Changelog of nse-madec progect

The changelog of nse-madec progect generated from git log. *The table of contents is at the end of the document.*

Initial commit

```
426aa7c6201fb274fdfd69ae92a5fa6a29bc947e
2021-06-12 19:22:57 +0000
Vladimir Onoprienko <vonopr@yandex.ru>
```

Files changed:

```
 README.md | 2 ++
 1 file changed, 2 insertions(+)
```

Create a minimal CMakeLists.txt

```
e3a6243b6866a042c6a56d60bd177ada4957f77d
2021-07-13 19:49:03 +0300
Vladimir Onoprienko <vonopr@ya.ru>
```

A minimal working CMakeLists.txt delcares a name of the project, compilers' languages and a minimal cmake version possible to process the CMakeLists.txt.

The choice of a minimal version of cmake was quite random. It is quite new to support most needed features, and quite old (about 3 years) to be possibly installed on a user's machine.

Hence this commit.

Files changed:

```
 CMakeLists.txt | 3 +++
 1 file changed, 3 insertions(+)
```

Enable CI/CD

```
888939db09c9ed80f1e9cc71720dc0e2889b75b8
2021-07-14 12:03:55 +0000
Vladimir Onoprienko <vonopr@yandex.ru>
```

I'd like to start CI/CD, and the only thing we can check at this moment is if cmake works correctly.

It is better to build the code in a clean directory to avoid potential interference with the project's files. I'd prefer to see build directory outside the repo tree. Like that: . ├── build └── nse-madec └── README.md

But that seems to deteriorate the gitlab's CI/CD scenario. As the automatically cloned project's repo is the root directory for the CI/CD scripts. You CAN create a build directory on the same level with the project's repo directory, but that is discouraged due to possible interference with other jobs of the same runner. There are no guarantees it will be cleaned out after the job is executed.

I think the best compromise is to create a clean build directory inside the project's directory: nse-madec ├── build └── README.md

Files changed:

```
 .gitlab-ci.yml | 4 ++++
 1 file changed, 4 insertions(+)
```

Add build instructions to download needed sources

```
fc02e4d73882ecfddd2117eb9669537d2d758d3c
2021-07-16 11:27:29 +0300
Vladimir Onoprienko <vonopr@ya.ru>
```

Currently we want a project that downloads source files from other NSE projects, builds them and the resulted executable should be able to run advection experiments on the sphere.

The desired sequence of actions to be perfomed with the project is: download -> build -> run This commit deals with the download part.

The downloading will be perfomed with cmake. As it is convenient to have one tool both for download and build steps. Meanwhile, the building capabilities of cmake for C++ projects seem to be supreious over current analogs.

The list of files to be downloaded was taken from the analagous project nse-sphere2d-adv, namely from http://tesla.parallel.ru/emortikov/nse-

sphere2d-adv/-/blob/ a617c882c0404d3ff1e1c1c7ea8dd576d8019933/nse-sphere2d-adv/cpall.sh file

Things that should be considered during this commit:

1) Why multiple *.cmake files are better than one? It is desired that used NSE projects could be perceived with a glance at the projects' directory tree. That is especially useful as multiple analagous projects are expected to come.

There should be an nse-*.cmake file per downloaded nse project.

2) Why downloading external projects via ssh, not http? It is faster for a skilled user. He is required to register an ssh-key and forget about logins. Otherwise, when using http entering username/password pair would be nessassary at every rebuild that would be really annoying. Providing them with cmake config options is unsecure.

In future, to lower down the requirement to the user's bash and git skills, one could add the possibility just to point to the preliminary downloaded external projects.

3) Why not displaying download progress? Well, this feature is desired. It would be very convient to see that download progress to be sure that it was perfomed.

We are going to use standard cmake tool to download external git project at configuration step – 'FetchContent'. It is convinient but the only way to force it to show the download progree is to set FETCHCONTENT_QUIET to FALSE. And that enables printing too much information making the output unsatisfactorily noisy.

So, the lesser evil is to keep it silent.

Files changed:

```
CMakeLists.txt              |  3 +++
deps/CMakeLists.txt         |  3 +++
deps/nse-sphere2d-adv.cmake | 18 ++++++++++++++++++
deps/nselibx-common.cmake   | 20 ++++++++++++++++++++
deps/nselibx-sphgrid.cmake  | 16 ++++++++++++++++
5 files changed, 60 insertions(+)
```

Enable status messages for dependencies' download

```
ceac54426f7ec118f7e37d3e7821e9915901af61
2021-07-16 14:45:51 +0300
Vladimir Onoprienko <vonopr@yandex.ru>
```

If the external projects were downloaded/updated the success message should be produced. The user must be informed about external dependencies during configuration step.

As the currently downloaded projects are small it is better not to show the progress bar.

This commit realizes consideration #3 of the previous commit message (fc02e4d73882ecfddd2117eb9669537d2d758d3c).

Formatting considerations:

1) include() must hit the eye

The main lines are include() commands. But they are barely seen if they are tightly surrounded with status messages. Thus, I add one empty line before and after each inlude() command.

2) download sections for each project must be easily distinguished

Thus, I add quite lot of empty lines between project's parts. The reader should see three chunks at first glance.

Files changed:

```
 deps/CMakeLists.txt | 36 ++++++++++++++++++++++++++++++++++++
 1 file changed, 36 insertions(+)
```

Add cmake instructions to build the project

```
d531f83e122a66af1b071262de18d014ca15f019
2021-07-16 15:23:26 +0300
Vladimir Onoprienko <vonopr@yandex.ru>
```

After downloading of source files was enabled we are able to compile them and produce the executable.

The downloaded files are copied into ${build_dir}/src at the configuration step. The build process is trivial for cmake.

The packages MPI and OpenMP are required for the project to be built.

Files changed:

```
 CMakeLists.txt     |  6 ++++++
 src/CMakeLists.txt | 13 ++++++++++++
 2 files changed, 19 insertions(+)
```

Enable source compilation to CI/CD pipeline

```
7443f55c3ed387ae2ce3dde9c7ef236847232b99
2021-07-19 13:16:30 +0300
Vladimir Onoprienko <vonopr@yandex.ru>
```

The project has cmake building instruction. We should check their consistancy with gitlab's CI/CD pipeline.

Files changed:

```
 .gitlab-ci.yml | 1 +
 1 file changed, 1 insertion(+)
```

Add compatibility with GNU 4.8.5 compiler

```
ffa1a1216f0e0c8a1187bcb03833d0a952293b9c
2021-07-19 13:35:07 +0300
Vladimir Onoprienko <vonopr@yandex.ru>
```

The CI/CD pipeline of the previous commit 7443f55c3ed387ae2ce3dde9c7ef236847232b99 was broken due to the old version of the used compiler (GNU 4.8.5).

Adding the compiler option "-std=c++0x" fixes the problem.

Files changed:

```
 src/CMakeLists.txt | 1 +
 1 file changed, 1 insertion(+)
```

Add list of dependencies to README.md

```
4d34c8ae2cb166ad64f48bf2fba5cf67d42e6915
2021-07-19 13:52:23 +0300
Vladimir Onoprienko <vonopr@yandex.ru>
```

After we started to actually compile the code during CI/CD pipline we should provide the list of dependencies in the project's README.md

Currently two software packages are used: C++ GNU compiler and CMake.

The oldest tested version of C++ compiler is GNU 4.8.5, see the preious commit 8e7158d8a63592fd4ee1cf3c28f5ac11cd43f9c2.

The minimal cmake version is 3.12 is specified in the project's CMakeLists.txt.

Files changed:

```
README.md | 5 +++++
1 file changed, 5 insertions(+)
```

Add CMake installtion instructions

```
70904135cf9232d92538fdd8b71f9f7ca3e983cb
2021-07-19 14:15:58 +0300
Vladimir Onoprienko <vonopr@yandex.ru>
```

After the project's executable is built we should be able to install it.

Alongside with the exec we should install the library. It is useless when static linking is applied. But if the user perfoms a dynamic linking the executable will need the library when launched.

Files changed:

```
src/CMakeLists.txt | 3 +++
1 file changed, 3 insertions(+)
```

Enable installation in CI/CD pipeline

```
f3b39dbdf0501f6cde49228f2f1be88872a45e23
2021-07-19 14:26:39 +0300
Vladimir Onoprienko <vonopr@yandex.ru>
```

We should include installtion check into CI/CD pipeline.

The executable file does not work wihtout the configuration file. As we do not provide the config file yet we can't check the executable's operability. We just check that it is sucessfully copied into the proper place.

Install job will rely on the results of the build job. To order the jobs corectly we register stages. The build stage should provide the built files to install job using the artifacts mechanism.

Files changed:

```
.gitlab-ci.yml | 19 ++++++++++++++++++-
1 file changed, 18 insertions(+), 1 deletion(-)
```

Add configuration file for the executable

```
ee00799a3e343ff5b83d9e5de9797c755609a789
```

```
2021-07-19 17:38:13 +0300
Vladimir Onoprienko <vonopr@yandex.ru>
```

The proper configuraiton file can be downloaded from the project “nse-sphere2d-adv”.

The file’s URL is too long and is splitted into two lines:

http://tesla.parallel.ru/emortikov/nse-sphere2d-adv/-/blob/ a617c882c0404d3ff1e1c1c7ea8dd576d8019933/sphere2d-adv/config-ex.txt

To launch the project’s executable rename the copy the config file into the running directory. The config’s filename must be “config.txt”. Rename if needed.

Files changed:

```
 configs/config.txt | 142 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 142 insertions(+)
```

Add installation instructions for exec’s config

```
55d28e797c6524a602d7c9bf3971c45ae8421539
2021-07-21 09:55:33 +0300
Vladimir Onoprienko <vonopr@yandex.ru>
```

There should be two copies of “config.txt” in the installation directory: in the project’s root and in the config directory.

The project’s exec is supposed to be launched from the project’s root directory. If the config is copied there the exec can be successfully launched right after the installation.

The user may want to restore the original config after some changes he has made. It is handy to have the copy of the original config in the project’s directory.

Files changed:

```
 CMakeLists.txt         | 3 +--
 configs/CMakeLists.txt | 2 ++
 2 files changed, 3 insertions(+), 2 deletions(-)
```

Enable CI/CD check if exec works

```
35adbbd6823f6978bd309c69a0e9550551fa40fa
2021-07-21 10:11:47 +0300
Vladimir Onoprienko <vonopr@yandex.ru>
```

Files changed:

```
 .gitlab-ci.yml | 12 ++++++++++++
 1 file changed, 12 insertions(+)
```

Adjust config.txt for CI/CD runs

```
4732eb248108c1857161b999147929520d6b74bf
2021-07-21 10:47:02 +0300
Vladimir Onoprienko <vonopr@yandex.ru>
```

The CI/CD job runs the executable with the default configuartion from "config.txt". It produces too much text output.

Also it uses quite a high resoluation by default: 360×180, that is 1°×1° degree. That makes a job run for an unnessessary long time: almost 3 minutes.

I think we can reduce the default frequency of the output from 1 hour to 24 hours. The grid resolution can be reduced to 90×90 (4°×2°). That is still a reasonable resolution and the calculation time is expected to reduce from 3 minutes to 20-30 seconds.

Files changed:

```
 configs/config.txt | 8 ++++----
 1 file changed, 4 insertions(+), 4 deletions(-)
```

Add descriptions of how to install into README.md

```
868c453cc5d3c4290d4ec537a6626bfe1f8d429f
2021-07-21 12:09:53 +0300
Vladimir Onoprienko <vonopr@yandex.ru>
```

After we enabled checking the project's download, build, installation and run during CI/CD pipelines we should write down the instructions into the project's README.md

Note that "git clone" downloads a default branch. Thus the instructions from this commit won't be fully operational till this commit is merged into the default branch.

That is done intentionally to keep README.md simple.

Files changed:

```
 README.md | 39 +++++++++++++++++++++++++++++++++++++++
 1 file changed, 39 insertions(+)
```

Add a python tool for reading dsq timeseries

```
c5b4607699c4860e6b5a235d97402d94ad062df8
2021-07-22 10:42:01 +0300
Vladimir Onoprienko <vonopr@yandex.ru>
```

Some of the experiment's results are stored as 1D timeseries in a special "dsq" format. It was developed for nse projects.

We need a python tool that deals with reading dsq files.

Files changed:

```
 tools/read_dsq.py | 55 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 55 insertions(+)
```

Add a python summary generator for dsq timeseries

```
789437609ca8b861ceb89116555336122ebee866
2021-07-22 10:48:31 +0300
Vladimir Onoprienko <vonopr@yandex.ru>
```

The quickest way to get what the timeseries were about is to read a small summary table.

This commit provide a tool to generate this summary table.

Files changed:

```
 tools/read_dsq.py | 23 +++++++++++++++++++++
 1 file changed, 23 insertions(+)
```

Add a python exp's summary pusher to git server

```
74214ae81b704f6ab4441d66edeeedd33e4a4c79
2021-07-22 11:06:58 +0300
Vladimir Onoprienko <vonopr@yandex.ru>
```

After changes in the source files the output of the experiment may also change.

I think it is useful to have a short summary of changes of output of the default experiment in git server after every commit.

This commit realizes a kind of git-bot that writes such comments on a git server during CI/CD pipelines. It requires that the ACCESS_TOKEN environment variable is defined for the gitlab-runner. This can be adjusted in the project's settings.

Files changed:

```
 tools/gitlab-api.py | 17 ++++++++++++++++++
 1 file changed, 17 insertions(+)
```

Register a git-server bot

```
2d812dab75cba40e432fa70eeda210a409a0ad51
2021-07-22 11:26:58 +0300
Vladimir Onoprienko <vonopr@yandex.ru>
```

A git-server bot will send comments to git commits on the server containing short summary results of the default experiment's output.

See the previous commit ed539d54d06e7445a038df51d6239647b03f3c0b

Files changed:

```
 .gitlab-ci.yml | 9 +++++++++
 1 file changed, 9 insertions(+)
```

Add a python reader of 2D output files

```
c050f8a021c78d01912fd862c67f6d3cc685a024
2021-07-22 13:02:08 +0300
Vladimir Onoprienko <vonopr@yandex.ru>
```

The experiments produce 2D data output. The files are written in the tecplot's plt format.

This commit introduces a python script that reads plt files generated by the project.

Note that it cannot read all the files allowed by tecplot's standard. At least, it recognizes files generated by the default project's experiment specified with configs/config.txt file.

Files changed:

```
 tools/read_plt.py | 81 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 81 insertions(+)
```

Add a python script for plotting model's 2D output

```
0a2f8034448cb3887f8e768a2078784c63cd4199
2021-07-22 14:43:18 +0300
Vladimir Onoprienko <vonopr@yandex.ru>
```

The script points to the output directory and generates png images from .plt files into pics directory inside the OUTPUT directory.

The intuitive behaviour would be pointing to a single.plt file (or a list of them) to produce an image inside the current directory.

I prefer this a bit counter-intuitive way to simplify the experiment's post-processing. The output directory is thought to be the atomisitic container of all single experiment's artifacts.

In this sence the script continues with the experiment's workflow. Hence, the chosen behavior.

Files changed:

```
 tools/plot.py | 88 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 88 insertions(+)
```

Add a python tool for plotting 1D timeseries

```
e1c2de036c1e665f32998d224cd3a5a61fd0e76e
2021-07-22 15:36:10 +0300
Vladimir Onoprienko <vonopr@yandex.ru>
```

The script reads the "misc.dsq" file inside the output variable and writes produce the image into the pics directory inside the OUTPUT directory, not inside the working directory. See commit 0a2f8034448cb3887f8e768a2078784c63cd4199 for more explanations.

The variables "c-integral", "c-integral-error" contained in "misc.dsq" are not plotted. Only L1, L2 and L∞ errors are plotted. The variables sharing the same plot should be homogeneous: they must have same units and similar values. We have two groups of variables with the same units.

I see no reason in plotting "c-integral", "c-integral-error" during every experiment's processing routine as they are not informative as timeseries: during such a short 1 week experiment they are almost constant.

Actually, indication of how constant these variable are might be helpful in dsq's summary. Consider adding the column with initial values to the summary table.

Files changed:

```
 tools/plot.py | 22 ++++++++++++++++++++++
 1 file changed, 22 insertions(+)
```

Add installation instructions for python tools

```
9ad9fe35934378e0ecb5f61d6ee51f8032fdad18
2021-07-27 12:40:11 +0300
Vladimir Onoprienko <vonopr@yandex.ru>
```

Files changed:

```
 CMakeLists.txt       | 1 +
 tools/CMakeLists.txt | 1 +
 2 files changed, 2 insertions(+)
```

Enable CI/CD check of experiment's post-processing

```
54052016ff248c48cf9ea1b70168a0acc4df490f
2021-07-27 13:15:02 +0300
Vladimir Onoprienko <vonopr@yandex.ru>
```

We should check if the pictures based on the experiment's output are generated with the project's python tools.

Files changed:

```
 .gitlab-ci.yml | 9 +++++++++
 1 file changed, 9 insertions(+)
```

Contents