Comparison of the ARDC method of Agocs and Barnett with the Windowing Algorithm of Bremer

The preprint of Agocs and Barnett [1] claims their ARDC method achieves a ten times speedup over the windowing method of Bremer [2]. We replicate their experiments here and find that, in fact, the method of Bremer is faster.

We used the implementation of the method of [2] found at

`https://github.com/JamesCBremerJr/Phase-functions`.

Tmplementation of [1] we used is available at

`https://github.com/fruzsinaagocs/riccati`

A mirror of the repository of Agocs with the code we used to perform these experiments added can be found at:

`https://github.com/JamesCBremerJr/Aagocs_Riccati`

The experiments were run on a laptop equipped with an Intel 12th generation Core i5-1250P CPU and 16GB of memory. The code of [2] was compiled with GNU Fortran version 13.2.1 using the compiler flags "-Ofast -march=native".

1 Experiment of Section 4.2 of [1]

This experiment concerned the initial value problem

$$\begin{aligned} y''(t) + \nu^2(1 - t^2\cos(3t))y(t) &= 0, \qquad -1 < t < 1, \\ y(-1) &= 0 \\ y'(-1) &= \nu. \end{aligned} \tag{1}$$

For each $\nu = 2^0, 2^1, \ldots, 2^{20}$, we solved (1) using each of the two methods.

Owing to the limitations of the implementation of [1], we settled for a rather unsatisfactory method to measure the error in each obtained solution. While the algorithm[2] produces a piecewise Chebyshev expansion which allows for the rapid evaluation of the solution at any point on $[-1, 1]$, the implementation of [1] returns the values of the solution at a collection of points which are determined by the algorithm and it does not offer a facility to calculate the value of the solution at an arbitrary point in $[-1, 1]$ using this output. It does allow the user to specify a collection of points at which the solution should be evaluated as an input, but we didn't want to use this option in case it affects the running time of the procedure. Instead, we used an option present in the code of [1] which ensures the value of the solution at the right-hand endpoint of the solution interval is returned. The reported times were obtained by measuring the running time of each procedure 100 times and averaging the results. Table 1 gives the results. The columns under the heading "Kummer Phase" pertain to [2] and those under the heading "ARCDC" pertain to [1].

At lower frequencies, [2] appears to be both much faster and more accurate than [1]. At higher frequencies, the algorithms achieve similar levels of accuracy but [2] is slightly faster than [1].

2 Legendre Polynomials

In this experiment, we used each method to calculate the value of the Legendre polynomial $P_\nu(t)$ at $t = 0.9$ by solving Legendre's differential equation

$$(1 - t^2)y''(t) - 2ty'(t) + \nu(\nu + 1)y(t) = 0 \tag{2}$$

	Kummer Phase		ARDC	
ν	Time in Seconds	Error in $y(1)$	Time in Seconds	Error in $y(1)$
2^0	2.09×10^{-04}	7.21×10^{-16}	2.57×10^{-3}	8.18×10^{-13}
2^1	3.13×10^{-04}	2.22×10^{-16}	2.71×10^{-3}	4.71×10^{-13}
2^2	3.91×10^{-04}	2.49×10^{-16}	5.71×10^{-3}	3.82×10^{-12}
2^3	6.20×10^{-04}	4.88×10^{-15}	1.07×10^{-2}	4.29×10^{-13}
2^4	5.99×10^{-04}	3.33×10^{-15}	1.77×10^{-2}	6.59×10^{-12}
2^5	5.95×10^{-04}	7.66×10^{-15}	7.54×10^{-3}	1.69×10^{-12}
2^6	3.93×10^{-04}	3.50×10^{-14}	8.65×10^{-4}	7.82×10^{-14}
2^7	3.94×10^{-04}	3.96×10^{-14}	7.06×10^{-4}	3.77×10^{-14}
2^8	3.95×10^{-04}	1.14×10^{-14}	6.39×10^{-4}	6.71×10^{-14}
2^9	4.00×10^{-04}	9.79×10^{-14}	5.91×10^{-4}	9.51×10^{-14}
2^{10}	3.98×10^{-04}	1.38×10^{-13}	5.74×10^{-4}	4.16×10^{-13}
2^{11}	3.98×10^{-04}	3.38×10^{-13}	5.45×10^{-4}	9.33×10^{-13}
2^{12}	4.03×10^{-04}	7.21×10^{-14}	5.28×10^{-4}	4.80×10^{-13}
2^{13}	4.05×10^{-04}	3.08×10^{-12}	5.15×10^{-4}	1.61×10^{-12}
2^{14}	4.05×10^{-04}	1.56×10^{-12}	5.17×10^{-4}	9.50×10^{-13}
2^{15}	4.04×10^{-04}	8.68×10^{-13}	5.09×10^{-4}	2.50×10^{-13}
2^{16}	4.04×10^{-04}	5.38×10^{-12}	4.98×10^{-4}	1.60×10^{-11}
2^{17}	4.04×10^{-04}	3.38×10^{-12}	4.99×10^{-4}	3.89×10^{-11}
2^{18}	4.05×10^{-04}	7.20×10^{-11}	5.06×10^{-4}	4.32×10^{-11}
2^{19}	4.06×10^{-04}	2.40×10^{-10}	5.00×10^{-4}	2.78×10^{-12}
2^{20}	4.17×10^{-04}	6.18×10^{-12}	4.93×10^{-4}	8.65×10^{-12}

Table 1: The results of the experiment of Section 1.

over the interval $[0.0, 0.9]$. This corresponds to the experiment of Section 4.3 of [1]. Table 2 reports the absolute errors in the obtained values and the times taken by the algorithms. The reported times were obtained by measuring the running time of each procedure 100 times and averaging the results. For some values of ν, the algorithm of [1] failed with an error message. Corresponding entries of Table 2 are marked with "-".

We refer the reader to [3] for an algorithm which allows for the numerical solution of the differential equation defining the associated Legendre functions, something neither [1] or [2] can accomplish.

References

[1] Agocs, F. J., and Barnett, A. H. An adaptive spectral method for oscillatory second-order linear odes with frequency-independent cost, 2022.

[2] Bremer, J. On the numerical solution of second order differential equations in the high-frequency regime. *Applied and Computational Harmonic Analysis 44* (2018), 312–349.

[3] Bremer, J. Phase function methods for second order linear ordinary differential equations with turning points. *Applied and Computational Harmonic Analysis 65* (2023), 137–169.

	Kummer Phase		ARDC	
ν	Time in Seconds	Error in $P_\nu(0.9)$	Time in Seconds	Error in $P_\nu(0.9)$
2^0	5.09×10^{-04}	1.11×10^{-16}	8.60×10^{-4}	1.50×10^{-12}
2^1	5.08×10^{-04}	3.33×10^{-16}	1.71×10^{-3}	1.03×10^{-12}
2^2	5.32×10^{-04}	3.05×10^{-16}	1.75×10^{-3}	3.60×10^{-13}
2^3	7.96×10^{-04}	1.11×10^{-16}	2.87×10^{-3}	8.23×10^{-13}
2^4	1.30×10^{-03}	5.55×10^{-17}	6.63×10^{-3}	1.37×10^{-12}
2^5	1.23×10^{-03}	4.71×10^{-16}	1.29×10^{-2}	6.58×10^{-13}
2^6	1.29×10^{-03}	$0.00\times10^{+00}$	-	-
2^7	7.15×10^{-04}	5.84×10^{-15}	-	-
2^8	7.05×10^{-04}	1.31×10^{-15}	-	-
2^9	7.07×10^{-04}	5.92×10^{-15}	1.45×10^{-3}	7.03×10^{-15}
2^{10}	7.11×10^{-04}	4.51×10^{-16}	1.32×10^{-3}	1.49×10^{-14}
2^{11}	7.24×10^{-04}	3.50×10^{-15}	1.19×10^{-3}	1.01×10^{-14}
2^{12}	7.17×10^{-04}	5.49×10^{-15}	1.31×10^{-3}	4.67×10^{-14}
2^{13}	7.17×10^{-04}	1.75×10^{-15}	1.10×10^{-3}	1.48×10^{-14}
2^{14}	7.15×10^{-04}	2.74×10^{-16}	1.05×10^{-3}	3.22×10^{-14}
2^{15}	7.17×10^{-04}	3.36×10^{-15}	1.04×10^{-3}	2.75×10^{-14}
2^{16}	7.18×10^{-04}	4.52×10^{-15}	1.03×10^{-3}	4.80×10^{-14}
2^{17}	7.25×10^{-04}	5.37×10^{-14}	1.02×10^{-3}	1.34×10^{-13}
2^{18}	7.30×10^{-04}	2.70×10^{-14}	1.00×10^{-3}	2.22×10^{-13}
2^{19}	7.20×10^{-04}	9.73×10^{-14}	1.00×10^{-3}	1.40×10^{-13}
2^{20}	7.28×10^{-04}	1.21×10^{-14}	9.71×10^{-4}	4.74×10^{-13}

Table 2: The results of the experiment of Section 2 regarding Legendre's equation. Entires marked with "-" indicate cases in which the implementation of [1] failed.