



How to contribute to the DualSPHysics project using the public repository

O. GARCÍA-FEAL, G. FOURTAKAS

DualSPPhysics software



Current Developers:

- Dr José M. Domínguez (jmdominguez@uvigo.es). Universidade de Vigo, Spain
- Dr Georgios Fourtakas (georgios.fourtakas@manchester.ac.uk). The University of Manchester, UK
- Dr Alejandro J.C. Crespo (alexbece@uvigo.es). Universidade de Vigo, Spain
- Dr Benedict D. Rogers (benedict.rogers@manchester.ac.uk). The University of Manchester, UK
- Dr Renato Vacondio (renato.vacondio@unipr.it). Università degli studi di Parma, Italy
- Dr Corrado Altomare (corrado.altomare@upc.edu). Universitat Politècnica de Catalunya – BarcelonaTech, Spain
- Dr Angelo Tafuni (atafuni@njit.edu). New Jersey Institute of Technology, US
- Dr Orlando García Feal (orlando@uvigo.es). Universidade de Vigo, Spain
- Dr Joseph O'Connor (joseph.oconnor@imperial.ac.uk) Imperial College London, UK
- Iván Martínez Estévez (ivan.martinez.estevez@uvigo.es). Universidade de Vigo, Spain
- Professor Peter Stansby (p.k.stansby@manchester.ac.uk). The University of Manchester, UK
- Professor Moncho Gómez Gesteira (mggesteira@uvigo.es). Universidade de Vigo, Spain

COLLABORATIVE PROJECT

FREE OPEN-SOURCE (LGPL)

HIGHLY PARALLELISED

PRE- & POST-PROCESSING

REAL-LIFE PROBLEMS

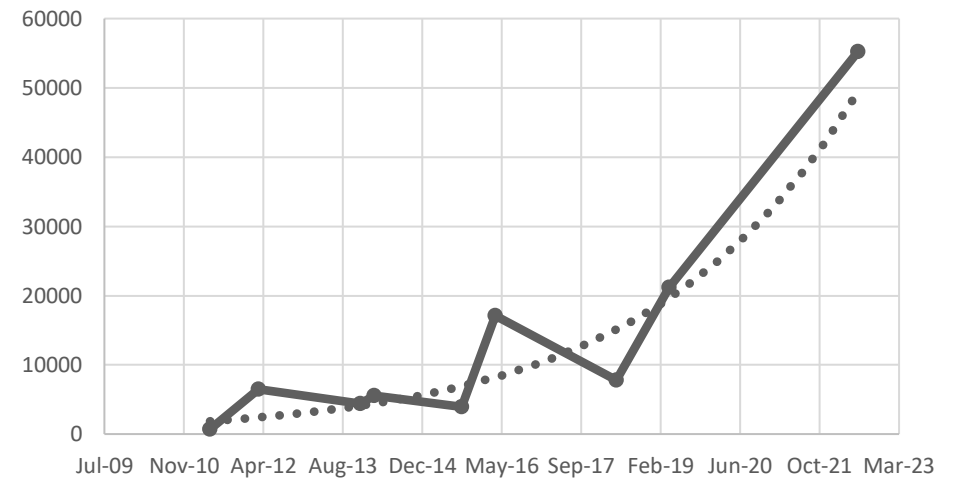
JOURNAL PUBLICATIONS

Current state of DualSPHysics

The DualSPHysics project has grown

- >130k downloads* through the web page
- Dedicated forum with:
 - >10k users
 - 2k discussion topics
- In 2017 a dedicated GitHub project was created which is very active with:
 - 190 forks
 - Many GitHub “Issues” and *pull requests*
 - Active community

DualSPHysics Downloads



- 📄 LGPL-2.1 license
- 📈 Activity
- ★ 509 stars
- 👁️ 56 watching
- 🍴 190 forks

*downloads (local clones) from GitHub are not being tracked

Collaborative development model

Fork and pull model

- The DualSPHysics GitHub project (<https://github.com/DualSPHysics>) contains the public repositories:
 - **DualSPHysics** and **DesignSPHysics**
- DualSPHysics repositories can only be modified by maintainers.
- Anyone can **fork** the repository and modify their personal repository.
- Those changes can be requested to be incorporated into the **upstream** repo (DualSPHysics) by opening a **pull request**.



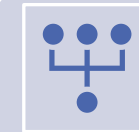
Git – basic concepts



Repository: location that stores all the project files and their revision history.



Clone: create a copy of a repository on your local machine.



Branch: A separate line of development for a specific feature or bug fix.



Commit: create a snapshot of your code modifications.



Push: upload local commits to a remote repository.

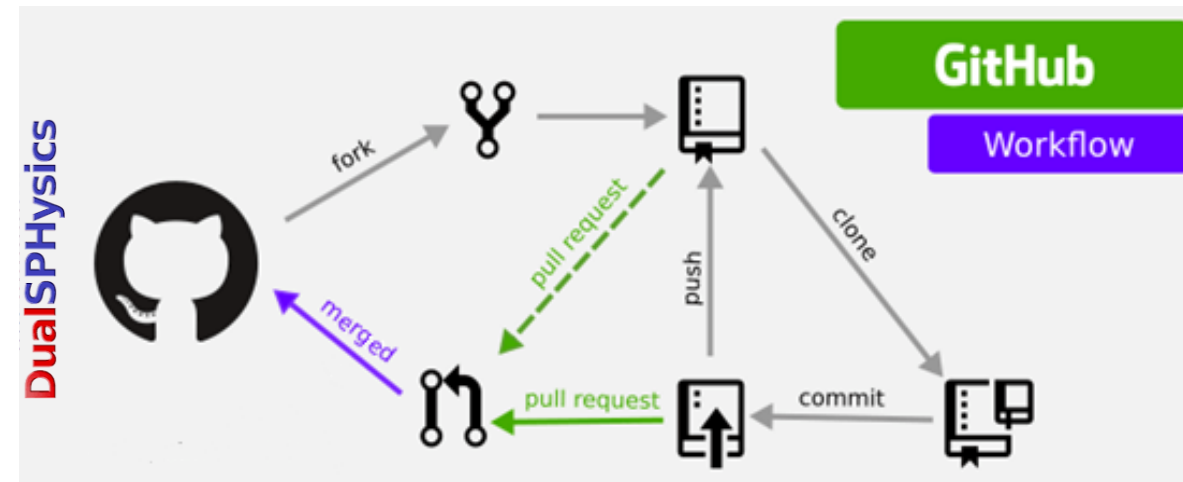


Pull: download the latest changes from the remote repository.

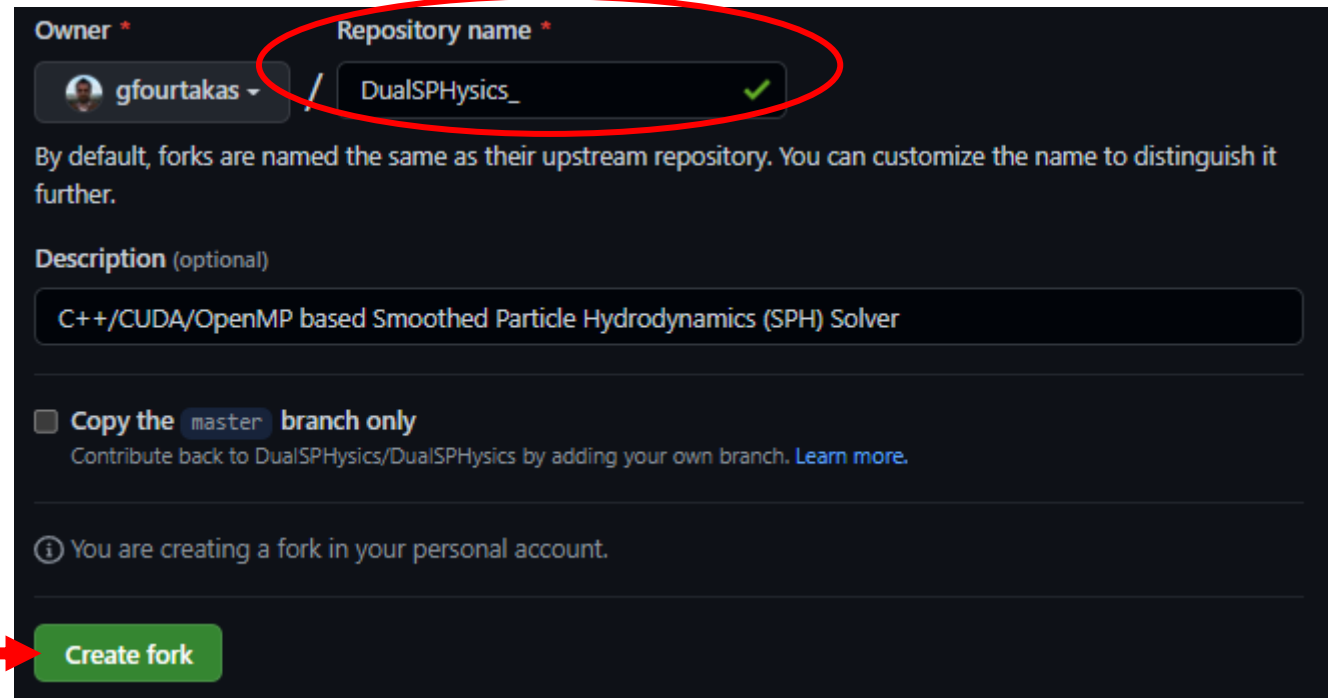
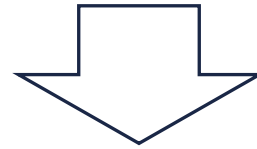
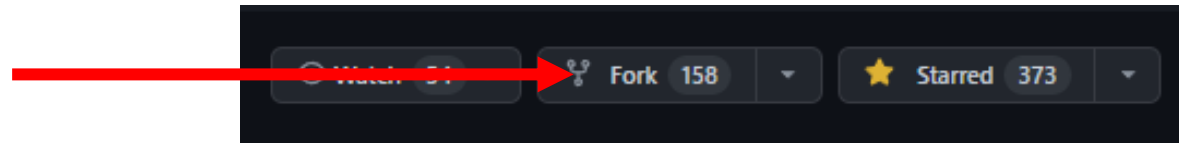
Collaborative development model

About Forks:

- Forking creates a copy the (DualSPHysics) repository into your GitHub account.
- Does not affect the **upstream** repo (DualSPHysics)
- You can **fetch** updates from the upstream repo
- You can use a **pull request** to suggest changes
 - Configure different remotes for the upstream (DualSPHysics) repo and your own (fork) repo.



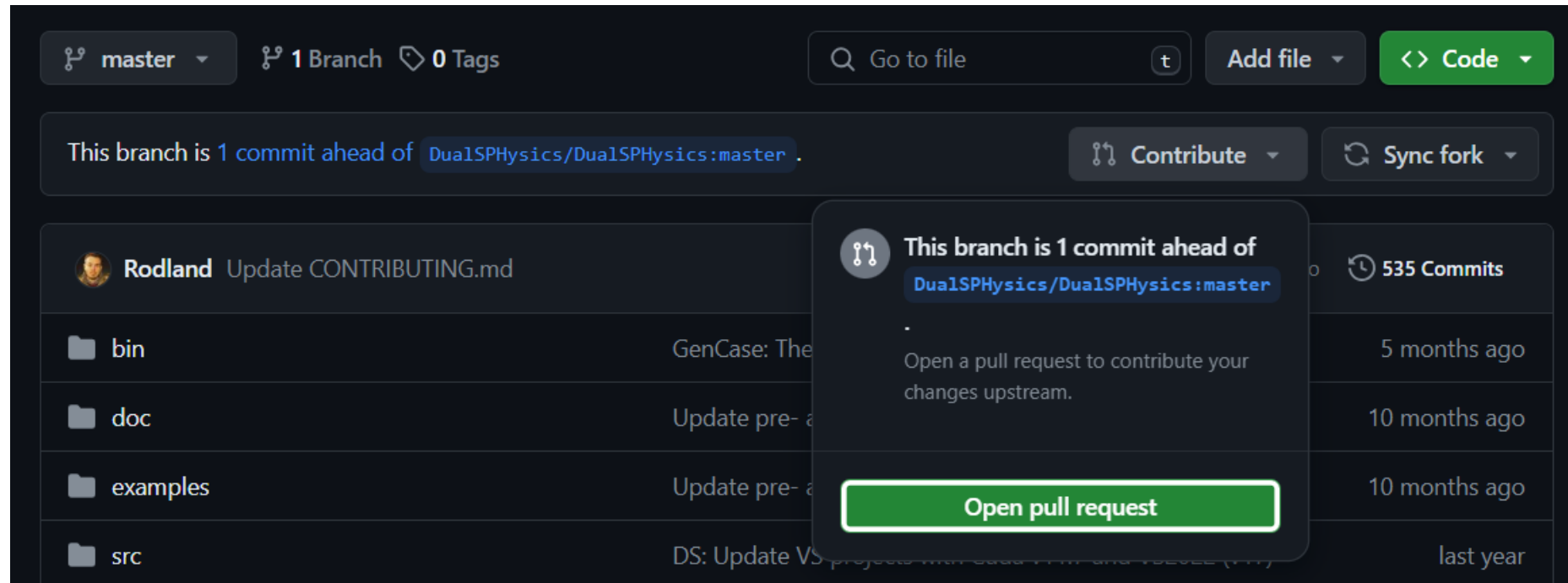
Collaborative development model



Collaborative development model

Creating a pull request from a fork

- We are accepting pull requests on the *develop* and *develop_nn* branch of our repo **only**



Collaborative development model

Creating a pull request from a fork

- We are accepting pull requests on the *develop* and *develop_nn* branch of our repo **only**

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about diff comparisons](#).



base repository: DualSPHysics/DualSPHysics ▾

base: develop ▾



...

head repository: Rodland/DualSPHysics ▾

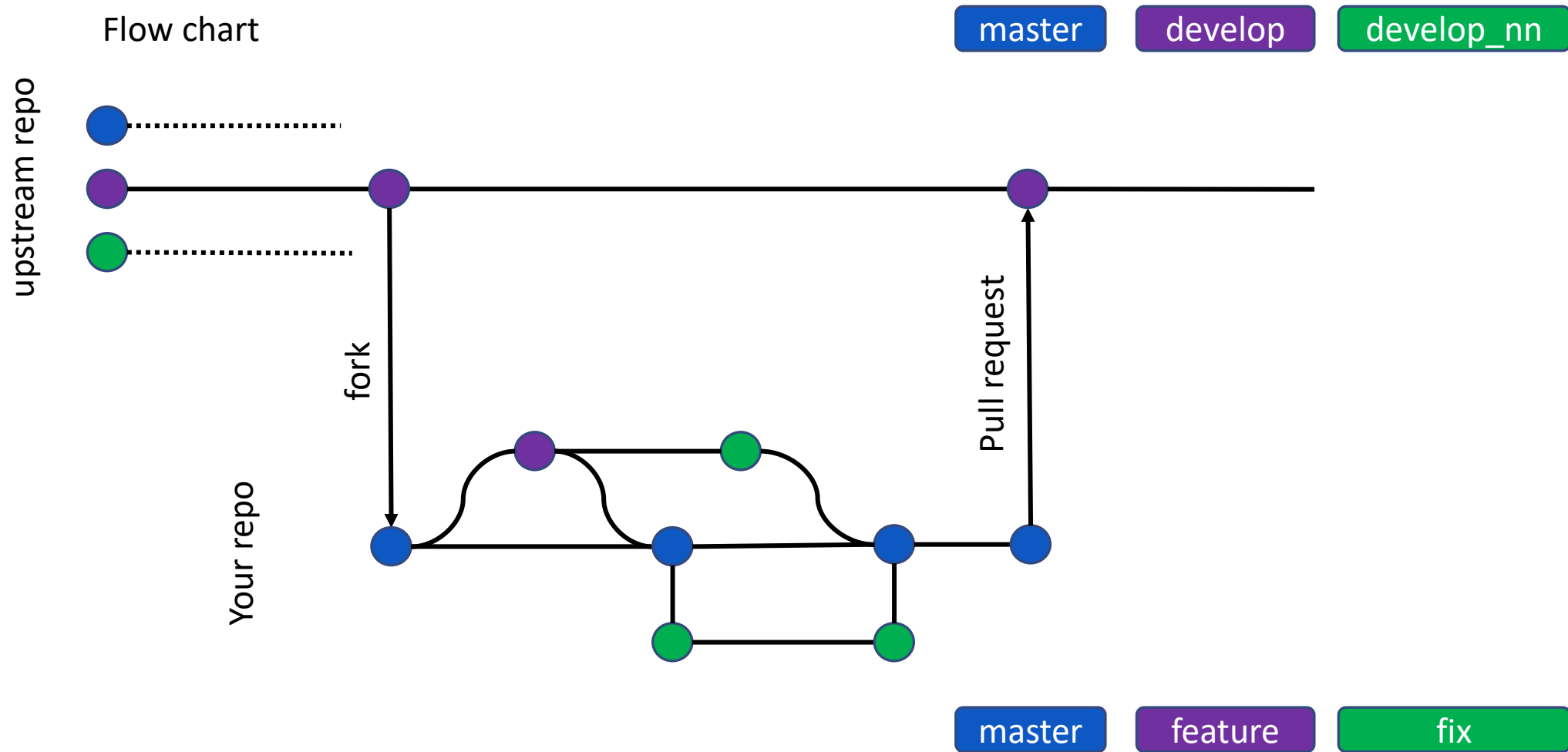
compare: master ▾

✓ **Able to merge.** These branches can be automatically merged.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#)

Create pull request

Contribution to GitHub workflow



Code requirements and tests

Maintainers will review pull requests and communicate with core developers prior accepting or rejecting.

Your implementation **must** conform with:

- Code structure and format (UseOurVariableNames please)
- Configurable XML switches/options (***no hard coding***)
- 2-D and 3-D
- CPU and **GPU**
- **Warnings** for features your modifications **are not compatible with** (see JSph.cpp for examples)

Code requirements and tests

- Full **compatibility** with at least **one fluid solver** (single phase, multiphase, flexstructures, etc)
 - Time stepping: **Verlet & Predictor-Corrector**
 - Wall boundaries: **DBC** and **mDBC**
 - **Moving wall boundaries**
 - **Floating objects**
 - **Density diffusion terms**
 - **Shifting** algorithms

Code requirements and tests

Your pull request **must include**:

- **Tests case(s)** that show how fix/feature are improving the results
 - at folder “*./examples/main/feature*”
 - with a **batch** (including pre- and post-processing) and **xml** file
- **Documentation “*./doc*”**
 - **fix**: a short pdf document highlighting the issue and fix/solution
 - **feature**: a pdf which discusses the computational/numerical advances and implementation (or journal paper), functionality and options (i.e., XML)

Contribution models

Minor contribution - GitHub:

- Small code changes which do not impact on the general structure of the solver (i.e., bug fixes, 1-10 lines of features, etc)

Major contribution - GitHub:

- Code changes are significant and span beyond one file, introduction of new functions and calls, hardware acceleration...
- Major improvements/reformulation of the scheme and/or models which improve or add extra functionality to the solver (i.e., higher accuracy, a new phases, new coupling techniques, etc)
- Normally already published in peer review journals

Contribution as collaborator through private GitLab repo:

- At least one of the core developers must be associated.
- The contributions must be published when merged into a release package
- If you are interested contributing through this model, speak to a developer
- **Requires long term commitment** (usually for PhD student or similar)

Take away message



<https://github.com/DualSPHysics>

- DualSPHysics is an open-source solver with LGPL
- It is a **collaborative project**
- The developers and users pool is increasing continuously
- Our resources are limited
- Community contributions help DualSPHysics to grow even further!

Contribute to the project through our repo!

