

SWIG: Simplified Wrapper and Interface Generator

ME 4953/5013 - Introduction to High-Performance Computing



- SWIG “automatically” generates wrappers for C and C++ programs into
 - Tcl
 - Python
 - Perl
 - Java
 - Ruby
 - PHP
 - R
 - Octave
 - and More!

- Simple interfaces can be generated right from header files.

fib.h

```
int fib(int a);
```

fib.c

```
#include "fib.h"

int fib(int a){
    if (a <=0 )
        return -1;
    else if (a == 1)
        return 0;
    else if ((a==2)|| (a==3))
        return 1;
    else
        return fib(a-2) + fib(a-1);
}
```

Generate wrapper

```
>swig -python -module fib fib.h
>gcc -fPIC -c fib*.c
>gcc -shared fib*.o -o _fib.so
```

Generates a python wrapper fib.py and a shared object library _fib.so

total.h

```
int total(double* x, int n);
```

total.c

```
#include "total.h"
int total(double* x, int n){
    int i;
    double count = 0;
    for(i=0;i<n;i++){
        count += x[i];
    }
    return count;
}
```

total.i

```
%module total
%{
#define SWIG_FILE_WITH_INIT
#include "total.h"
%}
#include "numpy.i"
%init %{
import_array();
%}
%apply (double* IN_ARRAY1, int DIM1) {(double* x, int n)};
#include "total.h"
```

Generate wrapper

```
>swig -python total.i  
>gcc -fPIC -c total*.c -I/path/to/python/include/files  
>gcc -shared total*.o -o _total.so
```

Generates a python wrapper `total.py` and a shared object library `_total.so`

To use:

```
>>import total  
>>x=range(10)  
>>total.total(x)  
45
```