

Advanced Computational Techniques

LAB 11 *Pseudo transient continuation method*

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). Równania Naviera-Stokesa będące matematycznym modelem przepływu płynu nieściśliwego są równaniami nieliniowymi – pęd (iloczyn gęstości i prędkości), występujący w drugiej zasadzie dynamiki Newtona, jest unoszony polem prędkości, iloczyn skalarny pędu i prędkości zawiera wyrazy ze składowymi prędkości w kwadracie.

Zagadnienia modelowania przepływów można podzielić na problemy niestacjonarne i stacjonarne. W problemach niestacjonarnych przedmiotem modelowania są stany pól fizycznych w kolejnych krokach czasowych symulacji, w zagadnieniach stacjonarnych poszukiwany jest stan układu fizycznego, w którym nie zachodzą zmiany w czasie. Równania dla problemów niestacjonarnych zawierają pochodne czasowe i wymagają dyskretyzacji czasowej, gdzie na każdym kroku czasowym poszukiwane jest (z zadaną dokładnością) rozwiązanie w danej chwili czasu.

W problemach stacjonarnych zakłada się, że pochodne czasowe znikają, a wszystkie pola stają się funkcjami wyłącznie położenia w przestrzeni.

Symulacje stacjonarnych przepływów płynów metodą elementów skończonych, ze względu na nieliniowość równań, wymagają zastosowania procedur rozwiązywania układów równań nieliniowych. Dla wielu problemów standardowe procedury (Newtona, Picarda) zawodzą, rozwiązania w kolejnych iteracjach nie zbiegają się.

Alternatywą jest przyjęcie poszukiwanego stanu stacjonarnego jako stanu ustalonego procesu niestacjonarnego. W takim wypadku można uzyskać rozwiązanie startując z pewnego stanu początkowego i przeprowadzając symulacje w czasie, zmierzające do stanu ustalonego, gdzie nie zachodzą już zmiany w kolejnych krokach czasowych.

Dla rozważanego zagadnienia niestacjonarnego stosuje się dowolną dyskretyzację czasową, rozwiązując na każdym kroku czasowym pewien problem (dla modelowania przepływów problem jest nieliniowy). Różnica w stosunku do zagadnień niestacjonarnych polega na tym, że dokładność rozwiązań w kolejnych krokach czasowych jest bez znaczenia – liczy się tylko ostateczny stan ustalony.

Dlatego najczęściej preferuje się bardziej stabilne metody całkowania w czasie (np. niejawną metodę Eulera) nad bardziej dokładne metody (jak np. metoda Cranka-Nicolson). Ponadto procedury rozwiązania problemu w pojedynczym kroku czasowym dostosowuje się do wymagań zbieżności do stanu ustalonego – można stosować tylko jedną iterację solwera nieliniowego (lub co najwyżej kilka), a rozwiązując każdy z powstałych układów równań liniowych metodami iteracyjnymi założyć relatywnie niską dokładność rozwiązania (np. względną redukcję błędu rzędu 10^{-3}).

W przypadku rozwiązywania sprzężonego problemu przepływu płynu i transferu ciepła istnieje szereg możliwych strategii rozwiązania. Jedną z nich jest sprzęgnięcie na poziomie solwera nieliniowego i dyskretyzacji czasowej, przy użyciu niezależnych sformułowań słabych i układów równań liniowych dla każdego z problemów.

Dla zadań stacjonarnych ważne jest uzyskanie sprzężenia (konwekcji ciepła polem prędkości płynu, na które wpływ ma temperatura) dla ostatecznego stanu ustalonego.

1. Zadanie 1 (obowiązkowe). Przeprowadzenie symulacji stacjonarnego sprzężonego zagadnienia przepływu płynu i transferu ciepła (*convective heat transfer*) poprzez zastosowanie sekwencji kroków czasowych z rozwiązaniami zbieżnymi do stanu ustalonego (*pseudo transient continuation method*)

a) Proszę utworzyć katalog *lab_11*

b) Utworzony katalog będzie katalogiem roboczym dla rozwiązania sprzężonego zadania przepływu płynu i transferu ciepła w obszarze obliczeniowym odpowiadającym siatkom typu A utworzonym w ramach laboratorium 2

- zadanie jest zadaniem dwuwymiarowym, dla programu ModFEM obszar powinien być pojedynczą, cienką warstwą, z warunkami symetrii dla dolnej i górnej warstwy
- **siatki wykorzystywane w niniejszym laboratorium mogą różnić się przypisaniem warunków brzegowych – na stronie przedmiotu znajdują się rysunki pokazujące poprawne przypisanie warunków brzegowych dla aktualnego zadania (wymagane są tylko zmiany w plikach *bc_heat.dat*, pliki siatek nie wymagają modyfikacji)**
- na brzegu oznaczonym kolorem czerwonym i strzałkami skierowanymi do wewnątrz obszaru znajduje się wpływ płynu z zadaną prędkością 1.0, posiadającego temperaturę 333 (temperatura początkowa płynu w całym obszarze zadana jest jako równa 300)
- na brzegach oznaczonych kolorem niebieskim znajdują się ściany zapewniające brak przepływu płynu i przekazywanie ciepła zgodnie z zadanym współczynnikiem transferu ciepła (*heat transfer coefficient*)
- na brzegach oznaczonych kolorem czerwonym i strzałkami skierowanymi na zewnątrz obszaru znajduje się wypływ, o ciśnieniu ustalonym na wartość 0 (w symulacjach równań Naviera-Stokesa ciśnienie ma wartość wyłącznie względną, z dokładnością do pewnego stałego nieokreślanego ciśnienia) i nieustalonej temperaturze, dla której zakłada się, że nie zachodzi w niej dyfuzja ciepła prostopadle do brzegu obszaru (zerowy normalny strumień ciepła – zmiana temperatury tylko na skutek unoszenia przez pole prędkości)

c) Rozwiązywanie zagadnień sprzężonych w programie ModFEM wymaga stworzenia plików konfiguracyjnych dla obu zagadnień, w tym wypadku przepływu płynu i transferu ciepła. Symbolem stosowanym dla problemu przewodzenia ciepła jest, tak jak dotychczas **heat**, natomiast dla problemu przepływu płynu, ze względu na zastosowanie stabilizacji SUPG przy dyskretyzacji równań Naviera-Stokesa, jest **ns_supg**. W katalogu, który będzie katalogiem roboczym przeprowadzania symulacji należy umieścić pliki:

- ***problem_heat.dat*** – w pliku problemowym zagadnienia transferu ciepła należy zastosować szereg standardowych ustawień znanych z poprzednich laboratoriów, m.in.:
 - standardowe typy i nazwy plików siatki i pola temperatury (przygotowanie siatki omówione jest poniżej)
 - warunek początkowy – opcja 'i', zadanie początkowej temperatury jako **ambient_temperature** (także określonej pliku, np. jako 300)
 - dane materiałowe w pliku ***materials.dat*** (używany także przez problem przepływu)
 - dane całkowania po czasie:
 - krok czasowy – gwarantujący stabilność symulacji (przykładowo, dla innych przykładowych danych problemu, 0.1)
 - liczba kroków czasowych – 1000 (jako warunek zakończenia symulacji, gdy zbieżność jest zbyt powolna lub jej nie ma)

- tolerancja błędu (jako wartość normy różnicy pomiędzy rozwiązaniami w dwóch kolejnych krokach czasowych poniżej której uznaje się proces za ustalony (przykładowo 1))
- częstość zapisu plików z rozwiązaniem – stosownie do wymagań wizualizacji wyników
- dane solwera nieliniowego
 - maksymalna liczba iteracji, np. 5
 - tolerancja błędu (jako wartość normy różnicy pomiędzy rozwiązaniami w dwóch kolejnych iteracjach solwera, poniżej której uznaje się proces za zbieżny (przykładowo 10))
 - ustalenie tolerancji błędu dla solwera nieliniowego wyższej niż dla całkowania po (pseudo)czasie oznacza, że zakłada się, że proces początkowo (daleko od stanu ustalonego) może być silnie nieliniowy i może wymagać pewnej zbieżności solwera nieliniowego; blisko stanu ustalonego symulacja na jednym kroku czasowym może być bliska liniowej, co oznacza, że przyjmuje się jedną iterację solwera nieliniowego jako wystarczającą do uzyskania zbieżności w czasie
 - w trakcie symulacji brak zbieżności solwera nieliniowego może być wskazówką, że problemy na jednym kroku czasowym są zbyt silnie nieliniowe, na co można próbować zaradzić zmniejszając długość kroku czasowego
- dane solwera liniowego
 - solver bezpośredni, bez pliku konfiguracyjnego
- dane adaptacji
 - typ – adaptacja w oparciu o oszacowanie błędu techniką Zienkiewicza-Zhu
 - bez automatycznej adaptacji w trakcie całkowania po czasie
 - tolerancja błędu: 0.11 – wartość powodująca podział elementów, w których błąd jest większy niż 0.11 największego błędu dla wszystkich elementów siatki
 - taka wartość tolerancji powoduje realizację strategii równego rozkładu błędu w elementach siatki (*equidistribution of error*), adaptacja przestaje dzielić nowe elementy w momencie kiedy największy błąd jest ok. 9 razy większy od najmniejszego (dla tolerancji 0.9, adaptacja przestaje dzielić elementy kiedy największy jest tylko ok. 1.1 razy większy od najmniejszego – to pokazuje zakres możliwego sterowania liczbą elementów)
- **problem_ns_supg.dat** – w pliku problemowym zagadnienia przepływu należy powtórzyć wiele z parametrów problemu transferu ciepła (ze względu na sposób sprzężenia wspólnym kodem całkowania po czasie i rozwiązywania układów równań nieliniowych), ustawiając parametry specyficzne dla problemu przepływu, takie jak:
 - zerowanie pola prędkości w warunku początkowym
 - własne nazwy plików dla pól ciśnienia i prędkości
 - przyciąganie ziemskie – założone jest pominięcie przyciągania przy rozwiązywaniu problemu
 - tolerancja błędu całkowania po czasie – założona przykładowo na 0.01
 - tolerancja solwera nieliniowego – założona przykładowo na 0.01

[Powyższe ustalenia pokazują, że wiele z parametrów kontrolnych obu sprzężonych symulacji jest wspólnych (np. większość parametrów całkowania po czasie i solwera równań nieliniowych – w ich przypadku, jeśli parametry w obu plikach są różne, przyjmowane są wartości z pliku problemowego przepływu, co wskazywane jest na wydruku przed pojawieniem się menu głównego)

- parametrami różnymi dla obu symulacji (które można zmieniać niezależnie dla każdego ze sprzężonych problemów) są m.in.:
 - tolerancja błędu całkowania po czasie
 - tolerancja błędu solwera nieliniowego
 - wybór solwera liniowego i jego parametry

]

- **materials.dat** – wspólne przykładowe dane materiałowe dla zagadnienia przepływu płynu i transferu ciepła
 - **density = 7500.0;**
 - **specific_heat = 650.0;**
 - **thermal_conductivity = 3500;**
 - **dynamic_viscosity = 1.0e2;**
- **bc_heat.dat** – zadawane parametry, dla fragmentów brzegu o określonej roli powinny być następujące:
 - wpływ: **isothermal:{temp = 333.0; };**
 - wpływ: **normal_heat_flux:{flux =0.0;};**
 - ściana: **radconv:{alfa=10000.0; eps = 0.0;};**
 - symetria (górze i dół): **normal_heat_flux:{flux =0.0;};**
- **bc_ns_supg.dat** – zadawane parametry, dla fragmentów brzegu o określonej roli powinny być następujące:
 - wpływ: warunek parabolicznego rozkładu prędkości dla przekroju np. kanału 2D, jako prędkość **v** należy zadać maksymalną prędkość w przekroju, natomiast parametrami **n1, n2, n3, n4** są kolejne wierzchołki prostokąta stanowiącego brzeg obszaru (parametry podane poniżej mogą pasować do siatek tworzonych w ramach laboratorium 2, mogą też wymagać drobnych korekt współrzędnych)


```

inflow_rect_2d:{ //bc type
  v = 1.5; //bc parameters
  n1 = [0.0, 0.0, 0.0];
  n2 = [0.0, 0.0, 0.01];
  n3 = [1.0, 0.0, 0.0];
  n4 = [1.0, 0.0, 0.01];
};
          
```

w przypadku kłopotów z powyższym warunkiem brzegowym, można zadać stałą prędkość na przekroju (co jednak powoduje zaburzenie przepływu przy styku ze ścianą): **velocity:{ v = [0.0, 1.0, 0.0]; };**
 - wpływ: **outflow:{ pressure = 0.0; };**
 - ściana: **noslip:{};**
 - symetria (górze i dół): **symmetry:{};**

- przykładowe pliki z danymi kontrolnymi obu problemów oraz z danymi materiałowymi i warunkami brzegowymi dla zadania przepływu znajdują się na stronie przedmiotu
- d) Kolejnym krokiem modelowania jest przygotowanie pliku siatki.
- W celu utworzenia pliku siatki, który mógłby być wykorzystywany w wielu symulacjach należy przeprowadzić następującą procedurę:
 - początkowo należy użyć siatek typu A utworzonych w ramach laboratorium 2.
[siatki wykorzystywane w niniejszym laboratorium mogą różnić się przypisaniem warunków brzegowych – należy zweryfikować przypisanie warunków brzegowych na podstawie plików siatek na stronie przedmiotu (wymagane są tylko zmiany w plikach bc_heat.dat, pliki siatek nie wymagają modyfikacji)
]
 - w plikach *problem_heat.dat* i *problem_ns_supg.dat* należy zadać jako typ pliku siatki 'j' i nazwę swojego pliku siatki typu A
 - proszę uruchomić program **MOD_FEM_ns_supg_heat_prism2d_std**
 - po wczytaniu danych sterujących i pliku siatki należy dokonać kilkukrotnej ręcznej, jednorodnej adaptacji siatki za pomocą opcji 'm' (nie są potrzebne żadne parametry)
 - docelowa siatka powinna mieć ok. 1000-4000 węzłów
 - po osiągnięciu zakładanej docelowej siatki proszę zapisać siatkę (zapisywane są także początkowe pola) za pomocą opcji 'd'
 - otrzymany plik siatki powinien mieć nazwę *mesh0.dmp* (plik jest dodatkowo kompresowany, jednak ta informacja jest pomijana w plikach problemowych)
 - nazwę otrzymanego pliku wraz z informacją o typie 'p' należy umieścić w obu plikach problemowych jako dane o siatce
- e) samo przeprowadzenie symulacji polega na uruchomieniu programu **MOD_FEM_ns_supg_heat_prism2d_std** oraz wybraniu opcji całkowania po czasie 't'

[przed uruchomieniem kodu należy upewnić się, że jest włączona stabilizacja (wyłączana w trakcie realizacji lab 10) - jej brak uniemożliwia poprawne rozwiązywanie większości zadań, w tym także zadań z lab 10]

- dla usprawnienia symulacji należy przed rozpoczęciem obliczeń ustalić w terminalu wartość zmiennej środowiskowej:
 - **export OMP_NUM_THREADS=4**

2. Zadanie 2 (4.0). Przeprowadzenie procesu poprawy dokładności wyniku symulacji przez zastosowanie adaptacji

- rozwiązanie uzyskane jako stan ustalony na założonej siatce początkowej może zostać poprawione przez zastosowanie adaptacji opartej na szacowaniu błędu
- procedura jest prosta i polega na jednokrotnym lub wielokrotnym zastosowaniu sekwencji wyborów opcji: 'a' i 't', powodującej podział siatki, zgodnie z oszacowaniem błędu 'a posteriori' metodą Zienkiewicza-Zhu, a następnie uzyskanie zbieżności do stanu ustalonego na siatce zaadaptowanej
 - szczegółowo procedura wymaga zastosowania następujących kroków:

- rozwiązanie na siatce początkowej – opcja '**t**' (po uzyskaniu stanu ustalonego można oszacować błąd aproksymacji – opcja '**e**')
- adaptacja siatki – opcja '**a**' (z parametrem 2 - oszacowanie błędu dla obu problemów determinuje dokonywane podziały elementów)
- rozwiązanie na nowej siatce (opcja '**t**') – do uzyskania stanu ustalonego (po zakończeniu całkowania po czasie można ponownie oszacować błąd aproksymacji – opcja '**e**' i porównać błąd na siatce zaadaptowanej z błędem na siatce początkowej)
- adaptacja siatki – opcja '**a**' (z parametrem 2 - oszacowanie błędu dla obu problemów determinuje dokonywane podziały elementów)
- rozwiązanie na nowej siatce (opcja '**t**') – do uzyskania stanu ustalonego
- itd. - na każdej nowej siatce przeprowadzane jest całkowanie po czasie do uzyskania stanu ustalonego i szacowany błąd – wykres pokazujący jak zmienia się błąd aproksymacji na kolejnych siatkach jest dobrą ilustracją procesu (należy jednak pamiętać, że dla zadań z dominującą konwekcją proces zbieżności niekoniecznie będzie tak regularny, jak dla zadań eliptycznych)
- każdorazowo uzyskane siatki i rozwiązania można przeglądać w ParaView oceniając efekty zastosowanych działań
- można manipulować parametrami tolerancji adaptacji (np. ustalając wartość na mniejszą niż 0.1 i przechodząc tym samym na ustalenie bezpośredniej (a nie względem maksymalnej dla wszystkich elementów) wartości błędu powodującego podział elementu) oraz tolerancji zbieżności w czasie (kolejne podziały siatki powodują coraz mniejsze modyfikacje rozwiązania, dlatego celowym może być zmniejszenie tolerancji błędu przy całkowaniu po czasie dla zapewnienia, że rozwiązanie rzeczywiście znajdują się w stanie ustalonym) w celu uzyskania najdokładniejszych wyników

3. Zadanie 3 (5.0). Przeprowadzenie zaawansowanej wizualizacji rozwiązania, obejmującej m.in. linie prądu (*streamlines*)