

Escritura en modo append

La rutina standard para escribir archivos de datos tiene la siguiente forma:

```
void write_vector(int J, double dx, vector<double>& v, string filename) {
    ofstream salida;
    salida.open(filename.c_str());
    for(int j=0 ; j<J ; j++) {
        double x = j*dx;
        salida << x << " " << v[j] << endl;
    }
    salida.close();
}
```

En este caso, el archivo de salida se abre y se cierra cada vez que llamamos a la rutina `write_vector` (en la segunda línea y en la última, respectivamente). Al abrir el archivo, se borra todo contenido que el archivo tuviera de antes. En un programa en el cual tenemos que llamar a `write_vector` varias veces sobre el mismo archivo, el problema que aparece es que sólo vamos a ver escrito en el archivo de salida el contenido escrito en la última llamada a la rutina.

Dos formas de evitar esto:

- Opción 1) En la instrucción que abre el archivo de salida, podemos indicar que el archivo se abra sin borrar el contenido previo, en lo que se llama el modo de escritura *append*. Esto lo hacemos así:

```
salida.open(filename.c_str(), ios::app);
```

Fíjense que si usamos siempre este modo de escritura, en ninguna parte del programa se borra el contenido del archivo. Esto puede dar lugar a que se mezclen datos de dos corridas distintas. No es difícil modificar el programa para evitar este problema.

- Opción 2) En lugar de abrir el archivo de salida cada vez que ejecutamos `write_vector`, podríamos abrir el archivo en otra parte del código (por ejemplo, la que llame a `write_vector`) y pasarle a `write_vector` (por referencia) la variable que necesita para escribir en el archivo, en lugar de pasarle el nombre de archivo (que ya no necesita).

```
void write_vector(int J, double dx, vector<double>& v, ofstream& salida) {
    for(int j=0 ; j<J ; j++) {
        double x = j*dx;
        salida << x << " " << v[j] << endl;
    }
}
```