Resonant Ultrasound Spectroscopy

En Yung Hoang

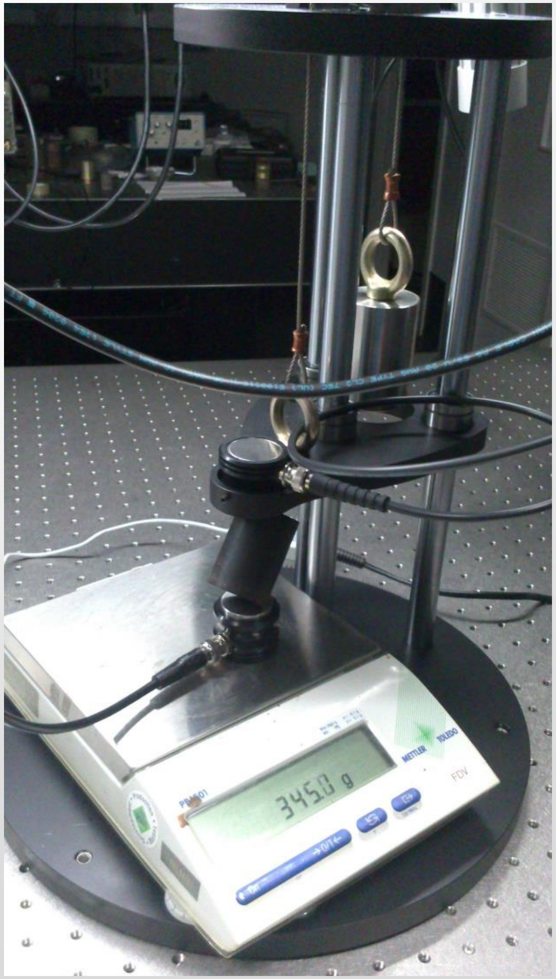
Supervisors:

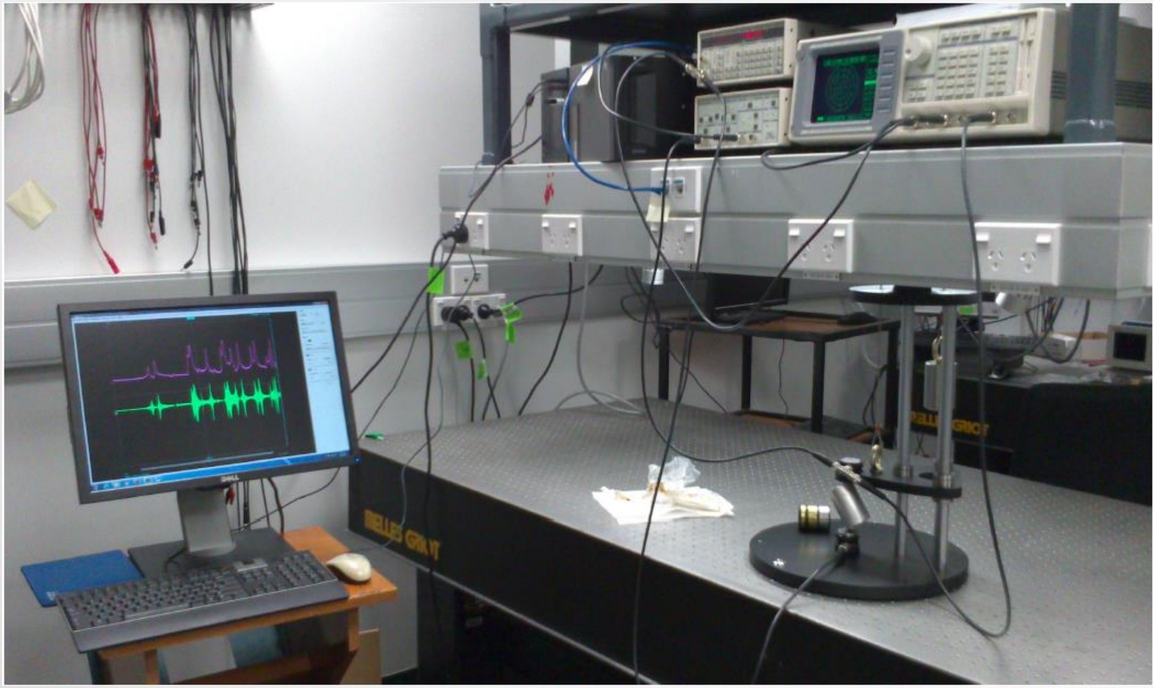
Paul Freeman

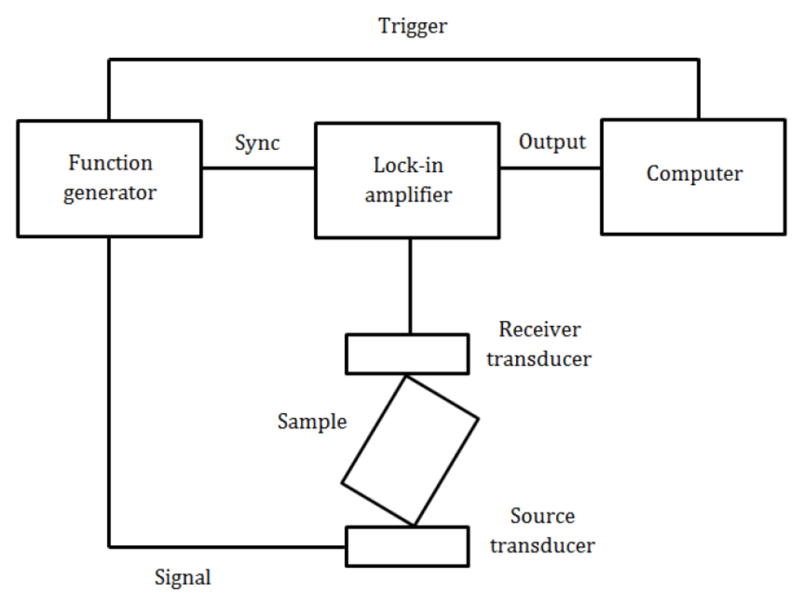
Kasper van Wijk

What is RUS?

- Determines material properties of solid objects
- General setup consists of:
  - Transducers
  - Function Generator
  - Amplifier







Trigger

Function generator

Sync

Lock-in amplifier

Output

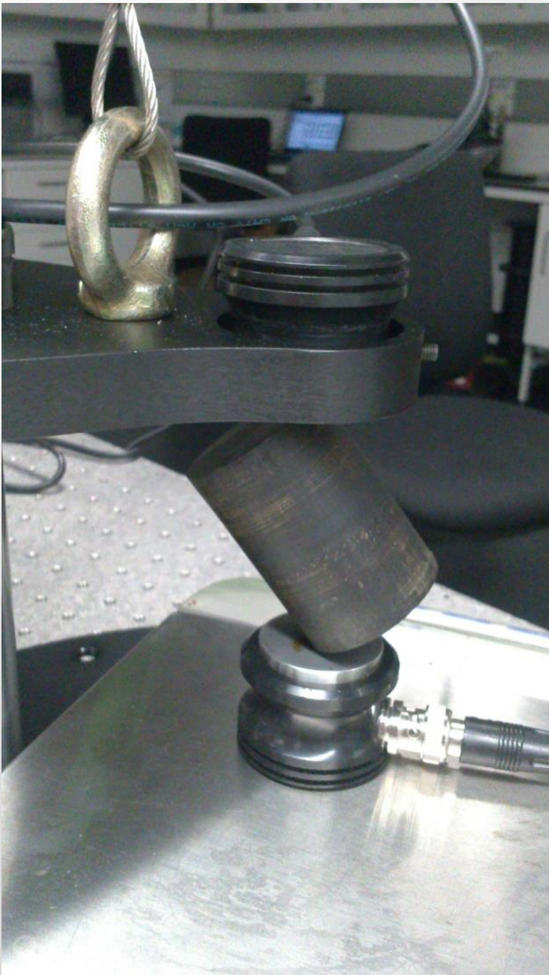
Computer

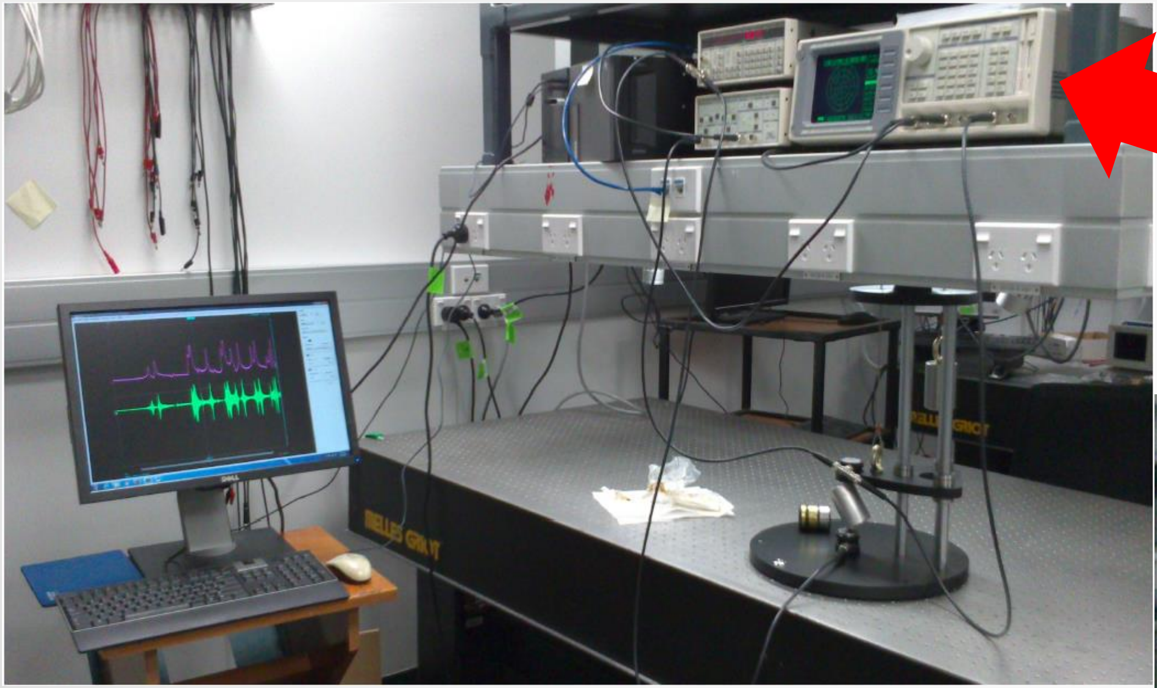
Receiver transducer

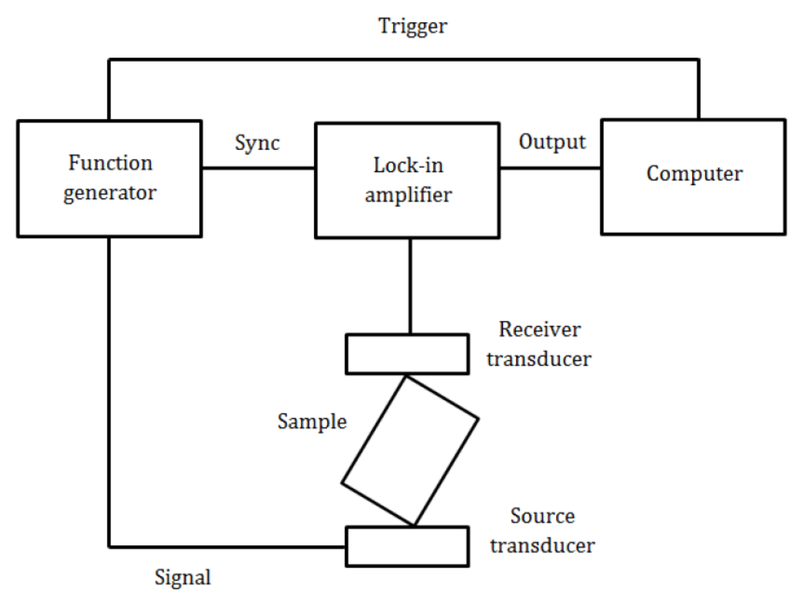
Sample

Source transducer

Signal






Trigger

Function generator

Sync

Lock-in amplifier

Output

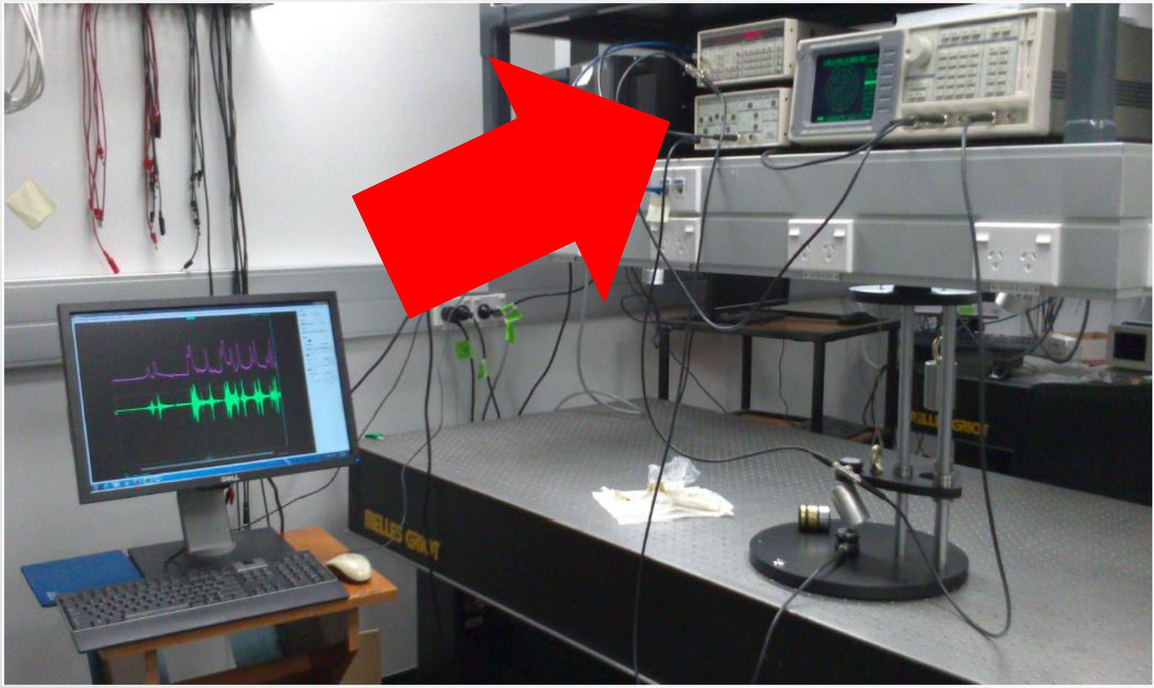
Computer

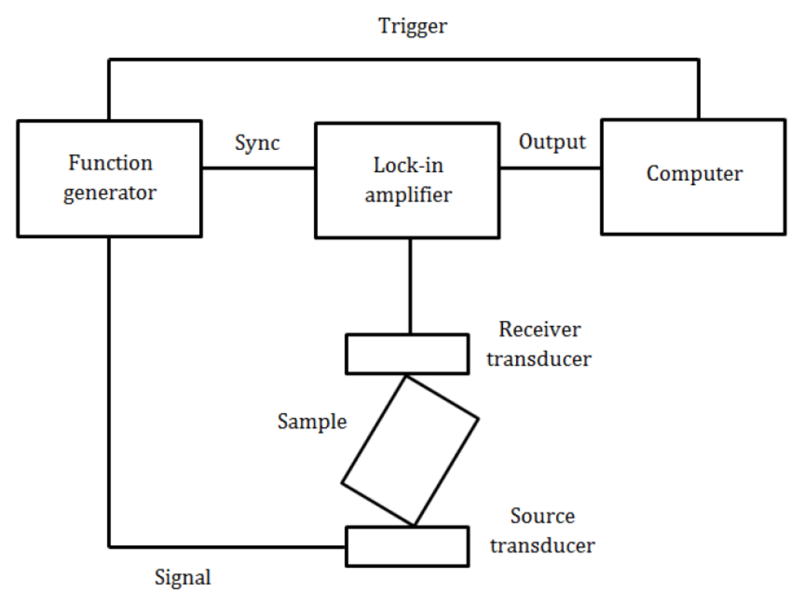
Receiver transducer

Sample

Source transducer

Signal





Trigger

Function generator

Sync

Lock-in amplifier

Output

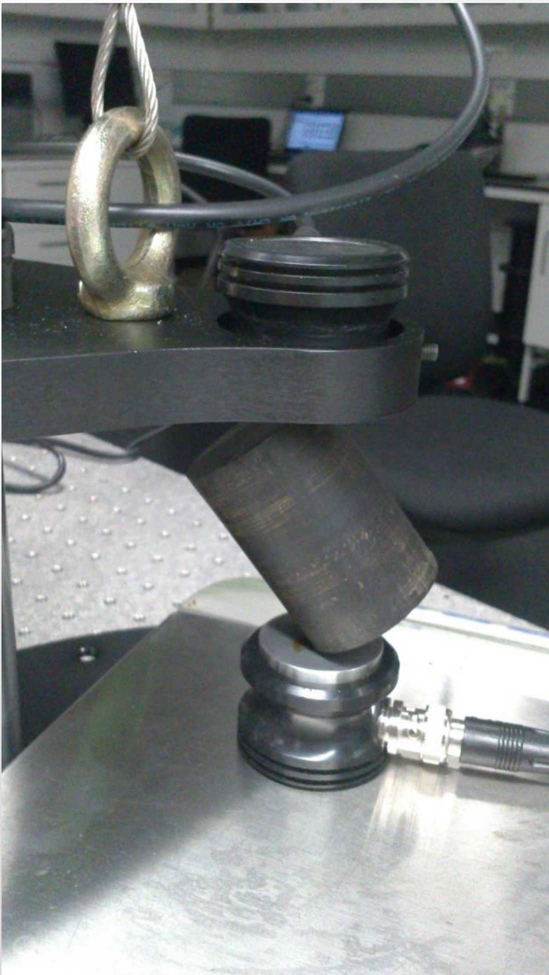
Computer

Receiver transducer

Sample

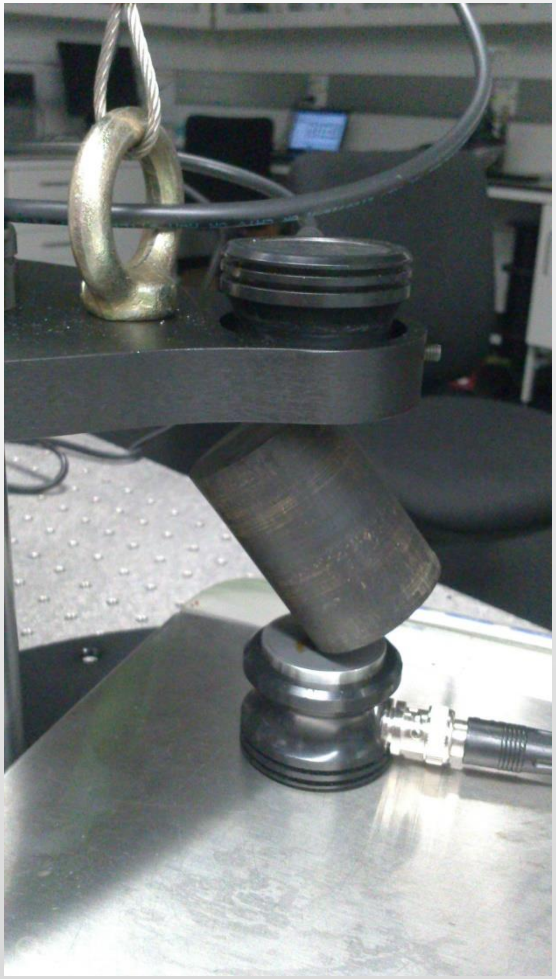
Source transducer

Signal



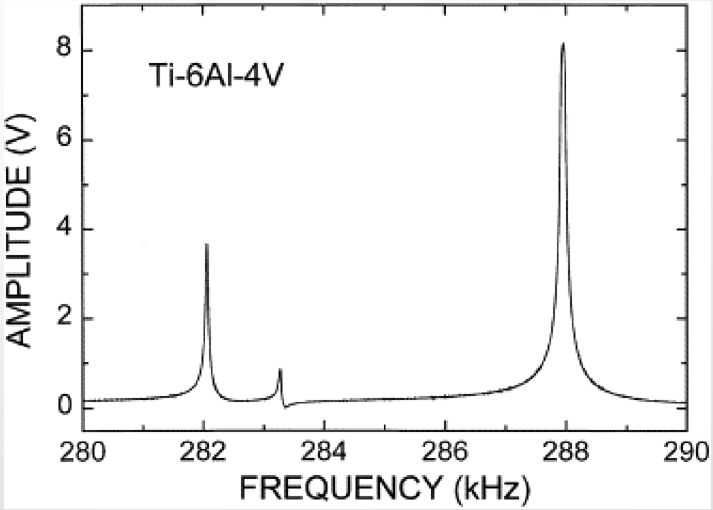
What is RUS?

- Solid object held together by two transducers
  - Uses honey to keep it in position
  - Held at an angle



What is RUS?

- Output are resonant peaks
  - Resonant peaks are caused by the shape, size and elastic parameters of the solid object
- Two algorithms
  - Forward Algorithm (Main focus)
  - Inverse Algorithm



The Problem

- C implementation is hard to install
- C implementation translated to Python
    - Reduces dependencies on hard to obtain and install libraries BUT made it slow

Tools considered

SWIG

ctypes

Weave

SIP

Cython

(Numba)

Chosen Tool

Cython

- What is it?
  - Optimization static compiler
  - Combines both Python and C into one file called a Pyrex file

Cython

- Why did I choose it?
  - Both languages in one file
  - Generates executable

Cython

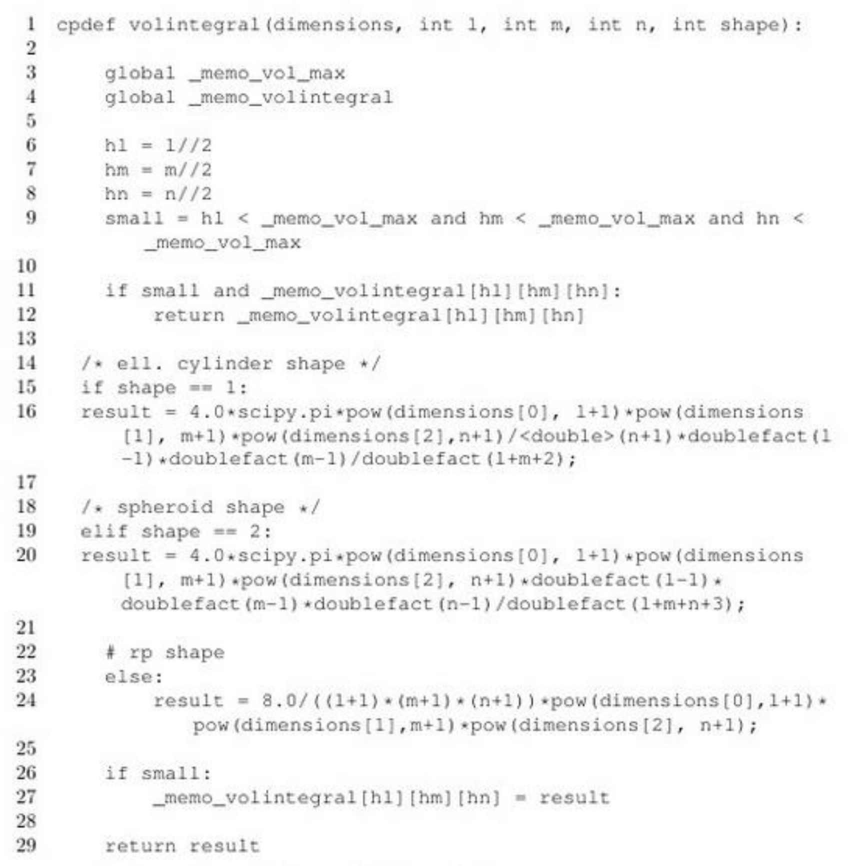
- How does it work?
  - Declare functions as def, cdef, cpdef
    - def and cdef cannot call each other
  - Uses a custom setup.py file to build the files

Simple Example of Cython

- Changes made:
    - Line 1: def → cpdef
    - Line 1: (kmax) → (int kmax)
    - Lines 2 and 3 were added in
- Increased while loop performance because it now uses a C Integer
- Line 16: A Python statement but its performance is increased because it uses a C integer

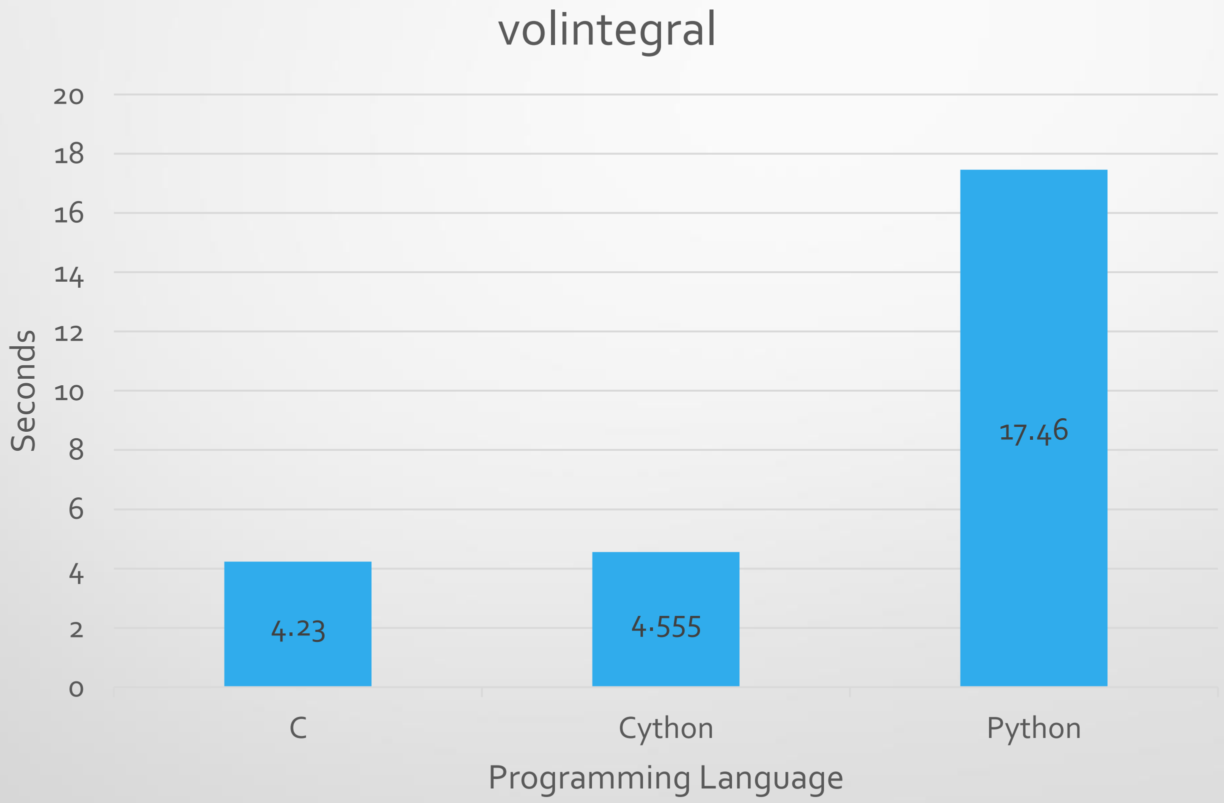
```
1  cpdef primes(int kmax):
2      cdef int n, k, i
3      cdef int p[1000]
4      result = []
5      if kmax > 1000:
6          kmax = 1000
7      k = 0
8      n = 2
9      while k < kmax:
10         i = 0
11         while i < k and n % p[i] != 0:
12             i = i + 1
13         if i == k:
14             p[k] = n
15             k = k + 1
16             result.append(n)
17         n = n + 1
18     return result
```

Simple RUS Cython Example



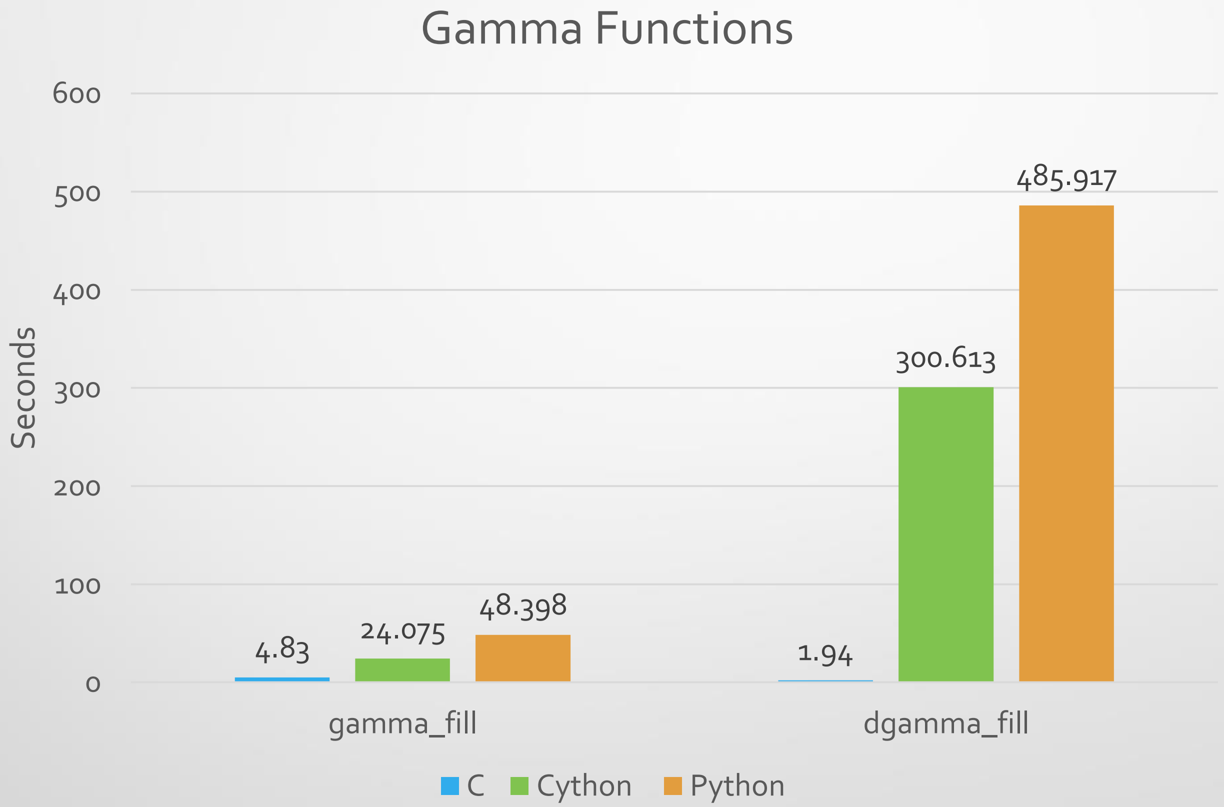
```
cpdef volintegral(dimensions, int l, int m, int n, int shape):

    global _memo_vol_max
    global _memo_volintegral

    hl = l//2
    hm = m//2
    hn = n//2
    small = hl < _memo_vol_max and hm < _memo_vol_max and hn < _memo_vol_max

    if small and _memo_volintegral[hl][hm][hn]:
        return _memo_volintegral[hl][hm][hn]

  /* ell. cylinder shape */
  if shape == 1:
  result = 4.0*scipy.pi*pow(dimensions[0], l+1)*pow(dimensions[1], m+1)*pow(dimensions[2],n+1)/<double>(n+1)*doublefact(l-1)*doublefact(m-1)/doublefact(l+m+2);

  /* spheroid shape */
  elif shape == 2:
  result = 4.0*scipy.pi*pow(dimensions[0], l+1)*pow(dimensions[1], m+1)*pow(dimensions[2], n+1)*doublefact(l-1)*doublefact(m-1)*doublefact(n-1)/doublefact(l+m+n+3);

    # rp shape
    else:
        result = 8.0/((l+1)*(m+1)*(n+1))*pow(dimensions[0],l+1)*pow(dimensions[1],m+1)*pow(dimensions[2], n+1);

    if small:
        _memo_volintegral[hl][hm][hn] = result

    return result
```

Results



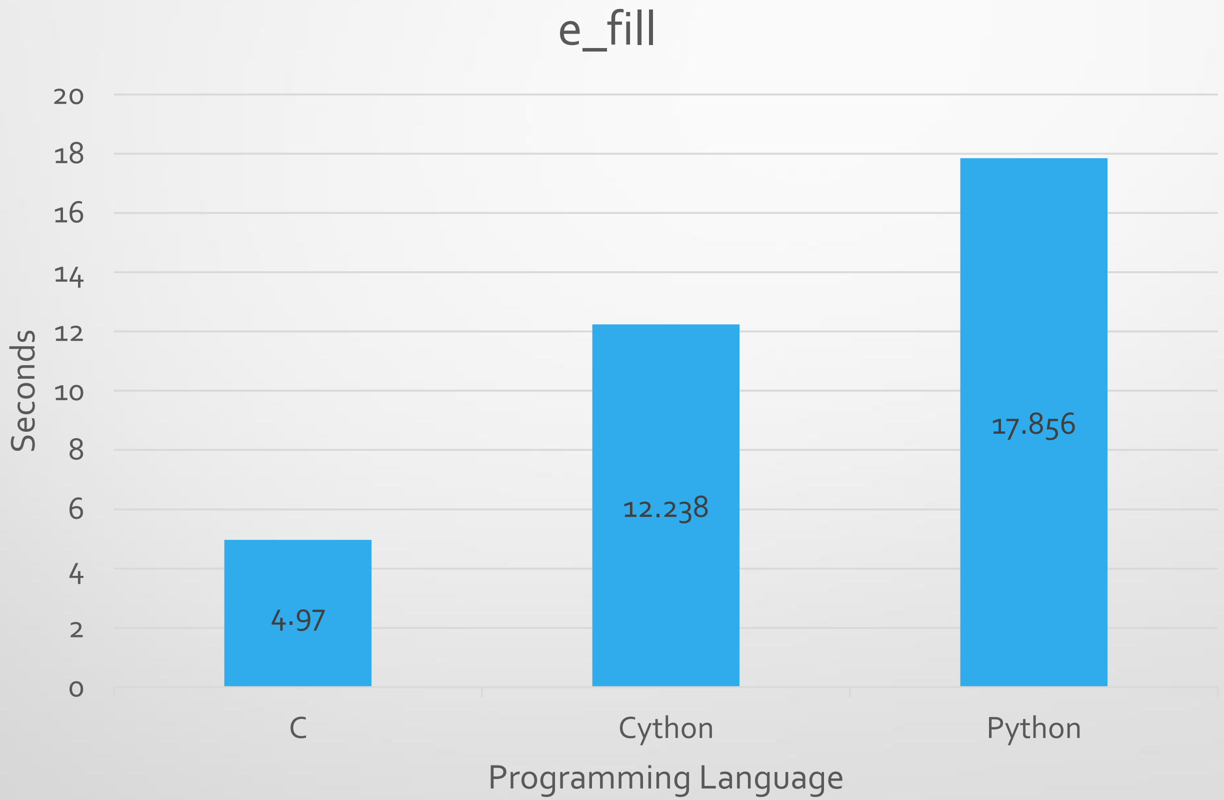
Ran at 100 iterations

Results



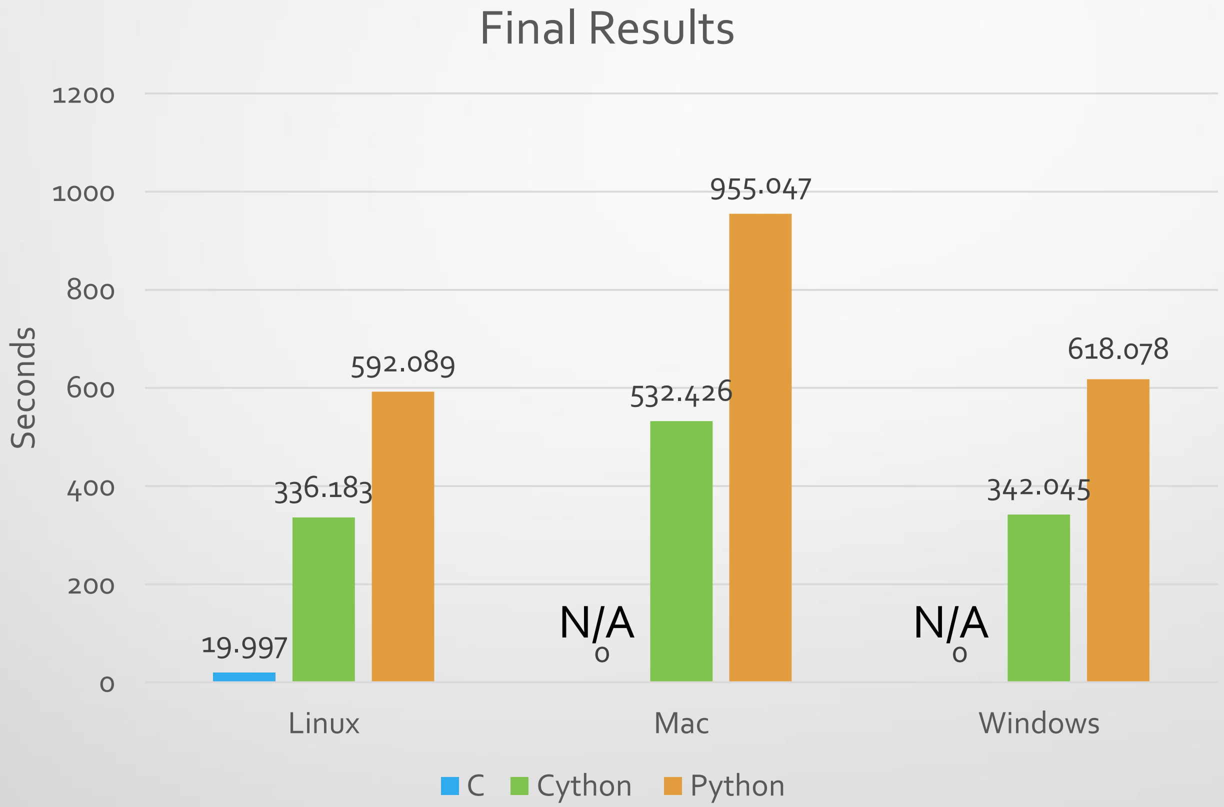
Ran at 100 iterations

Results



Ran at 100 iterations

Results



Ran at 100 iterations

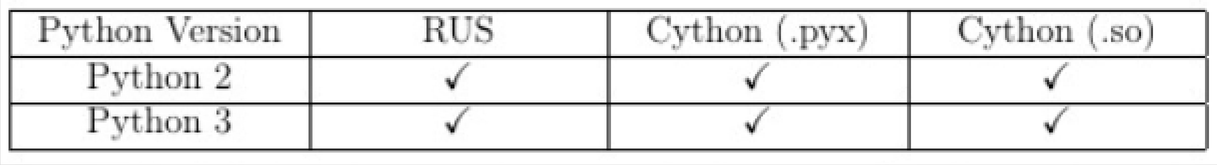
Problems with Cython

- Hard to create dimensional arrays
- Complications with Python and C objects

“Cannot convert 'int *' to a Python Object”

Program Compatibility

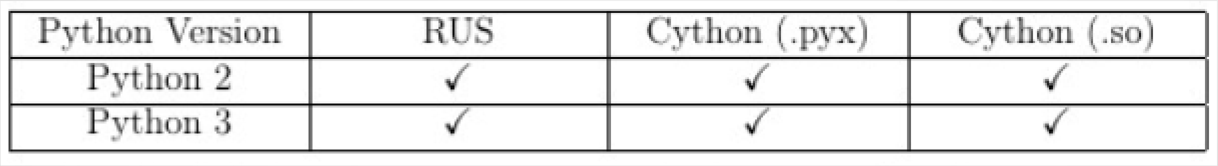
- Mac:
  - Pyrex files must be rebuilt
  - Executable built with Python 2 will not work with Python 3
  - Executable built with Python 3 is backwards compatible



Python Version	RUS	Cython (.pyx)	Cython (.so)
Python 2	✓	✓	✓
Python 3	✓	✓	✓

Program Compatibility

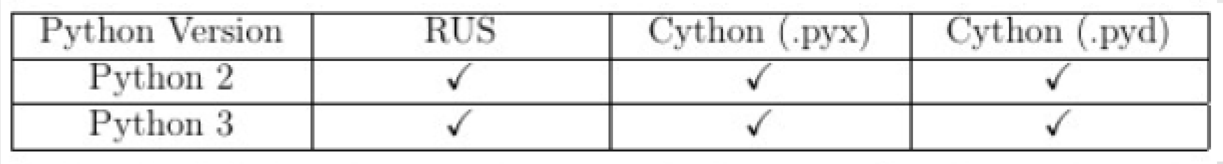
- Linux:
  - Pyrex files must also be rebuilt
  - Executable built using Python 2 can only work with Python 2
  - Executable built using Python 3 can only work with Python 3



Python Version	RUS	Cython (.pyx)	Cython (.so)
Python 2	✓	✓	✓
Python 3	✓	✓	✓

Program Compatibility

- Windows:
  - Everything works but the installation process is difficult compared to Mac and Linux



Python Version	RUS	Cython (.pyx)	Cython (.pyd)
Python 2	✓	✓	✓
Python 3	✓	✓	✓

My Contribution to RUS

- It is now easier to install and use
  - Can now be used on all 3 operating systems
- Faster than the Python implementation
  - Geoscientists will be happier ☺
- The generated executable makes it very portable
  - No extra installations
  - Only requires 1 command to run: “python –c “import rus” forward/inverse”

What I have gained from the project

- Got practice working in a team
- Gained confidence in communication via the weekly meetings and seminars
- Got better at reading someone else's code
- Learnt a new tool: Cython
- Learnt how to use GitHub
- Learnt how to write reports using LaTeX

Thank you!

Any Questions?