

Autor: Imię i nazwisko

Metody numeryczne w technice

(kierunek Matematyka)

Projekt 4

Metoda predyktor-korektor

Napisać procedurę realizującą algorytm metody predyktor-korektor (argumenty: f , x_0 , y_0 , b , n). Jako metodę startową wykorzystać metodę Rungego-Kutty rzędu trzeciego. Jako metodę predykcji wykorzystać trzy krokową metodę Adamsa-Bashfortha, a jako metodę korekcji trzy krokową metodę Adamsa-Moultona. W metodzie korekcji wykonać dwie iteracje metody iteracji prostej.

Korzystając z napisanej procedury wyznaczyć rozwiązanie przybliżone zagadnienia początkowego:

$$\begin{cases} y'(x) = 2 \sin x - y(x), & x \in [0, 15], \\ y(0) = 2. \end{cases}$$

Obliczenia wykonać dla 20 i 100 kroków.

Na wspólnym rysunku wykreślić rozwiązanie dokładne oraz uzyskane rozwiązania przybliżone. Wykreślić także, na jednym rysunku, błędy uzyskanych rozwiązań przybliżonych. Wyznaczyć także błędy maksymalne oraz średnie dla obu siatek.

Rozwiązanie

```
In[1]:= 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[109]:=

```

In[82]:= PredKor[function_, X0_, Y0_, B_, number_] :=
  Module[{f = function, x0 = X0, y0 = Y0, b = B, n = number},
    h = (b - x0) / n;
    k = 3;
    result = RungeKutty3[f, x0, y0, h, k];
    ListF = Table[f[result[[i, 1]], result[[i, 2]]], {i, 1, Length[result]};
    vectorBAB = {23/12, -16/12, 5/12};
    vectorBAM = {9/24, 19/24, -5/24, 1/24};
    For[i = k + 1, i ≤ n, i++,
      yn = result[[i, 2]] + h * Sum[vectorBAB[[j]] * ListF[[i + 1 - j]], {j, 1, 3, 1}];
      xn = result[[i, 1]] + h;
      phi[z_] := result[[i, 2]] +
        h * Sum[vectorBAM[[j + 1]] * ListF[[i + 1 - j]], {j, 1, 3, 1}] + h * vectorBAM[[1]] * f[xn, z];
      For[l = 1, l ≤ 2, l++,
        yn = phi[yn];
      ];
      AppendTo[ListF, f[xn, yn]];
      AppendTo[result, {xn, yn}];
    ];

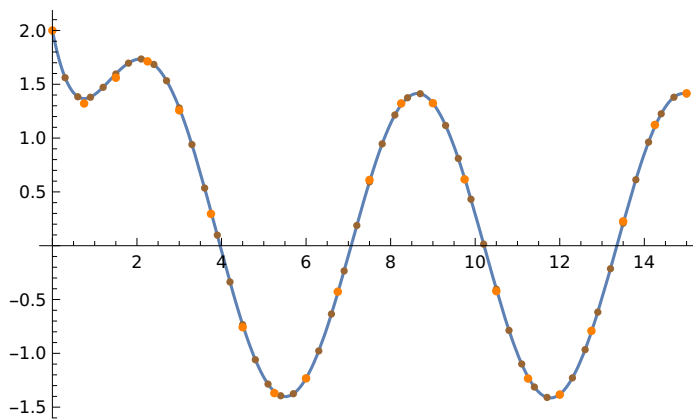
    Return[result]
  ]

```

Przykład dla $n=20$ i $n=50$

```
In[83]:= f[x_, y_] := 2 * Sin[x] - y  
acc = DSolve[{y'[x] == 2 * Sin[x] - y[x], y[0] == 2}, y[x], x][[1, 1, 2]];  
plot = Plot[acc, {x, 0, 15}];  
b = 15;  
x0 = 0;  
y0 = 2;  
PK20 = PredKor[f, x0, y0, b, 20];  
plot20 = ListPlot[PK20, PlotStyle -> Orange];  
PK50 = PredKor[f, x0, y0, b, 50];  
plot50 = ListPlot[PK50, PlotStyle -> Brown];  
Show[plot, plot50, plot20]
```

Out[93]=



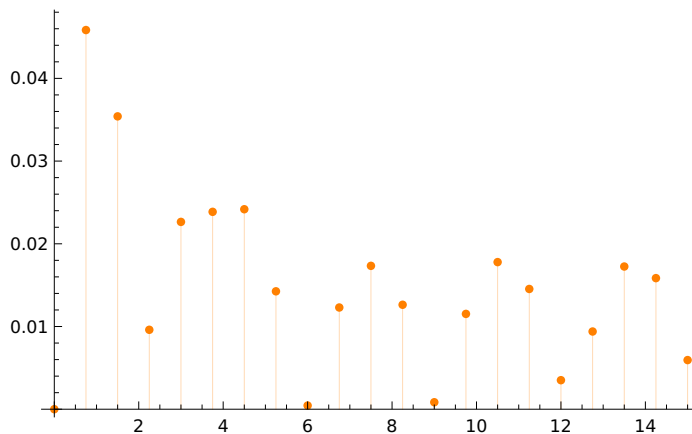
Błędy otrzymanych wyników dla $n=20$

```

In[94]:= ListX20 = Transpose[PK20][[1]];
ListY20 = Transpose[PK20][[2]];
resultPoints20 = Table[acc /. {x → ListX20[[i]]}, {i, 1, Length[ListX20]}];
bladbezwzględny20 = Abs[ListY20 - resultPoints20];
bar20 =
  ListPlot[Transpose[{ListX20, bladbezwzględny20}], PlotStyle → Orange, Filling → Axis]
Print["Błąd maksymalny: ", Max[bladbezwzględny20]]
Print["Średnia wartość błędu:", Mean[bladbezwzględny20]]

```

Out[98]=



Błąd maksymalny: 0.0458445

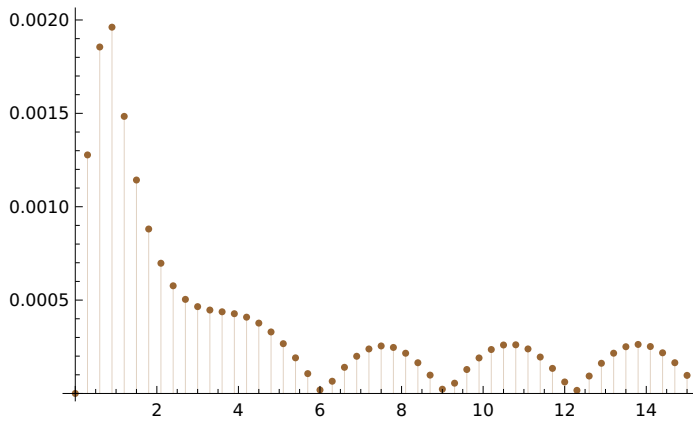
Średnia wartość błędu: 0.0150064

Błędy otrzymanych wyników n=50

In[101]:=

```
ListX50 = Transpose[PK50][[1]];
ListY50 = Transpose[PK50][[2]];
resultPoints50 = Table[acc /. {x → ListX50[[i]]}, {i, 1, Length[ListX50]}];
bladbezwzględny50 = Abs[ListY50 - resultPoints50];
bar50 = ListPlot[Transpose[{ListX50, bladbezwzględny50}],
  PlotStyle → Brown, Filling → Axis, PlotRange → Full]
Print["Błąd maksymalny: ", Max[bladbezwzględny50]]
Print["Średnia wartość błędu:", Mean[bladbezwzględny50]]
```

Out[105]=



Błąd maksymalny: 0.00196139

Średnia wartość błędu: 0.000372303

Błędy otrzymanych wyników

In[108]:=

Show[bar20, bar50]

Out[108]=

