

Sprawozdanie

Metody obliczeniowe – projekt 11-3

Aleksander Kowalczyk - 131954

Laboratorium komputerowe z przedmiotu “Metody Obliczeniowe”, prowadzący: dr hab. inż. L. Bieniasz

Ćwiczenie 11-3

Kowalczyk Aleksander

Zagadnienie z warunkiem początkowym i brzegowym obejmuje:

równanie różniczkowe cząstkowe $\frac{\partial U(x,t)}{\partial t} = D \left[\frac{\partial^2 U(x,t)}{\partial x^2} + \frac{2}{x} \frac{\partial U(x,t)}{\partial x} \right]$, określone dla współrzędnej

przestrzennej $x \in [r, +\infty)$ oraz czasu $t \in [0, t_{\max}]$,

warunek początkowy $U(x,0) = 1$, oraz

warunki brzegowe $U(r, t) = 0$, $U(+\infty, t) = 1$.

Zagadnienie to może opisywać transport ciepła, w ośrodku wokół kuli o promieniu r , przy współczynniku transportu ciepła D , po raptownym obniżeniu temperatury kuli w chwili $t = 0$.

Rozwiązanie analityczne tego zagadnienia ma postać: $U(x,t) = 1 - \frac{r}{x} \operatorname{erfc} \left(\frac{x-r}{2\sqrt{Dt}} \right)$, gdzie $\operatorname{erfc}(z) = 1 - \operatorname{erf}(z)$, a

$\operatorname{erf}(z)$ jest tzw. funkcją błędu: $\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z \exp(-w^2) dw$.

Do obliczeń numerycznych przedział nieskończony x należy zastąpić przedziałem skończonym $[r, r+a]$, a drugi

warunek brzegowy zastąpić warunkiem $U(r+a, t) = 1 - \frac{r}{r+a} \operatorname{erfc} \left(\frac{a}{2\sqrt{Dt}} \right)$. Do obliczenia funkcji $\operatorname{erfc}(z)$ z

dokładnością bliską dokładności maszynowej dla zmiennych typu **double** należy zastosować pakiet CALERF udostępniony przez prowadzącego zajęcia.

Należy rozwiązać to zagadnienie stosując zaznaczoną niżej kombinację algorytmów numerycznych oraz podane wartości parametrów. Należy przyjąć ustaloną wartość $\lambda = D \delta t / h^2$, możliwie najbliższą $\lambda = 0.4$ dla metod bezpośredniej lub $\lambda = 1$ dla metod pośrednich (uwaga na ograniczenia stabilności numerycznej!). Rozwiązania numeryczne należy porównać z analitycznymi i wyznaczyć błędy bezwzględne rozwiązań numerycznych. Jeżeli poniżej zaznaczono dwa alternatywne algorytmy, to wówczas w programie należy zrealizować oba, a uzyskane wyniki porównać.

Do zaliczenia projektu należy wykonać:

(1) Wykresy zależności maksymalnej wartości bezwzględnej błędu obserwowanej dla t_{\max} w funkcji kroku przestrzennego h (najlepiej w skali logarytmicznej, o ile to możliwe). Należy sprawdzić, czy zależność jest zgodna z teoretycznym rzędem dokładności i wyjaśnić ewentualne niezgodności. Do dalszych wykresów należy dobrać kroki czasowy (i przestrzenny) tak, aby uzyskać możliwie jak najlepszą dokładność rozwiązania w czasie obliczeń nie przekraczającym około jednej minuty, dla najszybszego z rozważanych wariantów obliczeń. Wyniki numeryczne oraz rozwiązania analityczne i błędy odpowiadające tej sytuacji należy zapisać w zbiorze, w postaci sformatowanej umożliwiającej przeglądanie wyników.

(2) Wykresy rozwiązań numerycznych i analitycznych dla kilku wybranych wartości czasu t z całego przedziału t (rozwiązania numeryczne punktami, rozwiązania analityczne linią ciągłą).

(3) Wykresy zależności maksymalnej wartości bezwzględnej błędu w funkcji czasu t . **Należy wyjaśnić ewentualnie obserwowane zmiany błędu w czasie.**

Algorytmy:

Dyskretyzacja:

- ☒ Klasyczna metoda bezpośrednia
- ☒ Metoda pośrednia Laasonen
- ☐ Metoda pośrednia Cranka-Nicolson

Rozwiązanie algebraicznych układów równań liniowych:

- ☒ Dekompozycja LU macierzy pełnej
- ☒ Algorytm Thomasa
- ☐ Metoda iteracyjna Jacobiego
- ☐ Metoda iteracyjna Gaussa-Seidela
- ☐ Metoda iteracyjna SOR (należy dobrać ω)

Parametry:

$t_{\max} = 2$, $r = 1$, $a = 10$, $D = 1$.

I). Wyznaczenie równania dla klasycznej metody bezpośredniej:

a) Obliczenia

☺ klasyczna metoda bezpośrednia ☺

$$\frac{\partial U(x,t)}{\partial t} = D \left[\frac{\partial^2 U(x,t)}{\partial x^2} + \frac{2}{x} \frac{\partial U(x,t)}{\partial x} \right]$$

$$\frac{U_{i,k+1} - U_{i,k}}{\delta t} = D \left[\frac{U_{i-1,k} - 2U_{i,k} + U_{i+1,k}}{h^2} \right] + \frac{2D}{x \cdot h} [U_{i+1,k} - U_{i,k}]$$

$$U_{i,k+1} = \lambda [U_{i-1,k} - 2U_{i,k} + U_{i+1,k}] + 2\lambda \frac{h}{x} [U_{i+1,k} - U_{i,k}] + U_{i,k}$$

b) Implementacja

```
void classical(double **U, int n, int m, double h) {
    double x;
    for (int k = 0; k < n - 1; k++) { // t
        x = r + h;
        for (int i = 1; i < m - 1; i++) { // x
            U[k+1][i] = lambda_direct * (U[k][i-1] - 2.0 * U[k][i] + U[k][i+1]) + (2.0 * lambda_direct * h / x) * (U[k][i+1] - U[k][i]) + U[k][i];
            x += h;
        }
    }
}
```

II). Wyznaczenie równania dla metody Laasonen:

a) Obliczenia

☺ Laasonen ☺

$$\frac{U_{i,k+1} - U_{i,k}}{\delta t} = D \left[\frac{U_{i-1,k+1} - 2U_{i,k+1} + U_{i+1,k+1}}{h^2} \right] + 2\frac{D}{x} \frac{U_{i+1,k+1} - U_{i-1,k+1}}{2h}$$

$$-U_{i,k} = \lambda [U_{i-1,k+1} - 2U_{i,k+1} + U_{i+1,k+1}] + \frac{(D\delta t)}{(x \cdot h)} \frac{U_{i+1,k+1} - U_{i-1,k+1}}{\lambda \frac{h}{x}} - U_{i,k+1}$$

$$-U_{i,k} = (\lambda - \lambda \frac{h}{x}) \cdot U_{i-1,k+1} + (-2\lambda - 1) U_{i,k+1} + (\lambda + \lambda \frac{h}{x}) U_{i+1,k+1}$$

$$= \underset{\uparrow}{\lambda(1-\frac{h}{x})} \cdot U_{i-1,k+1} - \underset{\uparrow}{[2\lambda+1]} \cdot U_{i,k+1} + \underset{\uparrow}{[\lambda(1+\frac{h}{x})]} U_{i+1,k+1}$$

$$\lambda = 1.0$$

b) Implementacja (Dekompozycja LU macierzy pełnej, alg. Thomasa)

W przypadku metody thomasa postanowiłem napisać osobną funkcję – (przygotowana prze zemnie na 6 laboratoria przyjmuje zgodnie z Pana poleceniem wektory a nie niepotrzebnie tracącą pamięć Macierz)

```
void laasonen_lu(double **U, int t, int x, double h) {  
  
    // tworze macierz i wektory dla równania A*y=b  
    double **A = alloc_0_matrix(x, x);  
    double *b = alloc_0_vector(x);  
    double *y = alloc_0_vector(x);  
    // dodatkowe macierze dla metody LU  
    double **L = alloc_0_matrix(x, x);  
    double **P = alloc_0_matrix(x, x);  
  
    // ZMIENNA x zależna od iteracji  
    double xx;  
  
    for (int k = 1; k < t; k++) {  
        // zaczynamy od k=1 więc wartość początkowa r + 1*h  
        xx = r + h;  
        A[0][0] = 1.0;  
        A[x - 1][x - 1] = 1.0;  
        for (int i = 1; i < x - 1; i++) {  
            A[i][i - 1] = lambda_indirect * (1.0 - h/xx);  
            A[i][i] = -(2.0*lambda_indirect + 1.0);  
            A[i][i + 1] = lambda_indirect * (1.0 + h / xx);  
            b[i] = -U[k-1][i];  
            xx += h;  
        }  
        b[0] = U[k][0];  
        b[x - 1] = U[k][x-1];  
  
        // rozwiązuje układ przy pomocy dekompozycji LU  
        gauss(A, L, P, x);  
        y = gauss_resoult(A, L, P, b, x);  
  
        // przypisanie wyników  
        for (int i = 1; i < x - 1; i++)  
            U[k][i] = y[i];  
    }  
}
```

```
void laasonen_thomas(double **U, int t, int x, double h) {  
  
    double *l = alloc_0_vector(x);  
    double *d = alloc_0_vector(x);  
    double *u = alloc_0_vector(x);  
    double *b = alloc_0_vector(x);  
    double *y = alloc_0_vector(x);  
  
    double xx;  
    for (int k = 1; k < t; k++) {  
        d[0] = 1.0;  
        d[x - 1] = 1.0;  
  
        xx = r + h;  
        for (int i = 1; i < x - 1; i++) {  
            l[i] = lambda_indirect * (1.0 - h / xx);  
            d[i] = -(2.0*lambda_indirect + 1.0);  
            u[i] = lambda_indirect * (1.0 + h / xx);  
            b[i] = -U[k - 1][i];  
            xx += h;  
        }  
        b[0] = U[k][0];  
        b[x - 1] = U[k][x - 1];  
  
        thomas_algorithm(l, d, u, b, y, x);  
        for (int i = 1; i < x - 1; i++)  
            U[k][i] = y[i];  
    }  
}
```

Jeżeli chodzi o Thomasa i dekompozycje LU korzystałem z napisanych przeze mnie wcześniej zadań laboratoryjnych:

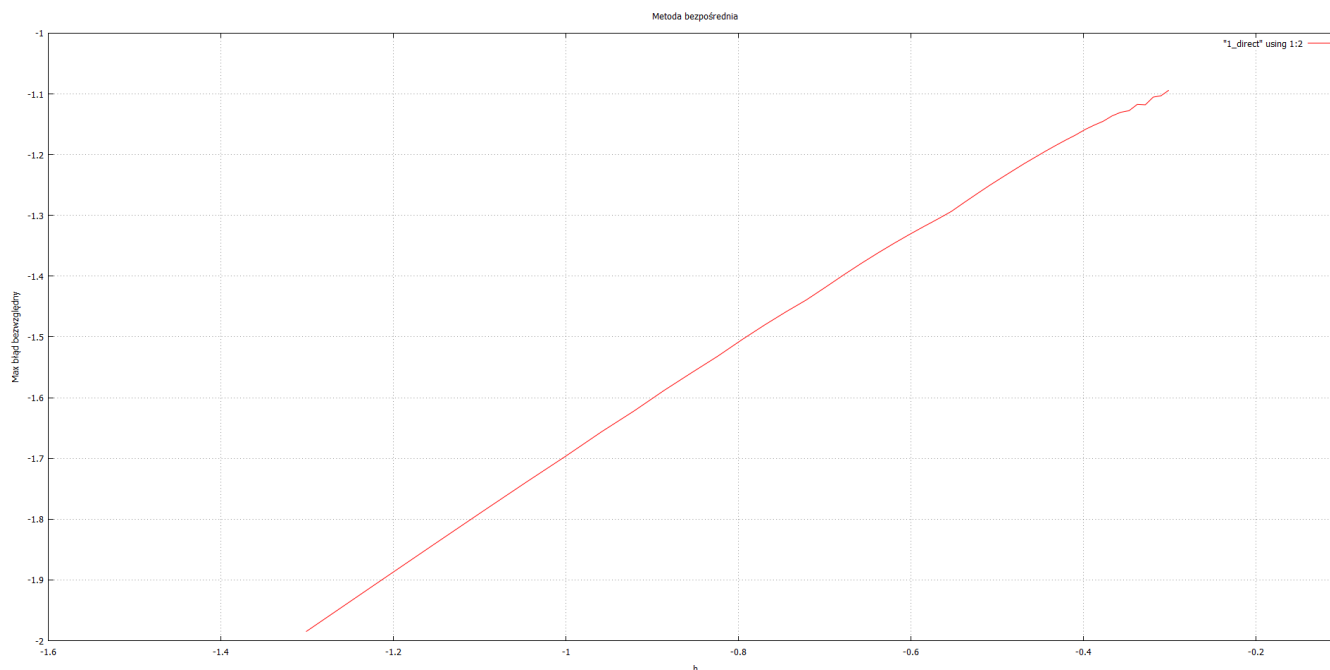
[Lab5 - Dekompozycję LU macierzy](#)

[Lab6 - Algorytm Thomasa dla macierzy trój-diagonalnej NxN](#)

Oczywiście wszystko znajduję się również w przesłanym kodzie.

(1) Wykresy zależności maksymalnej wartości bezwzględnej błędu obserwowanej dla t_{max} , w funkcji kroku przestrzennego h .

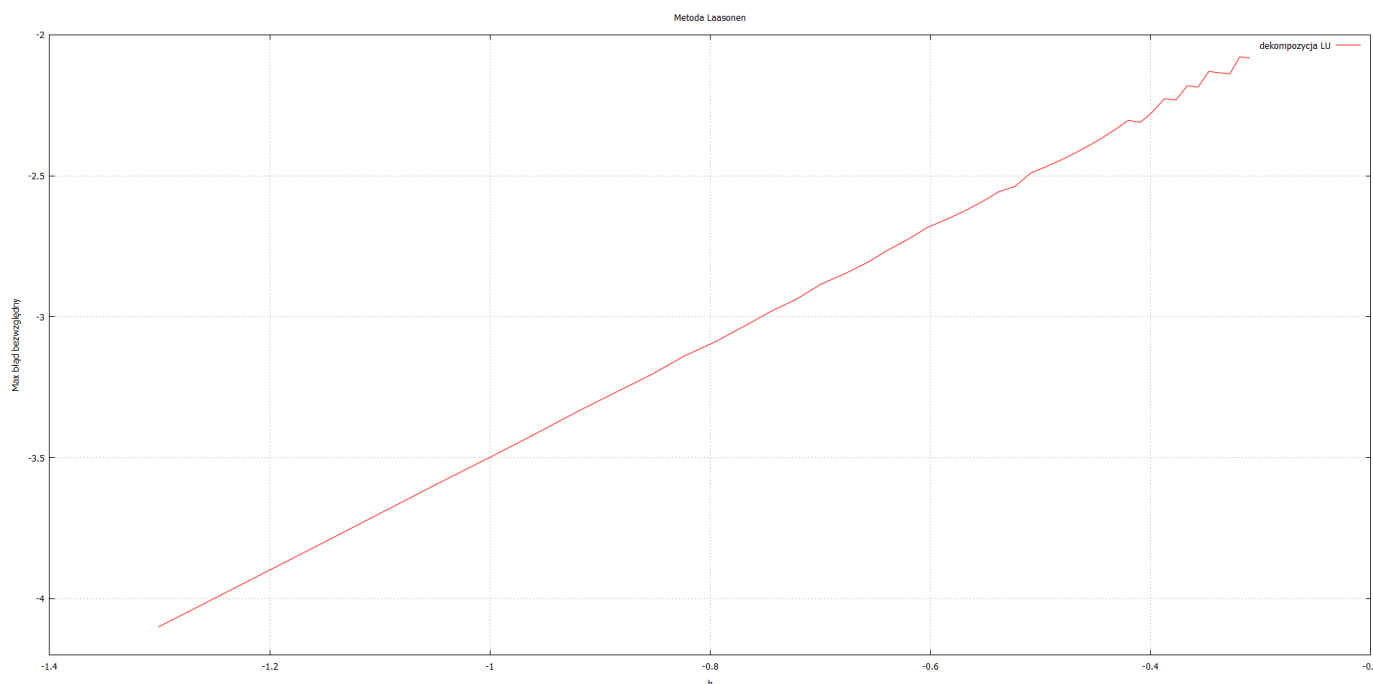
a) Metoda bezpośrednia klasyczna (wykres w skali logarytmicznej):



$$\text{Rząd dokładności} \approx \frac{(-1.11723 + 1.12756)}{(-0.337243 + 0.346741)} \approx 1.088 \approx 1$$

Teoretycznie spodziewałem się rzędu: 2, na niedokładność może wpływać np. wielokrotne dodawanie liczb o różnych znakach lub dana funkcja źle współgra z tą metodą. Przerobiłem funkcje grupując kolejne współczynniki U_{i-1} , U_i , U_{i+1} niestety otrzymany wynik był podobny więc wróciłem do pierwotnego stanu.

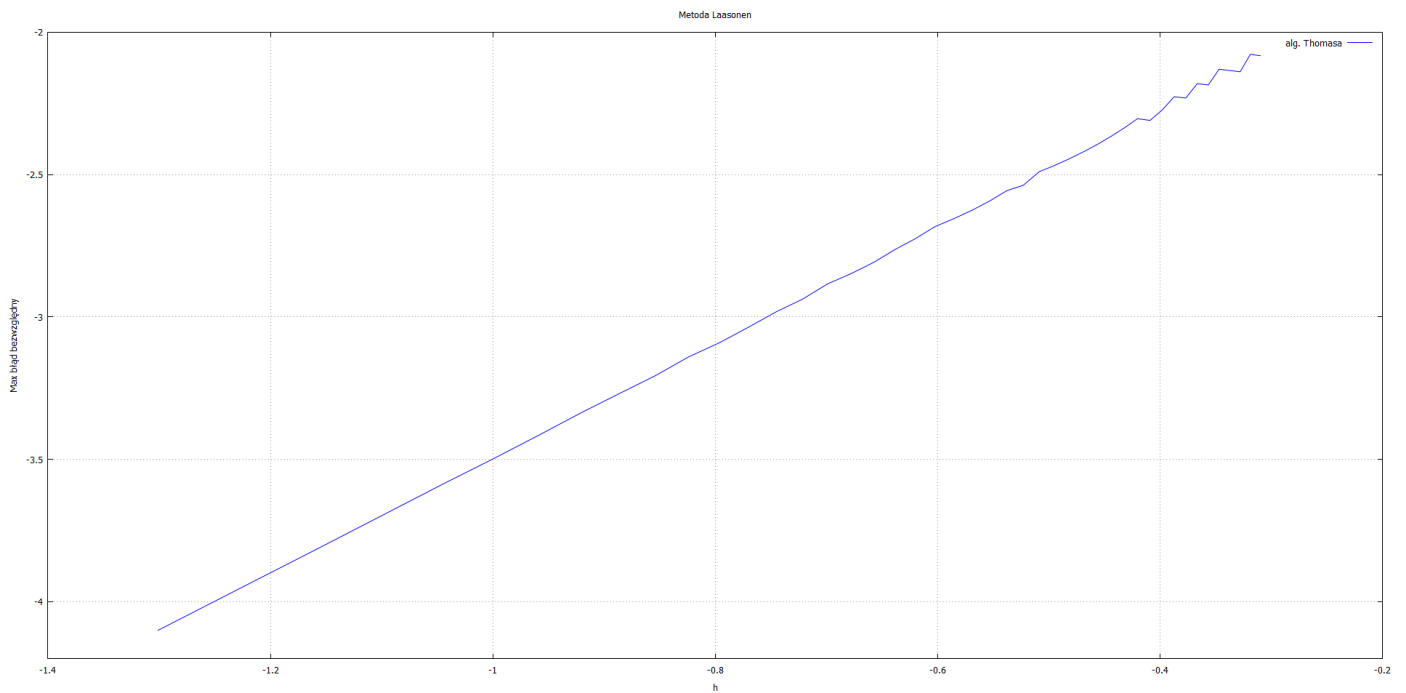
c) Metoda pośrednia Laasonen + Dekompozycja LU macierzy pełnej (wykres w skali logarytmicznej):



$$\text{Rząd dokładności} \approx \frac{(-2.34833 + 2.44599)}{(-0.436940 + 0.482155)} \approx 2.165 \approx 2$$

Wynik jest zgodny z oczekiwaniami.

d) Metoda pośrednia Laasonen + alg. Thomasa (wykres w skali logarytmicznej):



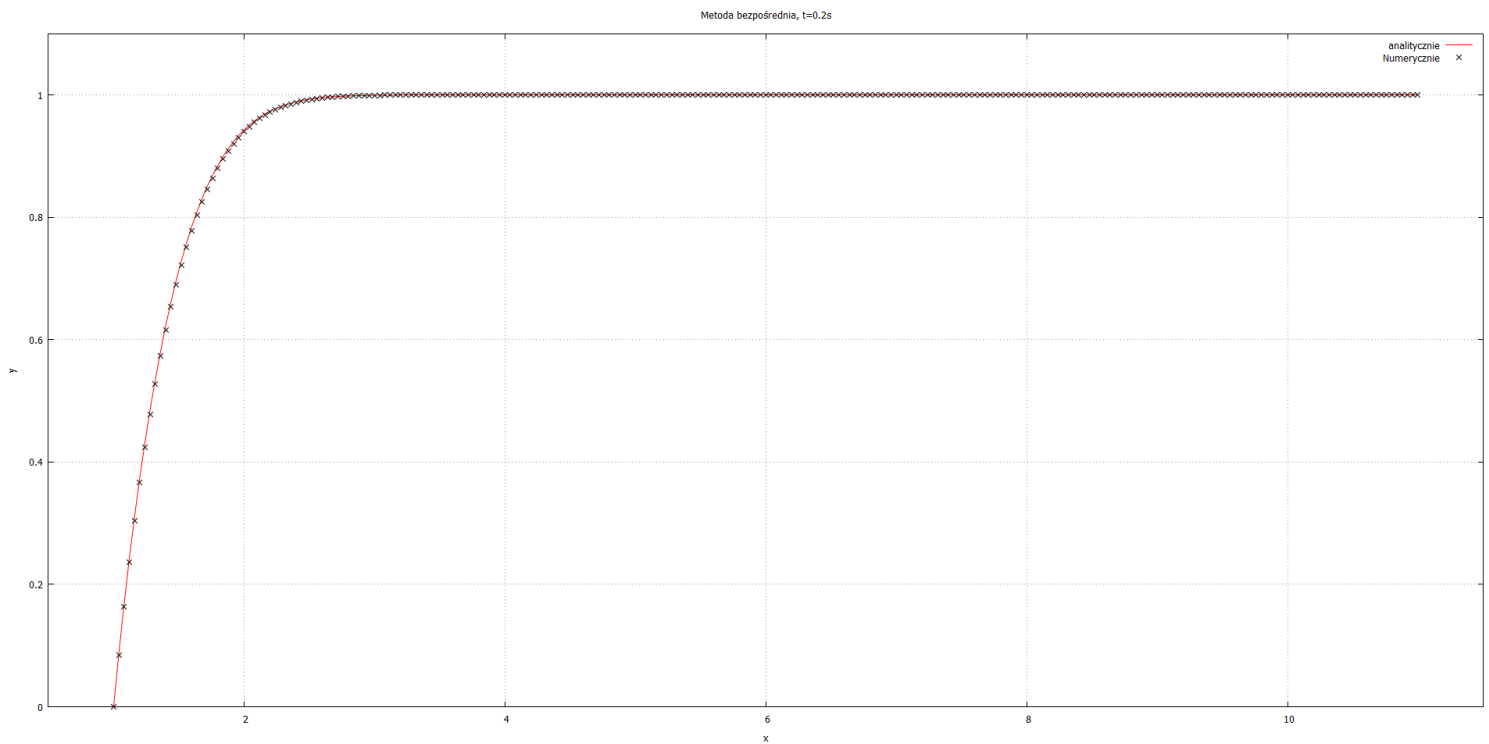
$$Rz\acute{a}d\ dok\l{a}dno\acute{s}ci \approx \frac{(-2.35023 + 2.45585)}{(-0.437862 + 0.488042)} \approx 2.105 \approx 2$$

Wynik jest zgodny z oczekiwaniami.

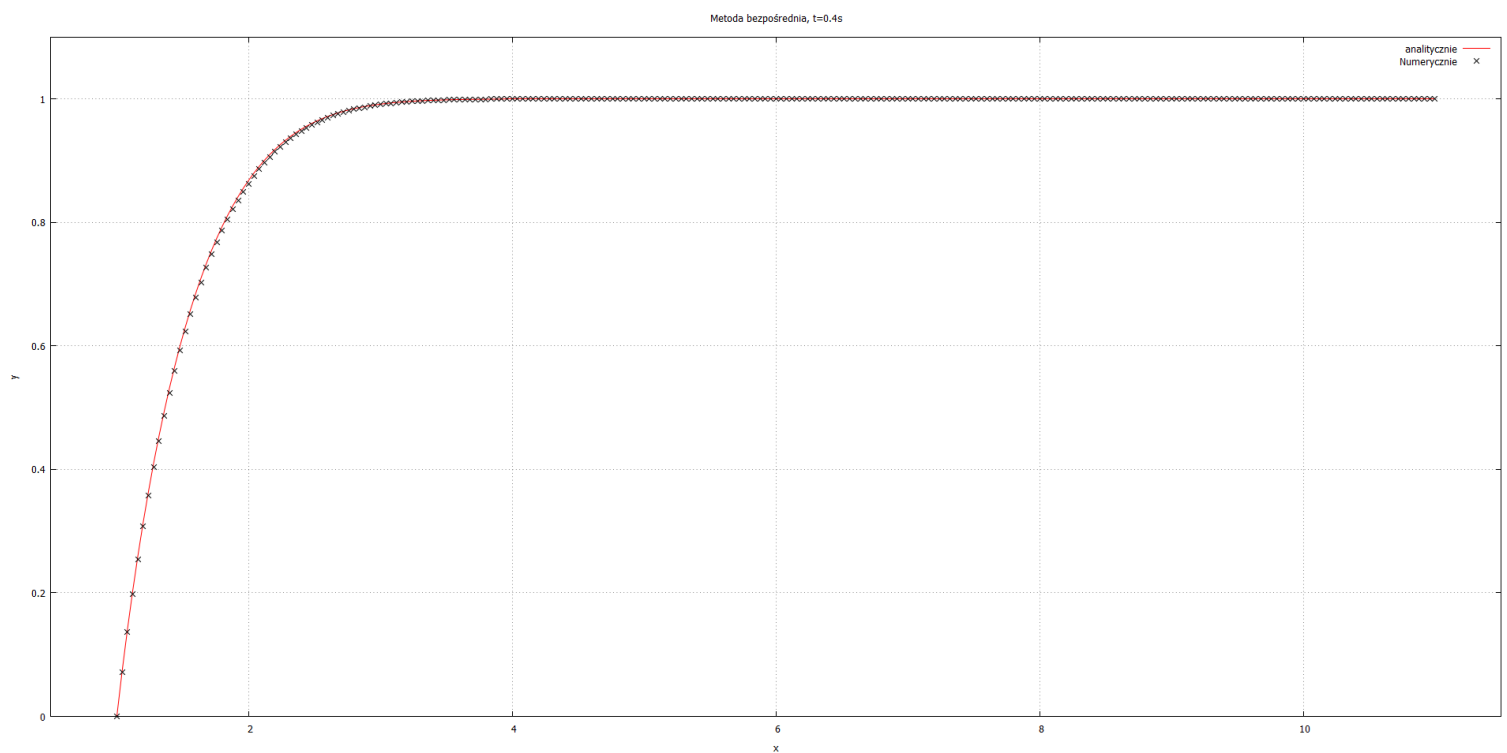
(2) Wykresy rozwiązań numerycznych i analitycznych dla kilku wybranych wartości czasu $t = (0.2s, 0.4s, 1s, 1.6s, 2s)$

a) Klasyczna metoda bezpośrednia

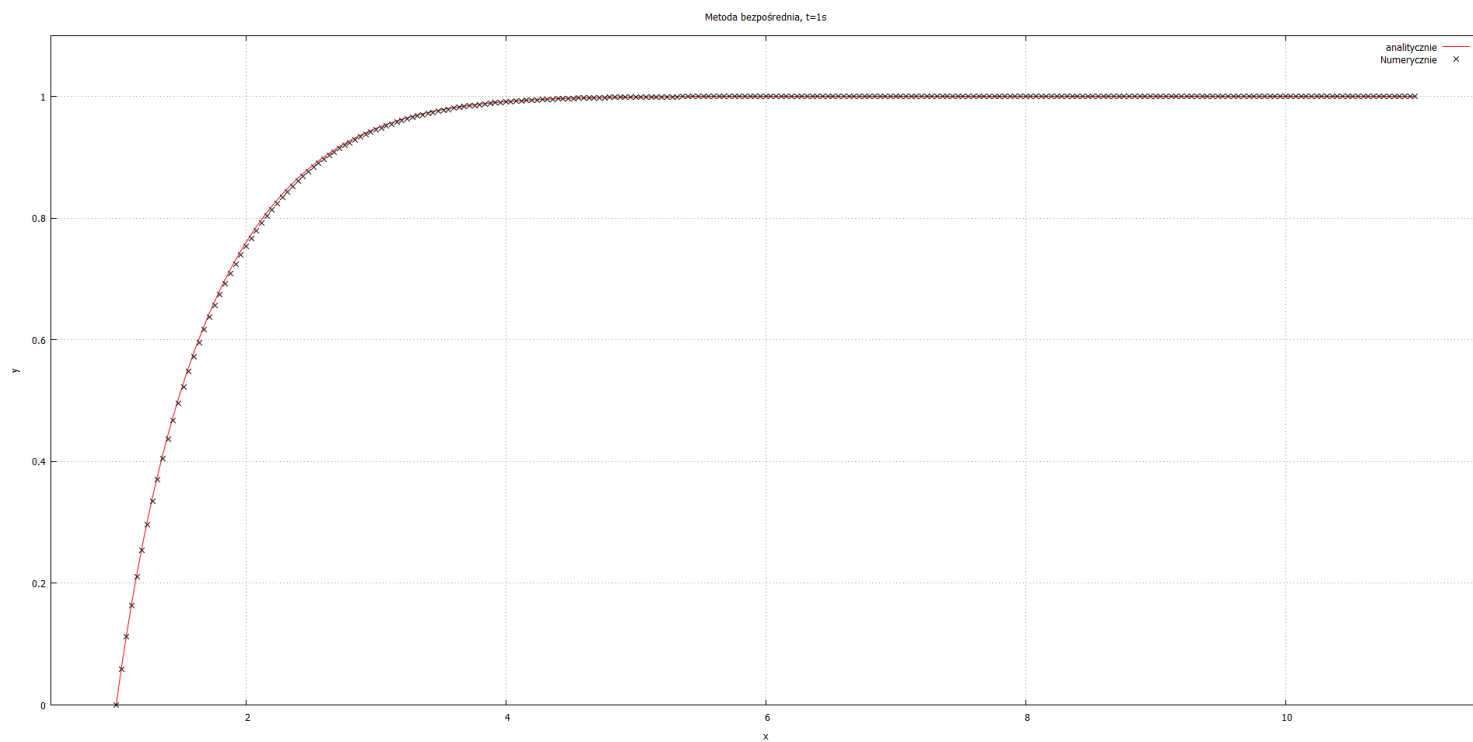
- $t=0.2s$



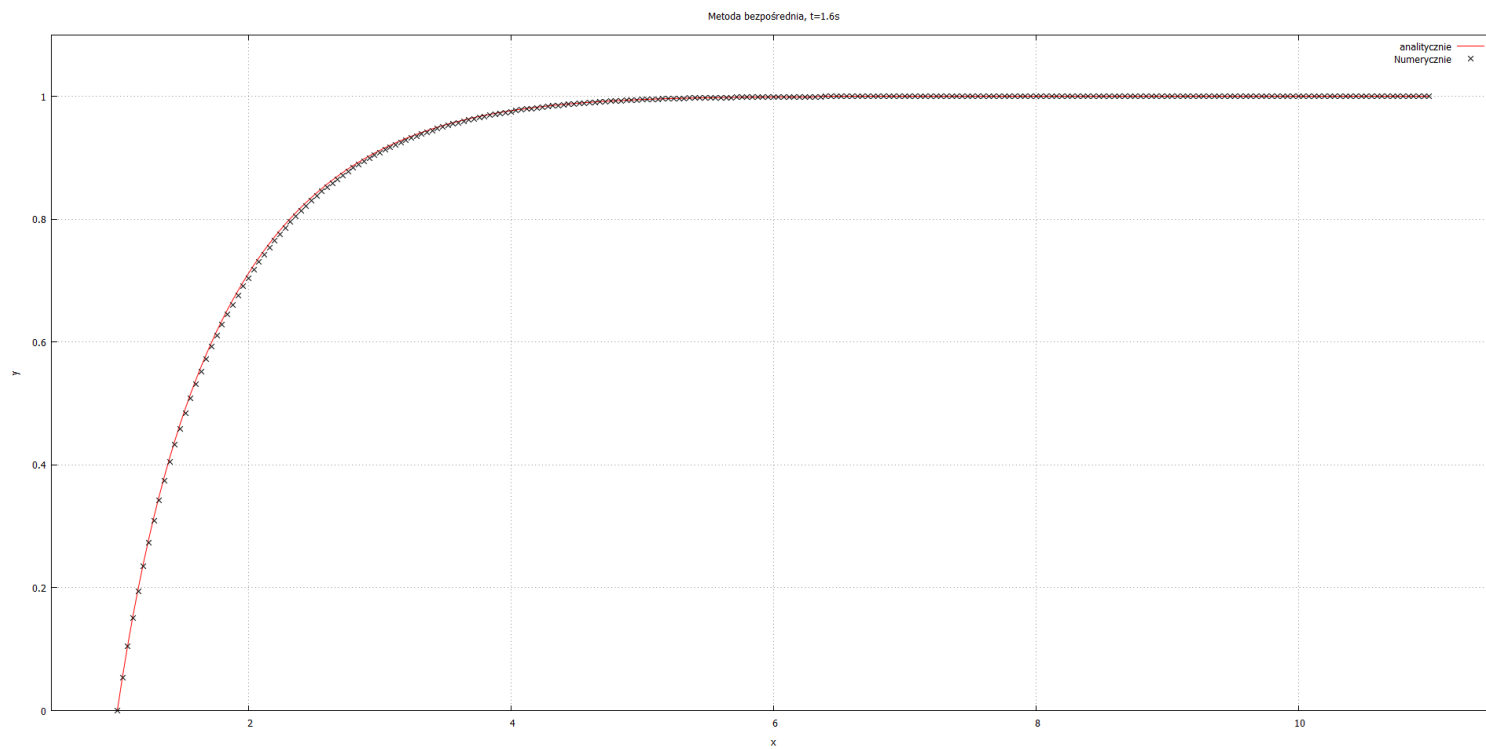
- $t=0.4s$



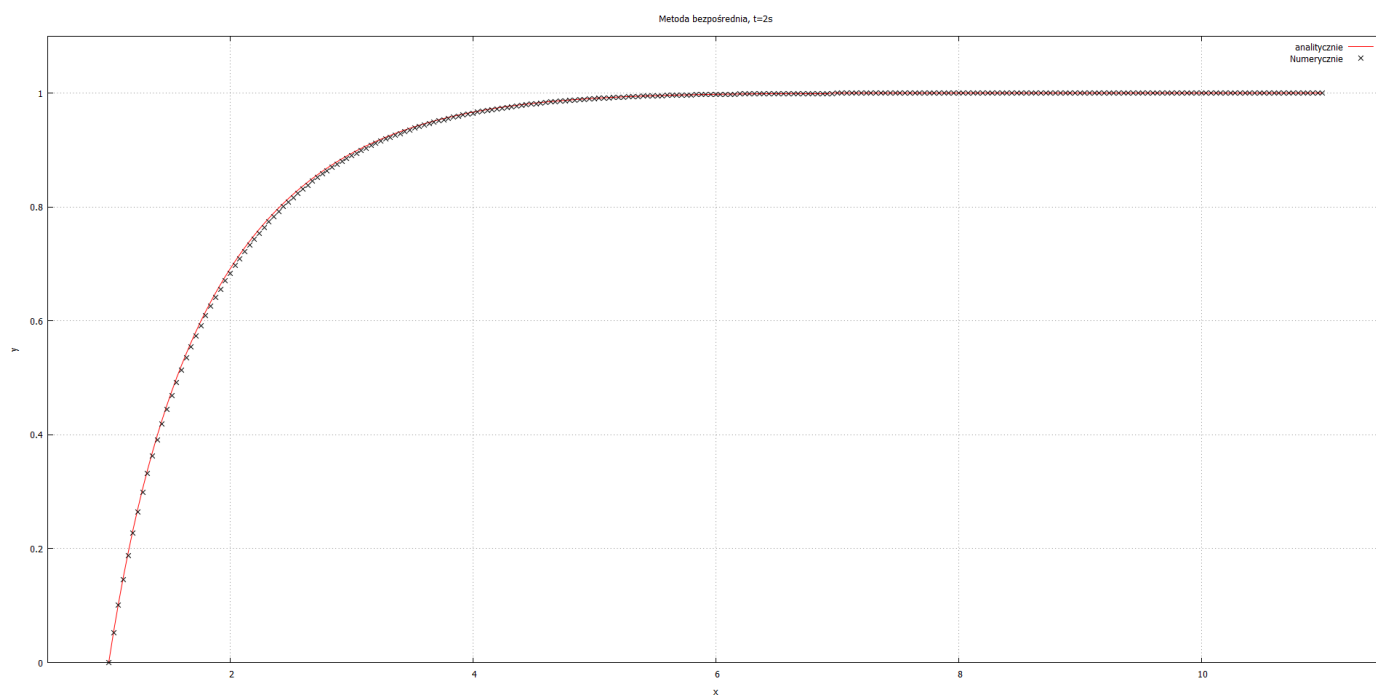
- **t=1s**



- **t=1.6s**

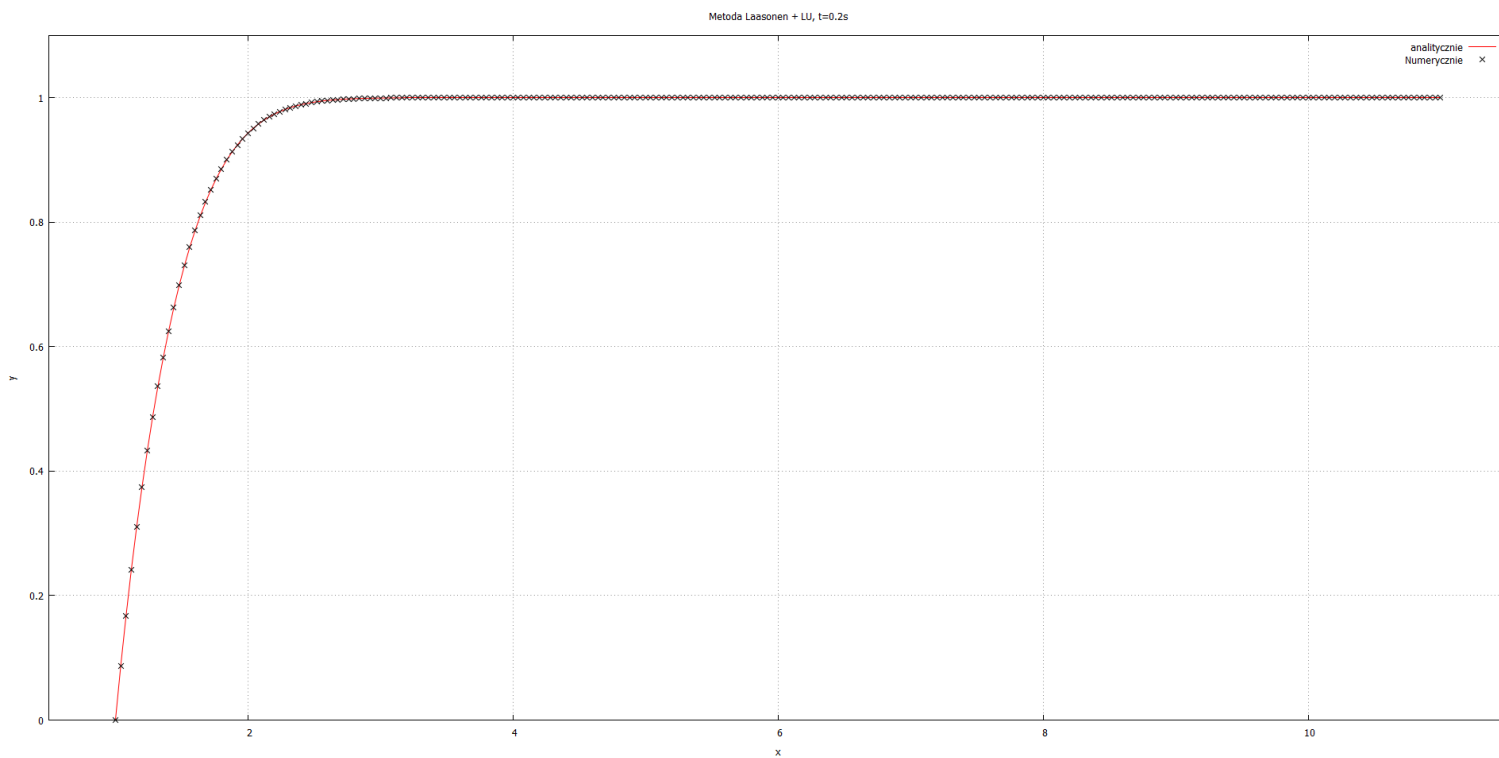


- $t=2s$

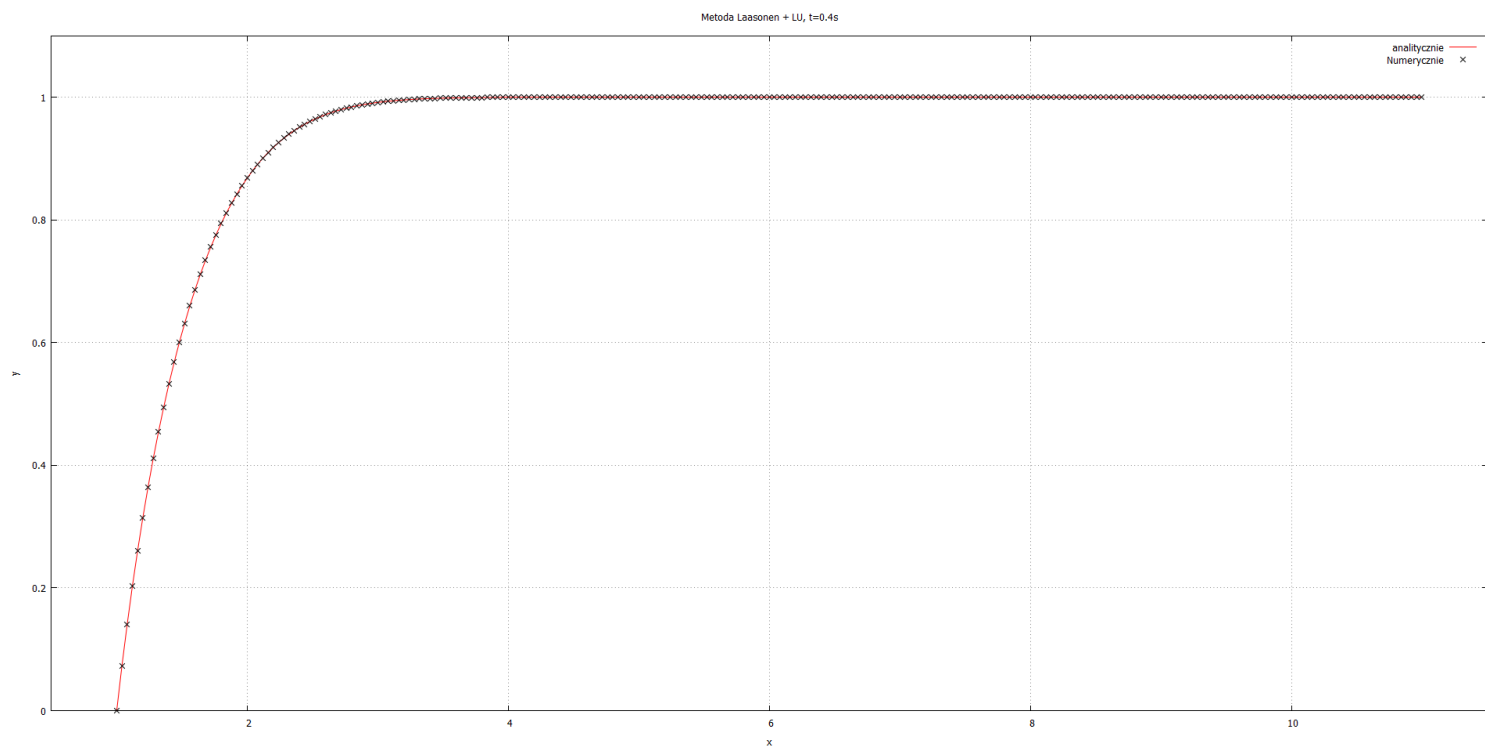


b) Metoda pośrednia Laasonen + Dekompozycja LU macierzy pełnej

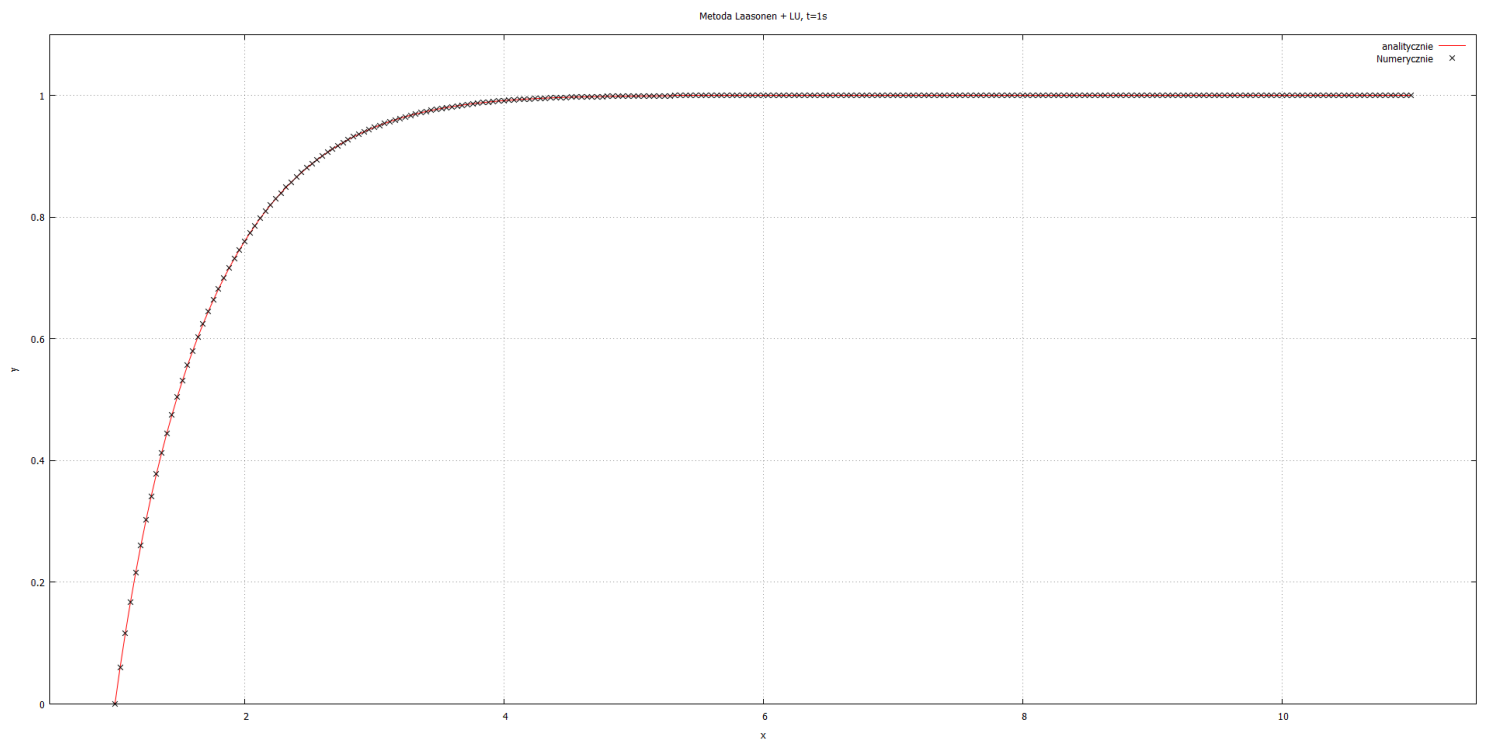
- $t=0.2s$



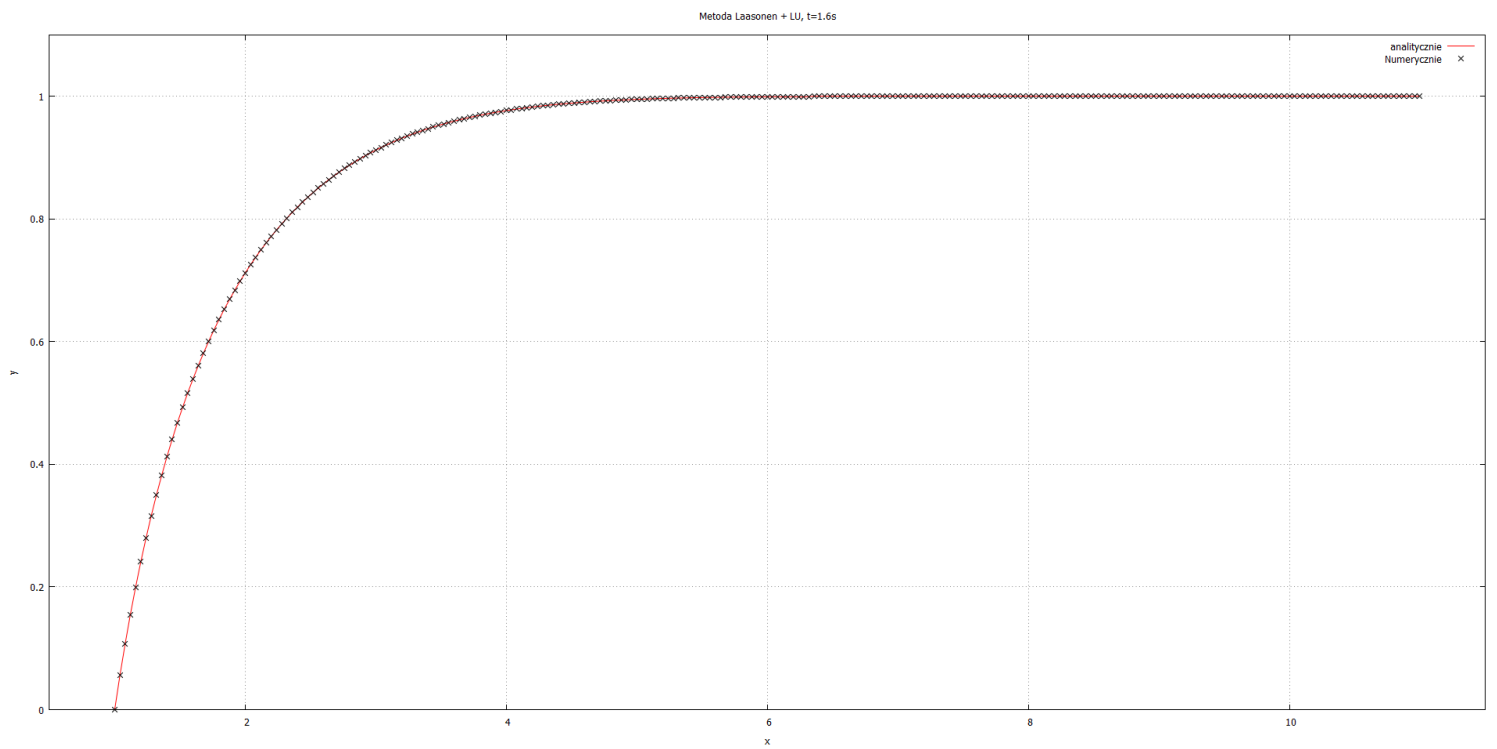
- $t=0.4s$



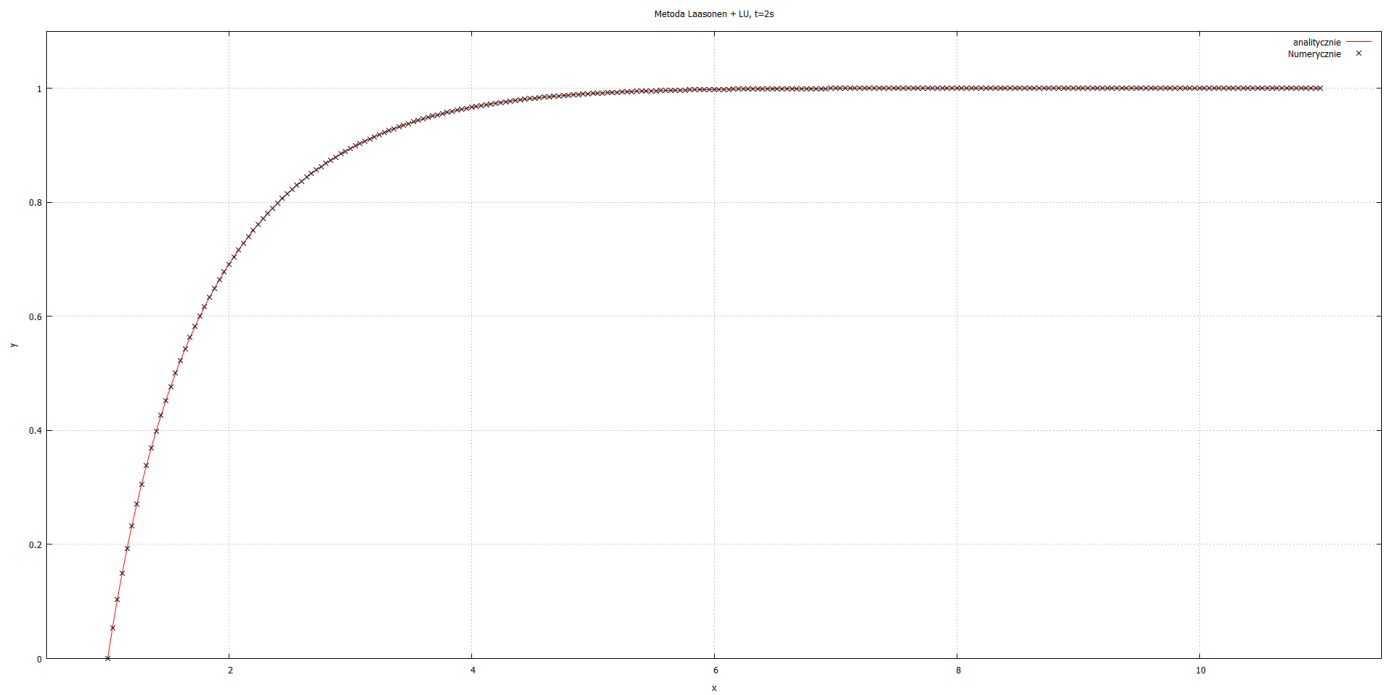
- **t=1s**



- **t=1.6s**

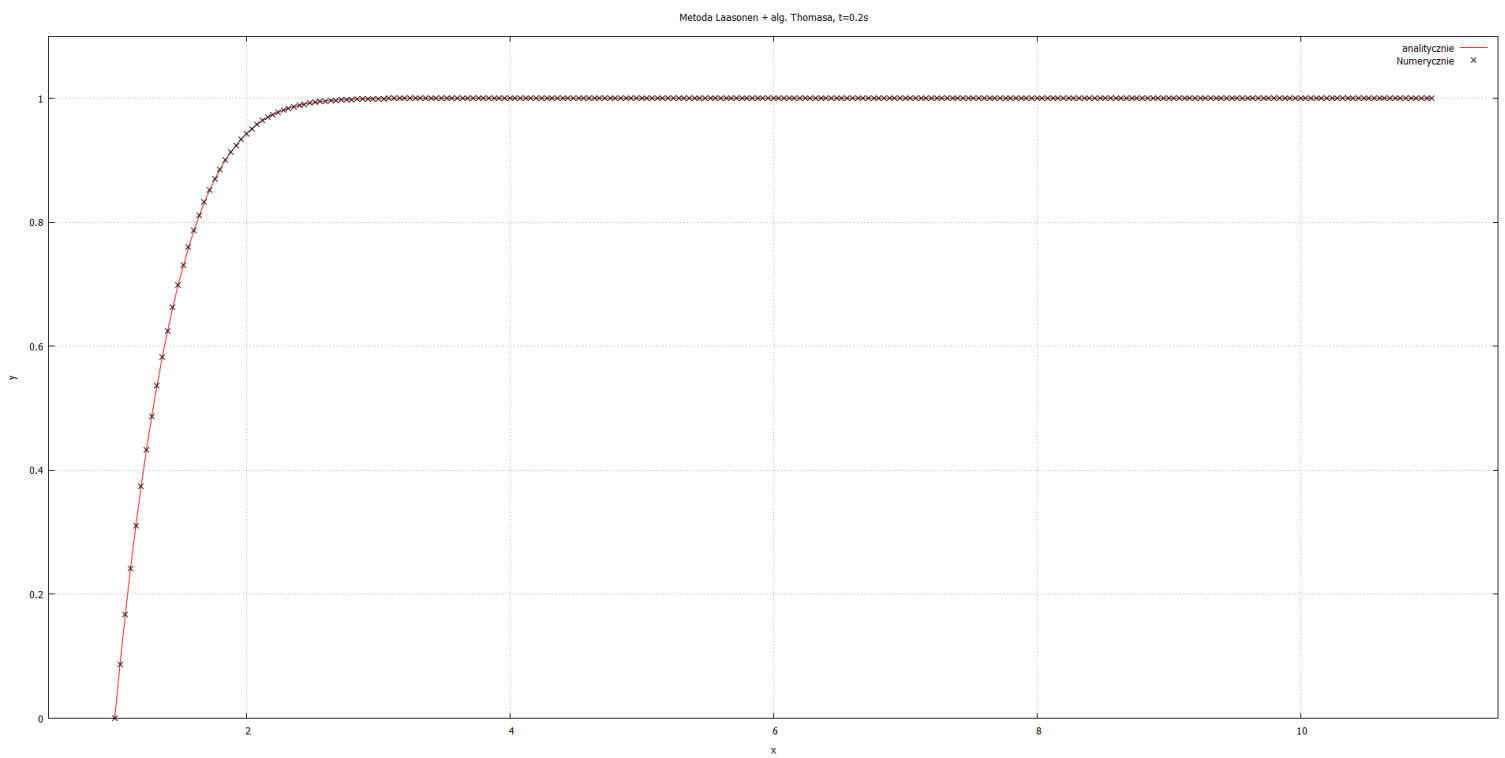


- $t=2s$

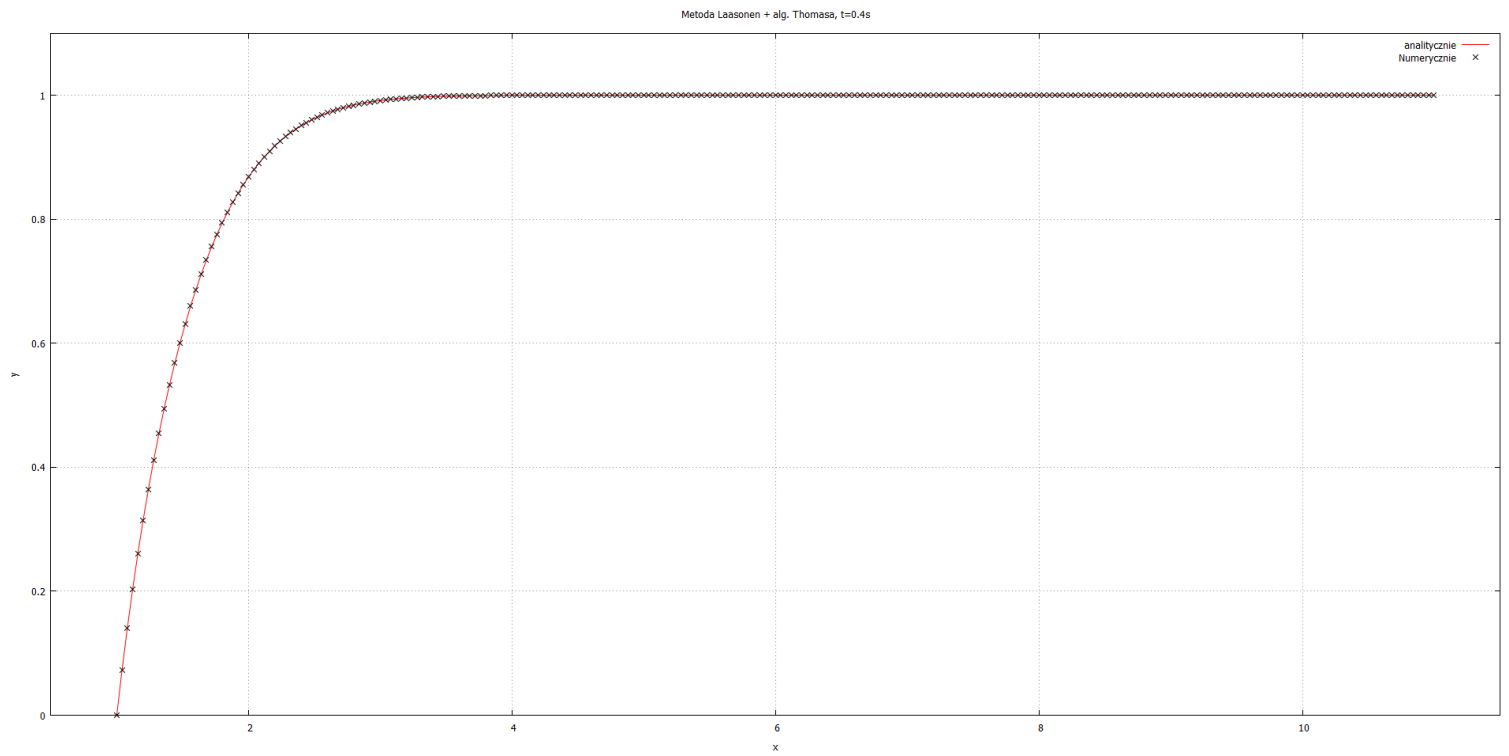


c) Metoda pośrednia Laasonen + alg. Thomasa

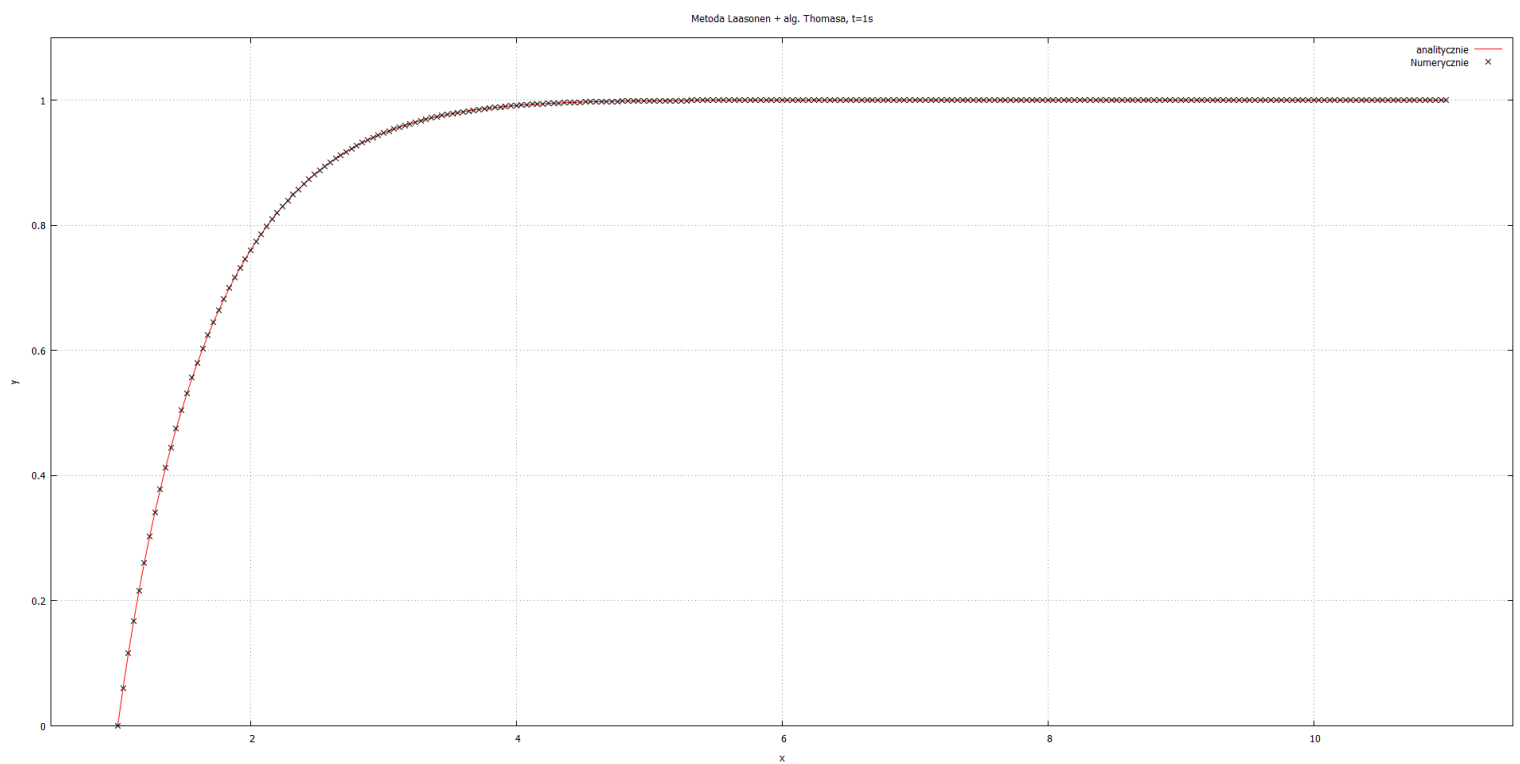
- $t=0.2s$



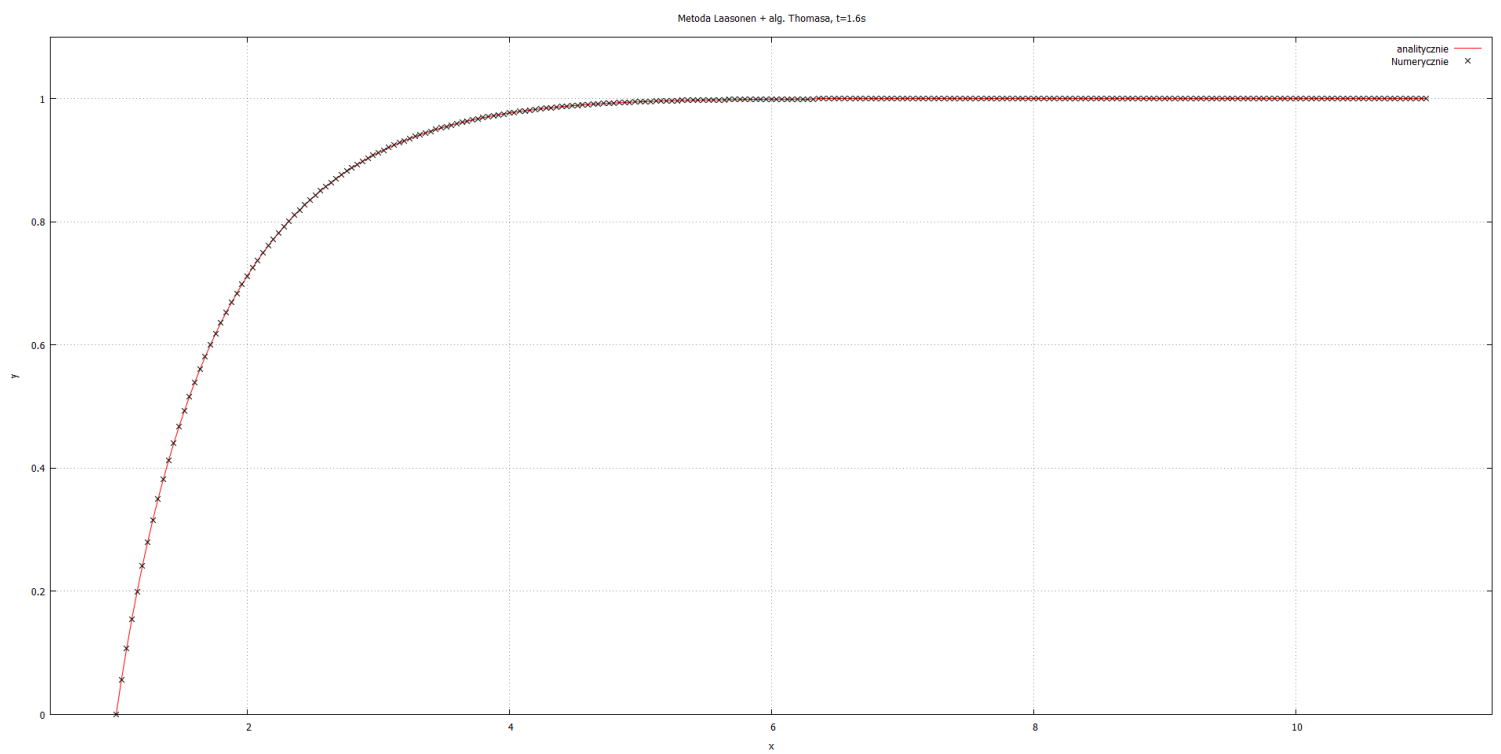
- **t=0.4s**



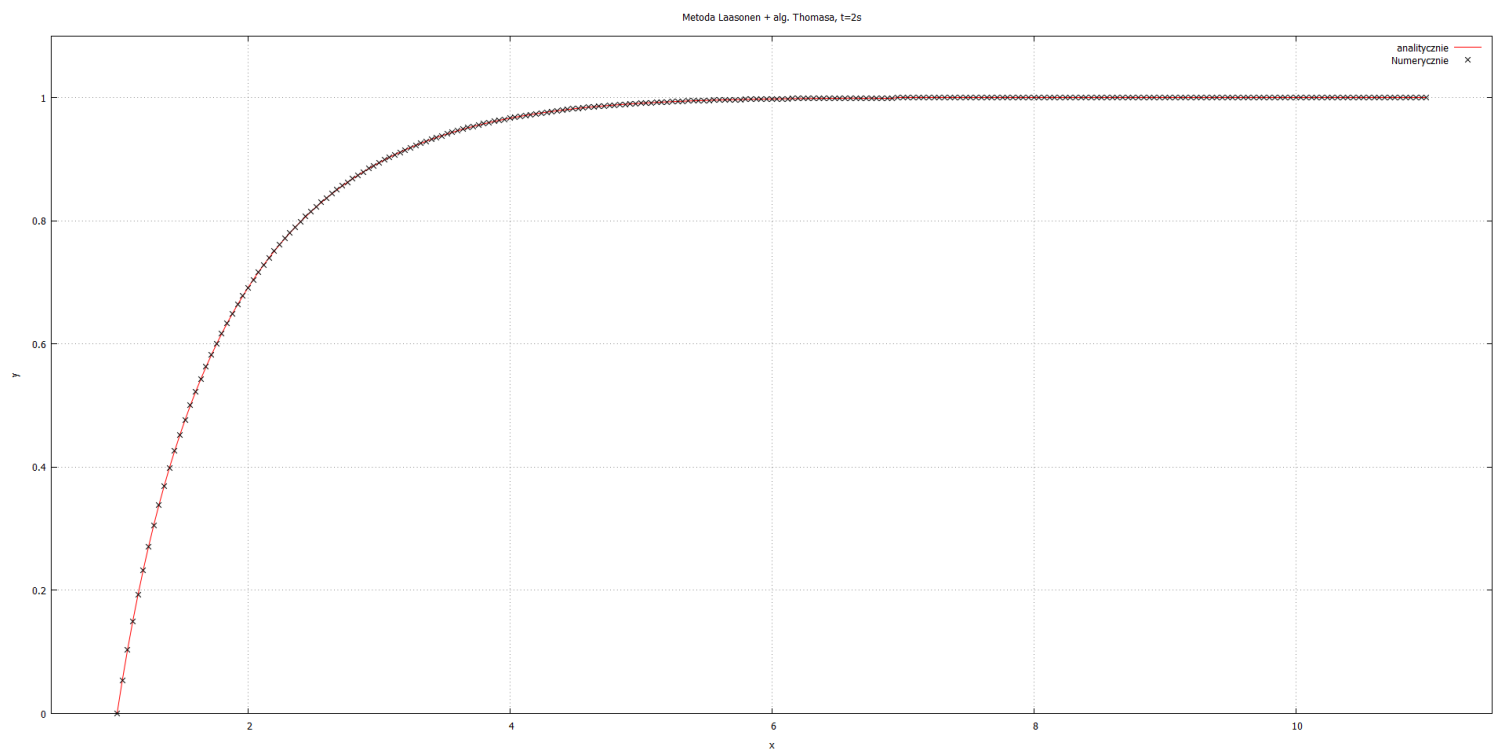
- **t=1s**



- **t=1.6s**

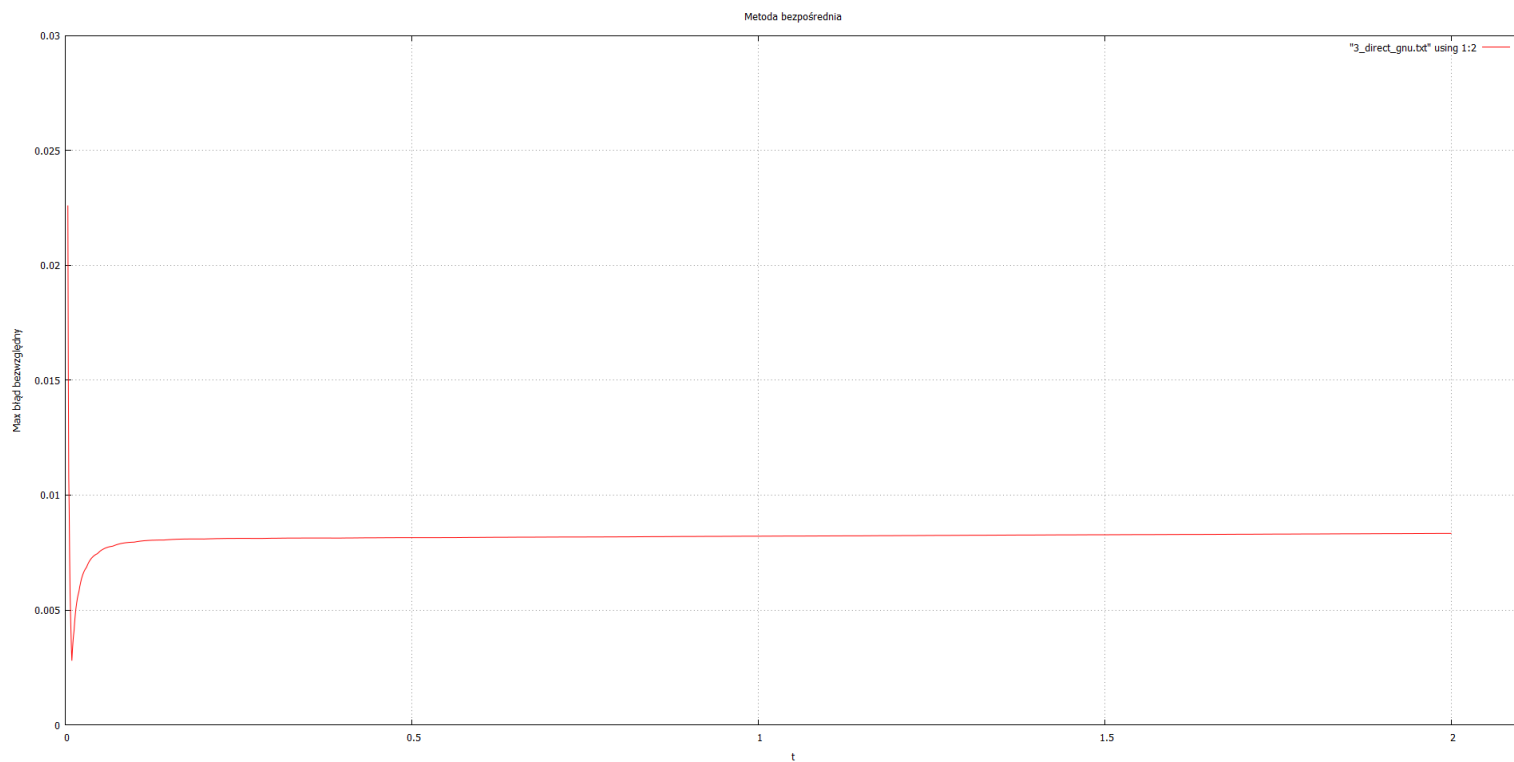


- **t=2s**

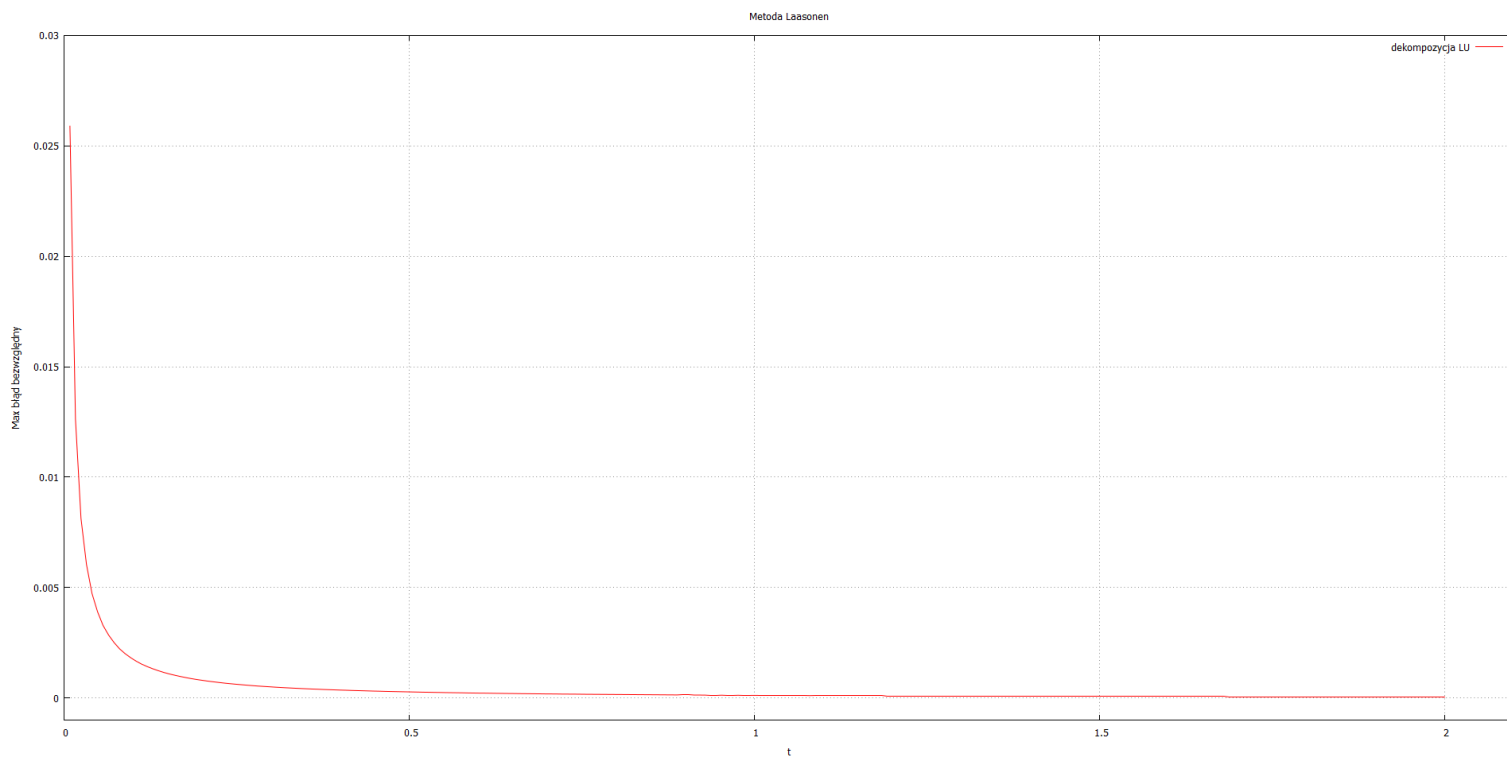


(3) Wykresy zależności maksymalnej wartości bezwzględnej błędu w funkcji czasu t

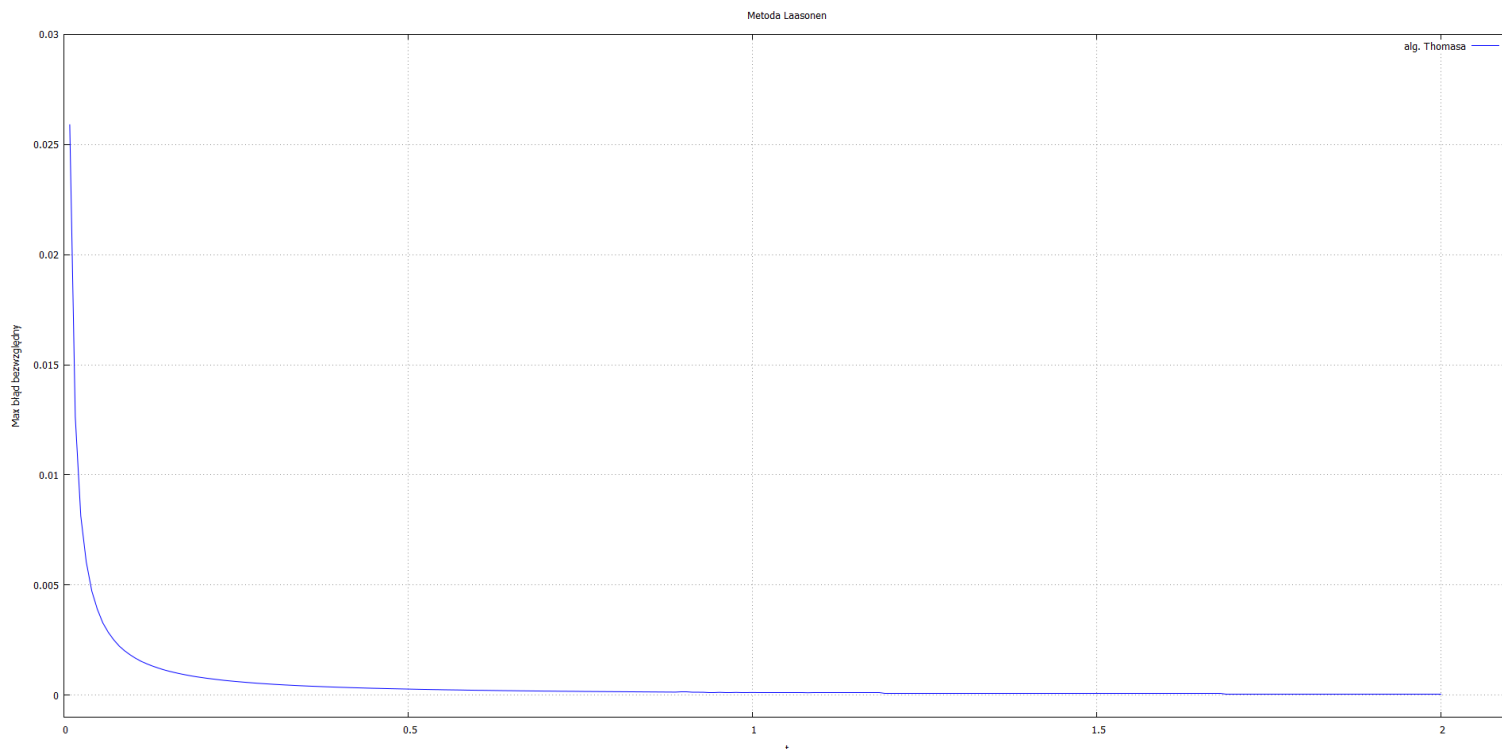
a) Klasyczna metoda bezpośrednia:



b) Metoda pośrednia Laasonen + Dekompozycja LU:



c) Metoda pośrednia Laasonen + alg. Thomasa:



Podsumowanie:

(1) Wykonane wykresy zależności max błęd bezwzględny od wielkości kroku h pozwoliły mi na odczytanie rzędów dokładności:

- Klasyczna metoda bezpośrednia $\rightarrow 1$
- Pośrednia metoda Laasonen + Dekompozycja LU $\rightarrow 2$
- Pośrednia metoda Laasonen + alg. Thomasa $\rightarrow 2$

W przypadku metody bezpośredniej nie udało się uzyskać tak dobrego wyniku jak dla metody pośredniej, jest ona za to dużo szybsza. Obliczenia dla klasycznej metody bezpośredniej były dużo mniej wymagające niż dla Laasonen.

(2) Dzięki ukazaniu wykresów w różnym czasie można lepiej zrozumieć co opisuje dane równanie. Dodatkowo dzięki porównaniu obliczeń analitycznych (linia) i numerycznych (punkty) można zauważyć, że punkty dla metody Laasonen (Thomasa i Dekompozycji) dość dobrze oddają rzeczywisty wykres, natomiast metoda bezpośrednia lekko odstaje (szczególnie na początku).

(3) Jak widać im dalej funkcja idzie w czasie tym dokładniejsze wyniki zwraca. Dla początkowych wartości czasu spadek jest najszybszy, później zwalnia – można wysunąć wniosek, że błędy z poprzednich poziomów są tłumione i wraz z kolejnymi poziomami czasu otrzymujemy coraz lepszy wynik.

Zarówno Metoda Thomasa jak i Dekompozycji są dokładne, i dla Metody pośredniej Laasonen dają bardzo podobne, jeżeli nie identyczne wyniki. Niepoprawności można doszukiwać się w szczególności w błędach maszynowych oraz błędach przybliżeń.