

Nest, NestList i metoda Newtona

W poniższych zadaniach będziemy używali funkcji `Nest` oraz `NestList`. Przykładowo, funkcja `Nest[f,a,3]` zwraca `f[f[f[a]]]`, a funkcja `NestList[f,a,3]` zwraca `{a, f[a], f[f[a]], f[f[f[a]]]}`

Zadanie 1. Uruchom polecenia:

```
NestList[2 #1 &, 1, 10]
Nest[1/(1 + #) &, x, 6]
Nest[Subsuperscript[#, #, #] &, x, 6]
```

Zastanów się, czy rozumiesz, co się stało.

Zadanie 2. Znajdź punkt stały funkcji `cos`

Metoda Newtona 1D

Zadanie 3. Wyznaczyć metodą Newtona $\sqrt{2}$ z dokładnością do 40 cyfr dziesiętnych.

Rozwiązanie:

```
f[x_] = x^2 - 2;
h[x_] = x - f[x]/f'[x];
x0 = 1.0;
```

```
NestList[h, x0, 10] // TableForm
```

 Wyświetla 10 pierwszych iteracji

$$x_0, h(x_0), \dots, h^{10}(x_0).$$

Dokładność ostatniego przybliżenia jest za mała.

Zwiększmy zatem dokładność obliczeń

```
NestList[h, SetPrecision[x0, 50], 10] // TableForm
```

 Porównajmy ostatni wynik z
`N[Sqrt[2], 40]`

Zadanie 4. Utwórz graficzną ilustrację metody tworzenia ciągu kolejnych przybliżeń w metodzie Newtona.

Rozwiązanie: Tak jak wyżej definiujemy (jeśli jeszcze tego nie zrobiliśmy)

```
f[x_] = x^2 - 2;
h[x_] = x - f[x]/f'[x];
x0 = 1.0;
```

Zacznijmy od poruszania punktem początkowym

```
a=1; b=3;
Manipulate[
  Plot[f[x], {x, a, b},
  Epilog -> {{PointSize[0.015], Point[{x0, 0]}}},
  ImageSize -> {600, 325}],
  {{x0, a}, a, b, Appearance -> "Labeled" }]
```

W sekcji Epilog wstawiliśmy polecenie wyświetlenia punktu $(x_0, 0)$
 Utworzymy teraz łamaną. Najpierw zdefiniujemy listę kolejnych przybliżeń:
`lista = NestList[h, x0, 5];`
 Otrzymamy ciąg

$$(x_0, x_1, x_2, x_3, x_4, x_5).$$

Do każdej liczby tej listy stosujemy funkcję $\# \rightarrow \{\{\#, 0\}, \{\#, f[\#]\}\}$:
`lista2 = Map[\{\{\#, 0\}, \{\#, f[\#]\}\} &, lista]`
 Otrzymamy

$$\left(((x_0, 0), (x_0, f[x_0])), ((x_1, 0), (x_1, f[x_1])), \dots, ((x_5, 0), (x_5, f[x_5])) \right)$$

Usuńmy teraz wewnętrzne nawiasy:

`lista3 = Flatten[lista2, 1]`

Otrzymamy

$$\left((x_0, 0), (x_0, f[x_0]), (x_1, 0), (x_1, f[x_1]), \dots, (x_5, 0), (x_5, f[x_5]) \right)$$

W liście tej ostatni element jest niepotrzebny:

`lista4 = Most[lista3]`

Otrzymamy ciąg

$$\left((x_0, 0), (x_0, f[x_0]), (x_1, 0), (x_1, f[x_1]), \dots, (x_5, 0) \right)$$

Teraz można narysować łamaną

`Graphics[Line[lista4]]`

lub łamaną której ostatni odcinek jest strzałką

`Graphics[Arrow[lista4]]`

Wpiszmy to na koniec do Manipulate

```
Manipulate[
  lista = NestList[h, x0, 5];
  lista2 = Map[\{\{\#, 0\}, \{\#, f[\#]\}\} &, lista];
  lista3 = Flatten[lista2, 1];
  lista4 = Most[lista3];
  Plot[f[x], {x, a, b},
    Epilog -> {Red, Thickness[.002], Arrowheads[.03], Arrow[lista4],
      PointSize[0.015], Point[\{x0, 0\}], ImageSize -> {600, 325}},
    {\{x0, a, Subscript["x", "0"]}, a, b, Appearance -> "Labeled" }
```

Zadanie 5. W przedziale $[-1.5, 2]$ funkcja $f(x) = 3x(x+1)(x-1.5)$ ma trzy miejsca zerowe $x_0 = -1$, $x_1 = 0$ i $x_2 = 1.5$. Używając Manipulate z poprzedniego zadania wyznaczyć w przybliżeniu obszary przyciągania układu dynamicznego $h(x)$ do punktów x_0 , x_1 i x_2 . Znaleźć metodą graficzną punkt x , którego orbita jest 2-elementowa. Wyznaczyć ten punkt numerycznie.

Metoda Newtona 2D

Zadanie 6. W obszarze $\{x + yi : |x| < 2, |y| < 2\}$ funkcja $f(z) = z^3 - 1$ ma trzy miejsca zerowe $z_0 = 1$, $z_1 = \cos 2\pi/3 + i \sin 2\pi/3$, $z_2 = \cos 2\pi/3 - i \sin 2\pi/3$. Wyznaczyć graficznie obszary przyciągania układu dynamicznego $h(z)$ do z_0 , z_1 i z_2 .

Rozwiązanie:

Zdefiniujmy najpierw

```
f[z_] = z^3 - 1;
h[z_] = z - f[z]/f'[z];
a1 = -2; a2 = 2; b1 = -2; b2 = 2;
```

oraz punkty z_0, z_1 i z_2 (pierwiastki trzeciego stopnia z 1).

Zamiast $\lim_{n \rightarrow \infty} h^n(z)$ weźmiemy w przybliżeniu $h^{20}(z) = \text{Nest}[h, N[x + y I], 20]$

Utwórzmy tabelę wartości tej funkcji na siatce $[a_1, a_2] \times [b_1, b_2]$ z krokiem 0.05.

```
n = 20;
delta = 0.05;
tablwart = Table[Nest[h, N[x + y I], n], {y, b1, b2, delta}, {x, a1, a2, delta}];
```

Wartości w tej tablicy to (w przybliżeniu) z_0, z_1, z_2 i być może jeszcze inne liczby. Sprawdźmy:

```
tablwart[[1, 1]]
tablwart[[-1, 1]]
tablwart[[-1, -1]]
```

Przygotowując się do wyświetlenia tablicy `tablwart` kolorami, zamienimy każdą wartość zbliżoną do z_0 na 0, do z_1 na 1, i do z_2 na 2, pozostałe wartości na -1 . W tym celu zdefiniujemy funkcję

```
eps = 0.001;
r[x_] = Piecewise[
  {{0, Abs[x - z0] < eps},
   {1, Abs[x - z1] < eps},
   {2, Abs[x - z2] < eps}}, -1];
```

Zdefiniujmy więc

```
tablwart2 = Table[r[Nest[h, N[x + y I], n]], {y, b1, b2, delta}, {x, a1, a2, delta}];
```

i sprawdźmy

```
tablwart2[[1, 1]]
tablwart2[[-1, 1]]
tablwart2[[-1, -1]]
```

Grafika

Do graficznego wyświetlenia tablicy liczb użyjemy polecenia `ArrayPlot`

Najpierw kilka przykładów

```
1. tab1 = {{1, -1, 0, 1}, {1, 1, 0, 1}, {1, 0, 1, -1}};
   ArrayPlot[tab1, ColorRules -> {1 -> Pink, 0 -> Yellow, -1 -> Red}]
```

2. Tablica 40×40 liczb losowych z odcinka $[0, 1]$:

```
tab2 = RandomReal[1, {40, 40}];
ArrayPlot[tab2, ColorFunction -> "Rainbow"]
```

```
3. ArrayPlot[Table[Sin[x y], {x, -40, 40}, {y, -40, 40}]]
```

Wracamy do zadania głównego. Wyświetlmy teraz `tablwart2` w postaci tablicy kolorów

```

ArrayPlot[
  tablwart2,
  ColorRules -> {0 -> Yellow, 1 -> Blue, 2 -> Red, -1 -> Black},
  DataRange -> {{a1, a2}, {b1, b2}},
  DataReversed -> True, Axes -> True, Frame -> True,
  FrameTicks -> True, ImageSize -> {400, 400}]

```

Obszar zaznaczony na żółto - obszar przyciągania do z_0 ,

Obszar zaznaczony na niebiesko - obszar przyciągania do z_1

Obszar zaznaczony na czerwono - obszar przyciągania do z_2

Obszar zaznaczony na czarno, to obszar rozbieżności ciągu przybliżeń.

Jeżeli wszystko poszło dobrze, zmienimy wartość kroku **delta** w tablicy z 0.05 na 0.01 i ponownie wyświetlmy obraz.

Zadanie 7. Wykonać „zoom” grafiki, np. w obszarze $[0.2, 0.6] \times [0.1, 0.5]$

Uwaga: Żeby obrazek ładnie wyglądał trzeba zmniejszyć **delta**

Zadanie 8. W obszarze $\{x + yi : |x| < 2, |y| < 2\}$ funkcja $f(z) = z^5 - 1$ ma pięć miejsc zerowych. z_0, z_1 i z_2, z_3, z_4 . Wyznaczyć graficznie obszary przyciągania układu dynamicznego $h(z)$ do z_i .

Zadanie 9 ().** Zróżnicować kolor w zależności od szybkości zbieżności ciągu kolejnych przybliżeń.