

CFD: ANSYS/Fluent/CFX

Krótki opis usługi

W ramach zasobów oferowanych przez infrastrukturę PL-Grid możliwe jest prowadzenie obliczeń obejmujących zagadnienia szeroko rozumianej obliczeniowej dynamiki płynów (*Computational Fluid Dynamics*). Dla użytkowników udostępniony został pakiet oprogramowania CFD firmy **ANSYS**, w szczególności obejmujący program FLUENT.

Aktywowanie usługi

Uruchamianie zadań możliwe jest w ramach różnych interfejsów graficznych do obsługi zadań. Poniższy przykład został opisany dla systemu UNICORE.

Poniżej opisane zostały najważniejsze kroki wymagane do rozpoczęcia obliczeń:

- **Rejestracja (założenie konta)**
Rejestracja dotyczy tylko osób które nie mają jeszcze konta w [portalu PL-Grid](#). Przy rejestracji wskazane jest utworzenie konta "zwykłego" użytkownika, chyba, że rejestracja następuje tylko na potrzeby szkolenia, zgodnie z [opisem w podręczniku użytkownika](#).
- **Wystąpienie o certyfikat**
Certyfikat jest niezbędny do uruchamiania zadań w PL-Grid. Do pracy wystarczy certyfikat SimpleCA. Sposób jego uzyskiwania opisany jest w [osobnym rozdziale](#) podręcznika – poprzez zakładkę *Moje konto* w portalu i przycisk "wygeneruj certyfikat SimpleCA".
- **Wystąpienie o grant obliczeniowy (opcjonalnie)**
Każdy użytkownik po założeniu konta otrzymuje automatycznie grant osobisty (czyli minimalną pulę zasobów - 1000 godzin obliczeniowych i 40 GB pamięci dyskowej). Grant osobisty pozwala na szybkie rozpoczęcie pracy. W miarę prowadzenia obliczeń wskazane jest [wystąpienie o grant](#) tzw. "właściwy".
- **Aktywowanie usługi UNICORE**
Niezbędna jest jednorazowa aktywacja dostępu do zasobów PL-Grid poprzez system UNICORE. W tym celu należy [aplikować o usługę](#) *Globalny dostęp do UNICORE* w portalu PL-Grid, po wybraniu menu *Usługi* a następnie kliknięciu w odnośnik *Zarządzaj usługą* obok jej nazwy.



Zachęcamy do skorzystania ze szkolenia on-line: *System UNICORE w obliczeniach gridowych*. Prezentuje ono, krok po kroku, w jaki sposób można pobrać i przygotować do pracy klienta systemu UNICORE oraz wysłać za jego pomocą zadania do wykonania na infrastrukturze PL-Grid. Na szkolenie zapisać się można za pośrednictwem Portalu lub wykorzystać materiały dostępne pod adresem: <http://alfred.mat.umk.pl/unicore> (dostęp jako "Gość").

Pierwsze kroki

Klient UNICORE

Do uruchamiania obliczeń potrzebny jest program dostępowy, tzw. klient. Jego instalacja jest bardzo prosta. Przedstawiono ją we wspomnianych wyżej szkoleniach oraz w rozdziale podręcznika pt. [UNICORE Rich Client](#).

Uruchomienie zadania

Sposób uruchomienia obliczeń w programie Fluent w systemie kolejkowym UNICORE jest przedstawiony w [prezentacji](#). Sposób przygotowania plików do obliczeń opisano w rozdziale poniżej.

Należy pamiętać, że czas obliczeniowy przyznany w ramach grantu osobistego może szybko się wyczerpać, a wtedy konieczne jest aktywowanie pełnego grantu obliczeniowego oraz zaznaczenie w portalu, że ma być grantem domyślnym do prowadzenia obliczeń. Inną metodą na wskazanie grantu, w ramach którego prowadzone są obliczenia, jest ustawienie tego [bezpośrednio w kliencie UNICORE](#).

Działanie programu FLUENT w trybie wsadowym

Obliczenia są prowadzone w trybie wsadowym, co oznacza, że zadanie, rozumiane jako dane, obejmujące siatkę obliczeniową, parametry modelu, warunki brzegowe, ustawienia solwera, itd, musi zostać całkowicie zdefiniowane **przed** rozpoczęciem obliczeń. Użytkownik nie ma możliwości komunikacji z programem Fluent od momentu uruchomienia aż do chwili zakończenia działania. Dlatego wszystkie ustawienia oraz czynności do wykonania, które podczas 'standardowej' interaktywnej pracy najczęściej są podawane za pośrednictwem wyboru poleceń z graficznego menu, muszą zostać zapisane w pliku/plikach sterujących wykonaniem zadania. Można to robić na wiele sposobów; w ramach tego podręcznika podajemy najprostsze metody.

Dane do obliczeń są dostarczane w plikach, w najprostszym przypadku jest to plik z siatką (.msh lub .cas) oraz plik zawierający polecenia sterujące obliczeniami (często nazywany plikiem *journal*). Przykłady pliku .cas oraz pliku wejściowego *journal* są dołączone do prezentacji, której adres podano powyżej. W kolejnej części tego rozdziału podane zostaną podstawowe informacje o tworzeniu plików sterujących obliczeniami w programie Fluent.

Zakładamy, że Użytkownik ma możliwość lokalnego korzystania z programu Fluent w celu przygotowania pliku .cas opisującego model (siatka, ustawienia modelu, warunki brzegowe, itd). W najprostszym przypadku zadaniem poleceń w pliku sterującym, poniżej oznaczanym nazwą *inputfile*, będzie wczytanie pliku .cas oraz uruchomienie obliczeń.

Uwaga: Plik *inputfile* może być plikiem 'journal' utworzonym podczas wcześniejszych sesji FLUENTA, lub też plikiem zawierającym sekwencję poleceń do wykonania utworzonym za pomocą edytora tekstowego. W każdym przypadku plik ten może zawierać tylko polecenia interfejsu tekstowego.

Przykładowy plik *inputfile* jest przedstawiony poniżej:

```
file/read-case example.cas solve/initialize/initialize-flow solve/iterate 50 file/write-data example.dat
exit
```

Plik ten opisuje następujące czynności:

- wczytanie pliku *.cas* zawierającego opis zadania (tu: *example.cas*).
- inicjalizacja (standard initialization). Inicjalizację hybrydowa (hybrid initialization) jest opisana poleceniem *solve/initialize/hyb-initialization*;
- wykonanie 50 iteracji.
- zapisanie wyników do pliku *example.dat*.
- polecenie *exit* powoduje zakończenie działania Fluenta.

Uwaga: Liczba faktycznie wykonanych iteracji zależy od ustawień dokonanych w pliku *.cas* oraz od przypadku obliczeniowego. W szczególności może okazać się, że kryterium zbieżności nie zostało osiągnięte.

Niektóre polecenia można zapisywać korzystając z aliasów poleceń, w szczególności powyższy ciąg poleceń można zapisać równoważnie w postaci:

```
rc example.cas solve/init/init it 50 wd example50.dat it 50 wd example100.dat exit
```

Powyższa sekwencja wykorzystuje standardowe aliasy (*it* jest aliasem dla sekwencji poleceń *solve/iterate*, *rc* jest aliasem polecenia *file/read-case*, *wd* jest aliasem równoważnym *file/write-data*, itd). Aliaszy pozwalają skrótoowo opisywać najczęściej wykorzystywane polecenia. W ogólnym przypadku polecenie Fluenta wymaga podania pełnej ścieżki poleceń, np. *solve/init/init*, zgodnie ze strukturą menu poleceń Fluenta. Jest to dokładniej omówione w części dla zaawansowanych.

Kolejny przykład dotyczy zagadnienia ewolucyjnego (*transient*).

```
file/read-case example.cas solve/init/init
solve/set/time-step
; dlugosc kroku czasowego
0.001 solve/dual-time-iterate
10
; liczba kroków czasowych
500
; maksymalna liczba iteracji w kroku czasowym
file/write-data model.dat
exit
yes
```

Plik ten opisuje następujące czynności:

- wczytanie pliku zawierającego opis zadania (tu: *example.cas*);
- zdefiniowanie długości kroku czasowego (0.001 s);
- wykonanie obliczeń dla 10 kroków czasowych; w każdym kroku wykona się maksymalnie 500 iteracji;
- zapisanie wyników po 10 krokach czasowych do pliku;
- Średnik ; poprzedza linie komentarza.
- *exit* powoduje zakończenie działania programu.

Ostatnia linia w przykładzie *yes* jest niezbędna, bowiem przed zakończeniem działania Fluent wyświetla komunikat:

```
The current case and data have not been saved. Ok to discard?
```

Odpowiedź na ten komunikat również musiała znaleźć się w pliku *inputfile*.

Uwaga: Podczas przygotowywania zadania wsadowego należy poprawnie przewidzieć ciąg pytań generowanych przez program w miarę jego wykonywania i przygotować odpowiedzi na wszystkie takie komunikaty. Każda zmiana w stosunku do założonego scenariusza (czyli rozbieżność między pytaniami generowanymi przez program - a odpowiedziami znajdującymi się w pliku wejściowym) powoduje niepoprawny przebieg zadania wsadowego.

Poniższy ciąg poleceń pochodzi z przykładu opisanego w prezentacji Zadanie Generic.

```
/file/confirm-overwrite?
no
file/read-case test.cas
solve/initialize/initialize-flow
solve/iterate
20
/file/export/ascii
test.dat
wylot
```

; 'wylot' nazwa zakresu komórek.

```
..... pusta linia
; pusta linia oznacza koniec podawania zakresów komórek. Brak zakresu (tylko pusta linia) oznacza cały obszar obliczeniowy
```

```
yes
;yes - przecinek jako znak rozdzielajacy wartosci, no - spacja
velocity-magnitude
q
; q oznacza koniec wprowadzania zmiennych do wypisania
no
;no - wypisywac wartosci w wezlach, yes - wartosci w srodkach komorek
exit
yes
```

Pojawiły się kolejne nowe elementy:

- sekwencja `/file/confirm/overwrite?` oraz `'no'` zabezpiecza przed sytuacją, kiedy podczas próby zapisania wyników do pliku program upewnia się, czy można nadpisać istniejący plik. Jak wcześniej wspomniano, w trybie wsadowym każde odstępstwo od zaplanowanego scenariusza najczęściej powoduje niepoprawny przebieg obliczeń, dlatego też dobrze jest się zabezpieczyć przed nieoczekiwanymi pytaniami o istniejące pliki.
- Sesja Fluenta składa się ze wczytania pliku `.cas`, wykonaniu 20 iteracji oraz wyeksportowanie wyników do tekstowego pliku w formacie `ascii`.
- Wpisanie pliku w formacie `ascii` wymaga kolejno podawania: nazwy pliku(tu: *test.dat*), zakresów komórek, z których wartości będą zapisane do pliku (tu: *wyloł*), na zakończenie tej sekwencji należy umieścić pustą linię. Brak podania nazwy zakresu (tylko pusta linia w pliku) oznacza, że należy uwzględnić wszystkie komórki obliczeniowe. Następnie definiuje się sposób rozdzielania wartości w pliku (*yes* - przecinek, *no* - spacja) i kolejno podaje się nazwy zmiennych do wypisania (np. *pressure*, *velocity-magnitude*, *x-velocity*, itd). Koniec ciągu nazw zmiennych zaznacza się podając `'q'`. Na zakończenie określa się, czy wartości będą odnosiły się do środków komórek (*yes*), czy też będą związane z węzłami (*no*).
- Plik wynikowy `ascii` w kolejnych kolumnach zawiera: współrzędne *x*, *y*, *z* (w 3D) punktów (współrzędne są wypisywane zawsze) oraz wartości wskazanych zmiennych w kolejności odwrotnej do kolejności, w jakiej zostały podane.

Dalsze informacje

Możliwe jest także uruchamianie zadań przez inne systemy dostępne, np. [QosCosGrid](#) lub [gLite](#).