

M E T O D Y N U M E R Y C Z N E	WYDZIAŁ ELEKTRYCZNY W-5
ĆWICZENIA LABORATORYJNE	II ROK

Zad. 6	Zadanie należy wykonać w jednym z wymienionych języków programowania :	Borland Pascal, Turbo C, Javascript
	<p>Opracować funkcję avesum(...) umożliwiającą poprawione sumowanie nieskończonych szeregów naprzemiennych z uśrednianiem sum cząstkowych (patrz wykład).</p> $suma := \sum_{i=-\infty}^{\infty} a(i)$ <p><u>Przykład w wersji Borland Pascal:</u></p> <pre> {\$F+} { przetłacznik kompilatora dla adresowania typu „far” na początku programu, } { niezbędny tutaj do przekazywania funkcji poprzez zmienną typu funkcyjnego } type { ... } float = single; element = function(i : longint): float; { deklaracja typu funkcyjnego, nie funkcji ! } var suma1, poprawka1 : float; d, g : longint; { ... } function alter(i:longint):float; {przykładowa funkcja: minus jeden do potęgi “i” } begin if (i mod 2)=1 then alter:= -1.0 else alter:=1.0; end; function sinusone(i:longint):float; {przykładowy i-ty element szeregu} begin sinusone:=alter(i)/silnia(2*i+1); {funkcję silnia trzeba opracować ...} end; { ... } function avesum(a: element; d, g: longint; var err : float): float; { WE: a - zmienna funkcyjna dla i-tego elementu szeregu } { WE: d - dolna granica zmienności indeksu dla a() } { WE: g - górna granica zmienności indeksu dla a() } { WY: err - wyprowadzenie poprawki } { WY: avesum - wyprowadzenie końcowej sumy szeregu } { ... } begin {należy zaadaptować podaną na wykładzie procedurę sumowania z } {uśrednianiem sum cząstkowych} { ... } v := a(i) ; { obliczanie i-tego elementu szeregu definiowanego podczas wywołania funkcji } { ... } err := p; { wyprowadzenie do miejsca wywołania poprawki G-M} avesum := { ... } { wyprowadzenie końcowej poprawionej sumy szeregu } end; *) { ... } - w tych miejscach można/należy odpowiednio program uzupełnić ... <u>Przykład wywołania:</u> begin d := 0; g := 15; suma1 := avesum(sinusone, d, g, poprawka1); writeln(‘Wartość sumy „avesum” = ‘, suma1:15 ,’ Poprawka1 = ‘, poprawka1:15); readln; {zakończenie programu dopiero po naciśnięciu klawisza Enter} end. </pre>	

Kierownik kursu: doc. dr inż. Jarosław Szymańda	Zespół dydaktyczny: dr inż. L. Ładniak, dr hab. inż. J. Rezmer, dr inż. J. Piotrowicz, dr inż. L. Woźny
--	--

Przykład wykonania:

Testować funkcje **avesum(...)** dla podanych poniżej szeregów z dolną i górną granicą obcięcia. Każdy z podanych i-tych elementów szeregu należy odpowiednio zaprogramować według podanego w opisie zadania przykładu **sinusone**.

Obliczane wyniki przedstawiać w następującym układzie (#9 – kod tabulatora):

```
{...}
writeln('arctan(1.0)');
writeln('dolna_granica',#9, 'górna_granica',#9, 'avesum',#9, 'wartosc_dokladna');
{...}
writeln(d:3,#9,g:3,#9, suma1:15,#9, arctan(1.0):15);
{...}
```

Przykładowe i-te elementy testowanych szeregów:

{uwaga: w zależności od przyjętej reprezentacji, obliczenia sum szeregów dla niektórych podanych w tabeli górnych granic „g” mogą przekroczyć dopuszczalny zakres wartości zmiennopozycyjnych. W takich przypadkach obliczenia należy zakończyć na największej dopuszczalnej wartości „g”}

a(i)	d	g	Wartość dokładna
$\frac{(-1)^i}{(2i+1)}$	0	2, 6, 10, 50, 100, 500	arctan(1.0)
$\frac{(-1)^i}{(2i+1)!}$	0	2, 6, 10, 50, 100, 500	sin(1.0)
$\frac{(-1)^i}{i!}$	0	2, 6, 10, 50, 100, 500	1.0/e
$\frac{(-1)^{i+1}(x-1)^i}{i}$	1	2, 6, 10, 50, 100, 500	ln x ; dla x= 2.0