

Aitken's Algorithm and its Application at WolframMathematica

Bilall I. Shaini

State University of Tetova, Tetovë, R. Macedonia, bilall.shaini@unite.edu.mk

Teuta Zenku

State University of Tetova, Kërçovë, R. Macedonia, teuta_zenku@yahoo.com

Abstract

In this paper we will submitted the problem of interpolation of functions with Aitken's algorithm, by examining its implementation in application software WolframMathematica.[1]
This algorithm is quite suitable to be programmed and requires a small number of arithmetic operations compared to the interpolation polynomial of Lagrange. This programming option will use the WolframMathematica where we will provide a program for finding the polynomial P_n of power no greater than n who will have the same values with the function f at the interpolation nodes x_0, x_1, \dots, x_n of the segment $[a, b]$ it mean $P_n(x_i) = f(x_i)$ for $i = 0, 1, \dots, n$ and we will find the value of this interpolation polynomial in a point x different of nodes, thus estimating also the error^{[2],[3]}.

Keywords: Aitken's Algorithm, interpolation polynomial, Wolfram Mathematica application software.

Introduction

Interpolation methods and mathematical modeling have become essential in many areas of modern life. Sophisticated numerical analysis software is being embedded in popular software packages, e.g. spreadsheet programs, allowing many people to perform modeling even when they are unaware of the mathematics involved in the process.

With the growth in importance of using computers to carry out numerical procedures in solving mathematical models of the world, an area known as scientific computing has taken shape during the 1980s and 1990s.

This area looks at the use of numerical analysis from a computer science perspective. It is concerned with using the most powerful tools of numerical analysis, computer graphics, symbolic mathematical computations, and graphical user interfaces to make it easier to set up, solve, and interpret complicated mathematical models of the real world^[5].

Polinomial interpolation

Through any two points there is a unique line. Through any three points, a unique quadratic. Et cetera. The interpolation polynomial of degree n through the $n+1$ points $y_0 = f(x_0), y_1 = f(x_1), \dots, y_n = f(x_n)$ is given explicitly by Lagrange's classical formula (where x_0, x_1, \dots, x_n are elements of a given segment $[a, b]$ and are called points or nodes of interpolation)

$$L_n(x) = \sum_{i=0}^n \frac{(x-x_0) \cdots (x-x_{i-1})(x-x_{i+1}) \cdots (x-x_n)}{(x_i-x_0) \cdots (x_i-x_{i-1})(x_i-x_{i+1}) \cdots (x_i-x_n)} y_i$$

Aitken's Algorithm

It is not terribly wrong to implement the Lagrange formula straightforwardly, but it is not terribly right either. The resulting algorithm gives no error estimate, and it is also somewhat awkward to program. A much better algorithm (for constructing the same, unique, interpolating polynomial) is *Aitken's algorithm*.

The Aitken interpolation method is well- known and it is considered as the state- of- the- art for the interpolation of functions over real numbers. Furthermore, Aitken interpolation method is constructive in a way that permits the addition of a new interpolation point with low computational cost. [3]

Let construct the Aitken's interpolation polynomial. Let give the interpolation nodes x_0, x_1, \dots, x_n and there corresponding values $y_0 = f(x_0), y_1 = f(x_1), \dots, y_n = f(x_n)$ of the function f . Now we construct the interpolation polynomial $P(x)$ of degree n , for those notes, where \hat{x} is a point different of nodes.

We write $P_{i,0} = f(x_i)$ for $0, 1, \dots, n$ and with recurrent formula we have:

$$P_{i,k+1}(\hat{x}) = \frac{P_{k,k}(x)(x_i - \hat{x}) - P_{i,k}(x)(x_k - \hat{x})}{x_i - x_k} = \frac{1}{x_i - x_k} \begin{vmatrix} P_{k,k} & x_k - \hat{x} \\ P_{i,k} & x_i - \hat{x} \end{vmatrix} \text{ for } k < n - 1.$$

[5],[6],[7]

Aitken's algorithm with WolframMathematica

WolframMathematica is applicative software that has the power to program on it. Now we will see the one example solved with this software^[3].

Example 1: Given $f(x) = e^x$ and the nodes $\left\{ x_i = 1.0 + \frac{i}{4} \right\}_{i=0,1,2,3}$ compute the entries in Aitken's tableau for

evaluating $f(x)$ at $\hat{x} = 1.6$.

Solution:

First we will give the program for solving this example then the result that we take.

➤ Here we give the function and nodes. And we take:

```
f[x_] = E^x;
```

```
x_i = 1.0 + i/4;
```

```
x_hat = 1.6;
```

```
nodes = Table[x_i, {i, 0, 3}];
```

```
Print["Approximate "];
```

Approximate

```
Print["f[x]=", f[x], " ne piken x_hat=", x_hat];
```

f[x]=e^x at the point x_hat=1.6

```
Print["Using the nodes "];
```

Using the nodes

```
Print["{x_i} = ", nodes];
```

{x_i} = {1., 1.25, 1.5, 1.75}

➤ At this part we compute the coefficients $P(\hat{x})$.

```

P0,0 = f[x0];
P1,0 = f[x1];
P2,0 = f[x2];
P3,0 = f[x3];
Print["P0,0[" , x̂, "]=", f[x0]];
Print["P1,0[" , x̂, "]=", f[x1]];
Print["P2,0[" , x̂, "]=", f[x2]];
Print["P3,0[" , x̂, "]=", f[x3]];

P3,1 = 1 / (x3 - x2) Det[[f[x2] x2 - x̂],
                        [f[x3] x3 - x̂]];
Print["P3,1[" , x̂, "]=", P3,1];

P0,1,2 = 1 / (x2 - x0) Det[[P1,1 x0 - x̂],
                        [P2,1 x2 - x̂]];
Print["P0,1,2[" , x̂, "]=", P0,1,2];

P1,2,3 = 1 / (x3 - x1) Det[[P2,1 x1 - x̂],
                        [P3,1 x3 - x̂]];
Print["P1,2,3[" , x̂, "]=", P1,2,3];

P0,1,2,3 = 1 / (x3 - x0) Det[[P0,1,2 x0 - x̂],
                        [P1,2,3 x3 - x̂]];
Print["P0,1,2,3[" , x̂, "]=", P0,1,2,3];
    
```

➤ Then we take:

```

P0,0[1.6]=2.71828   P2,1[1.6]= 4.87823
P1,0[1.6]=3.49034   P3,1[1.6]= 4.99085
P2,0[1.6]=4.48169   P0,1,2[1.6]= 4.93963
P3,0[1.6]=5.7546    P1,2,3[1.6]= 4.95707
P1,1[1.6]= 4.57123  P0,1,2,3[1.6]= 4.95358
    
```

➤ Finally those coefficients we put at one tableau.

```

Print["The Aitken's tableau is"];
Print[TableForm[
    {
        {0, x0, P0,0, "", "", ""},
        {1, x1, P1,0, P1,1, "", ""},
        {2, x2, P2,0, P2,1, P0,1,2, ""},
        {3, x3, P3,0, P3,1, P1,2,3, P0,1,2,3}
    },
    TableHeadings -> {None, {"i", "xi", "Pi,0=f[xi]", "Pi,j", "Pi,j,k", "Pi,j,k,m"}}]];
    
```

➤ At this example it look like:

The Aitken's tableau is

i	x _i	P _{i,0} =f[x _i]	P _{i,j}	P _{i,j,k}	P _{i,j,k,m}
0	1.	2.71828			
1	1.25	3.49034	4.57123		
2	1.5	4.48169	4.87823	4.93963	
3	1.75	5.7546	4.99085	4.95707	4.95358

➤ Here we see the approximation and the sequence of Aitken approximation.

```
Print["f[x]=", f[x]];
Print["We have te approximation"]
Print["f[" , x̂, "]~", P0,1,2,3];
Print["It is close to the true value"];
Print["f[" , x̂, "]=", f[x̂]];
Print[" x̂ = ", x̂];
Print[" The sequence of Aitken approximations"];
Print["{P0,0, P1,1, P0,1,2, P0,1,2,3}="];
Print[{P0,0, P1,1, P0,1,2, P0,1,2,3 }, "->", f[x̂]];
```

➤ And we take:

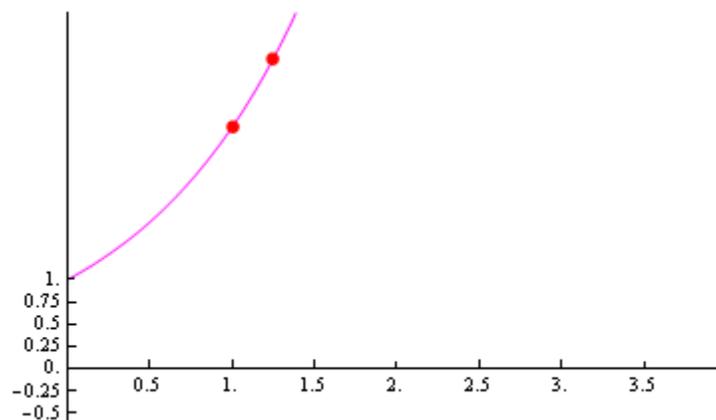
```
We have te approximation
f[1.6]~4.95358
It is close to the true value
f[1.6]=4.95303
x̂ = 1.6
The sequence of Aitken approximations
{P0,0, P1,1, P0,1,2, P0,1,2,3}=
{2.71828, 4.57123, 4.93963, 4.95358}->4.95303
```

➤ And at the end we give the error and the plot.

```
Print[" And the error"];
Print["{f[x̂]-P0,0, f[x̂]-P1,1, f[x̂]-P0,1,2, f[x̂]-P0,1,2,3 }="];
Print[f[x̂] - {P0,0, P1,1, P0,1,2, P0,1,2,3 }, "->0"];
dots = ListPlot[{{x0, f[x0]}, {x1, f[x1]}, {x2, f[x2]}, {x3, f[x3]}, PlotStyle -> {Red, PointSize[0.02]},
    DisplayFunction -> Identity];
gr10 = Plot[{f[x], P0,0[x], P1,1[x], P0,1,2[x], P0,1,2,3[x]}, {x, 0, 3.5}, PlotStyle -> {Magenta, Green, Blue, Red, Green, Blue},
    PlotRange -> {{0, 4}, {-0.6, 4}}, Ticks -> {Range[0, 3.5, 0.5], Range[-0.5, 1.0, 0.25]}, DisplayFunction -> Identity];
Show[gr10, dots, DisplayFunction -> $DisplayFunction]
```

➤ For our notes it is like this:

```
f[x]=e^
We have te approximation
f[1.6]~4.95358
It is close to the true value
f[1.6]=4.95303
x̂ = 1.6
The sequence of Aitken approximations
{P0,0, P1,1, P0,1,2, P0,1,2,3}=
{2.71828, 4.57123, 4.93963, 4.95358}->4.95303
And the error
{f[x̂]-P0,0, f[x̂]-P1,1, f[x̂]-P0,1,2, f[x̂]-P0,1,2,3 }=
{2.23475, 0.381804, 0.0134051, -0.000546169}->0
```



Conclusion

In connection with interpolation polynomial methods is possible with this application software to program other programs for their appointment for example with the method of Newton, divided differences etc. and their graphic presentation rapidly and accurately in a given interval according to our wishes. With those programs we save time and have fewer attempts.

References

- [1] John H., *Module for Aitken's and Neville's interpolation Methods*, Mathnews, 2005.
- [2] Stoer J., Bulirsch R., *Introduction to Numerical Analysis*, Springer-Verlag, Corrected 3rd printing, New York, 1996.
- [3] E.C. Laskari, G.C. Meletiou, M.N. Vrahatis, *Aitken and Neville Inverse Interpolation Methods over Finite Fields*, 2005.
- [4] Aitken scheme. M. K. Samarin, *Encyclopedia of Mathematics*. URL: http://www.encyclopediaofmath.org/index.php?title=Aitken_scheme&oldid=19113
- [5] Kendall E. Atkinson, *Numerical Analysis*, University of Iowa, Iowa.
- [6] Fatmir Hoxha, *Metoda të Analizës Numerike*, Tiranë, 2008.
- [7] Branko T., Naum C., *Elementi od Numerickata Matematika*, Skopje, 2002.