Autor: Karolina Tatarczyk

Wybrane metody modelowania matematycznego

(kierunek Matematyka)

Projekt 2

Rozwiązanie

```
cieplo[Alpha_, rangeA_, rangeB_, number_, M_, Tg_, function_, U0_, UA_, UB_, U_] :=
  Module[{f = function, alpha = Alpha, a = rangeA, b = rangeB,
    n = number, m = M, tg = Tg, u0 = U0, ua = UA, ub = UB, u = U},

(*Obliczenie parametrów *)
h = (b - a) / (n - 1);
k = tg / (m - 1);
w1 = alpha * k;
   w2 = -(2 * alpha * k + h ^ 2);
   w3 = h ^ 2;
   w4 = -k * h ^ 2;
(*uzupełnianie macierzy A i B
w  tym przypadku macierz U zostaje spłaszczona na potrzeby rozwiązania
wadome współczynniki z macierzy A zostają przerzucone do wektora B*)
A = {};
B = {};
   xt = Table[{a + (i - 1) * h, (j - 1) * k}, {i, 1, n}, {j, 1, m}];
   i = 2; j = 2;
   row = Table[0, {k, 1, n * m - n - 2 * m + 2}];
row[[(i - 2) * (m - 1) + j - 1]] = w2;
```

```
row[[(i - 1) * (m - 1) + j - 1]] = w1;
AppendTo[A, row];
AppendTo[B, w4 * f[xt[[i, j, 1]], xt[[i, j, 2]]] - w1 * ua[xt[[i, j, 2]]] - w3 * u0[xt[[i, j, 1]]]];
For[j = 3, j ≤ m, j++,
 row = Table[0, {k, 1, n * m - n - 2 m + 2}];
 row[[(i - 2) * (m - 1) + j - 1]] = w2;
 row[[(i - 1) * (m - 1) + j - 1]] = w1;
 row[[(i - 2) * (m - 1) + j - 2]] = w3;
 AppendTo[A, row];
 AppendTo[B, w4 * f[xt[[i, j, 1]], xt[[i, j, 2]]] - w1 * ua[xt[[i, j, 2]]]];
];
For[i = 3, i ≤ n - 2, i++,
 j = 2;
 row = Table[0, {k, 1, n * m - n - 2 * m + 2}];
 row[[(i - 3) * (m - 1) + j - 1]] = w1;
 row[[(i - 2) * (m - 1) + j - 1]] = w2;
 row[[(i - 1) * (m - 1) + j - 1]] = w1;
 AppendTo[A, row];
 AppendTo[B, w4 * f[xt[[i, j, 1]], xt[[i, j, 2]]] - w3 * u0[xt[[i, j, 1]]]];
 For[j = 3, j ≤ m, j++,
  row = Table[0, {k, 1, n * m - n - 2 * m + 2}];
  row[[(i - 3) * (m - 1) + j - 1]] = w1;
  row[[(i - 2) * (m - 1) + j - 1]] = w2;
  row[[(i - 1) * (m - 1) + j - 1]] = w1;
  row[[(i - 2) * (m - 1) + j - 2]] = w3;
  AppendTo[A, row];
  AppendTo[B, w4 * f[xt[[i, j, 1]], xt[[i, j, 2]]]];
 ];
];
i = n - 1; j = 2;
row = Table[0, {k, 1, n * m - n - 2 * m + 2}];
row[[(i - 3) * (m - 1) + j - 1]] = w1;
row[[(i - 2) * (m - 1) + j - 1]] = w2;
AppendTo[A, row];
AppendTo[B, w4 * f[xt[[i, j, 1]], xt[[i, j, 2]]] - w3 * u0[xt[[i, j, 1]]] - w1 * ub[xt[[i, j, 2]]]];
For[j = 3, j ≤ m, j++,
 row = Table[0, {k, 1, n * m - n - 2 * m + 2}];
 row[[(i - 3) * (m - 1) + j - 1]] = w1;
 row[[(i - 2) * (m - 1) + j - 1]] = w2;
 row[[(i - 2) * (m - 1) + j - 2]] = w3;
 AppendTo[A, row];
 AppendTo[B, w4 * f[xt[[i, j, 1]], xt[[i, j, 2]]] - w1 * ub[xt[[i, j, 2]]]];
```

```
    ];
    wynik = LinearSolve[A, B];
result = {};
For[i = 1, i ≤ n, i++,
For[j = 1, j ≤ m, j++,
If[i == 1, AppendTo[result, {xt[[i, j, 1]], xt[[i, j, 2]], ua[xt[[i, j, 2]]]}]];

If[i == n, AppendTo[result, {xt[[i, j, 1]], xt[[i, j, 2]], ub[xt[[i, j, 2]]]}]];
If[j == 1 && i ≠ n && i ≠ 1, AppendTo[result, {xt[[i, j, 1]], xt[[i, j, 2]], u0[xt[[i, j, 1]]]}]];
If[i ≠ 1 && i ≠ n && j ≠ 1,
        AppendTo[result, {xt[[i, j, 1]], xt[[i, j, 2]], N[wynik[[(i - 2) * m + (j - (i - 1))]]]}]];

];
];
    Return[N[result]]
  ];
```

Działanie programu dla wybranych parametrów

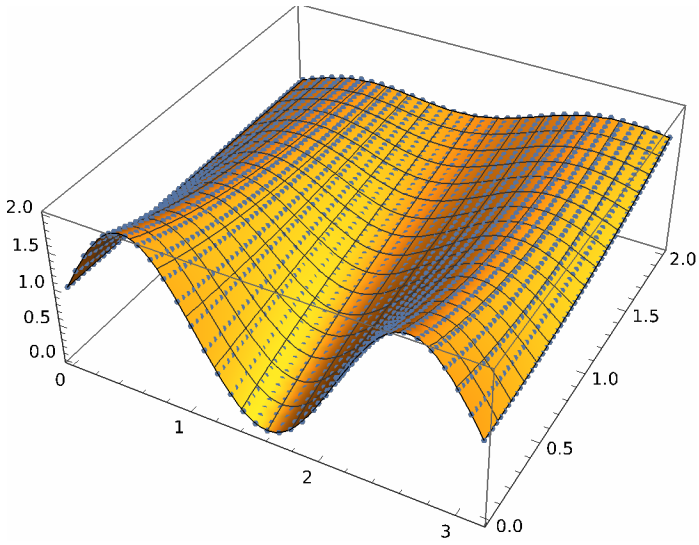
In[8]:=
```
(*Ustawienie parametrów wejściowych funkcji.*)
alpha = 1/9;
a = 0;
b = N[Pi];
n = 35;
m = 55;
tg = 2;
f[x_, t_] := x * t / 10;
u0[x_] := 1 + Sin[3 * x];
ua[t_] := 1;
ub[t_] := 1 + N[Pi] * t * t / 20;
u[x_, t_] := Exp[-t] * Sin[3 x] + x * t * t / 20 + 1;
```

In[24]:=
```
results = cieplo[alpha, a, b, n, m, tg, f, u0, ua, ub, u];
```

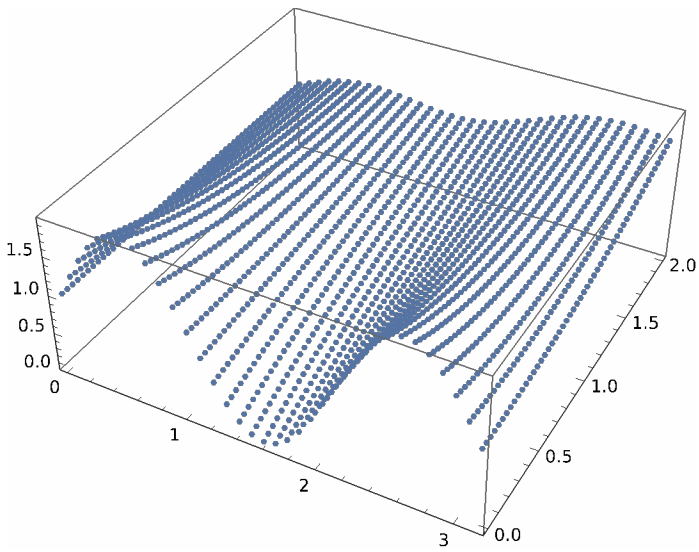
In[61]:=

```
pp = ListPointPlot3D[results];
pacc = Plot3D[u[x, t], {x, a, b}, {t, 0, tg}];
Show[pacc, pp]
Show[pp]
```

Out[63]=



Out[64]=



Błędy otrzymanego rozwiązania

In[52]:=

```
accResult = Table[{x, N[t], u[x, t]}, {x, a, b, (b - a)/(n - 1)}, {t, 0, tg, tg/(m - 1)}];
errorList = Table[{accResult[[i, j, 1]], accResult[[i, j, 2]],
    Abs[accResult[[i, j, 3]] - results[[(i - 1) * m + j, 3]]]}, {i, 1, n}, {j, 1, m}];
errors = 0;
errors = Join[errorList[[1]], errorList[[2]]];
For[i = 3, i ≤ Length[errorList], i++,
errors = Join[errors, errorList[[i]];
]
```

In[34]:=

```
ListPlot3D[errors]
```

Out[34]=