

# PSIDE – Parallel Software for Implicit Differential Equations

## Basic usage in GNU/Linux

---

## 1 Forewords

This document provides a short usage instructions of compiling and running a test problem with PSIDE. The suitable operational system where tests were performed was GNU/Linux. In the Table below I display the test environmental configurations.

|                         |   |
|-------------------------|---|
| Processor               | Intel® Core™ i5-7200U CPU @ 2.50GHz × 4 |
| RAM                     | 7,7 GiB DDR3                            |
| Operational System (OS) | Ubuntu 16.04 LTS                        |
| OS type                 | 64-bits                                 |

Table 1: Computer configurations.

The source files employed here were altered compared with the original ones. The directory **original** contains the latter, while **src** contains the ones which are under improvements, such as implementing modern parallelization techniques. The current changes concern in updating OpenMP directives. Further improvements will be carry out soon. The present state only adds new compilers optimization flags.

## 2 Compiling

The user can choose between some options. Currently, the available compilers are:

- GNU Compiler **gfortran**. Compilation can be achieved with the following script execution:

```
./compile_gfortran.sh test_problem.f
```

for actual source files and

```
./compile_original_gfortran.sh test_problem.f
```

for original PSIDE source file implementation. The **test\_problem.f** denotes the Fortran 77 file which the problem is described (ex.: **hires.f**).

- Intel Compiler **ifort** or PGI Compiler **pgfortran**: Analogously we have

```
./compile_###.sh test_problem.f flag
```

for actual source files and

```
./compile_original_###.sh test_problem.f flag
```

for the original source files. Here, **###** can be **intel** or **pgi** and **flag** denotes the sign of compiler optimization tools. If the user wants to use compiler's optimization, just have to place **1** where is **flag**. This argument is optional. If not provided, the compilation will be carried out as default.

A simple usage example is

```
./compile_intel.sh problems/Beam/beam.f 1
```

The output obtained was

```
Compiler version check:
+++ Version:

pgfortran 17.10-0 64-bit target on x86-64 Linux -tp haswell
PGI Compilers and Tools
Copyright (c) 2017, NVIDIA CORPORATION. All rights reserved.

+++ Status: OK

Clearing binaries:
+++ Status: OK
Compiling modules:
+++ Initializing...
    pgfortran -c -fast ../src/pside.f
    pgfortran -c -fast ../src/psidea.f
    pgfortran -c -fast ../src/report.f
    pgfortran -c -fast ../src/psided.f
+++ Linking...
    pgfortran -fast -o ../dotest ../problems/Beam/beam.f psided.o pside.o psidea.o report.o
+++ Status: OK
```

### 3 Running

Running PSIDE is straightforward due to gathering of test problem to the compiled solver is achieved by compilation script described in the previous section. Therefore, running PSIDE consists in executing the following command

```
./run_pside.sh
```

This script accepts a positive integer argument as `OMP_NUM_THREADS` setting up environmental variable manually, but this improvement is currently under development.

**Remark:** The tolerances reading during run-time were suppressed. This modification were employed aiming obtaining a more clear time execution record, suitable to realization of performances comparative studies. The values attributed are  $10^{-10}$  to both relative and absolute errors.