

Chylak_Maciej_305699_Maaco_pp1

Praca projektowa nr 1

Rozwiązanie zadan

Maciej Chylak 305699 IAD

1.Calka

Opis

Funkcja pozwalająca przybliżyć całkę oznaczoną funkcji **f** na przedziale **[a,b]** przy użyciu metody trapezów.

Użycie

`calka(f, a, b, n)`

`calka(f, a, b)`

Argumenty

f - Funkcja, której całkę chcemy przybliżyć.

a - Liczba rzeczywista. Lewy kraniec przedziału całkowania funkcji, według której chcemy policzyć całkę oznaczoną.

b - Liczba rzeczywista. Prawy kraniec przedziału całkowania funkcji, według której chcemy policzyć całkę oznaczoną.

n - Liczba naturalna większa od zera. Liczba punktów podziałowych przedziału **[a,b]**.

Wartość

Całka oznaczona funkcji **f** na przedziale **[a,b]**. *Integer* jeżeli to możliwe, w pozostałych przypadkach *double*.

Przykłady

Domyślne użycie `calka`:

```
calka(function(x){x^2 + 2}, 1, 6, 100)
```

```
## [1] 81.66875
```

Użycie `calka` bez podania argumentu *n* (domyślnie ustawiony *n* = 100):

```
calka(function(x){2*x - 10}, 1, 6)
```

```
## [1] -15
```

Użycie `calka` z wbudowaną funkcją *sin*:

```
calka(sin, 1, 6, 100)
```

```
## [1] -0.4197805
```

Złe użycie `calka`. n musi być większe od zera

```
calka(sin, 1, 6, -2)
```

Złe użycie `calka`. n musi być liczbą naturalną.

```
calka(sin, 1, 6, 0.5)
```

2. Sklej

Opis

Funkcja pozwalająca nam skleić **i-te** elementy wszystkich wektorów z listy **x**. Poszczególne elementy zostają oddzielone znakiem określonym przez parametr **sep**.

Użycie

```
sklej(x, sep)
```

```
sklej(x)
```

Argumenty

x - Lista zawierająca wektory napisów o tej samej długości.

sep - Napis oddzielający poszczególne elementy sklejanych napisów. Domyślnie ustawiony jako: " ".

Wartość

Wektor sklejonych napisów.

Przykłady

Domyślne użycie `sklej`:

```
sklej(list(c("a", "b", "c"), c("1", "2", "3")), ".")
```

```
## [1] "a.1" "b.2" "c.3"
```

Użycie `sklej` bez podania argumentu `sep` (domyślnie ustawiony `sep = ""`):

```
sklej(list(c("a", "b", "c"), c("1", "2", "3")))
```

```
## [1] "a1" "b2" "c3"
```

Przykład złego użycia funkcji `sklej`. x powinien być listą:

```
sklej(c("a", "b", "c"), ".")
```

Przykład złego użycia funkcji `sklej`. `sep` powinien być napisem:

```
sklej(list(c("a", "b", "c"), c("1", "2", "3")), 1)
```

3. Repr.macierz

Opis

Funkcja przekształcająca podaną macierz rzadką **x**, czyli taką, dla której większość komórek zawiera wartości zerowe, do macierzy składającej się z trzech kolumn reprezentujących wiersz i kolumnę, w którym niezerowy

element się znajduje, oraz jego wartość. Za element niezerowy uznaje się taki, który różni się co najmniej o **eps** od zera.

Użycie

```
repr_macierz(x, eps)
```

```
repr_macierz(x)
```

Argumenty

x - Macierz liczbowa, którą będziemy przekształcać.

eps - Liczba rzeczywista nieujemna. Najmniejsza liczba uznawana za niezerową co do wartości bezwzględnej. Domyślnie **eps** = 1e-16.

Wartość

Macierz posiadająca trzy kolumny: *row*, *col* oraz *value* oraz *k* wierszy, gdzie *k* oznacza liczbę elementów uznanych jako niezerowe. Wartości w kolumnach *row* oraz *col* reprezentują, w którym wierszu oraz kolumnie znajdują się niezerowy element, a *value* określa wartość tego elementu.

Przykłady

Domyślne użycie `repr_macierz`:

```
repr_macierz(matrix(c(0, 0.00005, 0.01, 0.00001, 2, 4, 1, 2, 0.004, 0.1), ncol = 2), 0.01)
```

```
##      row col value
## [1,]   3   1 0.01
## [2,]   5   1 2.00
## [3,]   1   2 4.00
## [4,]   2   2 1.00
## [5,]   3   2 2.00
## [6,]   5   2 0.10
```

Użycie `repr_macierz` bez podania argumentu *eps* (domyślnie ustawiony *eps* = 0.01):

```
repr_macierz(matrix(c(0, 0.00005, 0.01, 0.00001, 2, 4, 1, 2, 0.004, 0.1), ncol = 2))
```

```
##      row col value
## [1,]   2   1 5e-05
## [2,]   3   1 1e-02
## [3,]   4   1 1e-05
## [4,]   5   1 2e+00
## [5,]   1   2 4e+00
## [6,]   2   2 1e+00
## [7,]   3   2 2e+00
## [8,]   4   2 4e-03
## [9,]   5   2 1e-01
```

Przykład złego użycia funkcji `repr_matrix`. *x* powinien być macierzą:

```
repr_macierz(list(c(1,2, 4)), 0.01)
```

Przykład złego użycia funkcji `repr_matrix`. *eps* powinno być większe bądź równe 0:

```
repr_macierz(matrix(c(0, 0.00005, 0.01, 0.00001, 2, 4, 1, 2, 0.004, 0.1), ncol = 2), -1)
```

4. Logiderle

Opis

Funkcja sprawdzająca, czy istnieje takie \mathbf{p} , że element wektora $\mathbf{1:n}$ należy do przedziału $[\mathbf{i_p}, \mathbf{j_p}]$. W przypadku powodzenia zwraca *TRUE*, w przeciwnym przypadku *FALSE*.

Użycie

logiderle(i, j, n)

Argumenty

i - Wektor całkowitoliczbowy długości **k**, taki, że dla każdego p : $1 \leq i_p \leq j_p$ oraz $i_{p+1} > j_p$. Wartości tego wektora będą tworzyły lewy kraniec przedziału.

j - Wektor całkowitoliczbowy długości **k** taki, że dla każdego p : $1 \leq i_p \leq j_p$ oraz $i_{p+1} > j_p$. Wartości tego wektora będą tworzyły prawy kraniec przedziału.

n - Liczba całkowita dodatnia. Wartość ta przyczyni się do stworzenia wektora liczbowego postaci $1:n$ oraz określi długość wektora wynikowego.

Wartość

Wektor logiczny długości **n**.

Przykłady

Domyślne użycie logiderle:

```
logiderle(c(1, 3, 6), c(1, 4, 9), 11)
```

```
## [1] TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE TRUE FALSE FALSE
```

Przykład złego użycia funkcji logiderle. dla każdego p i_p ma być nie większe od j_p :

```
logiderle(c(1, 3, 6), c(1, 2, 9), 11)
```

Przykład złego użycia funkcji logiderle. dla każdego p i_{p+1} ma być większe od j_p :

```
logiderle(c(1, 1, 6), c(1, 1, 9), 11)
```