Autor: Karolina Tatarczyk

Metody numeryczne (Matematyka)

Projekt 1

Metoda bisekcji

Napisać procedurę realizującą algorytm metody bisekcji (argumenty : f, a, b, e).

Korzystając z napisanej procedury:

a) Wyznaczyć wszystkie rzeczywiste pierwiastki wielomianu: $f(x) = x^5 + 4x^3 + 2x - 8$.

b) Wyznaczyć (z dokładnością 10^{-6}) najmniejszy dodatni pierwiastek funkcji: $h(x) = \exp(-x^2)\sin x - \cos x$.

c) Znaleźć (z dokładnością do sześciu cyfr po przecinku) największą liczbę $x \in (0, 40]$, dla której jest spełniona nierówność: $10 \ln x + 5 \leq x$.

Rozwiązanie

Program

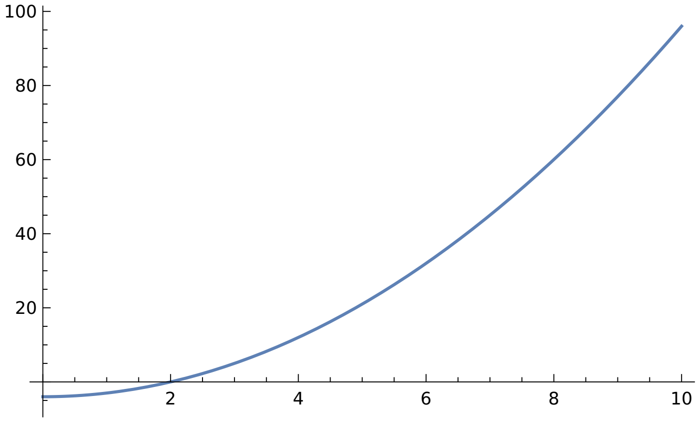
In[12]:=
```
Clear[bisection]
bisection[f_, a_, b_, e_] :=
 Module[{xsi, a1 = a, b1 = b, epsilon1 = e}, xsi = (a1 + b1) / 2;
  While[Abs[b1 - a1] > (2 * epsilon1),
   If[f[xsi] == 0, Return[xsi], If[f[a1] * f[xsi] < 0, b1 = xsi, a1 = xsi]];
   xsi = (a1 + b1) / 2;];
  Return[xsi];]
```

Przykład testowy:

In[45]:=
```
r[x_] := x ^ 2 - 4
```

In[46]:=
```
Plot[r[x], {x, 0, 10}]
```

Out[46]=



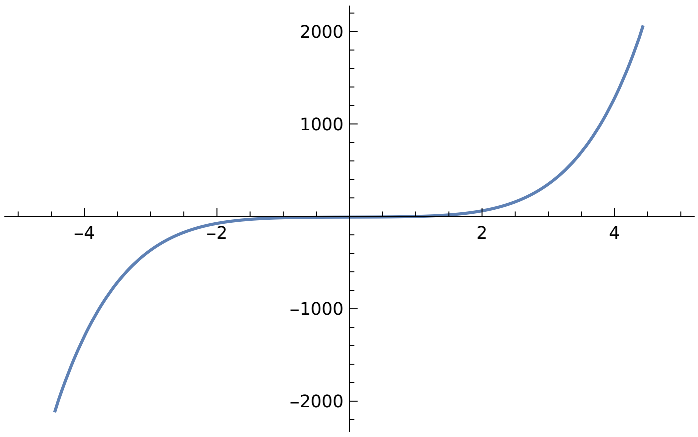
In[47]:=
```
bisection[r, 0, 10, 0.00001] // N
```

Out[47]= 2.

Zadanie a)

In[195]:=
```
f1[x_] := x^5 + 4 x^3 + 2 x - 8
Plot[f1[x], {x, -5, 5}]
```

Out[196]=



```
g1[x_] := D[f1[x], x]
Limit[g1[x], x → ∞] > 0
Limit[g1[x], x → -∞] > 0
f1[-∞] * f1[∞] < 0
```

Out[198]=

```
True
```

Out[199]=

```
True
```

Out[200]=

```
True
```

In[5]:=

```
bisection[f1, -5, 5, 0.0001] // N
```

Out[201]=

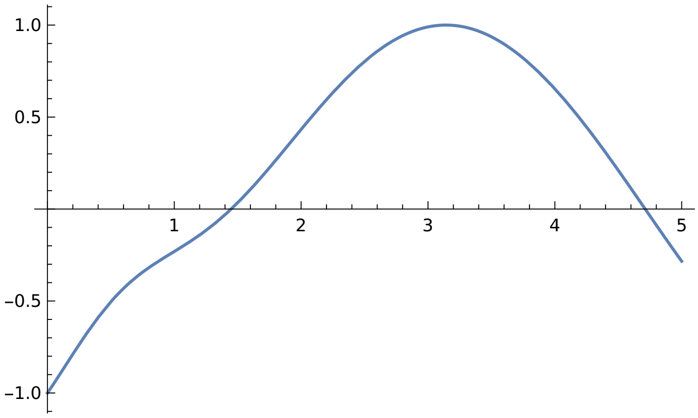
```
1.04973
```

Zadanie b)

In[63]:=

```
h[x_] := Exp[-x^2] * Sin[x] - Cos[x]
Plot[h[x], {x, 0, 5}] (*założenie o izolacji pierwiastka*)
```

Out[64]=



In[65]:=

```
bisection[h, 0, 3, 0.000001] // N
```

Out[65]=

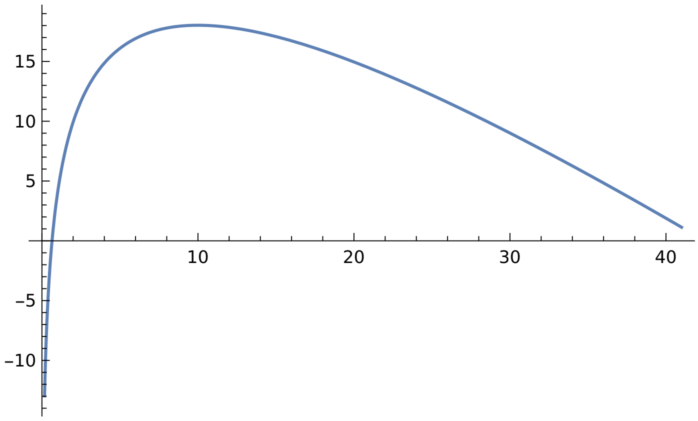
```
1.44884
```

Zadanie c)

In[52]:=

```
p[x_] := 10 Log[x] - x + 5
Plot[p[x], {x, 0, 41}]
```

Out[53]=



In[54]:=

```
bisection[p, 0, 40, 10^(-6)] // N
```

Out[54]= 0.647076

In[55]:=

```
p[bisection[p, 0, 40, 10^(-6)]] // N
```

Out[55]= 5.44703×10^{-6}

- (*wartość ta nie spełnia nierówności podanej w zadaniu*)

```
Clear[bisect];
bisect[f_, a_, b_, epsilon_] :=
 Module[{xsi, a1 = a, b1 = b, epsilon1 = epsilon}, xsi = (a1 + b1) / 2;
  While[Abs[b1 - a1] > (2 epsilon1),
   If[f[xsi] == 0, Return[xsi], If[f[a1] * f[xsi] < 0, b1 = xsi, a1 = xsi]];
   xsi = (a1 + b1) / 2;];

  If[f[xsi] > 0, xsi = xsi - epsilon];
  (* odejmuję przybliżenie by zgadzała się nierówność*)
  Return[xsi];]
bisect[p, 0, 40, 10^(-6)] // N
```

Out[58]= 0.647075

In[59]:=
```
p[bisect[p, 0, 40, 10^(-6)]] // N
```
Out[59]= -9.00713×10^{-6}

```
(*Przy podanych danych wartość przybliżonego pierwiastka,
który został wyliczony z procedury 'bisection' jest większa od 0,
co nie jest poszukiwaną przez nas wartością,
dlatego stworzyłam drugą procedurę z dodatkowym warunkiem,
że gdy wartość w naszym przybliżonym miejscu zerowym jest
 większa od zera to odejmujemy od niego wartość przybliżenia,
co pozwala na znalezienie możliwie najmniejszej wartości 'x'*)
```

In[60]:=
```
Plot[p[x], {x, 0.64, 0.65}]
```
Out[60]=