

# Fortran 2008

## Zadanie 3

### 1 Ogólne zasady zaliczenia

1. Zadanie powinno być napisane w Fortranie 2008
2. Kod powinien być kompilowany za pomocą `make` lub `rake`
3. Dopuszczalne jest używanie kompilatora ifort z odpowiednią opcją kompilacji wymuszającą standard 2008
4. Projekt powinien być opisany w pliku README.md (proszę o używanie składni Markdown)
5. Postęp prac nad projektem powinien być wersjonowany przy zachowaniu adekwatnych opisów commitów
6. Podstawę do zaliczenia projektu stanowi wysłany w terminie link do pobrania repozytorium z projektem
7. Zakazane jest używanie CVS, Team Foundation Server oraz płatnych systemów wersjonowania
8. Repozytoria mogą być hostowane w dowolnym miejscu (np. Github, Bitbucket, prywatny serwer itp.) pod warunkiem udzielenia odpowiedniego dostępu do kodu
9. Za serwer daty/godziny przyjmuje się serwer poczty AGH
10. Projekty powinny wykorzystywać jak największą ilość poznanych na zajęciach mechanizmów składni języka przy zachowaniu dobrych praktyk programistycznych

## 2 Opis problemu

Napisz własne funkcję całkującą metodami prostokątów, trapezów i kwadraturami Gaussa. Funkcja całkująca powinna mieć interfejs jak poniżej

```
interface
  function integrate (ibeg , iend ,myfun,p) result (value)
    implicit none
    ! beginning of integration interval
    real (kind=8), intent(in) :: ibeg
    ! ending of integration interval
    real (kind=8), intent(in) :: iend
    ! function to be integrated
    procedure(fun_int) :: myfun
    ! polynomial order
    integer(kind=4), intent(in) :: p
    ! result of integration
    real (kind=8) :: value
  end function integrate
end interface
```

```
interface
  function fun_int(x) result(y)
    implicit none
    real (kind = 8), intent(in) :: x
    real (kind = 8) :: y
  end subroutine fun_int
end interface
```

## 3 Zadanie

- Napisz każdą z funkcji całkujących w sposób równoległy z wykorzystaniem Coarrays
- Przetestuj działanie różnym metod całkowania dla funkcji ze zbioru funkcji typu:  $\{\sin x, e^x, \{\sum_{i=1}^p a_i x^i, p = 1, \dots, 10\}, \}$
- Porównaj wyniki numeryczne z wynikami uzyskanymi w sposób analityczny
- Przetestuj działanie programu z różnymi opcjami kompilacji Coarrays

### 3.1 Struktura kodu i plików

- W katalogu src/ powinny być kody źródłowe.
- W katalogu res/ mają być wyniki oraz wykresy (gnuplot?).
- W pliku README.md oprócz kompletnego opisu projektu powinno znajdować się sprawozdanie.

## 4 Termin i sposób nadsyłania rozwiązań

Za ostateczny termin oddania zadania uznaje się środę 12.06.2019, godzinę 23:59:59. Za każdy rozpoczęty dzień spóźnienia odejmowany jest jeden punkt.

Zadania wysyłane są na adres mailowy **macwozni@agh.edu.pl**. Temat wiadomości powinien być następującej postaci: **[FORTRAN] Nazwisko Imie Zad3** - np. **[FORTRAN] Kowalski Jan Zad3**.

Niepoprawnie zaadresowane rozwiązania nie zostaną ocenione (0 pkt.).