

Pracownia nr 1

10.10.2009

1 Zadanie : Ciąg arytmetyczny

Zacniemy od konstrukcji ciągu liczbowego. Chcemy dla danej liczby $n > 1$ i danego wyrazu a_1 utworzyć kolejne wyrazy a_2, \dots, a_n ciągu arytmetycznego o zadanej różnicy r .

1.1 Zaczynamy pracę

1. Tworzymy pierwsze wyrazy ciągu: a_1, a_2, a_3

```
>> n = 1;
>> n = 2;
>> n;
>> r = n;
>> a1 = 0;
>> a2 = r;
>> a3 = a2 + r;
```

2. Wyznamy kolejne wyrazy ciągu na podstawie wzoru. Przypomnijmy wzór na n -ty wyraz ciągu arytmetycznego:

$$a_n = a_1 + (n - 1) \cdot r$$

```
>> n = 4
>> an = a1 + (n-1)*r
```

3. Chcemy wyznaczyć kolejne wyrazy. Aby nie przepisywać ponownie wykonanych kroków, zaznaczamy interesujące nas komendy w oknie *Command History* i naciskając prawy klawisz myszy wybieramy *Create M-File*. Utworzymy *skrypt* - plik z listą komend, które chcemy wykonać. Nazwiemy go **skrypt_an.m**. Dopiszmy jeszcze na końcu każdej linii średnik - znak ';'. Po co? Zobaczmy.
4. Teraz wystarczy zmienić wartość zmiennej **n** w skrypcie aby uzyskać kolejne wyrazy. Wykonamy to kolejno dla $n = 5, 6$.

```
>> cls
>> skrypt_an
```

5. Przy okazji widzimy że polecenie **cls** wyczyści ekran z dotychczas wpisanych poleceń.

1.2 Ciąg jako funkcja matematyczna i jako funkcja w komputerze

Ciąg liczbowy to *funkcja matematyczna* czyli odwzorowanie f określone na zbiorze liczb naturalnych o wartościach w zbiorze liczb rzeczywistych:

$$f : N \rightarrow R \quad f(n) = a_n$$

A jak możemy opisać funkcję f w komputerze?

1. Posługiwanie się skryptem `skrypt_an` nie jest wygodne. Spróbujmy zrobić to samo sprytniej. Utworzymy *funkcję*, czyli krótki program wykonujący obliczenie wartości $f(n) = a_n$. Musimy w tym celu zmienić trochę nasz skrypt. Usuńmy linie:

```
n = 1;
n = 2;
n;
a2 = r;
a3 = a2 + r;
```

2. Następnie dopiszemy na początku pliku:

```
function an = f(n)
```

oraz na końcu:

```
end function
```

zapiszmy teraz plik jako `f.m`.

3. Teraz `f` jest funkcją, spróbujmy jej użyć:

```
>> a2 = f(2)
>> a11 = f(11)
```

Hmm... Wynik nie jest poprawny. Dlaczego?

4. Wyrazy ciągu arytmetycznego zależą także od wartości różnicy r . Powinna być ona także parametrem funkcji `f`. Zmieniamy plik z naszą funkcją usuwając linię przypisującą wartość zmiennej `r` oraz nagłówek funkcji:

```
function an = f(n, r)
```

5. Co jest jeszcze parametrem funkcji `f`? Dlaczego?
6. *Uwaga*: Nazwa pliku zawierającego funkcję i nazwa tej funkcji muszą być takie same! (tak jak: `f` i `f.m`)

1.3 Reprezentacja ciągu arytmetycznego w komputerze

Umiemy wyznaczyć n -ty wyraz ciągu arytmetycznego. Chcemy jednak mieć cały ciąg - wszystkie n wyrazów. Spróbujmy to zrobić.

1. Nasza funkcja `f` ostatecznie wygląda tak:

```
function an = f(n, r, a1)
    an = a1 + (n-1)*r;
end function
```

2. Użyjemy naszej funkcji do wyznaczenia n -elementowego ciągu a . Ciąg a zapiszemy jako *tablicę*:

```
>> a = f(1:11,2,0)
```

O rany! O co tu chodzi? Dlaczego to działa?!

3. Przekazaliśmy jako parametr funkcji cały zbiór liczb od 1 do 11. Dwukropek ':' jest tak zwanym operatorem zakresu, zobaczmy jak to działa:

```
>> 1:11
```

4. W wyniku tej sztuczki otrzymaliśmy ciąg 11 elementów zapisany w tablicy. Spójrzmy do okna przestrzeni roboczej **Workspace**. Co tam możemy zobaczyć?
5. Narysujemy elementy ciągu na wykresie:

```
plot(a, '+' )
```