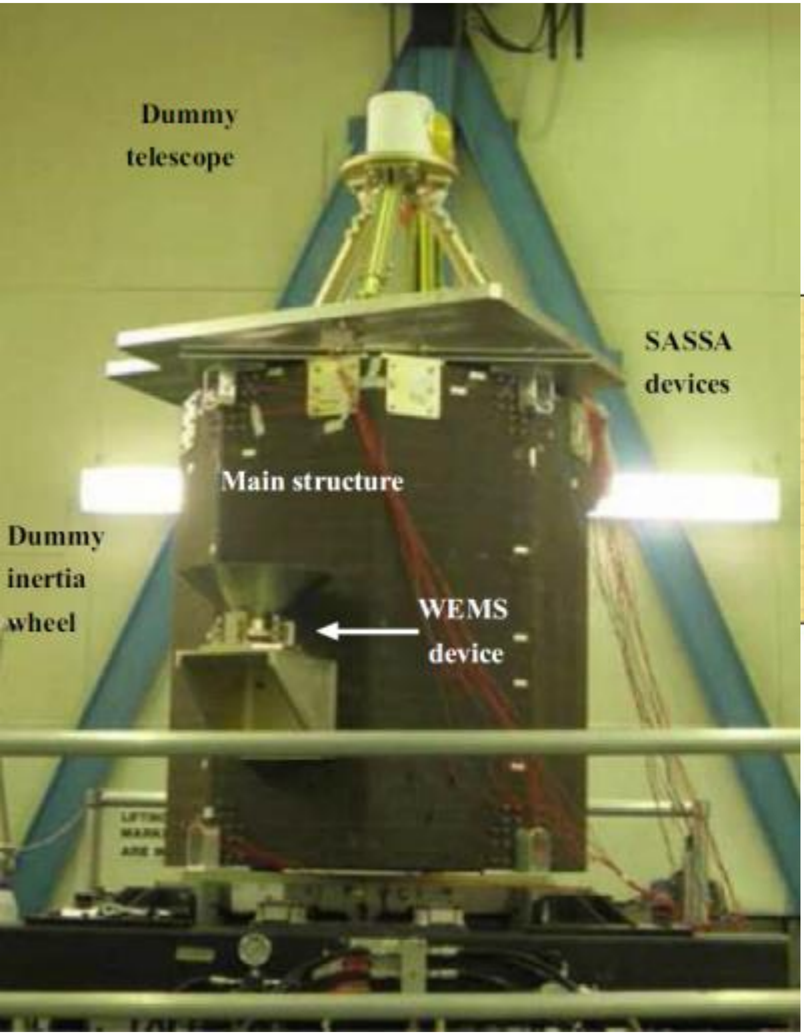
Validation

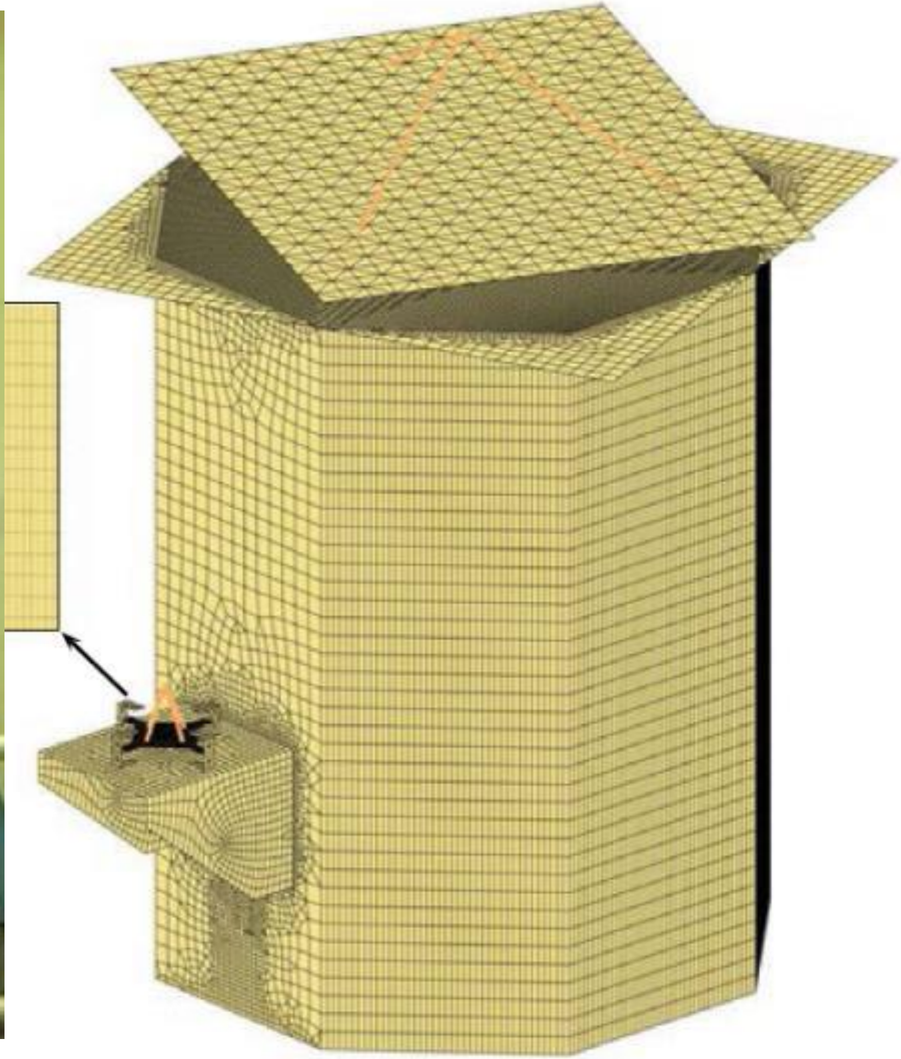
“Determine if the simulation results accurately represent the physical problem”

Confrontation with reality

Validation

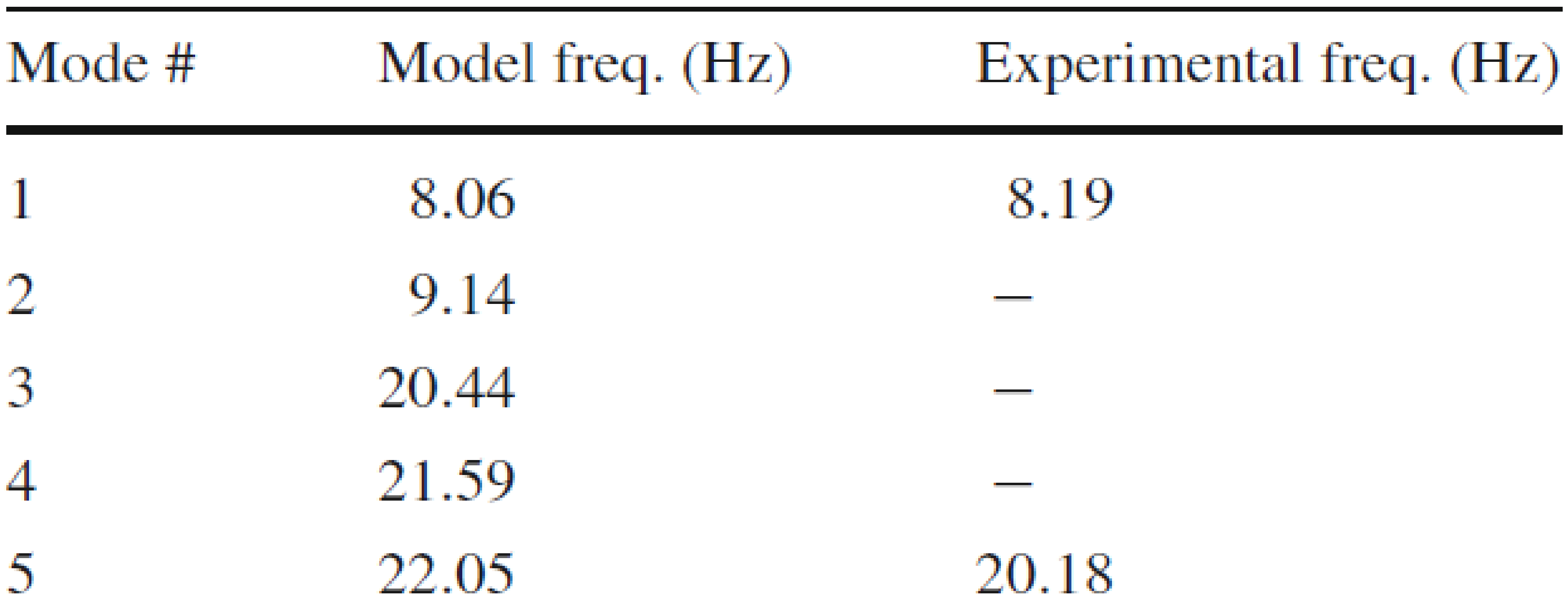
SmallSat structure, EADS-Astrium





Renson et al, (Nonlinear Dynamics, 2015)

Validation

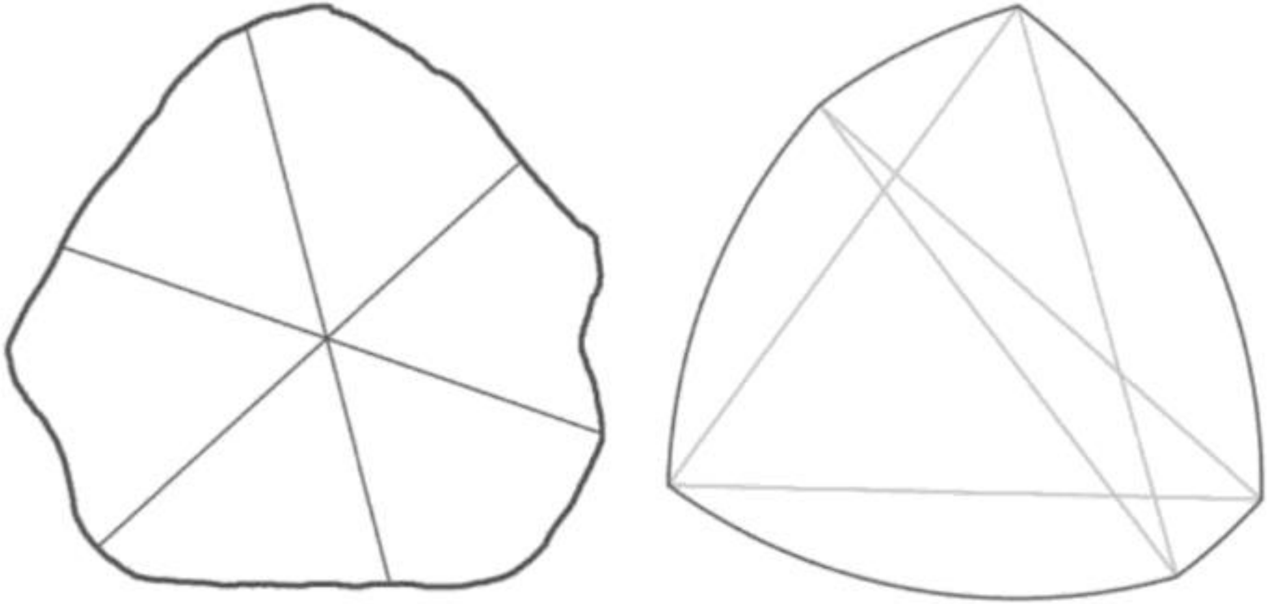


Mode #	Model freq. (Hz)	Experimental freq. (Hz)
1	8.06	8.19
2	9.14	–
3	20.44	–
4	21.59	–
5	22.05	20.18

Renson et al., (Nonlinear Dynamics, 2015)

Example insufficient validation

Is the diameter of solid rocket boosters still round?



Source: Humble Pi, by Matt Parker

“Finding #5: significant out-of-round conditions existed between the two segments”

How can you find errors?

Ask the right questions!

1. Did I use the correct theory?
2. Are all relevant phenomena taken into consideration?
3. Did I make an error in calculation?
4. Did I make an error in my computer program?
5. What is the effect of discretization? (e.g. mesh size, taking numerical derivatives, numerical integration)
6. What is the effect of my assumptions? (e.g. linear behavior)
7. How reliable is my input data?
8. What is the accuracy of the model prediction?
9. To what extent is the model validated?

How can you find errors?

Ask the right questions!

1. Did I use the correct theory?
2. Are all relevant phenomena taken into consideration?
3. Did I make an error in calculation?
4. Did I make an error in my computer program?
5. What is the effect of discretization? (e.g. mesh size, taking numerical derivatives, numerical integration)
6. What is the effect of my assumptions? (e.g. linear behavior)
7. How reliable is my input data?
8. What is the accuracy of the model prediction?
9. To what extent is the model validated?

Evaluation – Hypothesis testing

see WI2180-LR-II

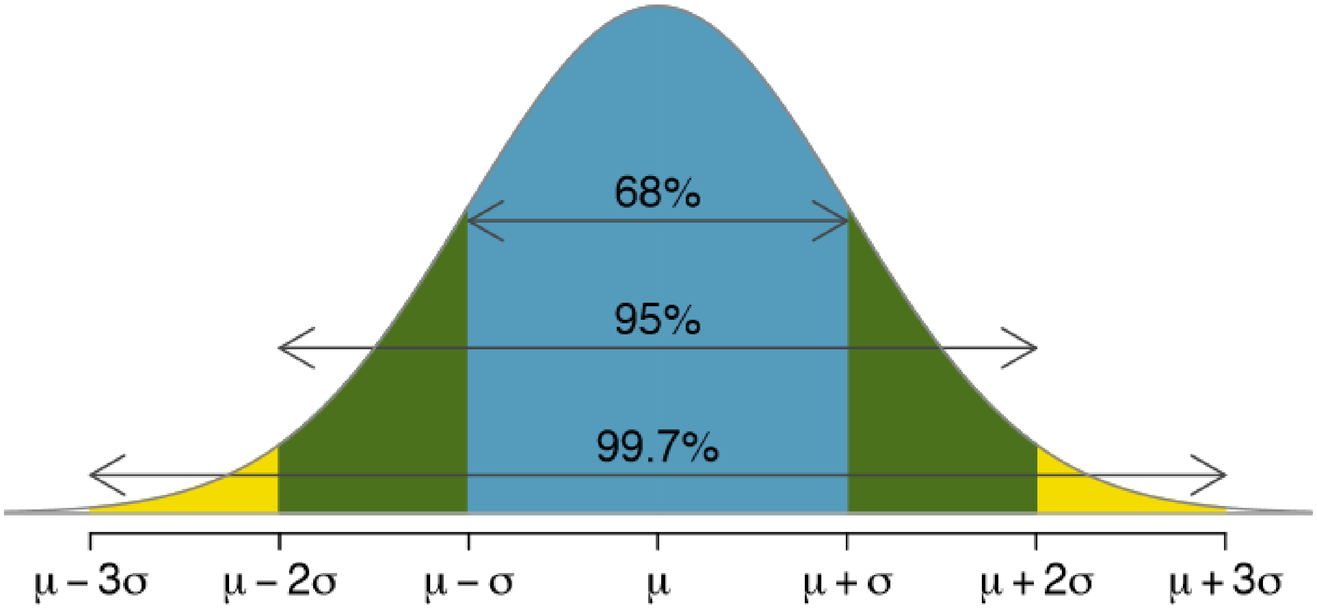
Example null Hypothesis for covid test:
H0: No confirmation of the virus.

	***H_0* is true**	***H_0* is false**
H_0 is not rejected	correct	False negative
H_0 is rejected	False positive	Correct (positive)

Hypothesis testing (2/3)

1. Formulate null hypothesis and alternative hypothesis
2. Fix significance (chance of false positive)
3. Find a test statistic
4. Find critical region

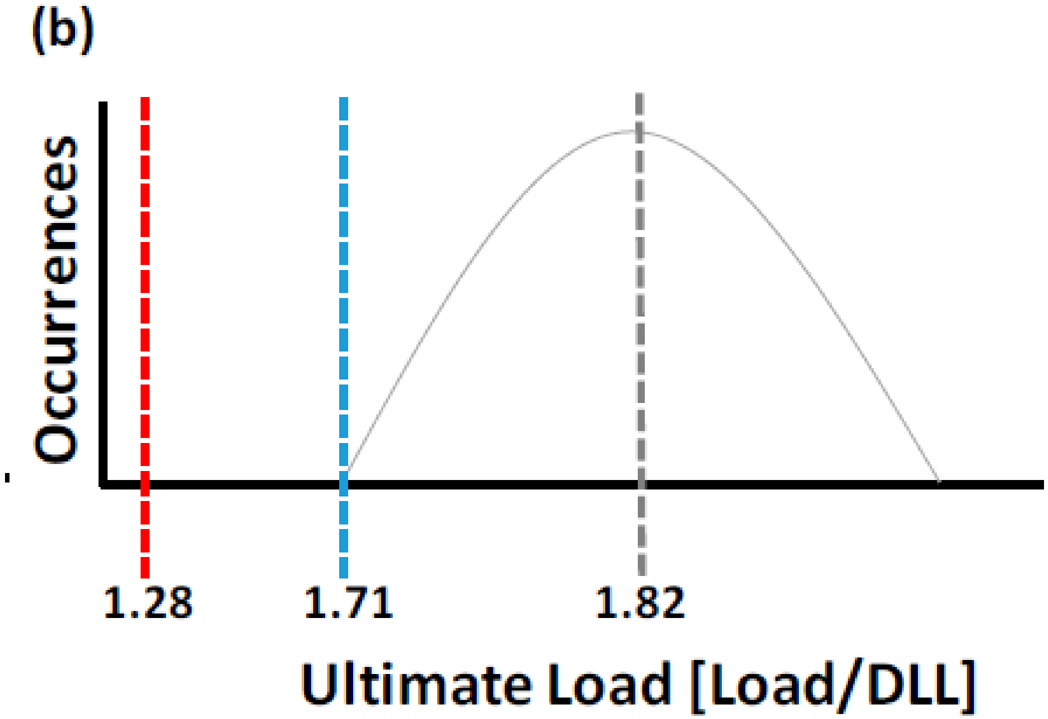
$$\boxed{\sigma = \sigma_{total} = \sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2}}$$



Hypothesis testing (3/3) – engineering example

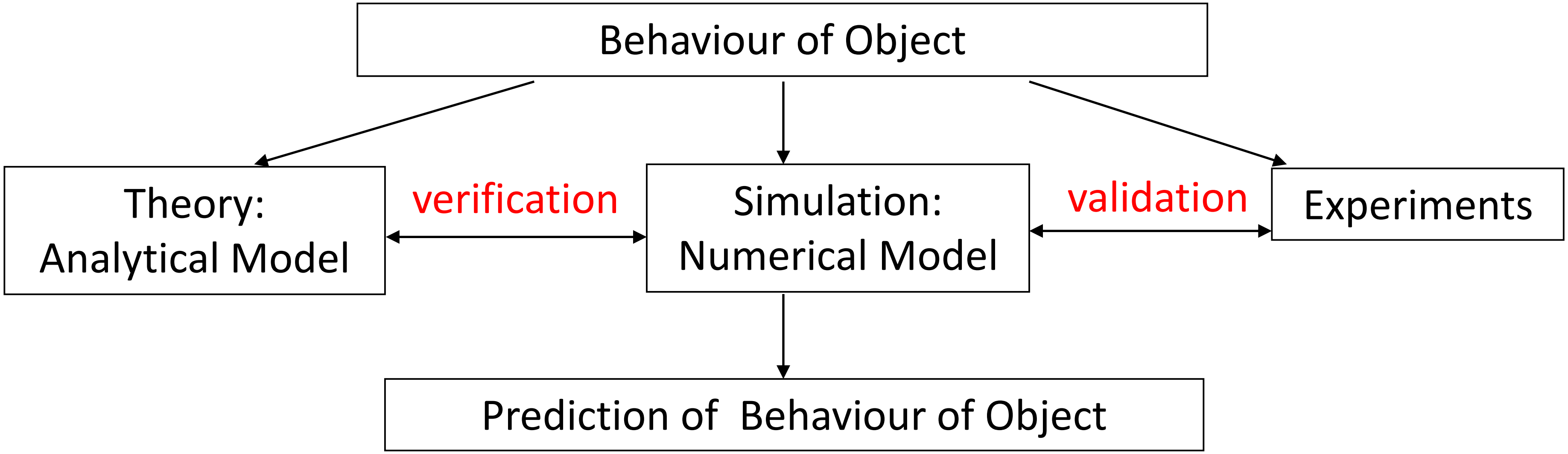
H0: mean is equal to 1.28 Design Limit Load

Alternative: ultimate crane load population mean < 1.28 DLL



Esola et al, Applied Sciences, 2018

Summary: Verification and Validation



Each step of the design cycle bugs cost 10 x more
Pentium FDIV bug cost Intel $475 million

Alan Hu, Computer Science, UBC

Verification and Validation in the BSc/MSc

- V & V procedures are to be defined in the midterm and final design of the DSE and in the MSc thesis

- V &V procedures are required for accreditation of a model or simulation
- V & V procedures are required for certification of aerospace vehicles with operational requirements

AIAA Guide for the Verification and Validation of Computational Fluid Dynamics Simulations

I SPENT THE PAST WEEK FIXING A CRITICAL BUG IN THE SOFTWARE THAT I FORGOT TO TELL YOU ABOUT.
Dilbert.com @ScottAdamsSays
IN A WAY, IT'S LIKE YOU NEVER EXISTED.
4-1-17 © 2017 Scott Adams, Inc./Dist. by Andrews McMeel
NO, IT'S NOT LIKE THAT AT ALL.
AND YOU HAVE A BAD ATTITUDE ON TOP OF ALL THAT.

Go to www.menti.com, code 781049
Quiz: Verification or Validation

1) On an exam you have to derive an equation. You check if the equation has the right units
2) You write a computer program for your MSc thesis and check that you get the same result as the MSc student that developed the code.
3) A self-driving car prototype driving on the road with a test driver on stand-by
4) You have built a simulation for the ocean and check that the difference between high and low tide is 12 hours and 25 minutes

SOFTWARE VERIFICATION

What is the program supposed to do?

When somebody gives you a program, can you check if it is functioning properly?

No, you need a *specification* of the program

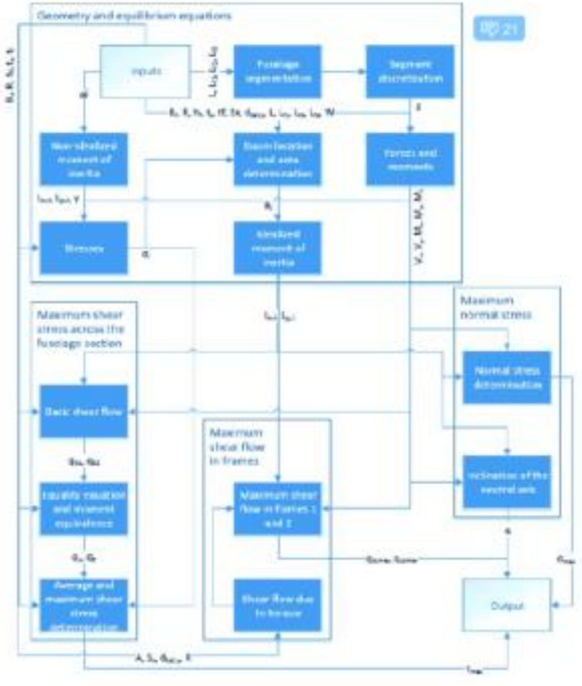
Language can be imprecise, mathematics is precise but not readable

Laski and Stanley (2009)

It is very difficult to be an objective tester of your own code. Working in a group has advantages!

Specification

```
Pseudocode:
begin
        Compute the set S of all monotonically increasing
        sequences in the subarray A [1 ... n ];
        Find the longest length of the sequences in S ;
end ;
```



Geometry and equilibrium equations
Inputs
Fuselage segmentation
Segment discretisation
Non-idealized moment of inertia
Boom location and area determination
Forces and moments
Stresses
Idealized moment of inertia
Maximum shear stress across the fuselage section
Basic shear flow
Equality equation and moment equivalence
Average and maximum shear stress determination
Maximum shear flow in frames
Maximum shear flow in frames 1 and 2
Shear flow due to torque
Maximum normal stress
Normal stress determination
Inclination of the neutral axis
Output

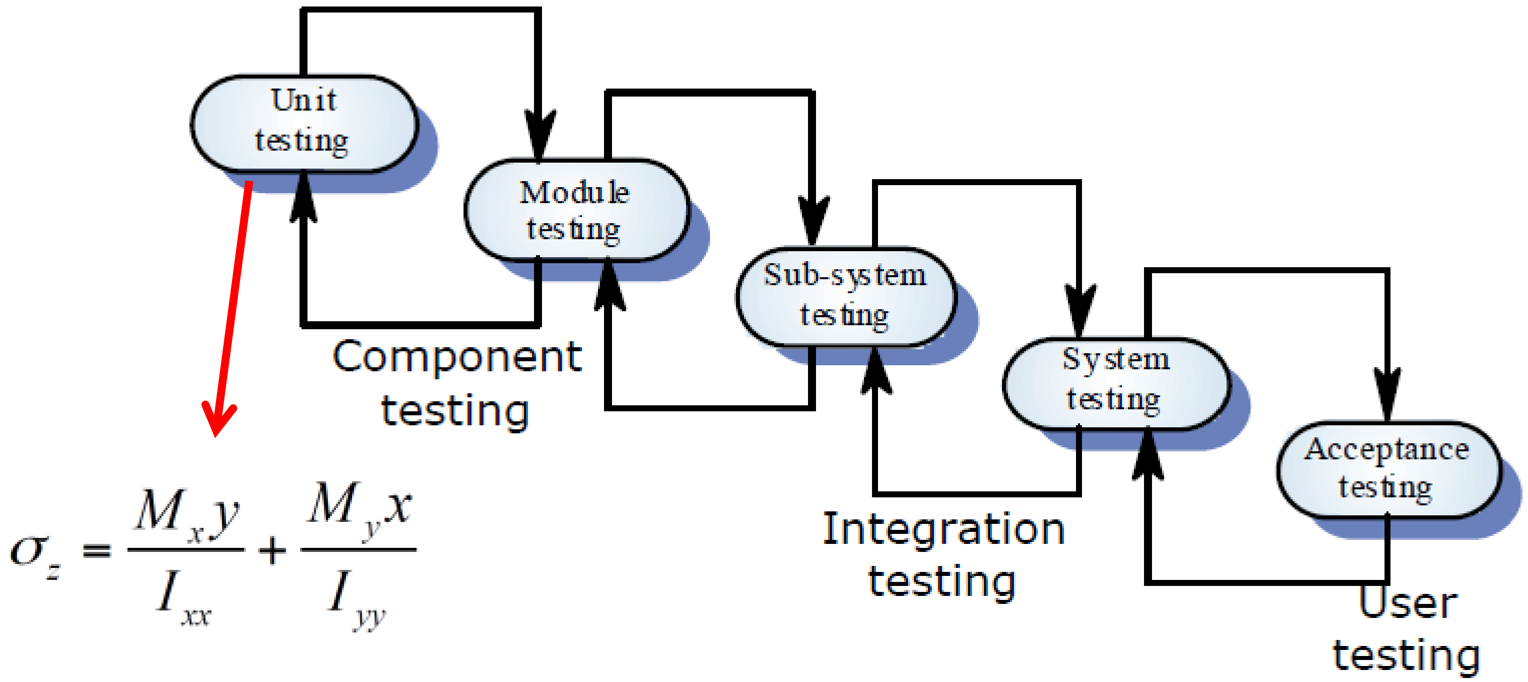
Figure 3.1: Flow chart

$$q_{s,f} = -\frac{I_{xx}V_x - I_{xy}V_y}{I_{xx}I_{yy} - I_{xy}^2}\sum_{r=1}^{n} B_r y_r - \frac{I_{yy}V_y - I_{xy}V_x}{I_{xx}I_{yy} - I_{xy}^2}\sum_{r=1}^{n} B_r x_r + q_{s0,f}$$

$$q_{s0f} = \frac{V_x\eta - V_y\varepsilon - \int_0^{\theta} r^2 q_{bf}\,d\theta}{2A}$$

Laski and Stanley (2009)

Software development

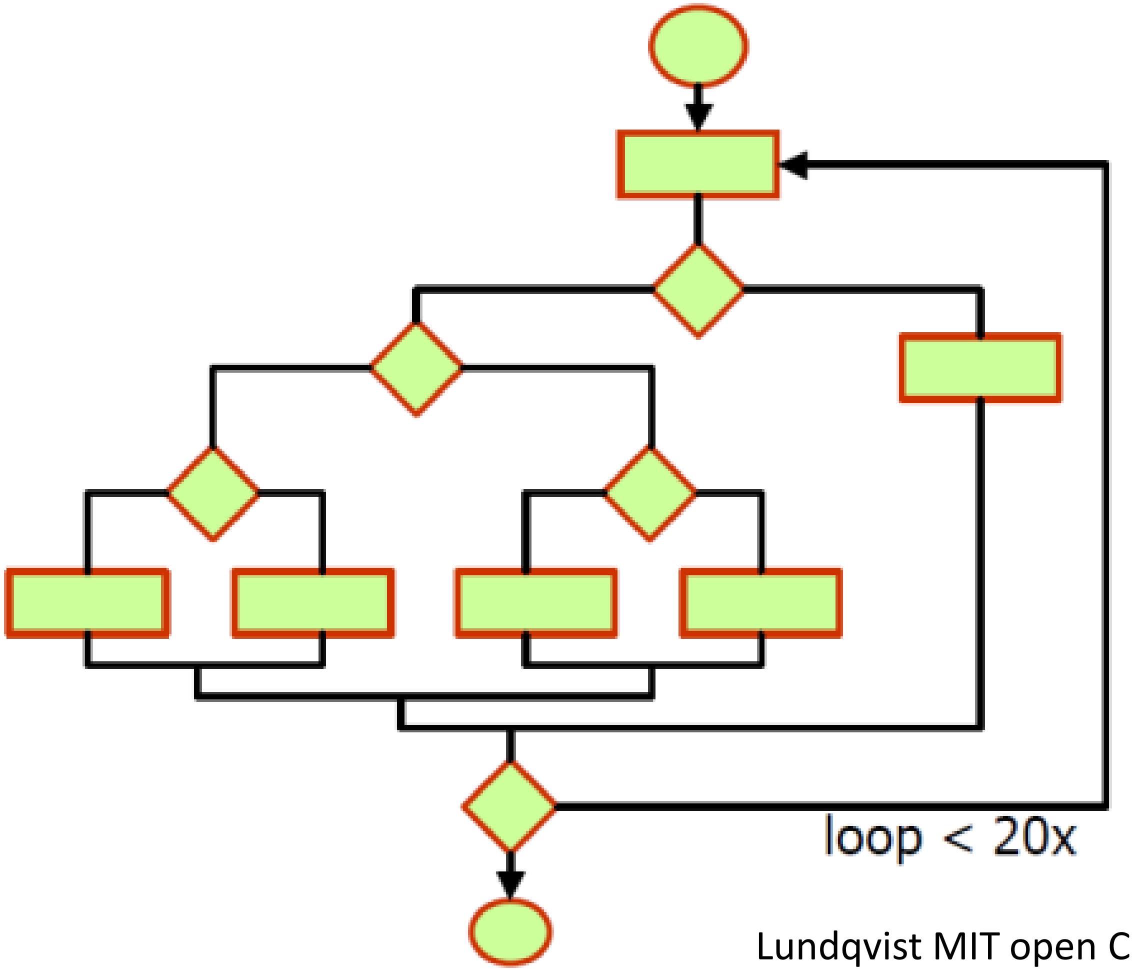


$$\sigma_z = \frac{M_x y}{I_{xx}} + \frac{M_y x}{I_{yy}}$$

Verification does not consist of only one comparison!

Lundqvist MIT open CourseWare

Verification of Software

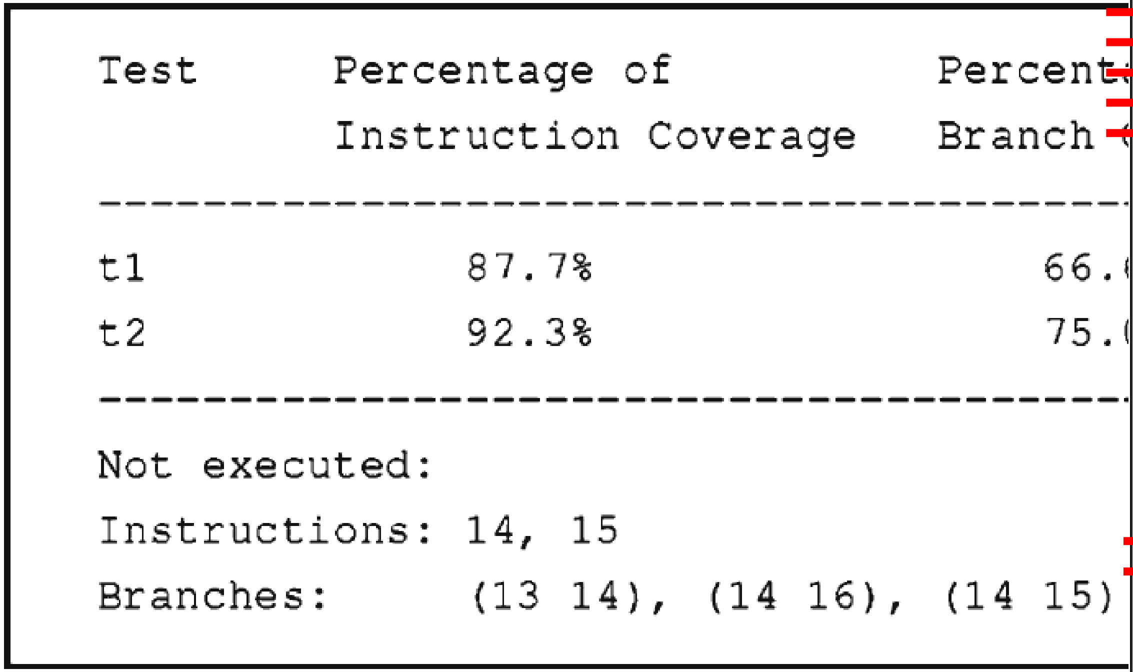


There are $5^{20}=10^{14}$ possible paths

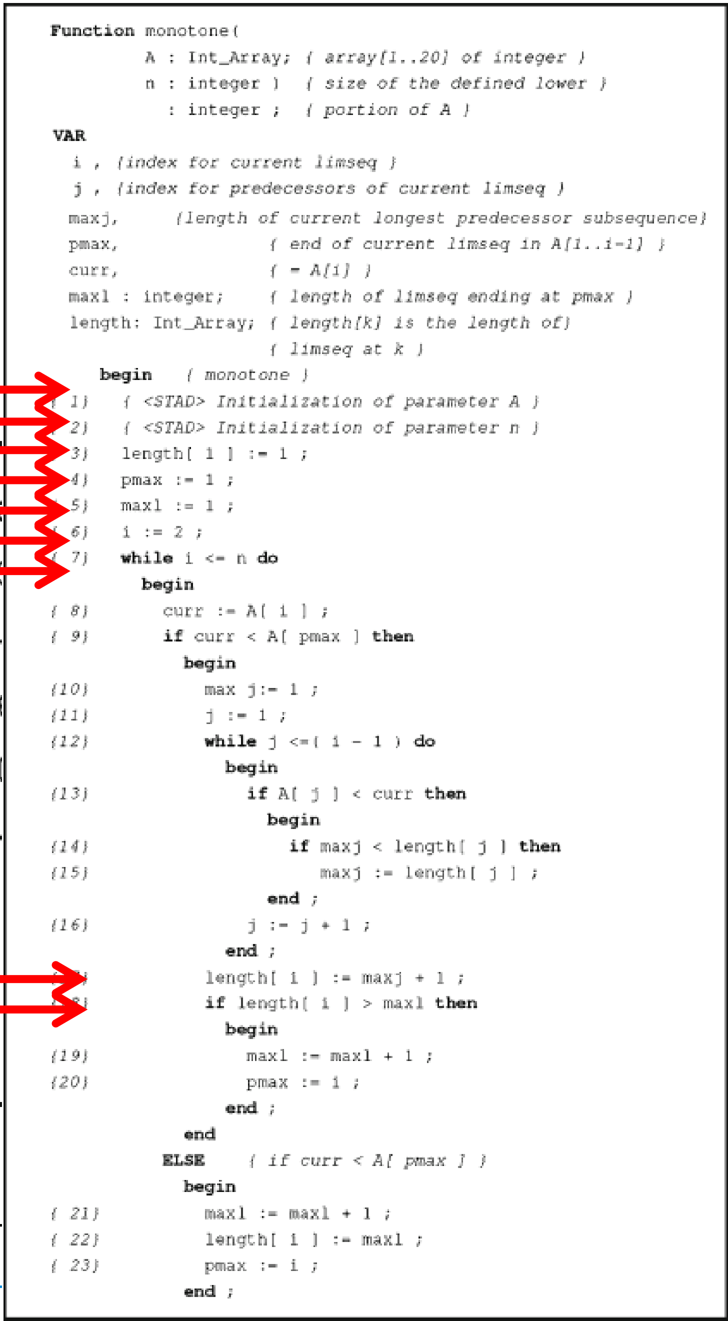
Lundqvist MIT open CourseWare

Structural testing (white box)

Code coverage: which parts of the code are executed?



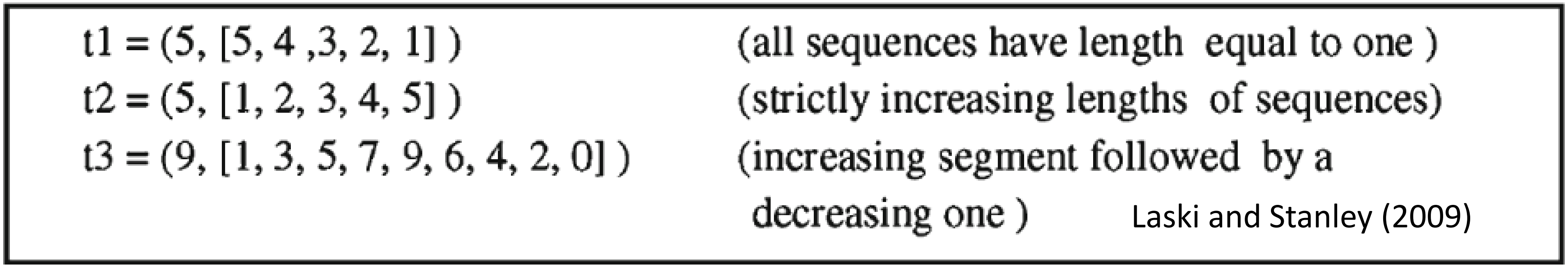
```
Test      Percentage of                Percent
          Instruction Coverage         Branch (
---------------------------------------------------
t1             87.7%                       66.(
t2             92.3%                       75.(
---------------------------------------------------
Not executed:
Instructions: 14, 15
Branches:      (13 14), (14 16), (14 15)
```



```
Function monotone(
         A : Int_Array; { array[1..20] of integer }
         n : integer )  { size of the defined lower }
           : integer ;  { portion of A }
VAR
  i , {index for current limseq }
  j , {index for predecessors of current limseq }
 maxj,      {length of current longest predecessor subsequence}
 pmax,                { end of current limseq in A[1..i-1] }
 curr,                { = A[i] }
 maxl : integer;      { length of limseq ending at pmax }
 length: Int_Array;   { length[k] is the length of}
                      { limseq at k }
    begin   { monotone }
{ 1}  { <STAD> Initialization of parameter A }
{ 2}  { <STAD> Initialization of parameter n }
{ 3}  length[ 1 ] := 1 ;
{ 4}  pmax := 1 ;
{ 5}  maxl := 1 ;
{ 6}  i := 2 ;
{ 7}  while i <= n do
        begin
{ 8}      curr := A[ i ] ;
{ 9}      if curr < A[ pmax ] then
            begin
{10}          max j:= 1 ;
{11}          j := 1 ;
{12}          while j <=( i - 1 ) do
                begin
{13}              if A[ j ] < curr then
                    begin
{14}                  if maxj < length[ j ] then
{15}                     maxj := length[ j ] ;
                    end ;
{16}              j := j + 1 ;
                end ;
{17}          length[ i ] := maxj + 1 ;
{18}          if length[ i ] > maxl then
                begin
{19}              maxl := maxl + 1 ;
{20}              pmax := i ;
                end ;
            end
          ELSE    { if curr < A[ pmax ] }
            begin
{ 21}         maxl := maxl + 1 ;
{ 22}         length[ i ] := maxl ;
{ 23}         pmax := i ;
            end ;
```

Testing (dynamic)

Program to find the largest increasing sequence in an array A

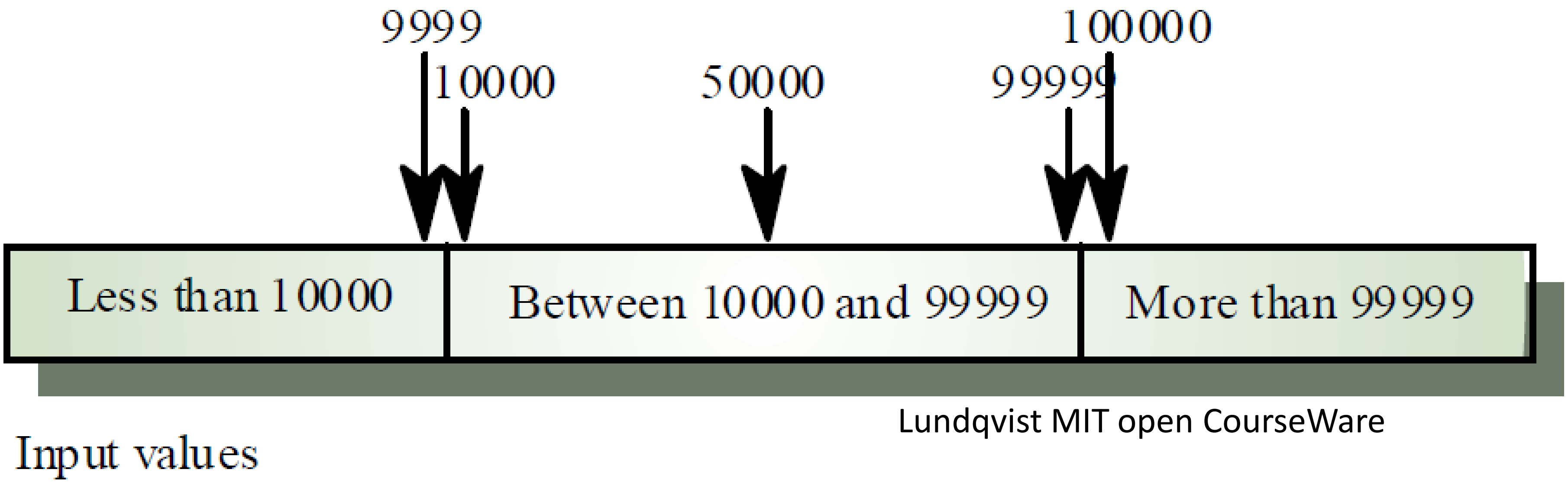


t1 = (5, [5, 4 ,3, 2, 1]) (all sequences have length equal to one)

t2 = (5, [1, 2, 3, 4, 5]) (strictly increasing lengths of sequences)

t3 = (9, [1, 3, 5, 7, 9, 6, 4, 2, 0]) (increasing segment followed by a decreasing one)

Laski and Stanley (2009)

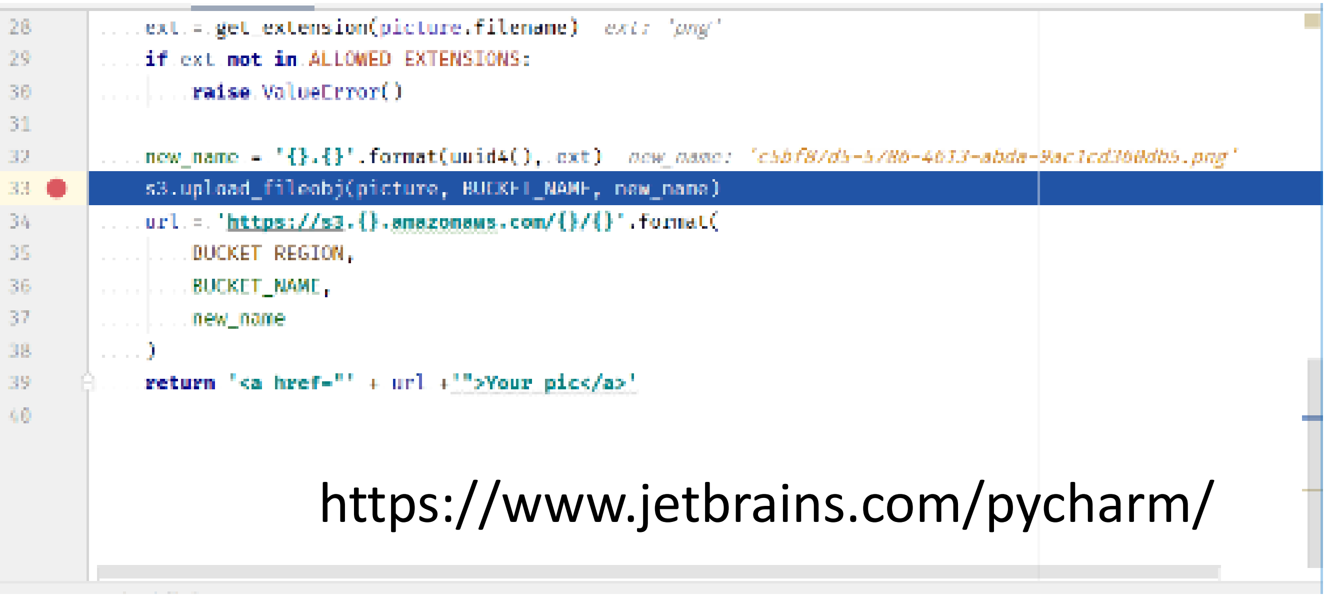


Lundqvist MIT open CourseWare

Debugging

- “walking through” the code vs intuition
- Binary search

After the fix, retest!



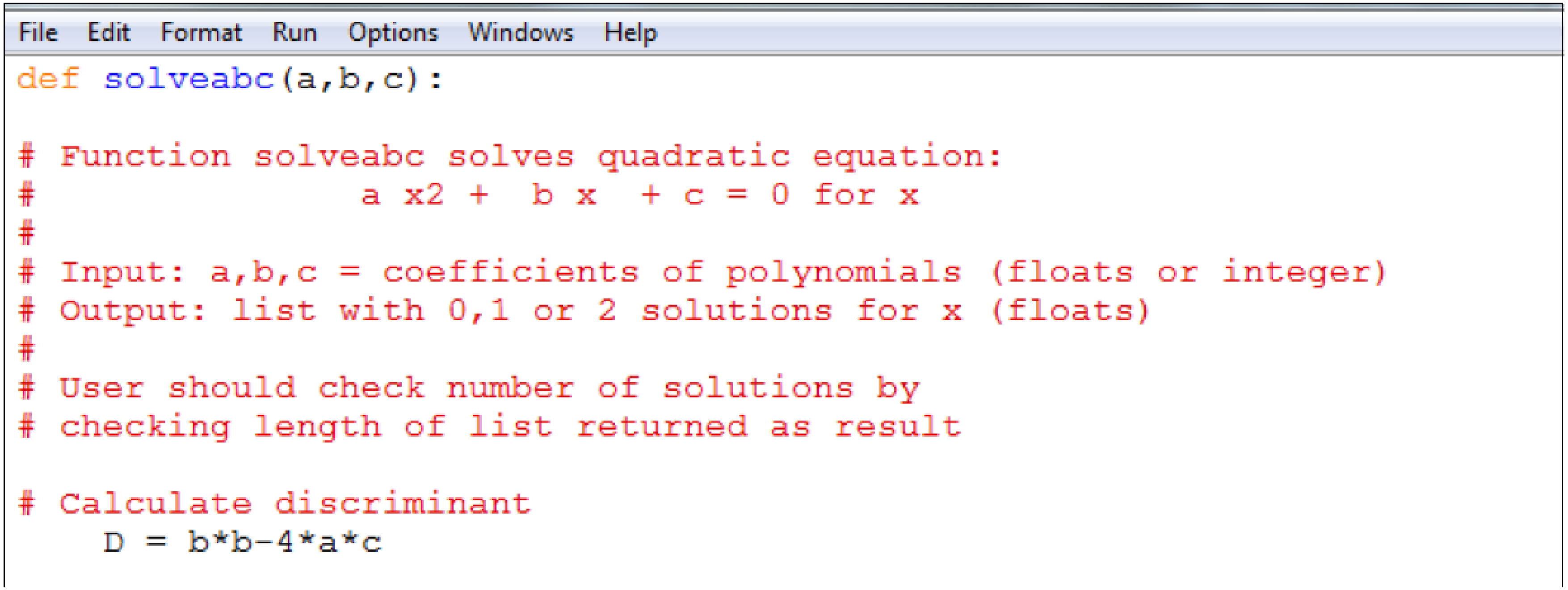
```
    ext = get_extension(picture.filename)  ext: 'png'
    if ext not in ALLOWED_EXTENSIONS:
        raise ValueError()

    new_name = '{}.{}'.format(uuid4(), ext)  new_name: 'c5bf87d5-5786-4633-abde-9ac1cd360db5.png'
    s3.upload_fileobj(picture, BUCKET_NAME, new_name)
    url = 'https://s3.{}.amazonaws.com/{}/{}'.format(
        BUCKET_REGION,
        BUCKET_NAME,
        new_name
    )
    return '<a href="' + url +'">Your pic</a>'
```

https://www.jetbrains.com/pycharm/

> *The sad truth is that debugging is the least researched and, consequently, least understood area in software engineering, despite the fact that it is most likely one of the most time-consuming and costly activities.* Laski and Stanley (2009)

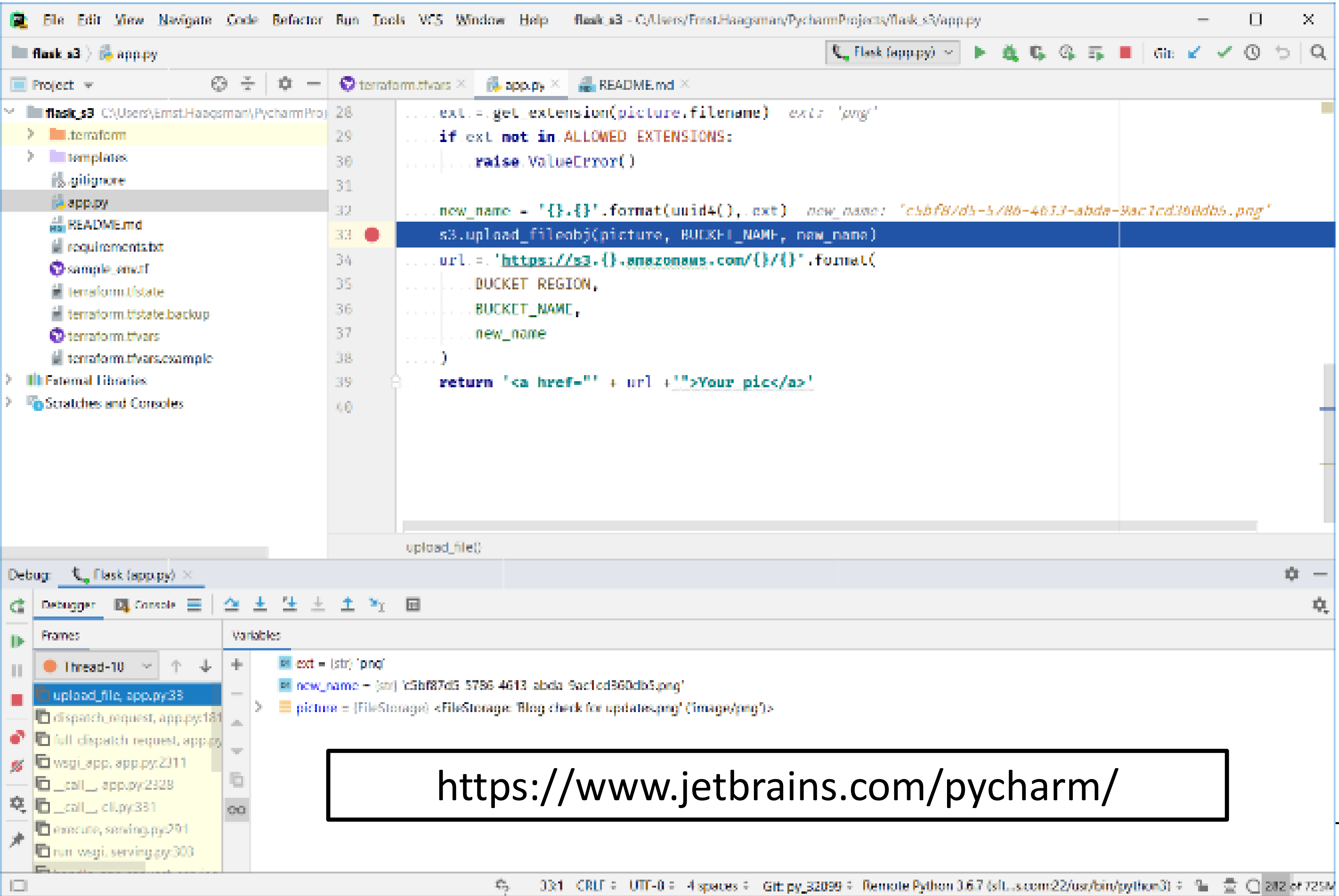
Tools – AE1205 lecture notes

4. Making your code reusable and readable



```
def solveabc(a,b,c):

# Function solveabc solves quadratic equation:
#               a x2 +  b x  + c = 0 for x
#
# Input: a,b,c = coefficients of polynomials (floats or integer)
# Output: list with 0,1 or 2 solutions for x (floats)
#
# User should check number of solutions by
# checking length of list returned as result

# Calculate discriminant
    D = b*b-4*a*c
```

Debugging tools - Pycharm



```
    ext = get_extension(picture.filename)   ext: 'png'
    if ext not in ALLOWED_EXTENSIONS:
        raise ValueError()

    new_name = '{}.{}'.format(uuid4(), ext)   new_name: 'c5bf87d5-5786-4613-abda-9ac1cd360db5.png'
    s3.upload_fileobj(picture, BUCKET_NAME, new_name)
    url = 'https://s3.{}.amazonaws.com/{}/{}'.format(
        BUCKET_REGION,
        BUCKET_NAME,
        new_name
    )
    return '<a href="' + url + '">Your pic</a>'
```

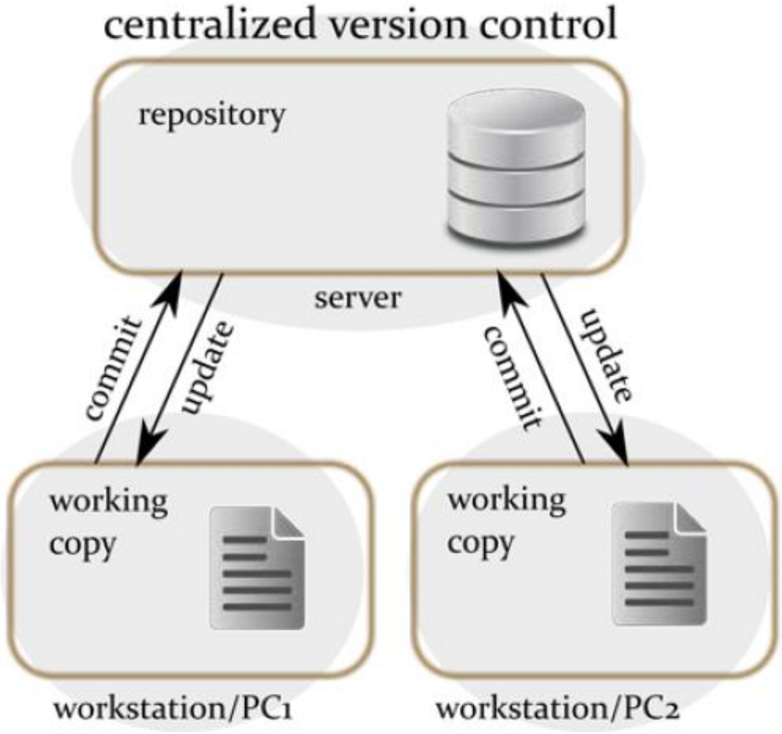
```
upload_file, app.py:33
dispatch_request, app.py:181
full_dispatch_request, app.py
wsgi_app, app.py:2311
__call__, app.py:2328
__call__, cli.py:331
execute, serving.py:291
run_wsgi, serving.py:303
```

```
ext = {str} 'png'
new_name = {str} 'c5bf87d5-5786-4613-abda-9ac1cd360db5.png'
picture = {FileStorage} <FileStorage: 'Blog check for updates.png' ('image/png')>
```

```
33:1  CRLF  UTF-8  4 spaces  Git: py_32099  Remote Python 3.6.7 (sft...s.com:22/usr/bin/python3)  282 of 725M
```

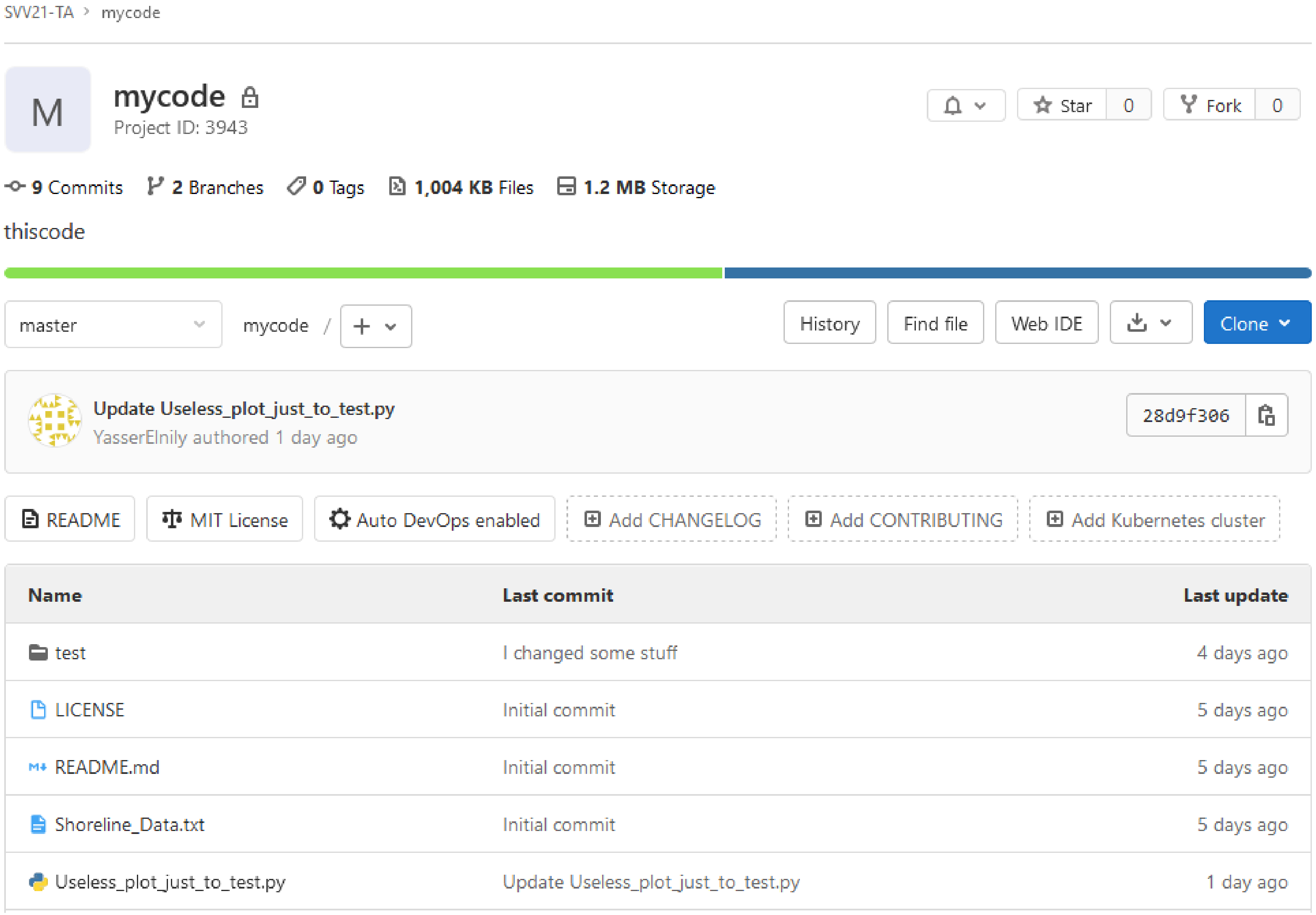
https://www.jetbrains.com/pycharm/

Tools – AE1205 Version Control



Debugging tools - Gitlab

gitlab.tudelft.nl



SVV21-TA > mycode

mycode

Project ID: 3943

9 Commits 2 Branches 0 Tags 1,004 KB Files 1.2 MB Storage

thiscode

master mycode /

Update Useless_plot_just_to_test.py

YasserElnily authored 1 day ago

28d9f306

Name	Last commit	Last update
test	I changed some stuff	4 days ago
LICENSE	Initial commit	5 days ago
README.md	Initial commit	5 days ago
Shoreline_Data.txt	Initial commit	5 days ago
Useless_plot_just_to_test.py	Update Useless_plot_just_to_test.py	1 day ago

Verification and validation is an important part of our specialized society. We require trust to rely on specialization and verification and validation can be seen as institutionalized trust.

ESA.com