

# Modelowanie matematyczne w nauce i technice

## LAB 04

### 1 Wstęp

Celem laboratorium jest zapoznanie z zaawansowaną edycją warunku początkowego w kodzie projektu „ModFEM” oraz wizualizacją danych z wykorzystaniem programu „ParaView”.

**W trakcie laboratorium dokonuje się trzech typów działań: edycji plików źródłowych kodu, kompilacji kodu, uruchomienia kodu. Najlepszym sposobem pracy jest otwarcie trzech okien, zalogowanie w każdym na serwer i praca w każdym z okien nad osobnym działaniem w odpowiednim katalogu (edycja w podkatalogach *src*, kompilacja w podkatalogu *bin\_cmake*, uruchamianie w katalogu roboczym, np. *lab\_04*)**

**Edycji kodu najlepiej dokonywać w edytorach z podświetleniem składni, takich jak *vi(m)*, *emacs*, *nano*.**

### 2 **Zadanie 1:** Przygotowanie warunku początkowego w kodzie „ModFEM” i przygotowanie pliku problemowego.

#### 2.1 Edycja kodu i jego ponowna kompilacja:

1. W katalogu **src/pdd\_heat/main/** otworzyć w edytorze plik **pds\_heat\_main.c** i znaleźć funkcję **pdr\_heat\_initial\_condition**
2. Zaraz po rozpoczęciu funkcji, następuje podstawienie do zmiennej **problem\_name** nazwy problemu określonej w pliku **problem\_heat.dat**:

```
char problem_name[300];
int problem_id = pdv_heat_current_problem_id;
pdr_problem_name(problem_id, problem_name);
```

W celu zadania określonego przez siebie warunku początkowego należy zmodyfikować kod od linii ok. 720 (wykorzystując zadaną przez siebie specyficzną nazwę problemu i w sposób nie zaburzający obliczeń dla problemów o innych nazwach) :

```
if(strcmp(problem_name, "...") == 0 ){
    double x = Xcoor[0];double y = Xcoor[1];double z = Xcoor[2];
    double value = ..; // tu należy wstawić wybraną przez siebie funkcję x,y,z
```

( w dalszej części kodu znajduje się wydruk kontrolny sprawdzający czy nastąpiło poprawne nadanie warunku początkowego – wydruk ten można po przetestowaniu kodu odpowiednio zakomentować, w całości lub - wykorzystując zmienną **initial\_setting** - dla wszystkich wywołań poza pierwszym):

```
static int initial_setting = 0;
if(initial_setting==0){
    printf("Problem ID: %d, problem_name %s\n", problem_id, problem_name);
}
//if(initial_setting==0){
printf("specified initial temperature at point %lf, %lf, %lf: %lf\n",
        Xcoor[0], Xcoor[1], Xcoor[2], value);
//}
initial_setting++;
```

3. Funkcja **pdr\_heat\_initial\_condition** zwraca do modułu aproksymacji wartość funkcji, która zostanie nadana w konkretnym węźle siatki obliczeniowej (funkcja jest wywoływana dla każdego węzła).  
[ Szczegółowy mechanizm polega na przesłaniu do modułu aproksymacji wskaźnika do funkcji **pdr\_heat\_initial\_condition** jako argumentu dla funkcji **utr\_initialize\_field** (w linii 2579 pliku **pds\_heat\_main.c**), która dalej przekazuje go do funkcji **apr\_init\_field** ]
4. napis **problem\_name** (zwracany przez funkcję **pdr\_problem\_name**) to nazwa problemu z pliku danych sterujących **problem\_heat.dat** (parametr **name**); - należy nadać wybrana przez siebie nazwę, a następnie umieścić ją także w pliku **pds\_heat\_main.c**
5. włączenie nadawania wartości początkowej za pomocą zadanej funkcji jest dokonywane przez zadanie w pliku **problem\_heat.dat** opcji: **field\_file\_in = "i";** (pozostałe dwie opcje służą do nadania wartości zerowych (**field\_file\_in = "z";**) lub wartości odczytywanych z pliku przekazanego jako parametr (**field\_file\_in = "nazwa\_pliku";**)  
• tylko dla opcji 'i' wywoływana jest funkcja **pdr\_heat\_initial\_condition**)
6. funkcja **pdr\_heat\_initial\_condition** otrzymuje jako argumenty współrzędne punktu (wektor **Xcoor**), które należy wykorzystać do obliczenia wartości zwracanej, oraz numer składowej, wykorzystywany w problemach z wektorową funkcją niewiadomą.

Po wykonaniu powyższych kroków należy, ponownie skompilować kod wykonując polecenia:

1. Przejść do katalogu:

```
~/ModFEM/bin_cmake/imie_nazwisko_nomp_i_none_gcc_g++
```

(*imie\_nazwisko* - to początek nazwy katalogu budowania kodu utworzonego w ramach laboratorium nr 1)

2. Wykonać polecenie tworzenia kodu binarnego:

```
make -j8
```

(w przypadku problemów z uwzględnieniem dokonanych zmian w skompilowanym kodzie może być konieczna pełna rekompilacja kodu - instrukcje na końcu tematu)

## 2.2 Przygotowanie pliku problemowego

- 2.2.1 Proszę utworzyć katalog **lab\_04** (na poziomie katalogów dla poprzednich laboratoriów) i skopiować pliki definiujące problem (3 pliki: dane kontrolne, siatka, warunki brzegowe) **utworzone i sprawdzone podczas poprzedniego laboratorium** (ewentualnie można pobrać pliki ze strony przedmiotu i odpowiednio je zmodyfikować )

- 2.2.2 Proszę wykonać edycję pliku problemowego: „**problem\_heat.dat**”, należy zmodyfikować wartość:  
**name = "nazwa\_problemu";**

- nazwa problemu powinna być taka jak użyta w punkcie 2.1.

- 2.2.3 W zadaniu należy wykorzystać siatkę utworzoną w ramach poprzednich zajęć laboratoryjnych - A... (opcjonalnie siatkę obliczeniową ze strony przedmiotu lub dołączoną do laboratorium nr 3 [co jednak powoduje obniżenie oceny z zajęć - w sprawozdaniu musi już znaleźć się własna siatka]) oraz plik z zadanymi podczas poprzedniego laboratorium warunkami brzegowymi.

2.2.4 Należy w pliku siatki założyć utworzenie co najmniej kilku warstw o grubości zbliżonej do rozmiaru pojedynczego trójkąta w siatce 2D (może to wymagać edycji pliku siatki w formacie `jk`, zmiany grubości obszaru obliczeniowego i liczby warstw)

2.2.5 Nazwa pliku z siatką ma zostać przekazana w pliku „`problem_heat.dat`” jako parametr `mesh_file_in` natomiast wartość parametru `field_file_in` ma zostać ustawiona na „`i`”

2.3 Uruchomienie programu „`ModFEM`” i wygenerowanie plików dla programu „`ParaView`”:

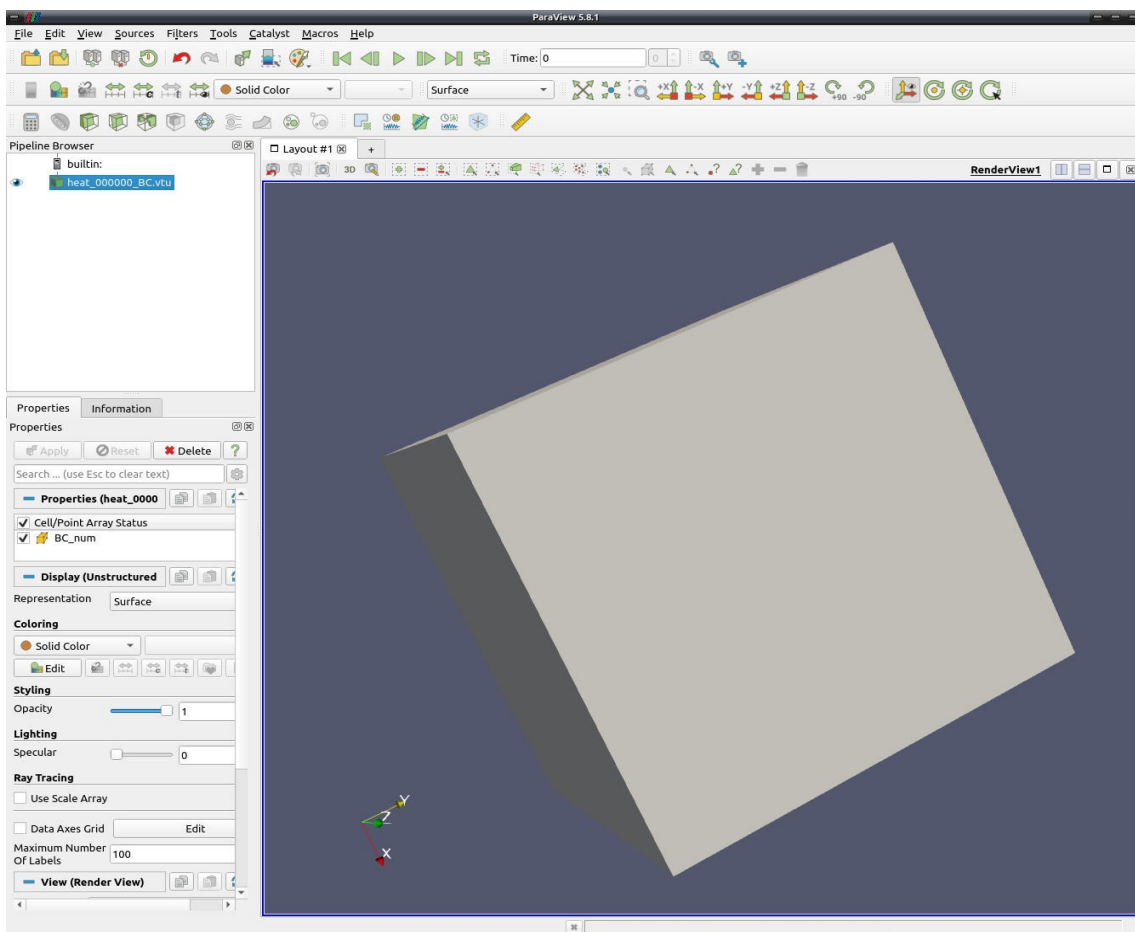
2.3.1 Proszę uruchomić program „`MOD_FEM_heat_prism_std`” w katalogu z zadaniem obliczeniowym (`lab_04`):

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

2.3.2 Należy sprawdzić (wykorzystując wydruk w funkcji `pdr_heat_initial_condition`) poprawność nadawania warunku początkowego. **W sprawozdaniu powinno się znaleźć kilka linii wydruku** (pojawiającego się w terminalu powyżej menu głównego `ModFEM`).

2.3.3 Proszę wygenerować pliki „`ParaView`” – klawisz ‘`v`’. Po wykonaniu powinny się pojawić pliki:

```
— heat_0000.pvd
— heat_000000.partmesh
— heat_000000.vtu
— heat_000000_BC.vtu
```



### 3 Zadanie 2: Wizualizacja danych w programie „`ParaView`”

## OKNO PROGRAMU

Program „**ParaView**” pozwala na zaawansowaną wizualizację wyników obliczeń numerycznych. Oprogramowanie jest rozwijane na zasadach Open Source i jest dostępne dla wszystkich popularnych systemów operacyjnych: Windows/Linux/MacOS. Oprogramowanie umożliwia:

- Obsługę danych strukturalnych i niestructuralnych
- Zaawansowane przetwarzanie danych z wykorzystaniem filtrów
- Tworzenie pól wektorowych, w celu lepszej wizualizacji zmiany danych w symulacji
- Wyodrębnianie danych z symulacji (wyświetlanie tylko wybranych danych)
- Wyświetlanie wyników w różny sposób jako zbiór punktów, linii itd.
- Obliczanie nowych wartości z wykorzystaniem wbudowanego kalkulatora
- Zaawansowane przetwarzanie danych z wykorzystaniem skryptów w języku Python
- Wiele innych...

Celem ćwiczenia jest zapoznanie z niektórymi możliwościami programu. Ćwiczenie należy wykonać dla utworzonego przez siebie obszaru obliczeniowego, traktując rysunki jako poglądowe ilustracje. Zrzuty ekranu w sprawozdaniu muszą dotyczyć własnego obszaru i zadanego warunku początkowego. **Pliki należy skopiować na maszynę lokalną i tam uruchamiać ParaView.**

### 3.1 Podstawowe operacje.

3.1.1 Proszę wczytać plik **heat\_000000.vtu** jako dane dla programu Paraview ( **File -> Open** – inne sposoby mogą wyłączać pewne opcje Paraview). Pierwszy zrzut ekranu powinien przedstawiać widok obszaru z zaznaczonymi elementami i kolorami odpowiadającymi zadanemu warunkowi początkowemu (wybrane pola: "temperature" – w miejsce "Solid Color" i "Surface with edges" w miejsce "Surface").

3.1.2 dla ułatwienia znalezienia zasięgu obszaru można w widoku obszaru w Paraview włączyć opcje widoczności siatki (Cube axes lub Data Axes Grid lub Axes Grid – w lewym panelu Properties – różne opcje w zależności od wersji Paraview, szczegółowe parametry widoku można dodatkowo edytować)

### 3.2 Drugi zrzut powinien zawierać przekrój obszaru

3.2.1 Guzik (ikona) "Clip", wprowadzenie "Plane Parameters" – "Normal" np. 0.5, 0.5, 0.5, "Apply"

3.2.2 Można eksperymentować z innymi wartościami parametrów, opcjami wyglądu i sterowaniem widokiem obiektu

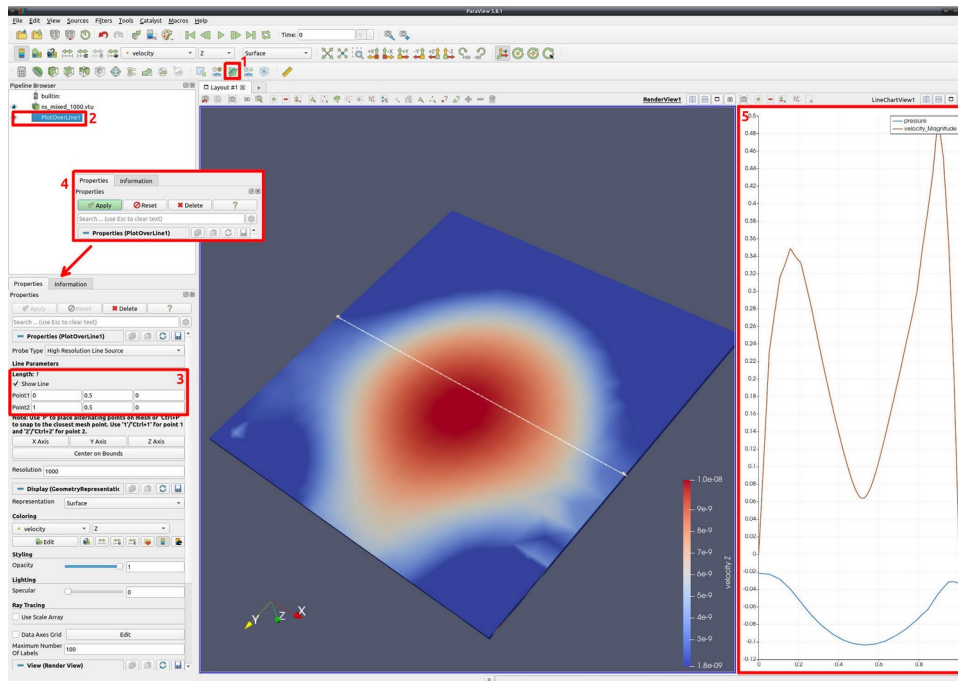
### 3.3 Tworzenie profilu rozwiązania wzdłuż zadanej linii:

(1) Ikona '**Plot Over Line**' – otwiera okno do konfiguracji profilu rozwiązania.

(2) Nowy obiekt – powstaje po kliknięciu '**Apply**'

(3) Konfiguracja parametrów linii, wzdłuż której zostanie utworzony profil rozwiązania.

(4) Panel akceptacji konfiguracji (należy zawsze kliknąć '**Apply**', aby wykonać akcję.



(5) Wygenerowany profil rozwiązania

**Zadanie 2a** : Dla wczytanej siatki z warunkiem początkowym proszę utworzyć profil rozwiązania wzdłuż wybranej przez siebie linii (proszę zwrócić uwagę, że linia wzdłuż której liczony jest profil rozwiązania jest rysowana na wizualizacji jako biały wektor). Proszę dobrać linię profilu w taki sposób, aby wyniki miały sens.

Obraz z wizualizacji wraz z wykresem profilu rozwiązania oraz parametrami narzędzia proszę umieścić w sprawozdaniu.

4 Pozostałe zadania są nadobowiązkowe, podwyższają ocenę ze sprawozdania i mogą także przydać się do kolejnych laboratoriów

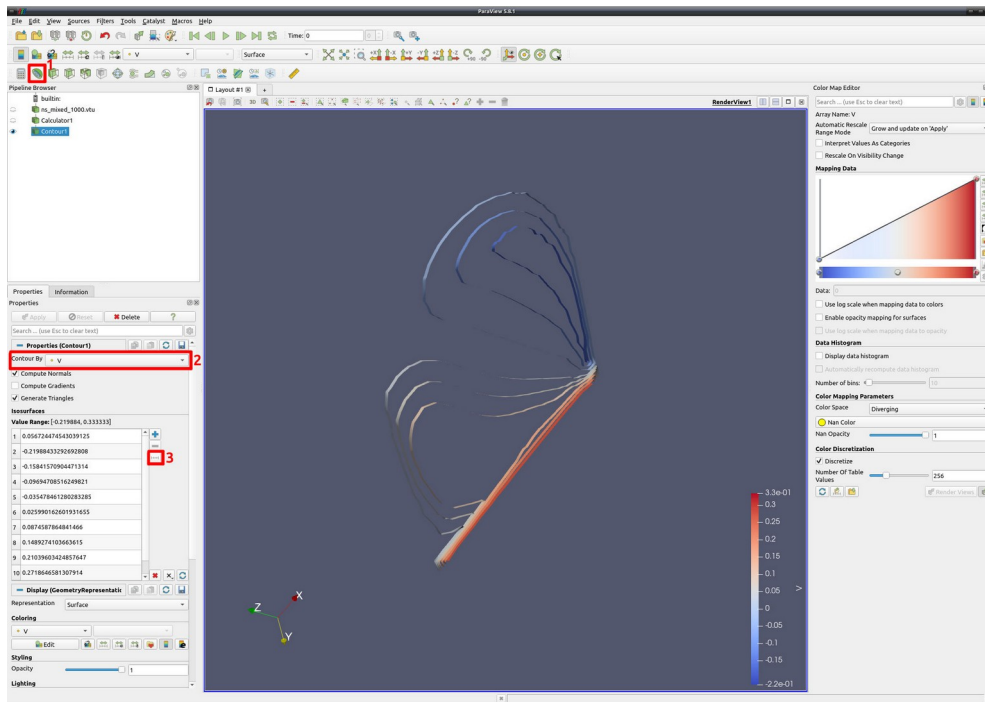
4.1 Narzędzie 'Contour' umożliwia wyświetlenie wykresów rozwiązania dla zadanych wartości.

(1) Ikona 'Contour' – otwiera okno narzędzia do rysowania konturów rozwiązania.

(2) Wybór rozwiązania.

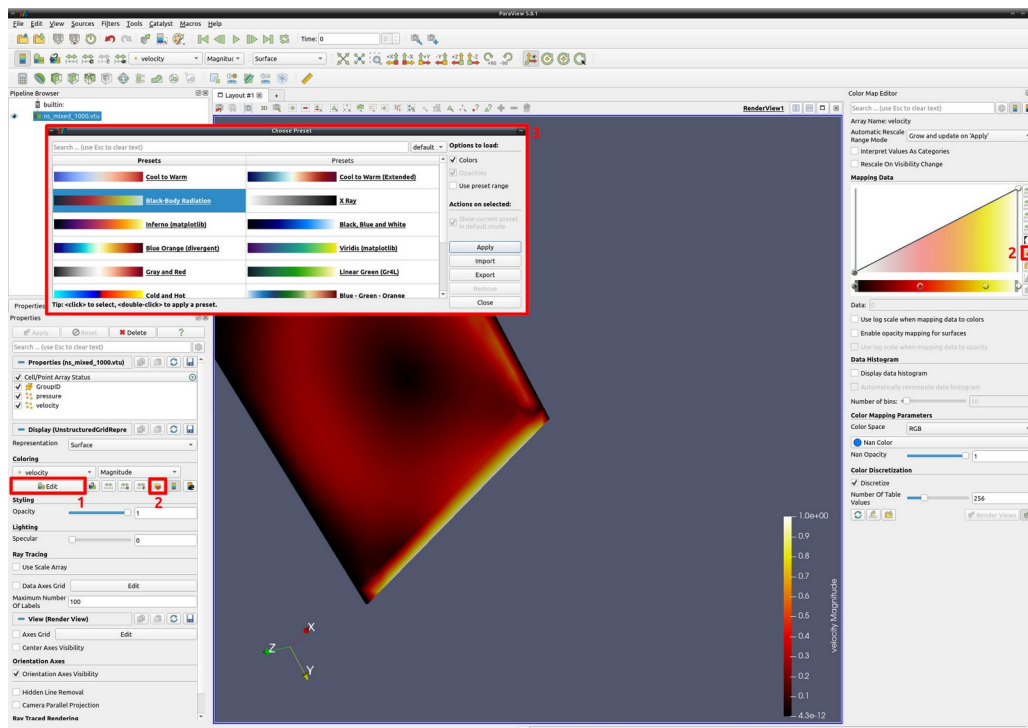
(3) Określanie zakresu rozwiązania oraz kroku.

**Zadanie 3c:** Proszę utworzyć kontur rozwiązania (warunku początkowego). Proszę ustalić liczbę przedziałów na 15. Obraz z wizualizacji oraz parametry narzędzia proszę umieścić w sprawozdaniu.



- 4.2 Modyfikacji palety kolorów dla wykresów
- Program 'ParaView' pozwala na dowolną edycję palety kolorów stosowanej przy wizualizacji danych. Udostępnia również wiele gotowych zestawów kolorów dla potrzeb wizualizowania danych.
- (1) Otwarcie okna edycji palety kolorów.
  - (2) Wybór zestawu kolorów ze zdefiniowanego wzorca.

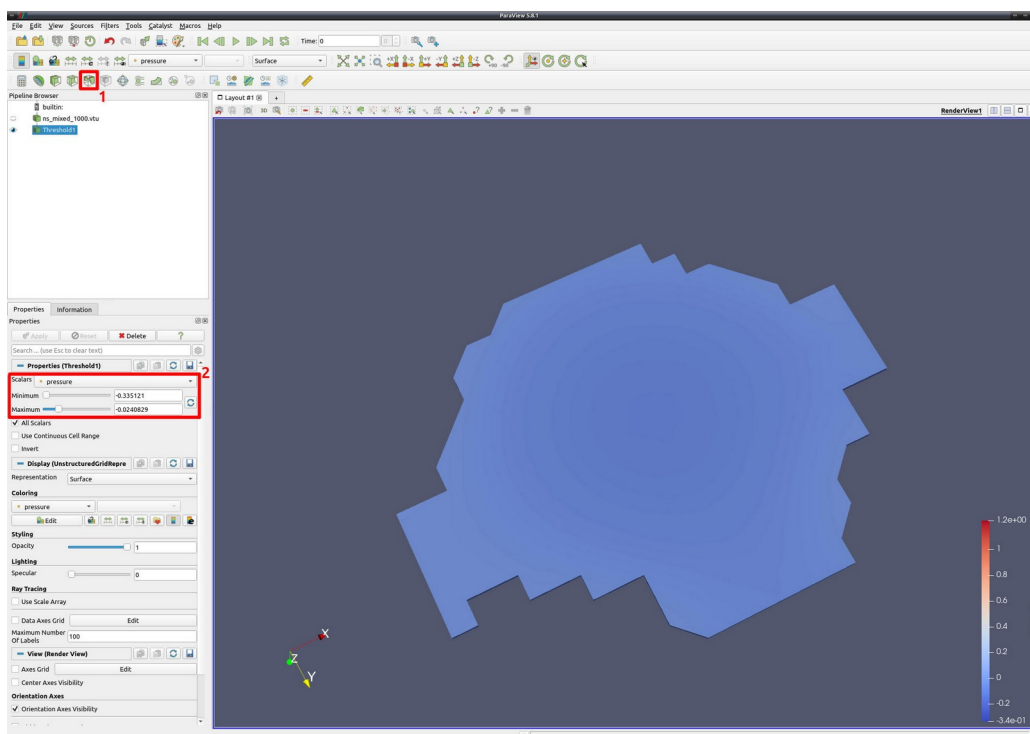
**Zadanie 3d:** Dla dwóch wybranych zadań **3.a-d** proszę zmodyfikować paletę kolorów (zestaw kolorów proszę dobrać w taki sposób, aby wyniki były czytelne). Otrzymane obrazy proszę umieścić w sprawozdaniu.



#### 4.3 Tworzenie wycinka rozwiązania dla zadanych zakresów wartości.

- (1) Ikona **'Threshold'** – otwiera okno do konfiguracji wycinka rozwiązania.
- (2) Ustalanie minimalna i maksymalnej wartości rozwiązania, dla której będzie tworzona wizualizacja.
- (3) Pamiętajmy o kliknięciu **'Apply'**

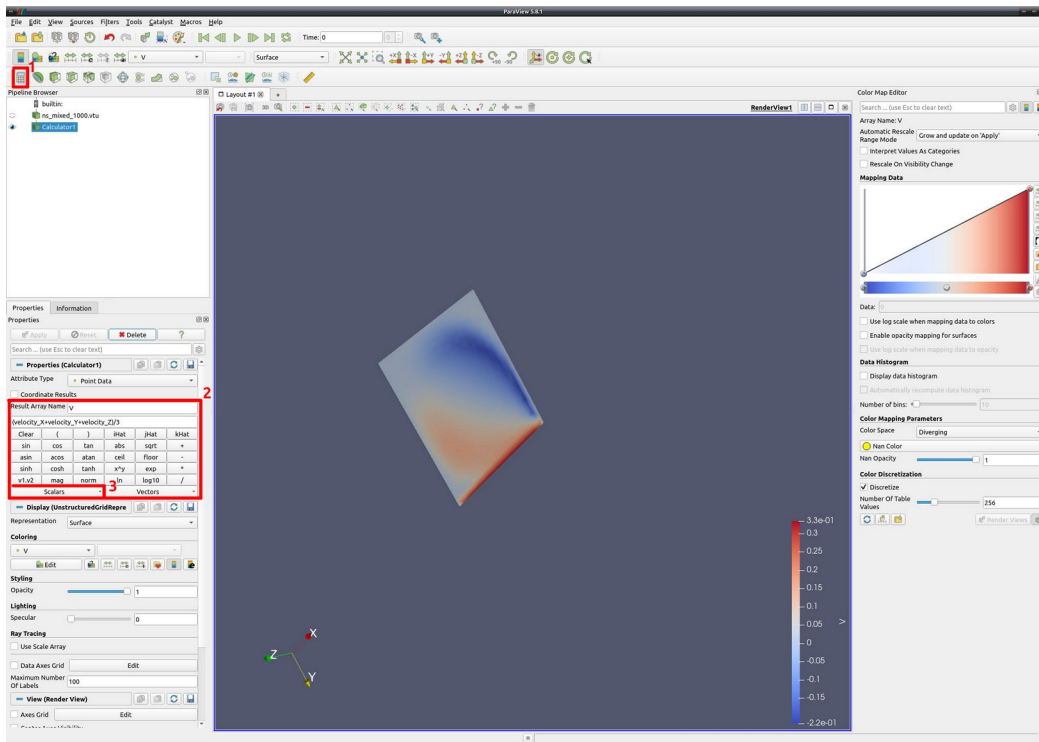
**Zadanie 3a:** Dla wczytanej siatki z warunkiem początkowym proszę utworzyć wizualizację z wykorzystaniem narzędzia **'Threshold'**. Parametry proszę dobrać tak, aby pojawił się wycinek rozwiązania. Obraz z wizualizacji oraz parametry narzędzia proszę umieścić w sprawozdaniu.



#### 4.4 Wykonywanie obliczeń z wykorzystaniem kalkulatora **'Calculator'**

- (1) Ikona **'Calculator'** – otwiera okno kalkulatora.
- (2) Okno kalkulatora.
- (3) **'Scalars'** – umożliwia pobranie wartości skalarnych z wczytanych plików symulacji / **'Vectors'** – umożliwia pobranie całych wektorów rozwiązania.

**Zadanie 3b:** Proszę z wykorzystaniem kalkulatora pobrać rozwiązanie (warunek początkowy symulacji) i obliczyć wartość  $\sin(1/x)$  lub innej wybranej funkcji ( $x$  – wartość rozwiązania pobrana za pomocą ikony **'Scalars'**). Wynik proszę zapisać w zmiennej RES. Obraz z wizualizacji oraz parametry narzędzia proszę umieścić w sprawozdaniu.



## 5 Podsumowanie realizacji zadań

Zadanie (skrótowy opis)	OCENA własna studenta w % (0-100)	OCENA prowadzącego w % (0-100)
Zad. 1A – edycja kodu i kompilacja kodu „ModFEM”		
Zad. 1B – przygotowanie pliku problemowego / uruchomienie kodu / wygenerowanie pliku ParaView		
Zad. 2A – utworzenie wizualizacji obszaru i rozwiązania - + opis narzędzia (dokumentacja ParaView)		
Zad. 2B – utworzenie widoku przekroju i wykresu profilu + opis narzędzia (dokumentacja ParaView)		
Zad. 3a – utworzenie wizualizacji + opis narzędzia (dokumentacja ParaView)		
Zad. 3b – utworzenie wizualizacji + opis narzędzia (dokumentacja ParaView)		
Zad. 3c – utworzenie wizualizacji + opis narzędzia (dokumentacja ParaView)		
Zad. 3d – utworzenie wizualizacji + opis narzędzia (dokumentacja ParaView)		
ŁĄCZNIE (900):		



Sprawozdanie powinno zawierać krótki opis zmian dokonanych w programie „ModFEM” oraz plikach problemowych (należy zamieścić w sprawozdaniu kilka modyfikowanych linii z **pds\_heat\_main.c** i z **problem\_heat.dat**). Dla zadań wizualizacji poza wykonaniem kroków w poszczególnych poleceniach i zamieszczeniem obrazów (zrzutów ekranu lub obrazów generowanych przez "ParaView") powinien się pojawić krótki opis narzędzi wykonany w oparciu o dokumentację „ParaView”.

6 Pomocne strony internetowe:

- **PARAVIEW** - [ <https://www.paraview.org/> ] - strona główna programu Paraview.
- **Kurs PARAVIEW** - [ [https://www.paraview.org/Wiki/The\\_ParaView\\_Tutorial](https://www.paraview.org/Wiki/The_ParaView_Tutorial) ] - kurs / tutorial – wprowadzenie do Paraview.

---

## Rekompilacja kodu

(Standardowa rekompilacja za pomocą *make* działa szybko, czasem jednak zawodzi z powodu wykorzystywania przez *cmake* plików tymczasowych *cache*. W przypadku kiedy zwykła kompilacja nie daje właściwego wyniku należy dokonać całkowitego przebudowania kodu. W tym celu należy usunąć całą zawartość katalogu z kodem binarnym poleceniem (**przed wykonaniem operacji proszę się upewnić czy jest się w odpowiednim katalogu poleceniem: *pwd***):

```
rm -rf *
```

a następnie wykonać konfigurację plików projektu, poleceniem ***cmake***:

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

i ponowną kompilację kodu (użycie opcji *-jx* skutkuje wykorzystaniem wielu wątków, co przyspiesza wykonanie, ale może spowodować, że wydruki nie będą czytelne, w szczególności miejsce wystąpienia błędu kompilacji dla danego wątku może zniknąć wśród wydruków innych wątków):

```
make -j4
```

całkowita przebudowa kodu może być w szczególności konieczna przy istotnej modyfikacji pliku ***imie\_nazwisko.cmake***