



Compilazione separata (I)

La discussione sulle funzioni necessarie per il metodo di Runge Kutta (e sulla classe proposta) fornisce l'opportunità di parlare di compilazione separata. Per un utile riutilizzo delle funzioni che create è conveniente che queste non debbano essere ricomplete ogni volta.

Supponiamo di avere la funzione (o una classe) `func` il cui prototipo è definito in `func.h` e il codice è in `func.cpp`.

Questa può essere compilata una volta per tutte con il comando

```
g++ -c func.cpp
```

che crea il file oggetto `func.o`.

Se a questo punto voglio riutilizzare la funzione `func` nel programma `main.cpp` basterà scrivere

```
#include "func.h"
...
int main(){
    func(...);
}
```

e poi, in fase di compilazione:

```
g++ -o main main.cpp func.o
```



Compilazione separata (II)

È possibile costruirsi una libreria (`libmylib.a`) con i file oggetto con il comando

```
ar libmylib.a oggetto1.o oggetto2.o
```

In compilazione, per utilizzare una funzione presente nella libreria, basterà

```
g++ -o main main.cpp -l mylib
```

Può essere utile mettere il file di libreria in una directory (che chiameremo `DirLib`), in questo caso il comando di compilazione diventerà:

```
g++ -o main main.cpp -L DirLib -l mylib
```

Lo stesso si può fare per gli header files

```
g++ -o main main.cpp -I DirInc -L DirLib -l mylib
```

(gli header files andranno inclusi come `#include<header.h>`, il comando, sostanzialmente, aggiunge la directory `DirInc` alla lista di directory in cui cercare gli headers)