Cancellation points in OpenMP (optional)

Tasks induced by exploratory decomposition can be terminated before finishing as soon as the desired solution is found

- `#pragma omp cancel [parallel | taskgroup]`: this directive activates the cancellation of the enclosing `[parallel | taskgroup]` region. The thread that finds the directive finishes its execution; the other threads continue their execution as normal.
- `#pragma omp cancellation point [parallel | taskgroup]`: introduces a point to check if cancellation has been activated. When found by a thread, if the enclosing `[parallel | taskgroup]` region has been already cancelled, then it finishes its execution.

Cancellation points in OpenMP: very simple example (optional)

```
#pragma omp taskgroup
for (i=0; i<1000; i=i+100)
    #pragma omp task firstprivate(i) private(j)
    {
        for (j=i; j<i+100; j++) {
            if (do_computation(j) == 0) {
                #pragma omp cancel taskgroup
            }
            #pragma omp cancellation point taskgroup
        }
    }
```

The first task with 0 as a result of `do_computation` will finalise the execution of all the tasks in the `taskgroup`