

Autor: Karolina Tatarczyk

Metody numeryczne w technice

(kierunek Matematyka)

Projekt 2

Metoda Adamsa-Bashfortha

Napisać procedurę realizującą algorytm trzy krokowej metody Adamsa-Bashfortha

(argumenty: f , x_0 , y_0 , b , n).

Zminimalizować liczbę obliczeń funkcji f . Jako metodę startową wykorzystać metodę Rungego-Kutty rzędu trzeciego.

Korzystając z napisanej procedury wyznaczyć rozwiązanie przybliżone zagadnienia początkowego:

$$\begin{cases} y'(x) = \sqrt[3]{\frac{y(x)}{x^2}}, & x \in [1, 50], \\ y(1) = 1. \end{cases}$$

Obliczenia wykonać dla 10, 20 i 50 kroków.

Na wspólnym rysunku wykreślić rozwiązanie dokładne oraz uzyskane rozwiązania przybliżone. Wykreślić także, na jednym rysunku, błędy uzyskanych rozwiązań przybliżonych.

Rozwiązanie

Procedury

```

In[27]:= RungeKutty3[function_, X0_, Y0_, H_, number_] :=
  Module[{f = function, x0 = X0, y0 = Y0, h = H, n = number, x, y},
    x = {x0};
    y = {y0};
    For[i = 1, i ≤ n, i++,
      AppendTo[x, x[[i]] + h];
      k1 = f[x[[i]], y[[i]]];
      k2 = f[x[[i]] + h * 0.5, y[[i]] + h * k1 * 0.5];
      k3 = f[x[[i + 1]], y[[i]] - h * k1 + 2 * h * k2];
      AppendTo[y, y[[i]] + h * (k1 + 4 * k2 + k3) * 1 / 6];
    ];
    Return[Transpose[{x, y}]]
  ]

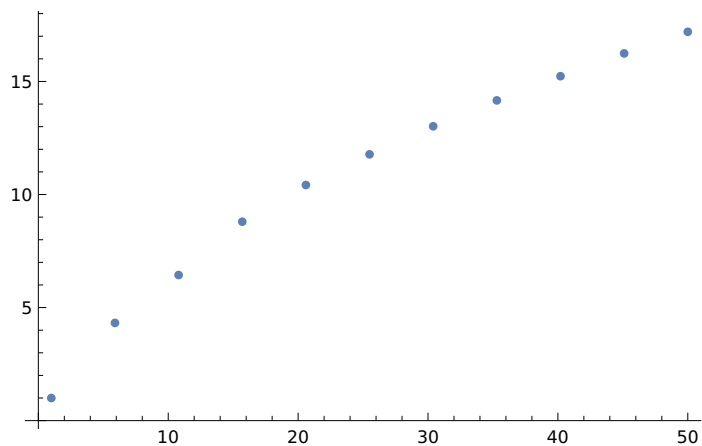
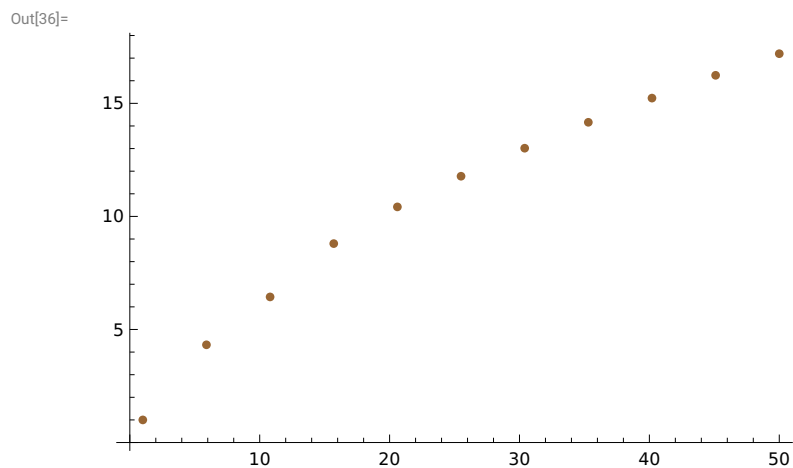
In[28]:= AdamsBashforth[function_, X0_, Y0_, B_, number_] :=
  Module[{f = function, x0 = X0, y0 = Y0, b = B, n = number, result},
    k = 3;
    h = (b - x0) / n;
    result = RungeKutty3[f, x0, y0, h, k - 1];
    listF = Table[f[result[[i, 1]], result[[i, 2]]], {i, 1, k, 1}];
    vectorB = {23 / 12, -16 / 12, 5 / 12};
    For[i = k, i ≤ n, i++,
      yn = result[[i, 2]] + h * Sum[vectorB[[j]] * listF[[i + 1 - j]], {j, 1, k, 1}];
      xn = result[[i, 1]] + h;
      AppendTo[listF, f[xn, yn]];
      AppendTo[result, {xn, yn}];
    ];
    Return[result]
  ]

```

Testy n=10

```
In[34]:= f[x_, y_] := Power[y/x^2, 1/3];  
AB10 = N[AdamsBashforth[f, 1, 1, 50, 10]]  
plot10 = ListPlot[AB10, PlotStyle -> Brown]
```

```
Out[35]= {{1., 1.}, {5.9, 4.32184}, {10.8, 6.43924},  
          {15.7, 8.79767}, {20.6, 10.4207}, {25.5, 11.7768}, {30.4, 13.0164},  
          {35.3, 14.1622}, {40.2, 15.2321}, {45.1, 16.2401}, {50., 17.1962}}
```

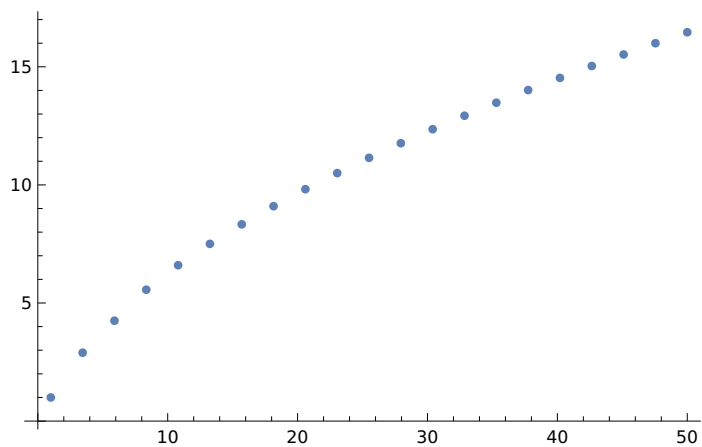


Testy n=20

```
In[7]:=
```

```
In[37]:= AB20 = AdamsBashforth[f, 1, 1, 50, 20];  
plot20 = ListPlot[AB20]
```

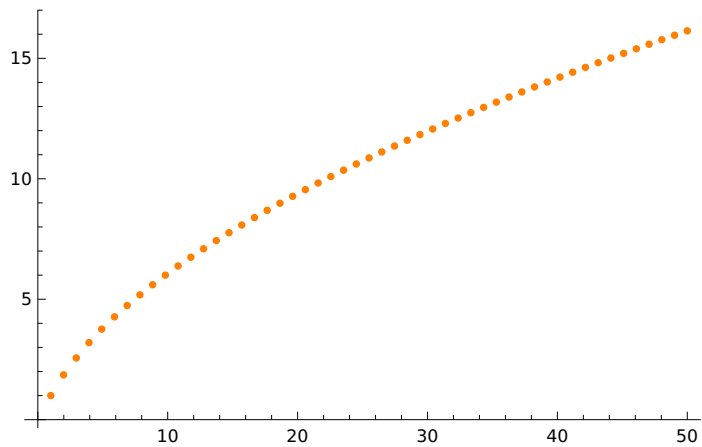
Out[38]=



Testy $n=50$

```
In[39]:= AB50 = AdamsBashforth[f, 1, 1, 50, 50];  
plot50 = ListPlot[AB50, PlotStyle -> Orange]
```

Out[40]=



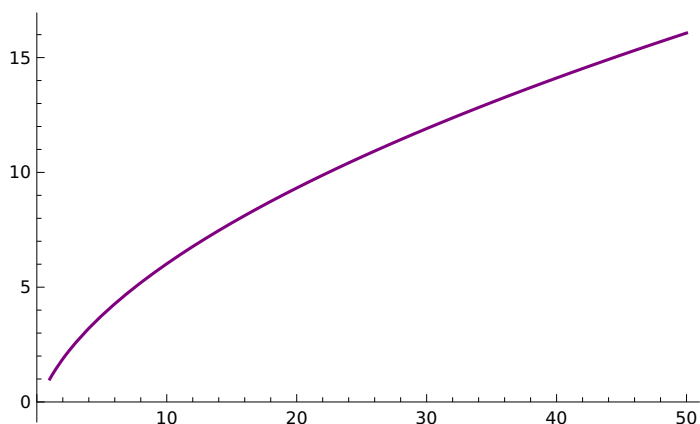
Dokładne rozwiązanie

```
In[41]:= result = DSolve[{y'[x] == Power[y[x]/x^2, 1/3], y[1] == 1}, y[x], x][[1, 1, 2]]  
plot = Plot[result, {x, 1, 50}, PlotStyle -> Purple]
```

Out[41]=

$$\left(-1 + 2 x^{1/3}\right)^{3/2}$$

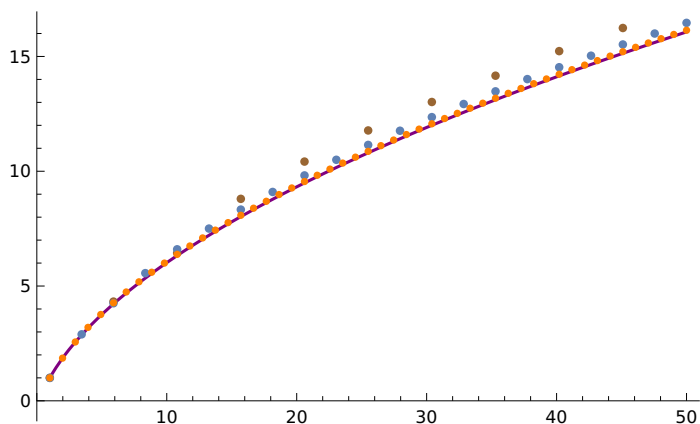
Out[42]=



Otrzymane wyniki

```
In[43]:= Show[plot, plot10, plot20, plot50]
```

Out[43]=

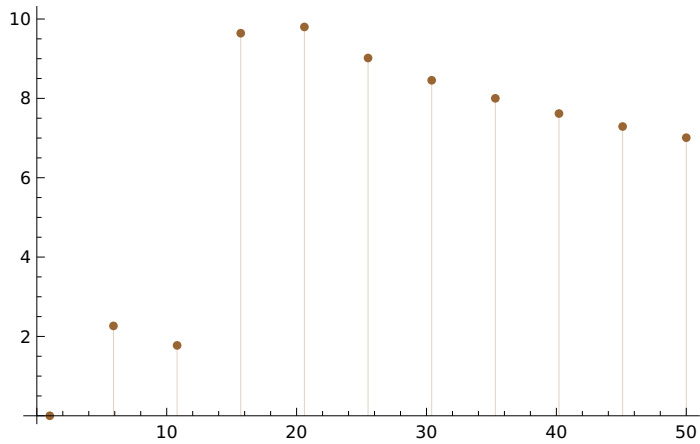


Błędy otrzymanych wyników

n=10

```
ListX10 = Transpose[AB10][[1]];
ListY10 = Transpose[AB10][[2]];
resultPoints10 = Table[result /. {x → ListX10[[i]]}, {i, 1, Length[ListX10]}];
bladbezwzględny10 = Abs[ListY10 - resultPoints10];
bladwzględny10 = 100 * bladbezwzględny10 / Abs[resultPoints10];
bar10 =
  ListPlot[Transpose[{ListX10, bladwzględny10}], PlotStyle → Brown, Filling → Axis]
```

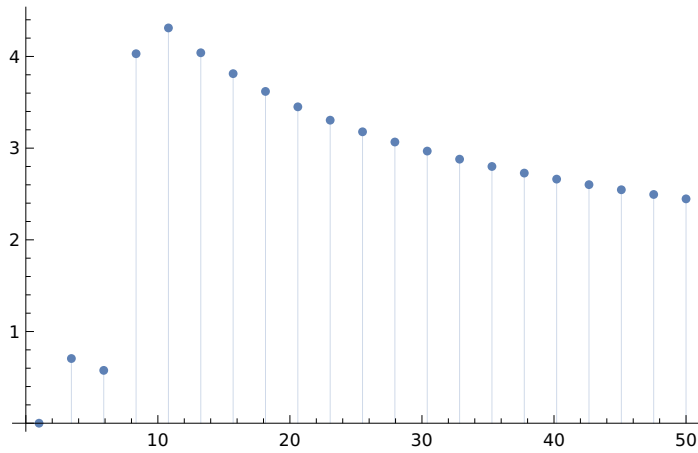
Out[49]=



n=20

```
In[50]:= ListX20 = Transpose[AB20][[1]];
ListY20 = Transpose[AB20][[2]];
resultPoints20 = Table[result /. {x → ListX20[[i]]}, {i, 1, Length[ListX20]}];
bladbezwzględny20 = Abs[ListY20 - resultPoints20];
bladwzględny20 = 100 * bladbezwzględny20 / Abs[resultPoints20];
bar20 = ListPlot[Transpose[{ListX20, bladwzględny20}], Filling → Axis]
```

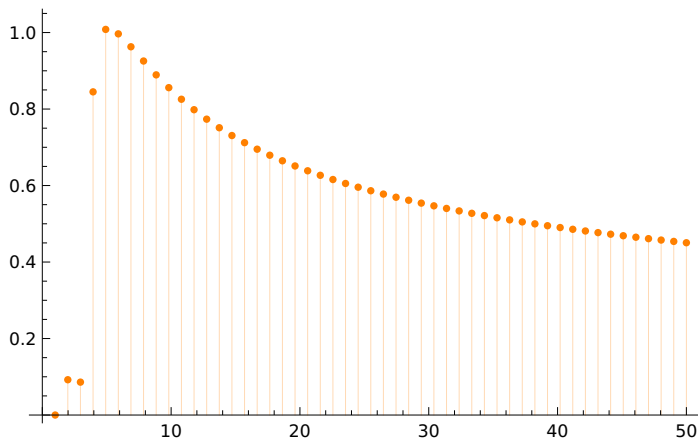
Out[55]=



n=50

```
In[56]:= ListX50 = Transpose[AB50][[1]];
ListY50 = Transpose[AB50][[2]];
resultPoints50 = Table[result /. {x → ListX50[[i]]}, {i, 1, Length[ListX50]}];
bladbezwzględny50 = Abs[ListY50 - resultPoints50];
bladwzględny50 = 100 * bladbezwzględny50 / Abs[resultPoints50];
bar50 =
ListPlot[Transpose[{ListX50, bladwzględny50}], PlotStyle → Orange, Filling → Axis]
```

Out[61]=



Wszystkie błędy na jednym rysunku

```
In[62]:= Show[bar10, bar20, bar50]
```

Out[62]=

