

Software Matemático Básico. Prácticas con GNU Octave

J. Rafael Rodríguez Galván

1 de julio de 2008

2

Capítulo

Segunda Práctica: Extensiones C++ de Octave

2.1. Planteamiento

En el curso hemos comentado la utilidad de Octave cuando es usado como “lenguaje pegamento”, utilizándolo para programar y controlar interactivamente la mayor parte del programa, pero implementando las funciones críticas en un lenguaje compilado como C++.

Realizar una extensión de Octave es tan sencillo como crear un fichero C++ siguiendo el siguiente patrón:

```
1 #include <octave/oct.h>
2 DEFUN_DLD(name, args, nargs , ``Documentation string'')
3 {
4     // (...)
5     // C++ code
6     // (...)
7 }
8 return octave_value()
```

Por ejemplo, crearemos una extensión muy sencilla en un fichero llamado `extension.cpp`, que simplemente imprima el mensaje “Hola Mundo”. Para ello, escribimos en el fichero el siguiente código:

```
1 #include <octave/oct.h>
2 DEFUN_DLD(extension, args, nargs , ``Documentation string'')
3 {
4     std::cout << ``Hola Mundo!' ' << std::endl;
5 }
6 return octave_value()
```

Y lo compilamos con el programa `mkoctfile`. Éste último es un simple script que encapsula al compilador de C++ del sistema junto a los

parámetros adecuados.

```
1 $ mkoctfile extension.cpp
```

Como resultado, debemos haber generado un fichero llamado `extension.oct`. Se trata de un fichero binario (en realidad, una librería de enlace dinámico) que contiene a la función C++ que hemos definido a través de la macro `DEFUN_DLD`, cual ser automáticamente utilizada desde Octave:

```
1 >> extension
2 Hola Mundo!
```

Para pasar argumentos desde Octave al código C++, deberéis utilizar el segundo parámetro de la macro `DEFUN_DLD` (al que hasta ahora he denominado `args`). Este parámetro es una variable de tipo `octave_value_list`, que define a cualquier lista de valores de octave (es decir a una lista de variables de tipo `octave_value`).

Para devolver valores desde la función hasta octave, deberemos devolver objeto de tipo `octave_value` o bien de tipo `octave_value_list`.

Por ejemplo, definamos la función $f(x) = x^2$:

```
1 #include <octave/oct.h>
2 DEFUN_DLD(xcuadrado, args, , \
3   "Calcula el cuadrado de un real, x'")
4 {
5   if(args.length() != 1) {
6     std::cerr << "ERROR: Se necesita tomar un valor" << std
7       ::endl;
8     return octave_value()
9   }
10  const double x = args(0).double_value();
11  octave_value y = x*x;
12  y = x*x;
13  return y;
}
```

2.2. Ejercicios Propuestos

Realizar una extensión para Octave utilizando código C o C++ que conozcáis o hayáis desarrollado en otras asignaturas, o bien por llamadas a una biblioteca de funciones C/C++ que conozcáis. Este código debe poder ser utilizable desde Octave.

Para ello, deberéis sustituir, en el ejemplo anterior, el código contenido en el cuerpo de la macro `DEFUN_DLD` por aquél código que implementéis.

Información adicional

Consultar el documento “Da Coda Al Fine. Pushing Octave’s Limits”, de Christoph L. Spiel. Disponible en internet, por ejemplo en formato PDF en <http://octave.sourceforge.net/coda/coda.pdf>