

## Modelowanie matematyczne w nauce i technice

### LAB 11. Rozwiązywanie problemów nieliniowych metodą elementów skończonych

#### Wstęp

Liniowe problemy brzegowe (i początkowo-brzegowe) to takie zadania, w których funkcje niewiadome (i ich pochodne) występują odrębnie w pierwszej potędze w poszczególnych wyrazach równania różniczkowego i w warunkach brzegowych, a współczynniki równań i warunków brzegowych nie zależą od funkcji niewiadomych (i ich pochodnych).

Wszystkie dotychczas rozważane zagadnienia były liniowe. Przykładami zagadnień nieliniowych mogą być problemy przewodnictwa cieplnego ze współczynnikiem przewodnictwa zależnym od temperatury lub problemy uwzględniające warunek brzegowy radiacji, w którym pojawia się temperatura w czwartej potędze.

Rozwiązanie zagadnienia liniowego za pomocą MES wymaga zastosowania standardowych procedur prowadzących do układu równań liniowych – rozwiązanie pojedynczego układu daje aproksymację rozwiązania dokładnego.

W przypadku zastosowania standardowych procedur (uzyskanie sformułowania słabego, zamiana sformułowania słabego na układ równań algebraicznych) dla zagadnienia nieliniowego, otrzymany układ równań jest układem równań nieliniowych. Rozwiązanie takiego układu wymaga zastosowania jednej z technik iteracyjnych, takich jak metoda Newtona, Picarda, lub jedna z wielu innych.

W ramach laboratorium będziecie Państwo badać uzyskiwanie rozwiązania przybliżonego MES dla nieliniowego problemu z współczynnikiem przewodnictwa cieplnego zależnym od temperatury. Procedura, będąca zastosowaniem metody iteracji Picarda, polega na zadaniu pewnej temperatury początkowej (może to być 0, może też dowolna inna funkcja, dla której przewiduje się, że będzie zbliżona do ostatecznego rozwiązania) i uruchomieniu procesu uzyskiwania kolejnych przybliżeń rozwiązania problemu nieliniowego. W każdej iteracji przyjmuje się współczynniki w układzie równań obliczone dla temperatury uzyskanej w poprzedniej iteracji i znajduje kolejne przybliżenie temperatury, poprzez rozwiązanie układu równań, który dzięki takiej strategii staje się układem liniowym (Lecture 3, slajd 12).

Celem procesu jest uzyskanie zbieżności – zbieżności sekwencji rozwiązań układów równań liniowych do rozwiązania układu równań nieliniowych. Po każdej iteracji obliczana jest norma (maksymalna) z różnicy pomiędzy rozwiązaniem aktualnym i rozwiązaniem z poprzedniej iteracji. Celem jest osiągnięcie sytuacji kiedy rozwiązania w kolejnych iteracjach nie ulegają już zmianie, czyli kiedy rozwiązania zbiegły się do rozwiązania problemu nieliniowego. W praktyce, nawet jeśli proces jest zbieżny, nie prowadzi się go do granic dokładności dla błędów zaokrągleń, ale przerywa, gdy norma różnicy pomiędzy rozwiązaniami w kolejnych iteracjach spada poniżej pewnej granicy. Granicę tę zadaje się w pliku konfiguracyjnym, dostosowując do specyfiki rozwiązywanego problemu.

Nie zawsze metody iteracyjne rozwiązywania zagadnień nieliniowych prowadzą do zbieżnych procesów. Dla problemów silnie nieliniowych proces może być rozbieżny – rozwiązania w kolejnych iteracjach nie zbliżają się do rozwiązania problemu nieliniowego, norma różnicy między nimi nie maleje, a czasami wręcz rośnie. Częstym rozwiązaniem w takich przypadkach jest potraktowanie rozwiązania problemu nieliniowego jako granicy, stanu ustalonego, dla pewnego procesu zmiennego w czasie. Uzyskanie takiego rozwiązania wymaga zastosowania metod symulacji zagadnień zmiennych w czasie (całkowania po czasie) i będzie przedmiotem jednego z kolejnych laboratoriów.

W ramach obecnego laboratorium nie będzie wykonywane rzeczywiste całkowanie po czasie, jednak ze względu na organizację kodów ModFEM, gdzie rozwiązanie problemu nieliniowego jest wbudowanego w procedurę całkowania po czasie, w celu rozwiązania zadania nieliniowego konieczne jest rozwiązanie zadania w pojedynczym kroku czasowym. Strategia taka była już wcześniej używana do uzyskania rozwiązania zadania liniowego.

# 1. Zadanie 1. (obowiązkowe) Rozwiązanie stacjonarnego nieliniowego problemu przewodnictwa cieplnego procedurą z całkowaniem po czasie i rozwiązaniem układu równań nieliniowych

1.1. Proszę utworzyć katalog *lab\_11*

1.2. Punktem wyjścia dla badania procedur rozwiązywania problemów nieliniowych jest proste zadanie przewodnictwa cieplnego z laboratorium 6. Pliki konfiguracyjne tego zadania (*problem\_heat.dat*, *bc\_heat.dat*) oraz **plik siatki** należy skopiować z katalogu *lab\_06* do katalogu *lab\_11*.

[ Siatka początkowa w rozwiązywanych zadaniach powinna mieć ok. 10000 stopni swobody. Dla mniejszych zadań nieliniowość może się "zgubić" wewnątrz elementów przy brzegu, większe zadania mogą zajmować zbyt dużo czasu. Zawsze powinno się używać tylko jednej warstwy elementów po grubości, można rozpoczynać rozwiązywanie od wykonania adaptacji z opcją *m*, tak żeby otrzymać właściwą liczbę węzłów (tak otrzymaną siatkę można sobie zapisać jako siatkę początkową dla kolejnych problemów). ]

1.3. W trakcie laboratorium do wszystkich zadań należy używać wersji kodu: *MOD\_FEM\_heat\_prism2d\_std* (modelowane jest zadanie 2D z pojedynczą, cienką warstwą elementów).

1.4. Jako wstęp do ćwiczeń należy uruchomić program dla konfiguracji z laboratorium 6, dokonując tylko (jeśli trzeba) kilku modyfikacji (parametry nie opisane poniżej pozostają bez zmian):

a) w pliku *problem\_heat.dat*:

- **nowa nazwa zadania, np. *problem\_nieliniowy*** – konieczne, żeby wyłączyć zmiany dokonane w poprzednich laboratoriach
- *reference\_temperature* i *ambient\_temperature* – 300 (w przypadku opcji :
- *field\_file\_in* = "i"; jako temperatura początkowa przyjmowana jest wartość *ambient\_temperature*)
- *materials\_file* = ""; - pusta nazwa pliku z danymi materiałowymi, co powoduje wczytanie zadanych poniżej w pliku *problem\_heat.dat* współczynników, w tym współczynnika przewodnictwa cieplnego
- *thermal\_conductivity* (w dalszej części przyjęta jest wartość przykładowa 35.0, pozostałe parametry materiałowe - *density*, *specific\_heat* - nie odgrywają żadnej roli w zadaniach niestacjonarnych)
- w parametrach całkowania po czasie (*time integration*)
  - *current\_timestep\_length* = 1.0e9;
  - *final\_timestep* = 1; - dla zadań w obecnym laboratorium zakłada się, że wystarczy jeden bardzo długi krok czasowy do rozwiązania zadania – także nieliniowego
- w parametrach rozwiązywania układów równań nieliniowych:
  - *max\_nonl\_iter* = 100; - maksymalna liczba iteracji przy rozwiązywaniu układu równań nieliniowych (powoduje przerwanie obliczeń po 100 iteracjach w przypadku braku lub zbyt wolnej zbieżności)
  - *nonl\_error\_tolerance* = 1.0e-6; - błąd w normie maksymalnej będący granicą uznania procesu iteracyjnego za zbieżny (w tym wypadku zakłada się, że proces zostanie przerwany, kiedy różnica w dowolnym węźle siatki MES pomiędzy rozwiązaniami w kolejnych iteracjach spadnie poniżej 1.0e-6 (co jest dość wygórowanym oczekiwaniem z praktycznego punktu widzenia)
  - *nonl\_monitoring\_level* = 3; - jak w wielu innych miejscach w ModFEM, parametr sterujący poziomem (liczbą) wydruków
- w parametrach rozwiązywania układów równań liniowych:
  - *linear\_solver\_type* = 0; - solver bezpośredni dla małych zadań powoduje najmniej kłopotów (nie wymaga złożonej konfiguracji)

- `solver_file = ""`; - solver bezpośredni najczęściej jest używany z ustawieniami domyślnymi
- `lin_solv_monitoring_level = 0`; - dla obliczeń w obecnym laboratorium nie potrzeba informacji i statystyk z solwera liniowego
- (jako alternatywę można pobrać ze strony przedmiotu przykładowy plik dla siatki testowej i podstawić tylko swoją nazwę pliku siatki)

b) w pliku `bc_heat.dat`

- należy zachować specyfikę warunków brzegowych (chłodzenie, grzanie, izolacja), jednak **dla grzania i chłodzenia przyjąć warunek stałej temperatury (isothermal), dla chłodzenia – 300, a dla grzania – 400**
- w szczególności warunki dla górnej i dolnej powierzchni powinny pozostać warunkami symetrii (zerowania strumienia ciepła)

1.5. Dla otrzymanej konfiguracji, proszę uruchomić program

`MOD_FEM_heat_prism2d_std` i rozwiązać zadanie posługując się opcją `'t'`. Powoduje to wejście do procedury całkowania po czasie (której elementem jest rozwiązanie układu równań nieliniowych).

```
***** BEGINNING TIME INTEGRATION *****
```

```
....
```

```
Solving time step 1 (step_length: 100000000.000000, initial
time: 0.000000)
```

Dzięki przyjęciu 1 kroku czasowego (o wielkim rozmiarze w celu modelowania zagadnienia stacjonarnego) procedura rozwiązania jest uruchamiana tylko raz (warunek początkowy jako temperatura startowa i efekt procedury jako rozwiązanie po jednym kroku czasowym). Po rozwiązaniu należy zapisać wynik opcją `'v'`

1.6. W wydruku pojawiającym się w trakcie obliczeń należy zwrócić uwagę na informacje w liniijkach:

- po każdej iteracji kodu rozwiązywania układów równań nieliniowych – solwera nieliniowego

```
After linear solver in nonlinear iteration 0
```

```
Solution difference norm (u_k, prev. u_k): 99.99 (limit 0.000001)
```

```
...
```

```
After linear solver in nonlinear iteration 1
```

```
Solution difference norm (u_k, prev. u_k): 0.0000 (limit 0.000001)
```

- oraz na zakończenie całego procesu iteracyjnego rozwiązania układu równań nieliniowych

```
After non-linear solver in time step 1
```

```
Solution difference norm (u_n, prev. u_n): 99.99 (limit 0.000001)
```

Pierwsza mówi o normie różnicy między rozwiązaniami w kolejnych iteracjach procesu rozwiązania układu równań nieliniowych.

Druga o przebiegu modelowania w czasie – normie różnicy między rozwiązaniami w kolejnych krokach czasowych.

Proszę zaobserwować, że dla rozwiązywanego problemu, który jest zadaniem liniowym, zbieżność procesu iteracyjnego rozwiązywania układu równań nieliniowych następuje już po pierwszej iteracji (norma różnicy rozwiązań po pierwszej i drugiej iteracji jest praktycznie równa 0), co komunikowane jest wydrukiem:

```
Convergence in non-linear iterations: 0.000000 < 0.000001 -breaking
```

1.7. Jako dodatkowe sprawdzenie można uruchomić zadanie z dwoma krokami czasowymi (`final_timestep = 2`);. Ze względu na to, że zadanie jest stacjonarne

*norma różnicy rozwiązań po jednym i po dwóch krokach czasowych także powinna być praktycznie równa 0 (odpowiedni zrzut ekranu może pojawić się w sprawozdaniu)*

## 2. Zadanie 2 (obowiązkowe – na 3.0). Badanie zbieżności rozwiązań w metodzie iteracyjnej Picarda dla problemu przewodnictwa cieplnego z prostą nieliniowością materiałową.

- 2.1. Nieliniowość materiałową (charakterystyki materiałowe zależne od wartości temperatury) można zadać w programach ModFEM przez wprowadzenie plików konfiguracyjnych z danymi materiałowymi `materials.dat`
- 2.2. Na stronie przedmiotu znajduje się plik z przykładowymi danymi, który należy skopiować do katalogu roboczego `lab_11`. Pierwszy materiał charakteryzowany w pliku (`Test_material`) będzie podstawą badań w ramach laboratorium, drugi jest przykładem rzeczywistych parametrów pewnego rodzaju stali
- 2.3. Zmiana w pliku konfiguracyjnym `problem_heat.dat` zmiennej `materials_file` (która powinna teraz wskazywać na plik `materials.dat` - dla pliku przykładowego na stronie oznacza to zakomentowanie linii 36 i odkomentowanie linii 35) , przy pozostawieniu w pliku `materials.dat` tych samych wartości, które były zadane w pliku `problem_heat.dat`, co powinno prowadzić do identycznego rozwiązania problemu jak w p.1. Taka postać pliku `materials.dat` będzie podstawą dalszych testów
  - a) identyczność taką należy sprawdzić jako wstępne zadanie – sprawdzenie poprawności składniowej pliku `materials.dat`
  - b) w wydruku danych wejściowych (przed wejściem do menu głównego) zamiast dotychczasowej informacji:

```
no file with materials configuration (data in problem_heat.dat)
single material data (not temperature dependent)
thermal_conductivity:      35.000000
```

...

powinien pojawić się wydruk:

```
materials configuration file:      materials.dat
HEAT material database or file read
material data at reference temperature
material number:      0
reference temperature:      300.000000
thermal_conductivity:      35.000000
```

(odpowiednie zrzuty ekranu z wydrukami powinny pojawić się w sprawozdaniu)

- 2.4. W pierwszym kroku wprowadzania nieliniowości do problemu rozwiązywane będzie zadanie z prostą liniową zależnością współczynnika przewodnictwa cieplnego od temperatury (ta liniowa zależność współczynnika prowadzi do nieliniowości równania różniczkowego i całego problemu).
- 2.5. Należy wykorzystać sposób zadawania zależności współczynników od temperatury w plikach `materials.dat` jako funkcji kawałkami liniowej. Pierwszy przypadek liniowej zależności dla całego zakresu temperatur w przykładzie (od 300 do 400) uzyskuje się przez zadanie:

```
thermal_conductivity = ( (300.0, 3.5), (400.0, 35.0) );
```

(przyjęty jest 10-krotny wzrost współczynnika – większe różnice mogą już prowadzić do trudności solwera nieliniowego)

[ właściwą postać dla pliku przykładowego `materials.dat` ze strony uzyskuje się przez zakomentowanie linii 22 i odkomentowanie linii 25 ]

- 2.6. Proszę uruchomić zadanie i zbadać proces zbieżności – odczytać kolejne wartości normy różnicy temperatury w poprzedniej i aktualnej iteracji, podawane w linii:  
**Solution difference norm (u\_k, prev. u\_k): . . . .**

Odczytane wartości należy zapisać w tabelce i stworzyć na ich podstawie odpowiedni wykres (na osi x numer iteracji, na osi y norma błędu, oś y ze skalą logarytmiczną).

**3. Zadanie 3 (na 4.0). Badanie zbieżności rozwiązań w metodzie iteracyjnej Picarda dla problemu przewodnictwa cieplnego z rosnącym stopniem nieliniowości materiałowej.**

- 3.1. Zadanie polega na rozwiązywaniu kolejnych przypadków, w których zmienia się charakter zależności współczynnika przewodnictwa cieplnego od temperatury, w taki sposób, że zależność staje się coraz bardziej nieliniowa. Zamiast jednostajnej zmiany współczynnika w zakresie od 300 do 400 stopni, współczynnik pozostaje stały w pewnym zakresie, po czym gwałtownie rośnie i znów pozostaje stały. Zadana jako funkcja kawałkami liniowa zależność powinna wyglądać następująco:

**thermal\_conductivity=( (300.0, 3.5) , (300.0+ . . . , 3.5) , (400.0- . . . , 35.0) , (400.0, 35.0) ) ;**

gdzie dla kolejnych zadań w miejsce kropek należy przyjmować kolejno jako parametr 10, 20, 30 (operacje dodawania i odejmowania należy wykonać przed wpisaniem wartości do pliku, w pliku powinny pojawić się już konkretne wartości np. 310 i 390 tak jak w przykładowym pliku **materials.dat** na stronie w linii 24)

- 3.2. Zaobserwuj zmianę szybkości zbieżności rozwiązań w kolejnych iteracjach metody Picarda w miarę rosnącej nieliniowości w zależności współczynnika przewodnictwa od temperatury. Ile potrzeba iteracji do uzyskania zbieżności dla przypadków wartości parametru 10, 20, 30 ?
- 3.3. *Jako dodatkowe zadanie pokazujące ograniczone możliwości metody iteracji Picarda można uruchomić program dla większych wartości parametru 40, 45 itd. (oczywiście mniejszych od 50) i zobaczyć, że w pewnym momencie proces rozwiązywania przestaje być zbieżny (wartość normy błędy przestaje się zmniejszać lub wręcz rośnie).*

- 3.4. Wyniki eksperymentów zamieść w tabelce o formacie:

Stopień nieliniowości	Liczba iteracji do uzyskania zbieżności
(300.0, 3.5) , (400.0, 35.0)	
(310.0, 3.5) , (390.0, 35.0)	
itd.	

Kolejne wiersze tabelki mają odpowiadać kolejnym wartościom parametru określającego stopień nieliniowości, w pierwszym należy umieścić wyniki z zadania 2, a w ostatnim wierszu wyniki dla pierwszej wartości parametru dla której nie udało się uzyskać zbieżności.

- 3.5. Uzupełnieniem tabeli powinny być szkice wykresu zależności współczynnika przewodnictwa cieplnego jako funkcji temperatury  $k(T)$ , w zakresie temperatur od 300 do 400 K, dla kolejnych przypadków testowych.
- 3.6. Elementem badania rozwiązywanych problemów jest prześledzenie w jaki sposób zwiększanie stopnia nieliniowości materiałowej wpływa na rozkład temperatury w obszarze obliczeniowym. W sprawozdaniu powinny znaleźć się ilustracje rozwiązania dla stałej wartości parametru  $k$ , dla jego liniowej zmiany oraz dla maksymalnego stopnia nieliniowości, dla którego udało się uzyskać zbieżność procesu iteracyjnego.

#### 4. Zadanie 4 (na 5.0). Zwiększanie dokładności rozwiązania problemu poprzez jednorodną i automatyczną adaptację siatki

4.1. Dla przypadku:

```
thermal_conductivity=( (300.0, 3.5) , (330.0, 3.5) , (370.0, 35.0) , (400.0, 35.0) );
```

przeprowadź proces zwiększania dokładności rozwiązania poprzez adaptację siatki. Procedura ta w ostateczności doprowadziłaby do zbieżności rozwiązania MES do rozwiązania dokładnego nieliniowego procesu opisywanego przez równania różniczkowe (z dokładnością do błędu zaokrążeń).

4.2. Jako pierwszą należy zastosować jednorodną adaptację siatki. Procedura polega na kilkukrotnym (np. pięciokrotnym – do uzyskania rozwiązania w fikcyjnym kroku czasowym 6, czyli w rzeczywistości na szóstej siatce) wykonaniu sekwencji wyborów z menu:

- a) 't' – rozwiązanie na aktualnej siatce, w przypadku kolejnych rozwiązań należy wskazać, że chcemy rozwiązać **jeden krok czasowy** o niezmienionej długości (długość kroku czasowego jest tak duża, żeby za każdym razem symulować uzyskanie stanu ustalonego – można traktować rozwiązania w kolejnych krokach czasowych, na kolejnych siatkach jako zupełnie odrębne procesy stacjonarne)
- b) 'z' – obliczenie oszacowania błędu aproksymacji metodą Zienkiewicza-Zhu
- c) 'm' – jednorodna adaptacja siatki (**podział każdego elementu na 4 mniejsze, ze względu na modelowanie zadania 2D**)  
[ **nie należy przekraczać liczby stopni swobody 100 000** ]
- d) po ostatniej adaptacji należy rozwiązać zadanie ('t'), obliczyć oszacowanie błędu ('z') i zapisać wynik w pliku do wizualizacji ('v' - plik `heat_000006.vtu`)
- e) w trakcie przeprowadzania obliczeń proszę zapisać w postaci tabeli dla każdej następnej siatki:
  - liczbę wierzchołków/węzłów/stopni swobody/niewiadomych/funkcji bazowych
  - oszacowanie normy błędu

4.3. W kolejnym kroku proszę przeprowadzić proces z użyciem automatycznej adaptacji. Polega ona na wykorzystaniu metody Zienkiewicza-Zhu do oszacowania błędu w każdym elemencie, a następnie dokonania podziału tych elementów, gdzie błąd jest największy. Tym razem procedura polega na kilkukrotnym (np. siedmiokrotnym, czyli aż do uzyskania rozwiązania w 8 (fikcyjnym) kroku czasowym na ósmej siatce) wykonaniu sekwencji wyborów z menu:

- a) 't' - rozwiązanie na aktualnej siatce,
- b) 'z' – obliczenie oszacowania błędu aproksymacji metodą Zienkiewicza-Zhu
- c) 'a' – automatyczna procedura adaptacji, z oszacowaniem błędu w elementach i podziale elementów z największym błędem (**na 4 mniejsze, ze względu na modelowanie zadania 2D**)  
[ **nie należy przekraczać liczby stopni swobody 100 000** ]
- d) po ostatniej adaptacji należy rozwiązać zadanie ('t'), obliczyć oszacowanie błędu ('z') i zapisać wynik w pliku do wizualizacji ('v' – np. plik `heat_000008.vtu`)
- e) w trakcie przeprowadzania obliczeń proszę zapisać w postaci tabeli dla każdej następnej siatki:
  - liczbę wierzchołków/węzłów/stopni swobody/niewiadomych/funkcji bazowych
  - oszacowanie normy błędu

4.4. Jako ostateczny efekt badania proszę umieścić w sprawozdaniu:

- a) wizualizację rozwiązania na siatce z jednorodnymi adaptacjami – dwa obrazki: rozwiązanie (opcje: "temperature" i "Surface" w Paraview) oraz siatki (opcje "Solid color" i "Surface With Edges")
- b) wizualizację rozwiązania na siatce z automatycznymi adaptacjami – dwa obrazki: rozwiązanie (opcje: "temperature" i "Surface" w Paraview) oraz siatki (opcje "Solid color" i "Surface With Edges")
- c) wykres zbieżności MES (zależności normy błędu od liczby stopni swobody) z dwoma krzywymi – jedna dla sekwencji siatek z jednorodną adaptacją i drugą dla sekwencji siatek z automatyczną adaptacją
- d) dla potwierdzenia, że adaptacje są dokonywane w miejscach dużej drugiej pochodnej rozwiązania można stworzyć wykres profilu rozwiązania wzdłuż prostej ("Plot Over Line"), tak dobranej, żeby przecinać miejsca gdzie dokonane zostały najgęstsze podziały
  - **ważne jest żeby linia przebiegała od jednego brzegu Dirichleta do drugiego, cały czas wewnątrz obszaru obliczeniowego**, co pozwala na wstępną kontrolę poprawności wyniku
  - **koniecznie należy wyświetlać tylko wartości temperatury** (wyłączyć group\_id, material\_id itp.)

**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)
Zadanie 1 - rozwiązanie zadania nieliniowego		
Zadanie 2 - zbieżność solwera nieliniowego		
Zadanie 3 - zbieżność solwera dla rosnącej nieliniowości		
Zadanie 4 - zbieżność aproksymacji MES		
<b>ŁĄCZNIE (400):</b>		
<b>OCENA KOŃCOWA:</b>	-----	

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 zawierać wnioski dotyczące całości tematu.