

Modelowanie matematyczne w nauce i technice

LAB 01

1. Wstęp

Celem zajęć jest wprowadzenie do obsługi zaawansowanych pakietów obliczeń naukowo-technicznych na przykładzie projektu "**ModFEM**" (ModFEM pozwala na obliczenia metodą elementów skończonych na klastrach z akceleratorami w postaci kart graficznych, podczas zajęć wykorzystywany będzie pojedynczy serwer "**HONORATA**"). W trakcie zajęć przypomniane zostaną zagadnienia związane z obsługą Linuksa z linii komend oraz kompilacja kodu źródłowego z wykorzystaniem skryptów narzędzia '**cmake**'. Na zakończenie uruchomiony zostanie program '**ModFEM**'.

2. Zadanie 1 [wykonywane lokalnie]

Proszę zapoznać się z podstawowymi komendami w systemie Linux, tj. **mkdir**, **cp**, **mv**, **rm** (w razie potrzeby proszę skorzystać z dokumentacji: **man komenda**) i wykonać poniższe polecenia [załączniki dla wszystkich poleceń powinny się znaleźć w sprawozdaniu]:

1. Utworzyć poniższą strukturę katalogów **testx** (można to zrobić jednym poleceniem **mkdir** z opcją **-p**), a następnie dodać pliki **x.c** (np. poprzez **touch**):

```
test1--|
      |--test2--|--plik1.c
      |      |--plik2.c
      |--test3--|
            |--plik3.c
```

2. Przenieść '**plik3.c**' do katalogu '**test1**'

3. Zadanie 2 [wykonywane lokalnie oraz na serwerze]

- Proszę zalogować się na indywidualne konto, wykorzystując protokół: **SSH** (*ssh konto@adres_ip*), wzór nazwy konta to:

'nazwisko_imie@10.156.112.164'

- Proszę utworzyć katalog o nazwie '**lab_01**' (w nazwach katalogów i plików proszę nie stosować polskich znaków)
- Proszę zmienić hasło dostępu poleceniem:

'passwd'

4. Zadanie 3 [wykonywane lokalnie oraz na serwerze]

Proszę przekopiować utworzoną strukturę plików na komputer '**Honorata**' do utworzonego wcześniej katalogu '**lab_01**', w tym celu należy wykorzystać polecenie '**scp**' z opcją '**-r**', szczegóły (*man scp*). (Standardowo polecenie **scp** wywoływane jest z komputera lokalnego - tak do kopiowania na serwer Honorata, jak i z serwera).

Przykładowe użycia polecenia '**scp**':

```
scp -r nazwisko_imie@10.156.112.164:lab_01/katalog1 ~/.
```

Polecenie kopiuje z komputera o adresie IP: **10.156.112.164** z konta **nazwisko_imie** katalog '**lab_01/katalog1**' do głównego katalogu użytkownika na komputer gdzie użytkownik jest aktualnie zalogowany.

```
scp -r katalog1 nazwisko_imie@10.156.112.164:lab_01
```

Polecenie kopiuje z lokalnego komputera katalog **katalog1** (będący podkatalogiem katalogu z którego wywoływane jest polecenie **scp**) na komputer o adresie IP: **10.156.112.164** do podkatalogu **lab_01** katalogu domowego użytkownika **nazwisko_imie**

W trakcie realizacji laboratoriów zamiast kopiowania całych katalogów wygodniejsze jest dokonanie najpierw kompresji wybranych plików (jeśli argumentem jest katalog dodawane z katalogu do archiwum są rekurencyjnie wszystkie pliki, podkatalogi, pliki w podkatalogach, podkatalogi podkatalogów itd.), np.:

```
tar cvzf pliki.tgz plik1 * plik2_a katalog_1 katalog_2* plik*.dat
```

a następnie skopiowanie skompresowanego pliku z serwera lub na serwer

```
scp nazwisko_imie@10.156.112.164:lab_01/pliki.tgz .
```

```
scp pliki.tgz nazwisko_imie@10.156.112.164:lab_01
```

(rozpakowanie dokonywane jest przez polecenie: **tar xvzf pliki.tgz**)

5. Zadanie 4 [wykonywane na serwerze]

Proszę do utworzonego na serwerze katalogu **~/ModFEM/** przekopiować plik **„~/home/students_mm/ModFEM/ModFEM_src_2023.tgz”**, a następnie rozpakować go za pomocą **tar**:

```
(tar xvzf ModFEM_src_2023.tgz)
```

1. katalog **~/ModFEM/** powinien pozostać katalogiem, w którym przez wszystkie zajęcia przechowujecie Państwo kod źródłowy programu, w którym dokonujecie modyfikacji kodu oraz jego kompilacji i rekompilacji
2. **wszystkie ścieżki dotyczące programu ModFEM w dalszej części opisu zadań są traktowane jako podkatalogi względem ~/ModFEM/**
3. w przypadku pełnej rekompilacji kodu z nowych dostarczanych źródeł, należy stary katalog **~/ModFEM/** przemianować na **~/ModFEM_vx** (**x = 1,2,3** itd., w katalogu tym mogą znajdować się zmodyfikowane prywatne pliki, których nie należy się pozbywać) i nową paczkę rozpakować ponownie do katalogu **~/ModFEM/**

(UWAGA: w dalszej części tematu w miejsce *imie_nazwisko* każdy podstawia własne dane)

6. Zadanie 5 [wykonywane na serwerze]

Proszę przygotować plik platformy

```
'src/cmake/Platforms/imie_nazwisko.cmake',
```

wykorzystując plik konfiguracyjny:

```
'REFERENCE_PLATFORM.cmake'
```

i zaczynając od przekopiowania:

```
cp REFERENCE_PLATFORM.cmake imie_nazwisko.cmake
```

Edytowany plik zawiera parametry konfiguracji programu ModFEM (kompilatory, opcje, biblioteki itp.). Należy starać się zrozumieć jak najwięcej z wprowadzanych zmian – w przypadku wątpliwości (np. do czego służy dana opcja, biblioteka itp.) pytać prowadzących.

Edycji własnego pliku `imie_nazwisko.cmake` można dokonywać dowolnym edytorem: `nano`, `vi`, `emacs` lub innym – najlepiej używać edytor, który zapewnia podświetlanie składni kodu źródłowego w C (w wersji na Honorata: `vi`, `emacs`, `nano`).

Kluczowe zmienne do ustalenia w pliku konfiguracyjnym dla komputera 'Honorata' to (kolorem zielonym zmiany do wprowadzenia, czerwonym do usunięcia, niebieskim kluczowe komentarze):

```
#####  
#                               ModFEM                               #  
#                               imie_nazwisko_grupa_lab01           #  
#####  
  
message("")  
message("*** Entering platform file with mkl and debug options ***")  
message("")  
  
# select DEBUG or RELEASE version (see options below)  
#set(CMAKE_BUILD_TYPE Debug)  
#set(CMAKE_BUILD_TYPE Release) // należy wybrać (odkomentować) opcję Release!  
  
# ----- User compiler flags ----- #  
if(CMAKE_BUILD_TYPE STREQUAL "Release") #Flags for release mode  
  set(USER_CMAKE_C_FLAGS "PUT_YOUR_FLAGS_HERE") #C compiler flags  
  set(USER_CMAKE_CXX_FLAGS "PUT_YOUR_FLAGS_HERE") #C++ compiler flags  
  set(USER_CMAKE_EXE_LINKER_FLAGS "-m64") #Linker flags  
elseif(CMAKE_BUILD_TYPE STREQUAL "Debug") #Flags for debug mode  
  
  # Debug flag info  
  # Warning flags: https://gcc.gnu.org/onlinedocs/gcc/Warning-Options.html#Warning-Options  
  # Debug flags: https://gcc.gnu.org/onlinedocs/gcc/Instrumentation-Options.html  
  # Program instrumentation options:  
https://gcc.gnu.org/onlinedocs/gcc/Instrumentation-Options.html  
  
  set(USER_CMAKE_C_FLAGS "-O0 -g") #C compiler flags  
  set(USER_CMAKE_CXX_FLAGS "-O0 -g") #C++ compiler flags  
  set(USER_CMAKE_EXE_LINKER_FLAGS "-O0 -g -m64") #Linker flags  
  
else() #Flags for other modes  
  set(USER_CMAKE_C_FLAGS "PUT_YOUR_FLAGS_HERE") #C compiler flags  
  set(USER_CMAKE_CXX_FLAGS "PUT_YOUR_FLAGS_HERE") #C++ compiler flags  
  set(USER_CMAKE_EXE_LINKER_FLAGS "-m64") #Linker flags  
endif()  
  
#Time measure  
#set(TIME_TEST TRUE)
```

```

set(TIME_TEST FALSE)

# ----- Libraries linking ----- #
# Choose TRUE to build static libraries (archives) and
# statically linked executables (slower);
# FALSE otherwise (faster)
set(MODFEM_USE_STATIC TRUE) #Set TRUE or FALSE

# ----- NEW MPI mmpd_adapter ----- #
#Choose to use generic mmpd_adapter (TRUE) or previous mmpd_prism (FALSE)
set(MODFEM_NEW_MPI FALSE) #Set TRUE or FALSE

# ----- ModFEM Solver ----- #
# Choose solver library used by executables
# Valid options are:
# - sil_mkb [iterative solver]
# - sil_lapack [direct solver]
# Note: For sid_pardiso option MODFEM_USE_MKL must be set to TRUE
# Note: Due to cyclic dependency in sid_krylow_bliter MODFEM_USE_STATIC
#       must be set to TRUE for that solver
set(MODFEM_ITER_SOLVER_MODULE sil_mkb)
set(MODFEM_DIRECT_SOLVER_MODULE sil_lapack)

# LSD MKB Extensions - default SuperLu
# Valid options are:
# - PARDISO // PARDISO nie PARADISO!
# - MUMPS
# - VIENNACL
# - SUPERLU
set(MODFEM_MKB_DIRECT_SOLVER_MODULE PARDISO)
#MKB Pardiso extensions, MKL needed, // PARDISO nie PARADISO!

# AMG (algebraic multigrid)
# Enable AMG for NS problem, module use PETSC
set(MODFEM_USE_PETSC FALSE)
# PETSC directory and architecture
#set(PETSC_HOME SELECT_YOUR_PETSC_DIRECTORY)
#set(PETSC_ARCH linux-openmpi) # Available architectures: linux-openmpi /
linux_mkl_openmpi

# Matrix print TOOL - ważne dla lab 8
set(ENABLE_MATRIX_PRINT FALSE)

# Mixed approximation matrix storage type
# ONLY ONE option could be enabled:
# MIXED_DOF_BY_DOF_STORAGE - store values from both fields together by dof
number
# MIXED_FIELD_BY_FIELD_STORAGE - first store values primary field after that
from second field
set(MIXED_DOF_BY_DOF_STORAGE TRUE) # DEFAULT option
set(MIXED_FIELD_BY_FIELD_STORAGE FALSE)

# Renumbering

```

```

# ONLY ONE option could be enabled:
# ENABLE_RENUMBERING - default renumbering
# ENABLE_INTERNAL_RENUMBERING - other renumbering algorithms
set(ENABLE_RENUMBERING TRUE) # DEFAULT option - ważne dla lab 8
set(ENABLE_INTERNAL_RENUMBERING FALSE)

# Save constraint values in dump field file
# ENABLE ONLY IF YOU KNOW WHAT ARE YOU DOING
set(SAVE_IN_DUMP_CONSTRAINT_VALUES FALSE)

# ----- Algebra library ----- #
# Select linear algebra libraries BLAS/LAPACK
# Set to MKL, ACML or GENERIC
set(MODFEM_BLASLAPACK MKL) # Set MKL or ACML or GENERIC

#-----#
#          EXTERNAL LIBRARIES CONFIGURATION SECTION          #
#-----#

# ----- Intel MKL configuration ----- #
# MKL paths and names
# (only relevant if MODFEM_USE_MKL set to TRUE)

# Paths to look for MKL include file
# (you can provide more than one - whitespace separated)
# Intel MKL OneAPI
set(MKL_INCLUDE_DIRS /opt/intel/oneapi/mkl/latest/include)

# Paths to look for MKL library files (you can provide more than one)
# and Intel Compiler library files (provide two paths)
# Available intel architecture subdirectories:
# - intel64 -> libraries for 64-bit system
# - ia32    -> libraries for 32-bit system
# - mic     -> libraries for MIC Xeon Phi architecture
set(MKL_LIBRARY_DIRS /opt/intel/oneapi/mkl/latest/lib/intel64
/opt/intel/oneapi/compiler/latest/linux/compiler/lib/intel64/) # 64-bit

# MKL library names (no extension) -
# - consult: http://software.intel.com/en-us/articles/intel-mkl-link-line-advisor/
# Select 32-bit or 64-bit system libraries by comment unused

# 32-bit system
#set(MKL_INTEL_LIB_NAME mkl_intel) #never use 'lib' prefix
#set(MKL_LAPACK_LIB_NAME mkl_lapack95) #usually not needed #never use 'lib'
prefix
#set(MKL_SOLVER_LIB_NAME mkl_solver) #never use 'lib' prefix
#set(MKL_IOMP5_LIB_NAME iomp5 libiomp5md) #never use 'lib' prefix
#set(MKL_THREAD_LIB_NAME mkl_intel_thread) #never use 'lib' prefix
#set(MKL_CORE_LIB_NAME mkl_core) #never use 'lib' prefix

# 64-bit system
set(MKL_INTEL_LIB_NAME mkl_intel_lp64) #never use 'lib' prefix
set(MKL_LAPACK_LIB_NAME mkl_lapack95_lp64) #usually not needed #never use

```

```

'lib' prefix
set(MKL_SOLVER_LIB_NAME mkl_solver_lp64) #never use 'lib' prefix
set(MKL_IOMP5_LIB_NAME iomp5 libiomp5md) #never use 'lib' prefix
set(MKL_THREAD_LIB_NAME mkl_intel_thread) #never use 'lib' prefix
set(MKL_CORE_LIB_NAME mkl_core) #never use 'lib' prefix

# ----- BLAS and LAPACK configuration ----- #
# Blas and Lapack paths and names
# (only relevant if MODFEM_USE_MKL set to FALSE)
# (when MKL in use blas/lapack from MKL will be used)
#
# Paths to look for Blas,Lapack libraries
# (usually can be left empty)
set(LAPACK_DIRS PUT_YOUR_LAPACK_DIRECTORY_OR_NOTHING)
set(BLAS_DIRS PUT_YOUR_BLAS_DIRECTORY_OR_NOTHING)

# Blas,Lapack library names (no extension)
set(BLAS_LIB_NAME blas) #never use 'lib' prefix
set(LAPACK_LIB_NAME lapack) #never use 'lib' prefix

# ----- LIBCONFIG configuration ----- #
# Libconfig paths and names
# (only important when building targets that use Libconfig)
set(LIBCONFIG_INCLUDE_DIRS /usr/include)
set(LIBCONFIG_LIBRARY_DIRS /usr/lib64)
set(LIBCONFIG_LIB_NAME config) #never use 'lib' prefix

# ----- BOOST configuration ----- #
# Boost paths and names
# (only important when building targets that use Boost)
set(BOOST_ROOT /usr/include/boost /usr/lib)
set(BOOST_INCLUDEDIR /usr/include/boost)
set(BOOST_LIBRARYDIR /usr/include/boost /usr/lib)
set(BOOST_VER_NO "1.54.0") #put the whole version number in ""

#-----
# MUMPS paths and names
# (uncomment only when MUMPS not found automatically)
#set(MUMPS_INCLUDE_DIRS PUT_YOUR_PATH_TO_INCLUDE_DIRECTORY) # Example:
# /usr/include/MUMPS
#set(MUMPS_LIBRARY_DIRS PUT_YOUR_PATH_TO_LIBRARY_DIRECTORY) # Example:
# /usr/lib64
#set(MUMPS_LIB_NAME dmumps) #never use 'lib' prefix

# ----- VORO++ ----- #
# ----- optional -----#
# Voro++ paths and names
# (only important when building targets that use Voro++)
# uncomment the below settings if you needed
#set(VOROPP_INCLUDE_DIRS PUT_YOUR_PATH_TO_INCLUDE_DIRECTORY)
#set(VOROPP_LIBRARY_DIRS PUT_YOUR_PATH_TO_BOOST_LIBRARY_DIRECTORY)
#set(VOROPP_LIB_NAME voro++) #never use 'lib' prefix

# ----- FreeGLUT library configuration ----- #

```

```

# FreeGLUT paths and names
set(GLUT_Xmu_LIBRARY Xmu) #Library: libXmu.so.{x}
                                     #{x} - version

# ----- OPENCL configuration ----- #
# OpenCL - setting opencl acceleration
# Available machine flag:
# cpu - central processing unit
# gpu - graphics processing unit
# phi - xeon phi acceleartor
# hsa - heterogeneous system architecture
set(OPENCIL_INCLUDE_DIRS YOUR_PATH_TO_OPENCIL_INCLUDE_DIRECTORIES) #Path to
directories where is CL/OpenCL folder
set(OPENCIL_LIBRARY_DIRS YOUR_PATH_TO_OPENCIL_LIBRARY_DIRECTORIES) #Path to
directories where is OpenCL library
set(OPENCIL_MACHINE "gpu") #Used machine cpu/gpu/phi/hsa
set(OPENCIL_USE_LIBRARY_ONLY FALSE) #Set TRUE or FALSE for use OpenCL library
set(OPENCIL_AUTO_TUNING FALSE) #Set TRUE or FALSE for use auto-tuning

# GPU Assembling [IS ENABLED BY DEFAULT]
#set(OPENCIL_GPU_ASSEMBLING TRUE) #Set TRUE or FALSE for assembling in kernel
(reference kernel is src/OpenCL_kernels/tmr_ocl_num_int_el.cl_assembling)

# ----- FUTURE changes ----- #
# TODO: ACML, PARMETIS
# TODO: PARALLEL BUILD (MPI)

#-----#
#                TARGETS CONFIGURATION SECTION                #
#-----#

# Specify which exe targets should be created
# (usage: "make target_name", "make" will try to build all of them )
# use TRUE or FALSE. Do _not_ comment them out.
# All available exe targets will be shown by main cmake,
# status of exe targets:
# TRUE          - target will be built
# FALSE         - target will not be built
# UNDEFINED    - target is not set in your platform file

# ----- CONV_DIFF -----#
set(CREATE_MOD_FEM_CONV_DIFF_PRISM_STD FALSE)      #Set TRUE or FALSE
set(CREATE_MOD_FEM_CONV_DIFF_PRISM_STD_QUAD FALSE) #Set TRUE or FALSE
set(CREATE_MOD_FEM_CONV_DIFF_HYBRID_STD FALSE)     #Set TRUE or FALSE
set(CREATE_MOD_FEM_CONV_DIFF_HYBRID_STD_QUAD FALSE) #Set TRUE or FALSE
set(CREATE_MOD_FEM_CONV_DIFF_PRISM2D_STD FALSE)    #Set TRUE or FALSE
set(CREATE_MOD_FEM_CONV_DIFF_PRISM2D_STD_QUAD FALSE) #Set TRUE or FALSE
set(CREATE_MOD_FEM_CONV_DIFF_PRISM_DG FALSE)       #Set TRUE or FALSE
set(CREATE_MOD_FEM_CONV_DIFF_HYBRID_DG FALSE)      #Set TRUE or FALSE
set(CREATE_MOD_FEM_CONV_DIFF_PRISM2D_DG FALSE)     #Set TRUE or FALSE

# ----- HEAT -----#
set(CREATE_MOD_FEM_HEAT_PRISM_STD TRUE)            #Set TRUE or FALSE

```

```

set(CREATE_MOD_FEM_HEAT_PRISM_STD_QUAD TRUE)           #Set TRUE or FALSE
set(CREATE_MOD_FEM_HEAT_PRISM2D_STD TRUE)             #Set TRUE or FALSE
set(CREATE_MOD_FEM_HEAT_PRISM2D_STD_QUAD FALSE)       #Set TRUE or FALSE
set(CREATE_MOD_FEM_HEAT_HYBRID_STD FALSE)             #Set TRUE or FALSE
set(CREATE_MOD_FEM_HEAT_HYBRID_STD_QUAD FALSE)        #Set TRUE or FALSE

# ----- NS_SUPG -----#
set(CREATE_MOD_FEM_NS_SUPG_PRISM_STD FALSE)           #Set TRUE or FALSE
set(CREATE_MOD_FEM_NS_SUPG_PRISM_STD_QUAD FALSE)      #Set TRUE or FALSE
set(CREATE_MOD_FEM_NS_SUPG_HYBRID_STD FALSE)          #Set TRUE or FALSE
set(CREATE_MOD_FEM_NS_SUPG_HYBRID_STD_QUAD FALSE)    #Set TRUE or FALSE
set(CREATE_MOD_FEM_NS_SUPG_PRISM2D_STD FALSE)        #Set TRUE or FALSE
set(CREATE_MOD_FEM_NS_SUPG_PRISM2D_STD_QUAD FALSE)   #Set TRUE or FALSE
set(CREATE_MOD_FEM_NS_SUPG_REMESH_STD FALSE)          #Set TRUE or FALSE
set(CREATE_MOD_FEM_NS_SUPG_REMESH_STD_QUAD FALSE)    #Set TRUE or FALSE

# ----- NS_MIXED -----#
set(CREATE_MOD_FEM_NS_MIXED_PRISM_STD_QUAD FALSE)    #Set TRUE or FALSE
set(CREATE_MOD_FEM_NS_MIXED_HYBRID_STD_QUAD FALSE)   #Set TRUE or FALSE
set(CREATE_MOD_FEM_NS_MIXED_PRISM2D_STD_QUAD FALSE)  #Set TRUE or FALSE

# ----- NS_SUPG_HEAT -----#
set(CREATE_MOD_FEM_NS_SUPG_HEAT_PRISM_STD FALSE)     #Set TRUE or FALSE
set(CREATE_MOD_FEM_NS_SUPG_HEAT_HYBRID_STD FALSE)   #Set TRUE or FALSE
set(CREATE_MOD_FEM_NS_SUPG_HEAT_PRISM2D_STD TRUE)    #Set TRUE or FALSE

# ----- NS_SUPG_HEAT_ALE -----#
set(CREATE_MOD_FEM_NS_SUPG_ALE_REMESH_STD FALSE)     #Set TRUE or FALSE

# ----- NS_SUPG_HEAT_VOF -----#
set(CREATE_MOD_FEM_NS_SUPG_HEAT_VOF_PRISM_STD FALSE) #Set TRUE or FALSE
set(CREATE_MOD_FEM_NS_SUPG_HEAT_VOF_HYBRID_STD FALSE) #Set TRUE or FALSE

# ----- PLAST_FLOW_SUPG -----#
set(CREATE_MOD_FEM_PLAST_FLOW_PRISM_STD FALSE)       #Set TRUE or FALSE
set(CREATE_MOD_FEM_PLAST_FLOW_HYBRID_STD FALSE)     #Set TRUE or FALSE
set(CREATE_MOD_FEM_PLAST_FLOW_REMESH_STD FALSE)      #Set TRUE or FALSE
set(CREATE_MOD_FEM_PLAST_FLOW_PRISM2D_STD FALSE)    #Set TRUE or FALSE
set(CREATE_MOD_FEM_PLAST_FLOW_PRISM_STD_QUAD FALSE) #Set TRUE or FALSE
set(CREATE_MOD_FEM_PLAST_FLOW_HYBRID_STD_QUAD FALSE) #Set TRUE or FALSE
set(CREATE_MOD_FEM_PLAST_FLOW_REMESH_STD_QUAD FALSE) #Set TRUE or FALSE
set(CREATE_MOD_FEM_PLAST_FLOW_PRISM2D_STD_QUAD FALSE) #Set TRUE or FALSE

# ----- PLAST_FLOW_MIXED -----#
set(CREATE_MOD_FEM_PLAST_FLOW_MIXED_PRISM_STD_QUAD FALSE) #Set TRUE or FALSE
set(CREATE_MOD_FEM_PLAST_FLOW_MIXED_HYBRID_STD_QUAD FALSE) #Set TRUE or FALSE
set(CREATE_MOD_FEM_PLAST_FLOW_MIXED_PRISM2D_STD_QUAD FALSE) #Set TRUE or FALSE

# ----- MOD_FEM_VIWER -----#
set(CREATE_MOD_FEM_VIEWER FALSE) #Set TRUE or FALSE
set(MOD_FEM_VIEWER_USE_AS_LIB FALSE) #Set TRUE or FALSE
set(MOD_FEM_VIEWER_GUI_LIN "WX") #Set 'WX' gui support

```

Po wykonaniu powyższego kroku należy przygotować katalog kompilacji.

7. Zadanie 6 [wykonywane na serwerze]

Proszę utworzyć katalog do kompilacji :

'**bin_cmake/imie_nazwisko_nompi_none_gcc_g++**' - gdzie:

_nompi – nie stosujemy obliczeń rozproszonych z MPI

_none – wyłączamy obliczenia równoległe z OpenMP, OpenCL, itp.

_gcc – kompilator języka C

_g++ - kompilator języka C++

następnie przejść do katalogu:

```
cd bin_cmake/imie_nazwisko_nompi_none_gcc_g++
```

8. Zadanie 7 [wykonywane na komputerze Honorata]

Kompilacja w utworzonym katalogu:

'**bin_cmake/imie_nazwisko_nompi_none_gcc_g++**'. Wykonujemy komendę (realizującą konfigurację środowiska budowania kodu):

```
cmake ../../src/
```

Po uruchomieniu **cmake** należy przeglądać powstały wydruk. Czy wszystkie opcje zostały poprawnie ustawione, np. czy poprawnie został wybrany program do rozwiązywania układów równań liniowych:

```
--> MKB INTERFACE selected direct solver: PARDISO
```

```
...
```

```
=> Direct solver via MKB interface: PARDISO
```

czy program nie będzie domyślnie zapisywał struktury macierzy układu równań liniowych:

```
Matrix print is disabled
```

oraz czy włączona jest opcja przenumerowania węzłów siatki MES: -

```
RENUMBERING is enabled
```

(dobrze jeśli program nie generuje też zbyt wielu wydruków dotyczących wydajności obliczeń):

```
ModFEM TIME TEST is disabled
```

[W kolejnych laboratoriach, dla kolejnych zmian w programie ModFEM, jeśli budowanie nie będzie prowadziło do spodziewanych rezultatów (np. nie uwzględni pewnych zmienionych opcji), można dokonać pełnego przebudowania z usunięciem starych plików tworzonych przez **cmake** przez **rm -rf *** (**uwaga: przebudowania należy dokonać w katalogu bin_cmake/imie_nazwisko_nompi_none_gcc_g++ istnieje ryzyko zmazania wszystkiego, przy uruchomieniu w złym katalogu!!!, najlepiej przed uruchomieniem komend sprawdzić bieżący katalog za pomocą komendy pwd**)]

Następnie wykonujemy kompilację kodu wykorzystując program **make**:

make -j4

Jeżeli wszystkie kroki wykonane były poprawnie to powinien powstać następujące pliki binarne:

**MOD_FEM_heat_prism_std
MOD_FEM_heat_prism2d_std
MOD_FEM_heat_prism2d_std_quad
MOD_FEM_ns_supg_heat_prism2d_std**

(w przeciwnym wypadku sprawdzić poprawność definicji w pliku 'src/cmake/Platforms/imie_nazwisko.cmake'):

9. Zadanie 8 [wykonywane na komputerze Honorata]

Uruchomienie kodu:

Proszę przekopiować **wszystkie** pliki konfiguracyjne przykładowego problemu ze strony przedmiotu do katalogu **~/lab_01/** np. będąc w docelowym katalogu:

wget http://ww1.metal.agh.edu.pl/~banas/MMNT/stationary_linear_problem/problem_heat.dat

W katalogu **~/lab_01/** uruchamiamy program wykonując polecenie:

~/ModFEM/bin_cmake/imie_nazwisko_nompi_none_gcc_g++/MOD_FEM_heat_prism_std .

oraz wykonujemy polecenia z menu głównego programu ModFEM:

s [enter] z [enter] q [enter]

- o poprawne zakończenie obliczeń sygnalizowane jest wydrukiem po wciśnięciu opcji **z** (obliczającej oszacowanie błędu aproksymacji):

Zienkiewicz-Zhu error estimator = 1.207858240564

(lub o innej bardzo zbliżonej wartości)

10. Podsumowanie realizacji zadań (poniższa tabelka ma znaleźć się w sprawozdaniu bezpośrednio po wnioskach, a przed załącznikami - numeracja punktów realizacji kolejnych kroków laboratorium i załączników ma odpowiadać numeracji poniższych zadań)

Zadanie (skrótowy opis)	OCENA własna w % (0-100)	OCENA prowadzącego w % (0-100)
Zad. 1 – tworzenie plików i katalogów w systemie Linux		
Zad. 2/3/4 – praca zdalna i przesyłanie plików		
Zad. 5 – konfiguracja plików ModFEM		
Zad. 6/7/8 – kompilacja i uruchomienie		
ŁĄCZNIE (400):		
OCENA KOŃCOWA:	----- -----	