

Modelowanie matematyczne w nauce i technice

LAB 12. Rozwiązywanie problemów niestacjonarnych metodą elementów skończonych

Wstęp

Zagadnienia zawierające w sformułowaniu różniczkowym pochodne czasowe (problemy paraboliczne i hiperboliczne) wymagają dyskretyzacji nie tylko w przestrzeni, ale także w czasie. Jedną z takich całościowych technik jest metoda linii, w której MES można zastosować do dyskretyzacji przestrzennej.

Podstawowym założeniem użycia MES w metodzie linii jest przyjęcie, że rozwiązania nadal poszukiwane są jako kombinacje liniowe funkcji bazowych (określonych tylko w przestrzeni), ale współczynniki kombinacji liniowej (stopnie swobody, w szczególnych przypadkach wartości w węzłach MES) są teraz funkcjami czasu.

Po zastosowaniu standardowych procedur MES do dyskretyzacji przestrzennej równań parabolicznych lub hiperbolicznych, otrzymuje się układ równań różniczkowych zwyczajnych, w którym niewiadomymi są współczynniki kombinacji liniowej funkcji bazowych, tworzące wektor $\mathbf{U}(t)$:

$$\mathbf{M} \cdot d\mathbf{U}/dt + \mathbf{K} \cdot \mathbf{U} = \mathbf{b}(t)$$

gdzie postać tzw. macierzy mas $\mathbf{M}$ oraz macierzy sztywności $\mathbf{K}$ i wektora prawej strony $\mathbf{b}$ wynikają ze sformułowania słabego problemu. Do takiego problemu (który można przedstawić także jako $d\mathbf{U}/dt = \mathbf{M}^{-1} * (\mathbf{b}(t) - \mathbf{K} \cdot \mathbf{U})$) należy zastosować dowolną z metod dyskretyzacji równań różniczkowych zwyczajnych. Najpopularniejsze z metod, stosujące prostą dyskretyzację $d\mathbf{U}/dt \approx (\mathbf{U}(t^{n+1}) - \mathbf{U}(t^n)) / \Delta t$ (gdzie $\Delta t = t^{n+1} - t^n$) to:

- metoda jawna Eulera: $\mathbf{M} \cdot \mathbf{U}(t^{n+1}) = \mathbf{M} \cdot \mathbf{U}(t^n) - \mathbf{K} \cdot \mathbf{U}(t^n) + \mathbf{b}(t^n)$
- metoda niejawna Eulera: $\mathbf{M} \cdot \mathbf{U}(t^{n+1}) + \mathbf{K} \cdot \mathbf{U}(t^{n+1}) - \mathbf{b}(t^{n+1}) = \mathbf{M} \cdot \mathbf{U}(t^n)$
- metoda Cranka-Nicolson:

$$\mathbf{M} \cdot \mathbf{U}(t^{n+1}) + 0.5 \cdot \mathbf{K} \cdot \mathbf{U}(t^{n+1}) - 0.5 \cdot \mathbf{b}(t^{n+1}) = \mathbf{M} \cdot \mathbf{U}(t^n) - 0.5 \cdot \mathbf{K} \cdot \mathbf{U}(t^n) + 0.5 \cdot \mathbf{b}(t^n)$$

W każdej z nich znajdowanie rozwiązania polega na rozwiązywaniu sekwencji układów równań dla niewiadomego wektora $\mathbf{U}(t^{n+1})$, zakładając znane rozwiązanie $\mathbf{U}(t^n)$, gdzie $n=0, 1, 2, \dots$ (dla pierwszego kroku znanym rozwiązaniem jest interpolacja warunku początkowego na siatce MES). Proces kontynuowany jest przez określoną liczbę kroków czasowych lub do momentu zbieżności w czasie, czyli do momentu kiedy norma różnicy rozwiązań dla dwóch kolejnych kroków czasowych spadnie poniżej zakładanej granicy (to ostatnie kryterium dotyczy zadań stacjonarnych rozwiązywanych za pomocą zbieżnej sekwencji rozwiązań problemu niestacjonarnego).

Wszystkie powyższe metody dają się przedstawić jako

$$\mathbf{M} \cdot \mathbf{U}(t^{n+1}) + \alpha \cdot \mathbf{K} \cdot \mathbf{U}(t^{n+1}) - \alpha \cdot \mathbf{b}(t^{n+1}) = \mathbf{M} \cdot \mathbf{U}(t^n) - (1-\alpha) \cdot \mathbf{K} \cdot \mathbf{U}(t^n) + (1-\alpha) \cdot \mathbf{b}(t^n)$$

z różnymi wartościami parametru α . W programie ModFEM można zastosować taką uogólnioną metodę α , w praktyce najczęściej z wartościami $\alpha = 0.5$ - odpowiadającej metodzie Cranka-Nicolson, jako najdokładniejszej (jedynej spośród metod α będącej metodą drugiego rzędu) lub $\alpha = 1.0$ - odpowiadającej metodzie niejawnej Eulera, jako najbardziej stabilnej (dla wielu problemów zastosowanie bardzo dużych kroków czasowych w niejawnej metodzie Eulera nie powoduje "uciekania" rozwiązania do nieskończoności).

W ramach laboratorium testowane będą różne warianty obu powyższych metod dyskretyzacji czasowej (całkowania po czasie) dla prostego problemu czystej konwekcji (zadania hiperbolicznego):

$$du/dt + \mathbf{v} \cdot du/dx = 0$$

(użyty w zadaniu obszar jest trójwymiarowy, ale przyjęte warunki brzegowe czynią go w praktyce zadaniem jednowymiarowego unoszenia z zadaną prędkością $\mathbf{v}$).

Zadanie nazwane jest "WAVE_IN_BOX".

0. Krokiem przygotowawczym laboratorium 12 jest dokonanie zmian w kodzie ModFEM. Należy przejść do katalogu, w którym każdy tworzy swoją wersję kodu:

```
cd ~/ModFEM/modfem2015
```

a następnie skopiować następujące pliki z katalogu

/home/students/ModFEM/modfem2015/ do odpowiednich podkatalogów **src/pdd_heat** :

```
cp /home/students/ModFEM/modfem2015/pdh_heat.h src/pdd_heat/include/
```

```
cp /home/students/ModFEM/modfem2015/pds_heat_main.c src/pdd_heat/main/
```

```
cp /home/students/ModFEM/modfem2015/pds_heat_adapt.c src/pdd_heat/adaptation/
```

```
cp /home/students/ModFEM/modfem2015/pds_heat_time_integration.c src/pdd_heat/time_integration/
```

Po skopiowaniu plików należy przejść do odpowiedniego podkatalogu **bin_cmake** , w którym tworzony jest kod i wykonać budowanie kodu za pomocą **make**. Przy poprawnym skopiowaniu plików, w trakcie kompilacji powinny pojawiać się następujące komunikaty:

```
Scanning dependencies of target MOD_FEM_heat_prism2d_std
```

```
[ 75%] Building C object pdd_heat/CMakeFiles/MOD_FEM_heat_prism2d_std.dir/main/pds_heat_main.c.o
```

```
[ 76%] Building C object pdd_heat/CMakeFiles/MOD_FEM_heat_prism2d_std.dir/time_integration/pds_heat_time_integration.c.o
```

```
[ 76%] Building C object pdd_heat/CMakeFiles/MOD_FEM_heat_prism2d_std.dir/linear_solver_interface/pds_heat_ls_intf.c.o
```

```
[ 77%] Building C object pdd_heat/CMakeFiles/MOD_FEM_heat_prism2d_std.dir/adaptation/pds_heat_adapt.c.o
```

```
[ 77%] Building C object pdd_heat/CMakeFiles/MOD_FEM_heat_prism2d_std.dir/input_output/pds_heat_materials_io.c.o
```

```
[ 78%] Building C object pdd_heat/CMakeFiles/MOD_FEM_heat_prism2d_std.dir/input_output/pds_heat_io.c.o
```

```
[ 78%] Building C object pdd_heat/CMakeFiles/MOD_FEM_heat_prism2d_std.dir/input_output/pds_heat_problem_io.c.o
```

```
[ 78%] Building C object pdd_heat/CMakeFiles/MOD_FEM_heat_prism2d_std.dir/input_output/pds_heat_paraview_io.c.o
```

```
[ 78%] Building C object pdd_heat/CMakeFiles/MOD_FEM_heat_prism2d_std.dir/materials/pds_heat_materials.c.o
```

```
[ 79%] Building C object pdd_heat/CMakeFiles/MOD_FEM_heat_prism2d_std.dir/weak_formulation/pds_heat_weakform.c.o
```

```
Linking CXX executable ../MOD_FEM_heat_prism2d_std
```

Zmodyfikowany kod jest konieczny do poprawnej realizacji laboratorium.

1. Zadanie 1 (obowiązkowe). Uruchomienie programu ModFEM dla zadania niestacjonarnej konwekcji dyskretyzowanego niejawnym schematem Eulera

1.1. Proszę utworzyć katalog dla aktualnego laboratorium, np. **~/lab_12**

1.2. Do katalogu należy skopiować pliki konfiguracyjne ze strony przedmiotu:

mesh_jk_bfs2111.dat, **bc_jk_heat_wave_in_box.dat** i **problem_heat.dat**.

- plik **mesh_jk_bfs2111.dat** zawiera siatkę dla modelowanego problemu, w którym obszarem obliczeniowym jest pojedyncza warstwa elementów o małej grubości (wzdłuż osi z), niewielkiej szerokości (wzdłuż osi y) i znacznie większej długości (wzdłuż osi x).
- plik **bc_jk_heat_wave_in_box.dat** określa warunki brzegowe dla rozwiązywanego problemu – w praktyce oznaczające zerowanie pochodnej normalnej do brzegu w każdym punkcie brzegu obszaru (nie zachodzi interakcja pomiędzy procesem modelowanym w obszarze obliczeniowym a otoczeniem)
- plik **problem_heat.dat** definiuje rozwiązywany problem "**WAVE_IN_BOX**", jako zagadnienie czystej konwekcji (unoszenia) pewnego zadanego w warunku początkowym kształtu wzdłuż osi x, z zadaną prędkością (w rozwiązywanym zadaniu równą 1 – warunek zastosowania takiej prędkości umieszczony jest jawnie w kodzie dla zadania "**WAVE_IN_BOX**")
 - zadanie jest niestacjonarne – jednak warunki powodują, że unoszony kształt po pewnym czasie dociera do brzegu obszaru i wydostaje się na zewnątrz, po czym rozwiązanie staje się równe 0 w całym obszarze i we wszystkich kolejnych krokach czasowych
- w definicji problemu najważniejsze składniki pliku **problem_heat.dat** to:
 - dane materiałowe: odpowiadają zadaniu równania konwekcji z prędkością **v**, co oznacza, że współczynnik przy wyrazach dyfuzji (współczynnik przewodnictwa cieplnego) wynosi 0, a pozostałe dane są równe 1
 - dane metody dyskretyzacji czasowej (wybrany niejawny schemat Eulera – $\alpha = 1.0$):**

```

/* TIME INTEGRATION PARAMETERS - GENERIC: FOR ALL PROBLEMS */
time_integration_type = 1; // int: 1 - alpha scheme
implicitness_parameter = 1.0; // alpha, double:
                        // 0.5-Crank-Nicolson, 1.0-implicit Euler
current_timestep = 0; // int: beginning of simulation
current_time = 0.0; // double: beginning of simulation

/* delta time */
current_timestep_length = 0.02; //double:
previous_timestep_length = 0.02; //double:

/* stopping criteria: simulation will stop whichever comes first*/
final_timestep = 500; // int
final_time = 10.0; // double
time_error_type = 0; // int: not used (max norm assumed)
time_error_tolerance = 1.0e-9; // double: convergence limit
time_monitoring_level = 0; // the level of output: (not used)

/* data dumps intervals (0 - no dumps) and accuracy settings */
full_dump_intv = 0; //int: interval for restart dumps
graph_dump_intv = 50; //int: interval for graphics dumps
graph_dump_accu = 0; //int: accuracy ( 0 - full accuracy)

```

- dane solwera nieliniowego: `max_nonl_iter = 0`; - wyłączenie algorytmu rozwiązywania równań nieliniowych, problem w każdym kroku jest traktowany jako liniowy
- dane solwera liniowego: `linear_solver_type = 0`; - w każdym kroku używany jest solver bezpośredni
- dane adaptacji (w zadaniu 1): `adapt_interval = 0`; - wyłączenie automatycznej adaptacji w trakcie całkowania po czasie

1.3. W celu zapewnienia sprawnego wykonywania obliczeń przez wiele osób konieczne jest ograniczenie liczby wątków przydzielanych programowi ModFEM poprzez wywołanie w terminalu polecenia

```
export OMP_NUM_THREADS=1
```

ograniczającego liczbę wątków do jednego

1.4. Następnie należy uruchomić program do symulacji zagadnienia konwekcji wzdłuż osi x (**program z adaptacją 2D – program standardowy wymagałby zbyt dużo mocy obliczeniowej!**):

```
MOD_FEM_heat_prism2d_std
```

1.5. Po uruchomieniu programu należy wybrać opcję 'v', co spowoduje zapis rozwiązania w kroku 0, czyli interpolowanego warunku początkowego

a) Na tym etapie można także sprawdzić poprawność definicji całego zadania poprzez przeglądanie wydruku poprzedzającego menu główne programu

1.6. Następnie proszę wybrać opcję 't', co spowoduje uruchomienie procesu całkowania w czasie

a) w wersji podstawowej zadanie zdefiniowane jest jako unoszenie w obszarze dyskretyzowanym za pomocą siatki początkowej (bez adaptacji)

b) liczba kroków ustawiona jest na 500 (co powinno pozwolić na symulację trwającą poniżej minuty), z zapisem plików do wizualizacji co 50 kroków.

- c) uzyskane wyniki (pliki `heat_0000...`) należy skopiować do katalogu (lokalnego) o nazwie wskazującej na użytą metodę (niejawna Eulera) i długość kroku czasowego (np. `NE_0.02`) – dzięki temu możliwe będzie w dalszych zadaniach porównywanie wyników uzyskanych różnymi metodami, przy różnych parametrach.

[*Przed wysłaniem pliki można spakować np. za pomocą*

`tar cvzf NE_0.02.tgz heat_000*.vtu`]

- 1.7. Wyniki należy zwizualizować w programie ParaView. W sprawozdaniu proszę zamieścić: widok siatki i rozwiązania po 50 oraz 500 krokach (czyli po 1 oraz 10 sekundach), wykres rozwiązania (Plot Over Line) wzdłuż osi x, także dla kroków czasowych 50 oraz 500.

[**Uwaga: przy wykresach Plot Over Line proszę zawsze używać skali na osi y od -0.2 do 1.2 (Left Axis Range -> Left Axis Use Custom Range)**]

- 1.8. Proszę zaobserwować znaczący spadek dokładności rozwiązania – zamiast pozostawać bez zmian, wyłącznie z przesunięciem wzdłuż osi x (zgodnie z rozwiązaniem dokładnym), unoszony kształt ma coraz mniejszą amplitudę, która w każdym kroku czasowym maleje (w ostateczności do zera).

2. Zadanie 2 (obowiązkowe). Uruchomienie programu ModFEM dla zadania niestacjonarnej konwekcji dyskretyzowanego schematem Cranka-Nicolson

- 2.1. W celu zwiększenia dokładności symulacji można zastąpić schemat całkowania po czasie pierwszego rzędu (metoda niejawna Eulera z poprzedniego punktu) schematem drugiego rzędu – metodą Cranka-Nicolson

- 2.2. Należy jedynie zmienić wartość parametru α , tak aby wynosił 0.5

`implicitness_parameter = 0.5; // alpha, double:`

- 2.3. Po zmianie proszę uruchomić symulację, otrzymane wyniki skopiować do odpowiedniego folderu (np. `CN_0.02`) i zwizualizować (w ten sam sposób jak niejawną metodą Eulera). Wykresy Plot Over Line dla obu przypadków najlepiej umieścić na jednym rysunku.

- 2.4. Proszę zaobserwować jak zmienia się rozwiązanie w metodzie Cranka-Nicolson w stosunku do niejawnej metody Eulera.

3. Zadanie 3 (4.0). Uruchomienie programu ModFEM dla zadania niestacjonarnej konwekcji dyskretyzowanego schematem Cranka-Nicolson na siatce zaadaptowanej

- 3.1. Innym z możliwych sposobów zwiększenia dokładności rozwiązania jest zastosowanie adaptacji siatki. Adaptacja zostanie użyta do poprawy rozwiązania metodą Cranka-Nicolson, które będąc znacznie dokładniejsze od rozwiązania metodą Eulera, wykazywało jednak mniejszą stabilność – na krańcach unoszonego kształtu pojawiły się oscylacje, jako błędy rozwiązania

- 3.2. W programie ModFEM można uruchomić automatyczną adaptację w ramach procedury całkowania po czasie – co określoną liczbę kroków czasowych dokonywane jest szacowanie błędu aproksymacji i przeprowadzane dzielenie elementów z dużym błędem oraz łączenie uprzednio podzielonych elementów, jeśli błąd w elementach potomnych jest mały. Tym sposobem można uzyskać lokalne zagęszczenia siatki w miejscach, w których jest ono wymagane i w ostateczności uzyskać rozwiązanie o dokładności zbliżonej do rozwiązania na siatce poddanej wielokrotnej jednorodnej adaptacji, jednakże znacząco mniejszym kosztem obliczeniowym.

- 3.3. W dotychczasowym pliku `problem_heat.dat` należy dokonać zmian w sekcji dotyczącej adaptacji, poza włączeniem adaptacji opartej na znajomości rozwiązania dokładnego (dostępnego dla zadania `WAVE_IN_BOX`), także interwału pomiędzy adaptacjami (w naszym przypadku adaptacja jest uruchamiana po każdym kroku czasowym):

```
/* ADAPTATION PARAMETERS - GENERIC: FOR ALL PROBLEMS */
adapt_type = 1; //int
```

```

/*
PDC_NO_ADAPT      0 // no adaptations
PDC_ADAPT_EXACT  1 // adaptations based on the knowledge of exact solution
PDC_ADAPT_ZZ      2 // adaptations based on Zienkiewicz-Zhu error estimate
*/
    adapt_interval = 1; //int: interval (in timesteps) for adaptations

```

Pozostałe parametry adaptacji:

adapt_maxgen = 2; // maksymalna liczba podziałów elementów siatki początkowej – czyli liczba generacji elementów w siatce (dla zadań 2D każdy podział jest podziałem na cztery mniejsze elementy)

adapt_maxgendiff = 1; // ograniczenie dopuszczalności podziałów: sąsiednie elementy nie mogą różnić się generacją o więcej niż 1

adapt_tolerance = 1.e-4; // granica normy błędu w elemencie (szacowanej lub rzeczywistej dla zadań ze znanym rozwiązaniem dokładnym) powyżej której element jest dzielony

adapt_deref_ratio = 0.2; // warunek ponownego łączenia uprzednio podzielonych elementów (suma błędów w elementach potomnych ma być mniejsza od określonego ułamka granicy błędu dla podziału elementów)

- 3.4. Od tego momentu badania dotyczyć będą tylko początku procesu konwekcji do chwili $t=1s$ (z powodu ograniczeń czasowych) – dla długości kroku 0.02s oznacza to 50 pierwszych kroków czasowych:

```
final_timestep = 50; // int
```

które porównywane będą z pierwszymi 50 krokami bez adaptacji

- 3.5. Dla nowych parametrów symulacji proszę powtórzyć obliczenia, zapisać wyniki w osobnym katalogu (np. CN_0.02_adapt) i dokonać wizualizacji

- a) szczególnie istotny jest teraz (poza wykresem Plot Over Line) widok zaadaptowanego fragmentu siatki (koniecznie w odpowiednim powiększeniu), uzyskiwany opcją wizualizacji "Surface With Edges"

- 3.6. Wyniki pokazują, że adaptacja nie usunęła oscylacji rozwiązania. Przyczyną jest fakt zwiększenia stosunku długości kroku czasowego Δt do rozmiaru elementów h , który decyduje o stabilności rozwiązania. Powszechnie stosowaną miarą tego stosunku jest tzw. liczba CFL (Couranta-Friedrichsa-Lewy'ego):

$$CFL = v * \Delta t / h$$

W celu utrzymania stabilności rozwiązania, w momencie przeprowadzania adaptacji siatki konieczne staje się także zmniejszenie kroku czasowego. W naszym przypadku, dla 2 generacji elementów, czyli 4-krotnie zmniejszonego rozmiaru elementów ostatniej generacji w stosunku do elementów siatki początkowej, rozwiązaniem utrzymującym liczbę CFL na niezmiennym poziomie jest zmniejszenie kroku czasowego z 0.02 do 0.005:

```
current_timestep_length = 0.005; //double:
```

```
previous_timestep_length = 0.005; //double:
```

```
final_timestep = 200; // int
```

```
graph_dump_intv = 200; //int: interval for graphics dumps
```

(ostatnie ustawienie pozwala porównywać wyniki kolejnych eksperymentów ze sobą)

- 3.7. Ostatnim krokiem standardowych testów jest przeprowadzenie symulacji na siatce poddawanej adaptacji z dostosowanym krokiem czasowym oraz wizualizacja rozwiązań

- a) podobnie jak poprzednio szczególnie istotny jest teraz (poza wykresem Plot Over Line) widok zaadaptowanego fragmentu siatki (koniecznie w odpowiednim powiększeniu), uzyskiwany opcją wizualizacji "Surface With Edges"

4. Zadanie 4 (5.0). Uruchomienie programu ModFEM dla zadania niestacjonarnej konwekcji z własnymi opcjami

- 4.1. Rozmaitość kombinacji opcji całkowania po czasie i adaptacji umożliwia uzyskiwanie rozwiązań o założonych cechach, dokładności, stabilności, ewentualnie optymalizowanych pod kątem stosunku jakości do czasu rozwiązania
- 4.2. W zadaniu niniejszym można zastosować dowolną strategię doboru parametrów do wybranego celu, a następnie opisać w sprawozdaniu samą strategię, motywację do jej użycia oraz ostateczny przebieg symulacji i ich wyniki.
- 4.3. Przykładowymi aspektami możliwymi do uwzględnienia w badaniach są np.
 - a) wpływ adaptacji na wyniki uzyskiwane metodą Eulera (czy możliwe jest stosowanie wyższych wartości liczby CFL niż dla schematu Cranka-Nicolson?)
 - b) wpływ fizycznej dyfuzji (w przypadku modułu **heat** programu ModFEM jest to przewodzenie ciepła i determinujący go współczynnik przewodnictwa cieplnego) na przebieg symulacji (dla wszystkich rozważanych powyżej opcji doboru metody całkowania po czasie i długości kroku czasowego – liczby CFL)
 - c) wpływ parametrów adaptacji: liczby generacji elementów i granicy błędu powodującego adaptację, na wyniki symulacji (w rozmaitych kombinacjach z innymi parametrami)
 - d) wpływ nieliniowości danych materiałowych na przebieg symulacji

Tabela kontrolna samooceny

Zadanie (skrótowy opis)	OCENA własna w % (0-100)	OCENA prowadzącego w % (0-100)
Zadanie 1		
Zadanie 2		
Zadanie 3		
Zadanie 4		
ŁĄCZNIE (400):		
OCENA KOŃCOWA:	-----	

Sprawozdanie powinno zawierać opis realizacji wszystkich zadań zawartych w temacie, wraz z omówieniem podstaw teoretycznych, odpowiedziami na pytania, wydrukami kodu i plików konfiguracyjnych oraz zamieszczonymi zrzutami ekranu. Opis realizacji każdego zadania może kończyć się wnioskami wynikającymi z przebiegu realizacji, całe sprawozdanie powinno kończyć się wnioskami dotyczącymi całości tematu.