NUMERICAL METHODS

APPROXIMATE SOLVING OF INTEGRAL EQUATIONS

Sergio Barbero, Mateusz Korusiewicz

Introduction to the problem

We will study the next Fredholm II equation:

$$y_{n+1}(x)=\lambda\sum_{i=1}^{j-1}k(x_j-t_i)\,y_n(t_i)\,\Delta+f(x_j)$$

Our task is analyze the error for a given set of points between this numerical method and the analytic solution given by Laplace transformation:

$$Y(s)=\lambda K(s)+F(s) \qquad Y(s)=\frac{F(s)}{1-\lambda K(s)}$$

Our problem

We are going to analyze two different cases, note that $z=x-t;$

1. $k(z)=\sin(z);\quad f(x)=\cos(x);\quad \lambda=1$
2. $k(z)=z \quad f(x)=\frac{x^2}{2} \quad \lambda=-1;$

For this given set of points $x\in[0,7] \quad \Delta=0{,}5$

In our first case the function obtained analytically is: $y(x)=1$

In our second case the function obtained analytically is $y(x)=1-\cos(x)$

Implementation

```java
public double[][] NumericalSolution(int n, int N){
    double[][] y = new double[n][N];

    //Initialization of our matrix y
    for(int i = 0; i < N; i++)
        y[0][i] = 0;

    //Performing iterations
    for(int it = 1, prev = 0; it < n; prev++, it++){
        for(int j = 0; j < N; j++){
            double sum = 0;
            for (int i = 0; i < j; i++) {
                if(type == 0)
                    sum += Math.sin(axis[j] - axis[i]) * y[prev][i] * delta;
                else
                    sum += (axis[j] - axis[i]) * y[prev][i] * delta;
            }
            sum *= lambda;
            if(type == 0)
                y[it][j] = sum + Math.cos(axis[j]);
            else
                y[it][j] = sum + (axis[j]*axis[j])/2.0;
        }
        System.out.println();
    }
```

This is the function which calculates the approximation of our integral, for type 0 we will run the first exercice, and for any other value we will perform the second one.

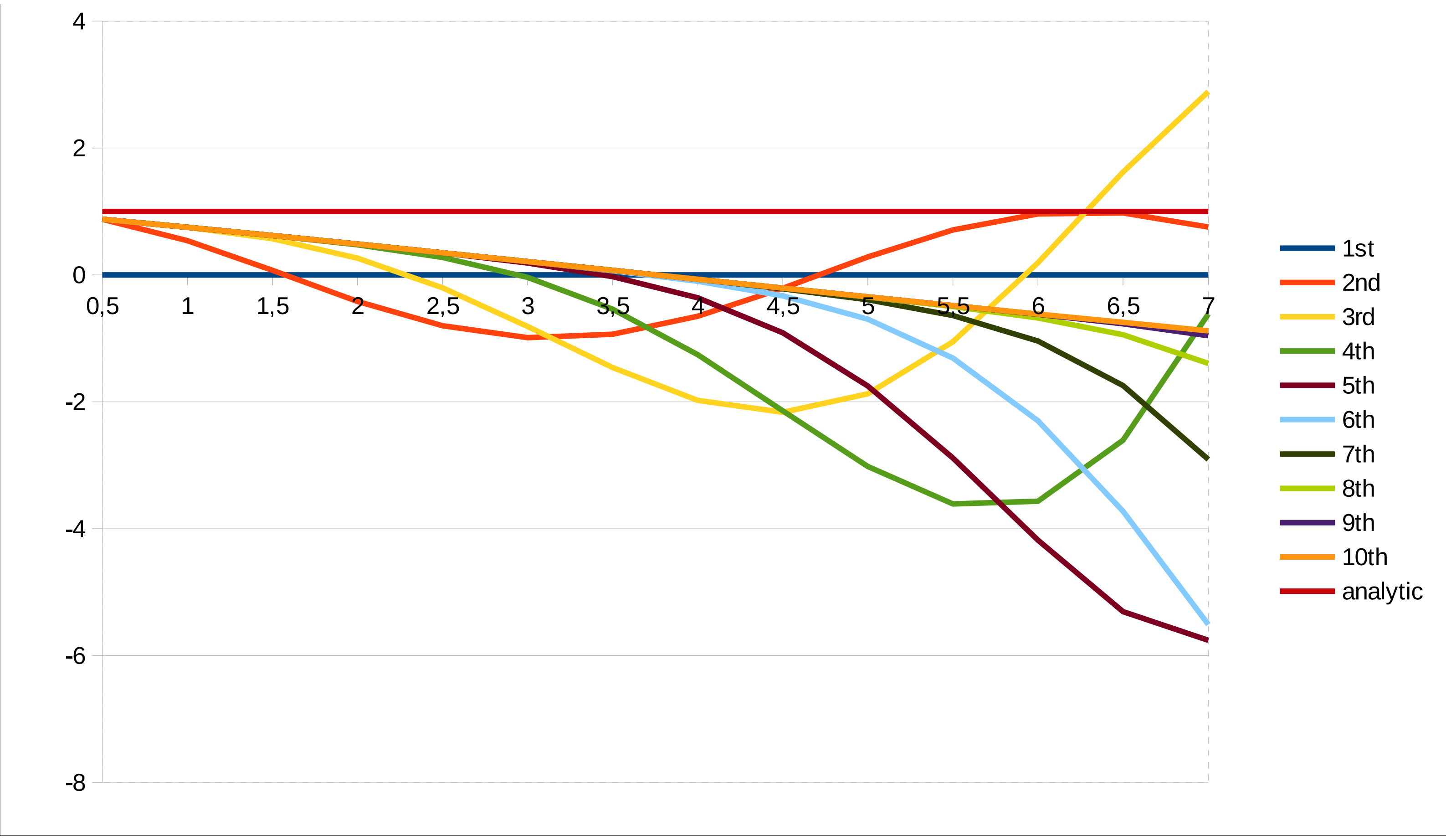
Results

We are going to analyze the result and also the absolute error

1. Results of first exercice

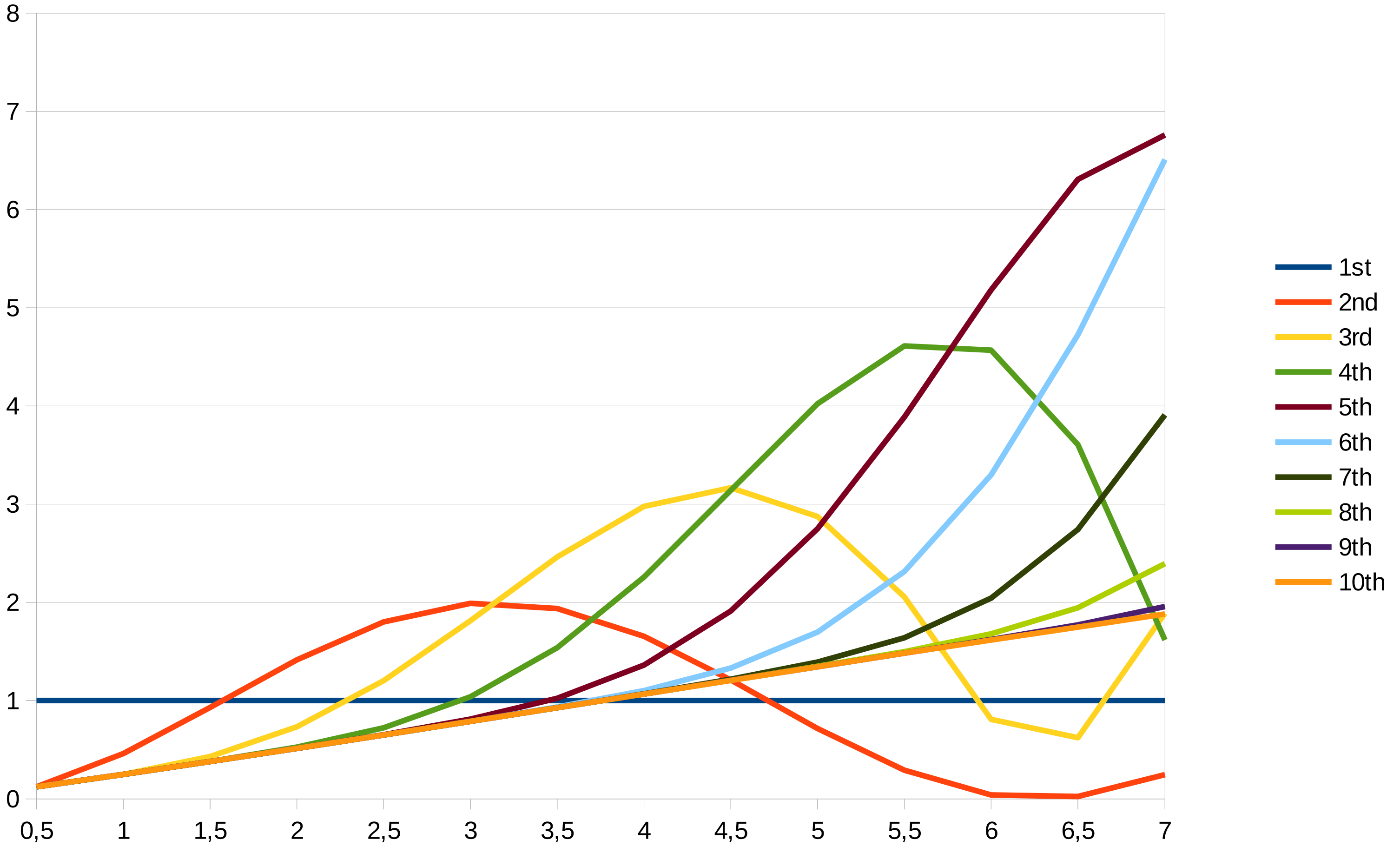
x	firstIt	secIt	thrIt,,,	nIt							Analytic
0,5	0	0,8775825619	0,8775825619	0,8775825619	0,8775825619	0,8775825619	0,8775825619	0,8775825619	0,8775825619	0,8775825619	1
1	0	0,5403023059	0,7506700521	0,7506700521	0,7506700521	0,7506700521	0,7506700521	0,7506700521	0,7506700521	0,7506700521	1
1,5	0	0,0707372017	0,569484695	0,61991253	0,61991253	0,61991253	0,61991253	0,61991253	0,61991253	0,61991253	1
2	0	-0,416146837	0,2658262336	0,4738915537	0,4859797496	0,4859797496	0,4859797496	0,4859797496	0,4859797496	0,4859797496	1
2,5	0	-0,801143616	-0,202671471	0,2755673401	0,3466600341	0,3495577291	0,3495577291	0,3495577291	0,3495577291	0,3495577291	1
3	0	-0,989992497	-0,813592487	-0,038808309	0,1885228623	0,2106506219	0,2113452364	0,2113452364	0,2113452364	0,2113452364	1
3,5	0	-0,936456687	-1,462631529	-0,538710838	-0,025074128	0,0653602333	0,0718837029	0,0720502109	0,0720502109	0,0720502109	1
4	0	-0,653643621	-1,978047988	-1,25732962	-0,361667965	-0,101943209	-0,069509786	-0,067653778	-0,067613864	-0,067613864	1
4,5	0	-0,210795799	-2,165856035	-2,142273622	-0,911110835	-0,330983827	-0,218321943	-0,207456145	-0,20694118	-0,206931613	1
5	0	0,2836621855	-1,873917433	-3,021897674	-1,751012339	-0,697392963	-0,393022325	-0,348800571	-0,345331966	-0,345191729	1
5,5	0	0,7086697743	-1,05518104	-3,61085772	-2,886592414	-1,312795996	-0,638671132	-0,498958921	-0,482785393	-0,481717354	1
6	0	0,9601702867	0,1917776666	-3,568694233	-4,181002696	-2,298733845	-1,042644936	-0,680052352	-0,621642943	-0,616031593	1
6,5	0	0,9765876257	1,62194759	-2,606818379	-5,307219278	-3,724101212	-1,742378752	-0,944580555	-0,771317853	-0,748535565	1
7	0	0,7539022543	2,8891087002	-0,617900922	-5,758931138	-5,509886951	-2,908232096	-1,39551293	-0,958575272	-0,882666663	1

For our set of values on the x-axis, this will be our table of approximations for each iteration, every column represent one single iteration and every row one value on x-axis.



This is the graphic representation, we can observe for every iteration we are getting closer of our analytical solution, being the 10th iteration not much better than the 9th, so we can assert from that iteration we don't have too much margin of improvement.

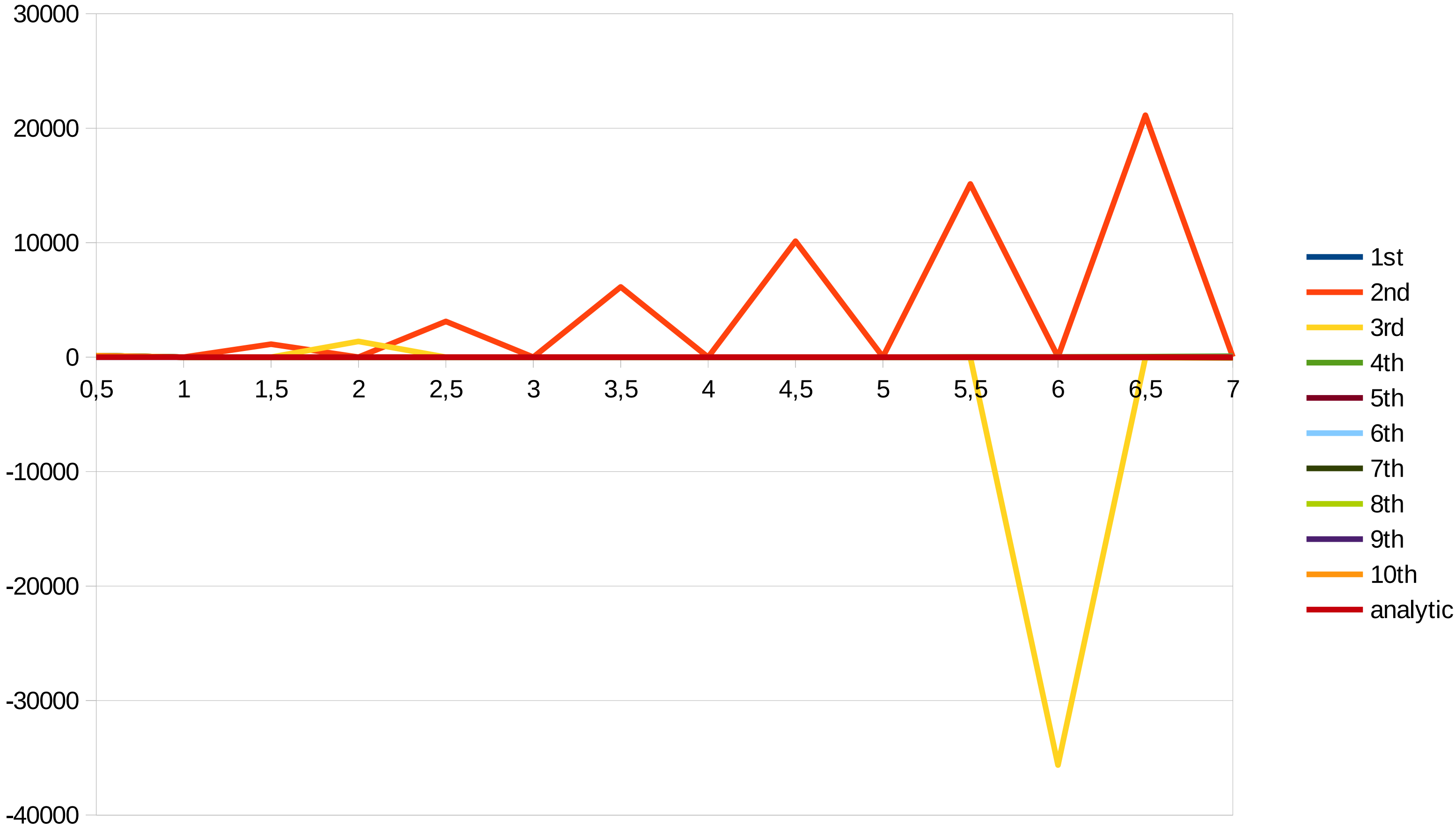
Taking into account this data we can get easily our absolute error graph:



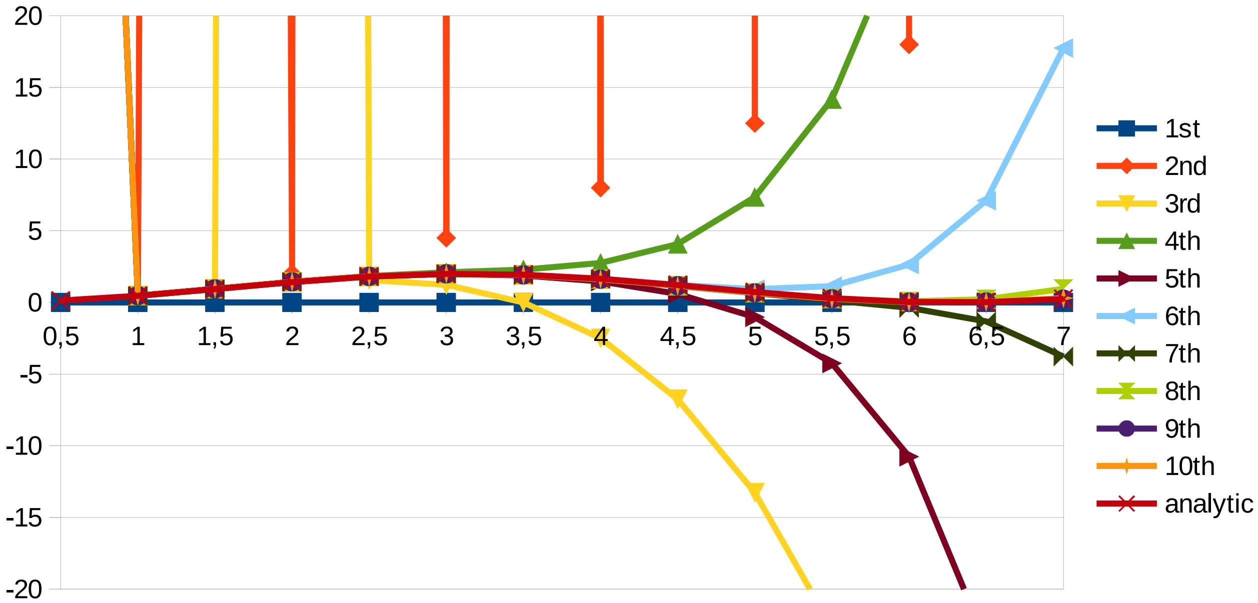
In this case the closer to 0 the more accurate, we can observe improvements for each single iteration.

2. Results of the second exercice

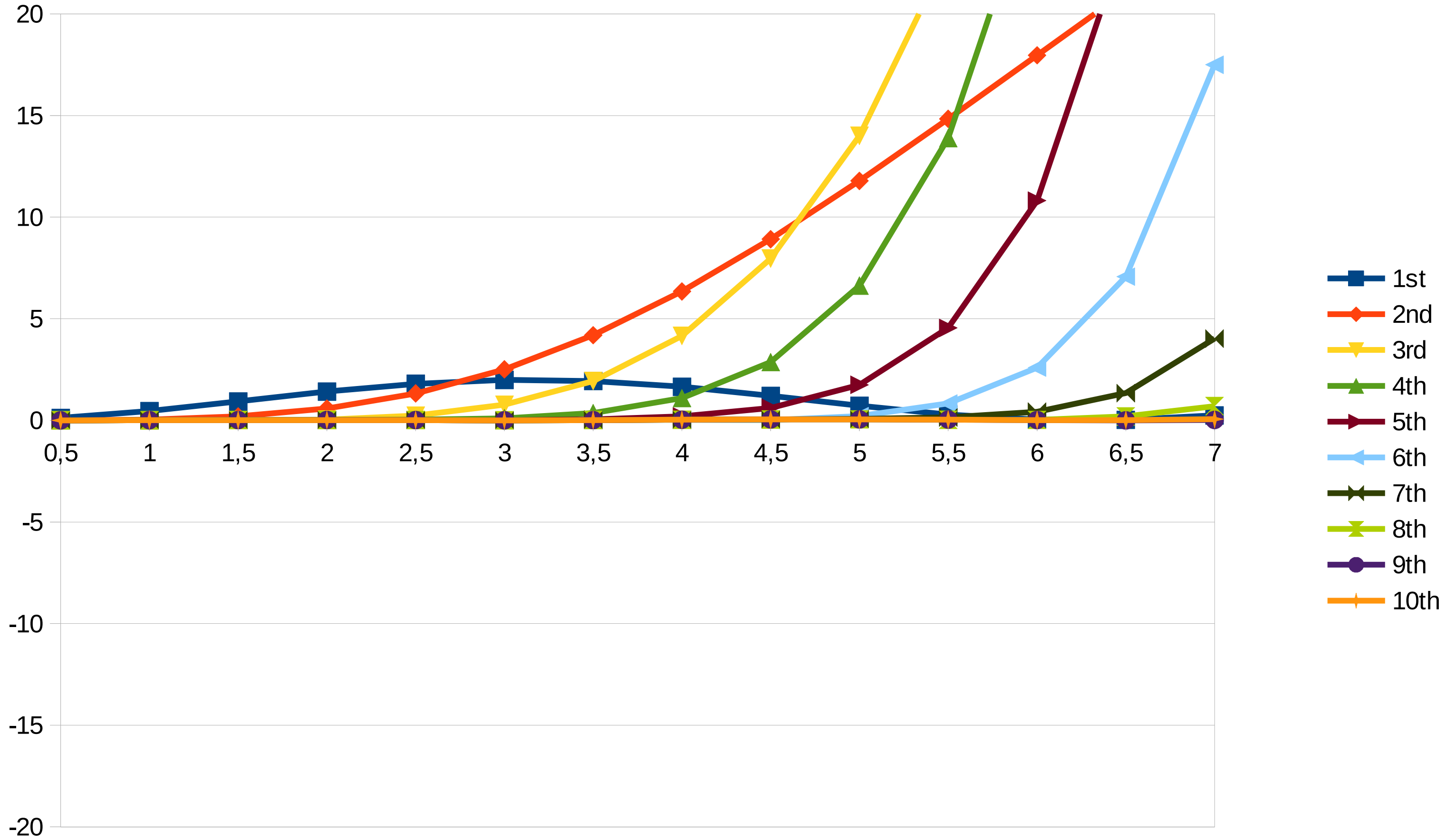
x	firstIt	secIt	thrIt,,,	nIt							analytic
0,5	0	125	125	125	125	125	125	125	125	125	0,1224174381
1	0	0,5	0,46875	0,46875	0,46875	0,46875	0,46875	0,46875	0,46875	0,46875	0,4596976941
1,5	0	1125	0,9375	0,9453125	0,9453125	0,9453125	0,9453125	0,9453125	0,9453125	0,9453125	0,9292627983
2	0	2	1375	1,4375	1,435546875	1,435546875	1,435546875	1,435546875	1,435546875	1,435546875	1,4161468365
2,5	0	3125	1,5625	1,8359375	1,81640625	1,8168945313	1,8168945313	1,8168945313	1,8168945313	1,8168945313	1,8011436155
3	0	4,5	1,21875	2,09375	1,98828125	1,994140625	1,9940185547	1,9940185547	1,9940185547	1,9940185547	1,9899924966
3,5	0	6125	0	2,296875	1,88671875	1,9243164063	1,9226074219	1,9226379395	1,9226379395	1,9226379395	1,9364566873
4	0	8	-2,5	2,75	1,4609375	1,6328125	1,6201171875	1,6206054688	1,6205978394	1,6205978394	1,6536436209
4,5	0	10125	-6,75	4,078125	0,59765625	1,2260742188	1,1594238281	1,1635437012	1,1634063721	1,1634082794	1,2107957994
5	0	12,5	-13,28125	7,34375	-1,03515625	0,919921875	0,6422119141	0,6666259766	0,6653289795	0,6653671265	0,7163378145
5,5	0	15125	-22,6875	14,1796875	-4,25390625	1,1225585938	0,1450195313	0,2591552734	0,2505950928	0,2509937286	0,2913302257
6	0	18	-35625	26,9375	-10,76757813	2,638671875	-0,3828125	0,0654296875	0,0210723877	0,0239715576	0,0398297133
6,5	0	21125	-52,8125	48,8515625	-23,765625	7,0966796875	-1,3203125	0,2174072266	0,0251922607	0,0416812897	0,0234123743
7	0	24,5	-75,03125	84,21875	-48,7265625	17,74609375	-3,781982422	0,9494628906	0,2249603271	0,3030929565	0,2460977457



In this case the first iterations are very inaccurate, for every iteration the improvement is really big, we cannot see the plots of last iterations properly, so let's zoom in the graph:



We could say the last iterations are very good approximations to our analytic solution, let's check it with the absolute error graph:



As we said, this approximation is very close to the analytical solution, being the 10th iteration a pretty good approximation, with an error between 0.02 and 0.05 for our points.