Instructions for developers

If you are a developer and want to use this code check the following guides.

Take into account that for pre- and post-processing you will need to use GenCase and the different post-processing tools included in the main DualSPHysics package, here. If you compile your own DualSPHyiscs version just overwrite the one in the package.

Developing a modified version to fit your own needs.

You can fork this repository and change or add anything you want. Keep in mind that your changes will not be taken into account into the main versions. If your objective is to implement your changes/improvements to the main code, check the next section.

Developing a modified or improved version to contribute to the project.

We appreciate your efforts! But please, if you are trying to develop/implement a functionality to be added to the main repository, be sure to follow the steps described in the CONTRIBUTING.md file.

Building the project

Microsoft Windows

This application is being developed in Visual Studio Community 2015 since it is free and compatible with CUDA 9.2 (download web). The repository contains project files.

Make sure that you install the CUDA SDK beforehand if you want to compile the GPU version, and configure the Visual Studio project to point to the CUDA libraries directory to compile (now prepared for CUDA 9.2).

You can also use CMake. It is possible that you'll need to edit it. Check the Alternative building method via CMAKE section of the wiki for more details.

Compiling DualSPHysics

Orlando Garcia-Feal edited this page on Jun 4, 2018 · 7 revisions

The code can be compiled for either CPU or CPU&GPU. Please note that both the C++ and CUDA version of the code contain the same features and options. Most of the source code is common to CPU and GPU, which allows the code to be run on workstations without a CUDA-enabled GPU, using only the CPU implementation.

To run DualSPHysics on a GPU using an executable, only a Nvidia CUDA-enabled GPU card is needed and the latest version of the GPU driver must be installed. However, to compile the source code, the GPU programming language CUDA and nvcc compiler must be installed on your computer. CUDA Toolkit X.X can be downloaded from Nvidia website http://developer.nvidia.com/cuda-toolkit-XX. CUDA versions from 4.0 till 9.2 have been tested.

Once the C++ compiler (for example gcc) and the CUDA compiler (nvcc) have been installed in your machine, you can download the relevant files from the directory **DualSPHysics_v4.2/src**.

Windows compilation

In **DualSPHysics_v4.2/src** there are also several folders:

- source: contains all the source files;
- lib/vs2013 & lib/vs2015: precompiled libraries for x64 (debug and release);
- VS: includes the project file *DualSPHysics4Re_vs2015.sln* is provided to be opened with Visual Studio Community 2015.

Different configurations can be chosen for compilation: a) Release: for CPU and GPU b) ReleaseCPU: only for CPU

The result of the compilation is the executable *DualSPHysics4.2_win64.exe* or *DualSPHysics4.2CPU_win64.exe* created in **DualSPHysics_v4.2/bin/windows**. The GPU codes were compiled for compute capabilities sm30, sm35, sm50, sm52 sm61, sm70 and with CUDA v9.2.

The Visual Studio project is created including the libraries for OpenMP in the executable. To not include them, user can modify *Props config* -> *C/C++* -> *Language* -> *OpenMp* and compile again. The use of OpenMP can be also deactivated by commenting the code line `#define OMP_USE` in **OmpDefs.h. **
In order to simulate more than 2000 and up to 65000 floating or solid objects, the user should activate or comment the code line `#define CODE_SIZE4` in **Types.h**.