

Maulana Advances in Inclusive Education

Vol. 2 No. 1 2026

ISSN: 3105-0107

Website : https://maulanapress.com/index.php/maulana

A Software Package for Modeling Unsteady and Oscillatory Motions of an Elastic Viscous Fluid in a Cylindrical Pipe

Rajabov Suxrob Xudoyberdiyevich*[1], Turdimurodov Otabek Farhod o'g'li[2], Tursunova Lola Bektosh qizi[3]

[1]Associate Professor, Department of Exact and Natural Sciences, Kattakurgan State Pedagogical Institute

[2]Assistant, Department of Algebra and Geometry, Samarkand State University

[3]2nd-year student, Faculty of Mathematics, Samarkand State University

Article information

DOI : xxx

*Correspondence : rajabovsuxrob72@gmail.com

Submission Track

Submission : July 30, 2026
Final Review : August 25, 2026
Accepted : September 31, 2026
Available online : October 31, 2026

Keywords

elastic-viscous fluid; unsteady flow; oscillatory flow; cylindrical pipe; Runge–Kutta method; Deborah number; Womersley number; software package.

Abstract

This paper describes the development of software tools intended for the mathematical modeling of unsteady and oscillatory flows of elastic-viscous fluids, for numerical computation, and for the graphical analysis of the results obtained. The developed programs make it possible to compute the discretized form of the mathematical model, to form the solution in time, to determine the influence of the main physical parameters on the flow process, and to present the calculation results in numerical and graphical form. Two independent programs are considered: the first computes the unsteady flow of an elastic-viscous fluid in a flat channel by the fourth-order Runge–Kutta method in time, and the second computes mathematical models of the flow in the radial coordinate, compares the numerical solution with the analytical one, and displays the results in two- and three-dimensional graphs.

This is an open access article under the CC BY-NC license
https://creativecommons.org/licenses/by-nc/4.0/

INTRODUCTION

Viscoelastic fluids combine viscous and elastic behavior, and the importance of elastic effects in a flow is usually characterized by the Deborah number, the ratio of the relaxation time of the fluid to the characteristic time of the flow [1], [2]. Constitutive equations for such media were formulated in a frame-invariant form by Oldroyd [3], and the rheology and fluid mechanics of viscoelastic liquids are treated in detail in the monographs [4]–[5].

Oscillatory flow in a circular tube under a periodic pressure gradient is characterized by the

dimensionless Womersley number; the classical solution for a Newtonian fluid was given by Womersley [6], and the subject is treated in the monographs [7], [8]. For viscoelastic fluids, unsteady and oscillatory flows in channels and tubes have been analyzed theoretically in [9]–[10].

The numerical modeling of such flows requires reliable schemes for the integration of the equations in time; one of the classical explicit one-step schemes is the fourth-order Runge–Kutta method [11]–[12], which is described in standard references on numerical methods [13]. Such calculations and the presentation of their results are conveniently organized in the Python programming language using the scientific libraries NumPy [14], SciPy [15] and Matplotlib [16].

The aim of this work is to describe the development of software tools for the mathematical modeling of unsteady and oscillatory flows of elastic-viscous fluids, for numerical computation, and for the graphical analysis of the results obtained.

METHODOLOGY

The two programs

The functional capabilities, computational algorithms and software architecture of two independent software products are considered in sequence. The first program is intended for computing the unsteady flow of an elastic-viscous fluid in a flat channel in time by the fourth-order Runge–Kutta method. The second program makes it possible to compute mathematical models of the flow of an elastic-viscous fluid in the radial coordinate, to compare the numerical solution with the analytical solution, and also to display the results in two- and three-dimensional graphs.

Implementation

The Python programming language was used to develop the program. The NumPy and SciPy libraries were used in the computational processes, the Matplotlib library for graphical visualization, and Tkinter technology for building the graphical user interface.

Functions of the second program

The interface of the second program contains several functional buttons for performing the calculation and analyzing the results.

- The "**Calculate**" ("Hisoblash") button starts the calculation process on the basis of the selected problem and the entered parameters.
- The "**2D graph**" button shows the change of the flow velocity at the central point in time in a two-dimensional graph.
- The "**3D graph**" button depicts the distribution of the flow velocity over the radial coordinate and time as a three-dimensional surface.
- The "**Heatmap**" button depicts the flow field as a color map. This function makes it possible to assess in general form the change of the flow in space and time.
- The "**Animation**" button shows the change of the radial flow in time in dynamic form.
- The "**Table**" function shows the calculation results in the form of a numerical table. This option is used to examine exactly calculated values.
- The "**Save CSV**" button makes it possible to save the calculation results in CSV format.

- The "**Save PNG**" function serves to save the obtained graphical results as an image file.

The program also makes it possible to clear the results or to start a new calculation, which makes it more convenient to carry out several experiments in succession.

RESULTS

Interface and parameters

Since this software contains two problems, the following interface appears when the program is launched (Fig. 1):

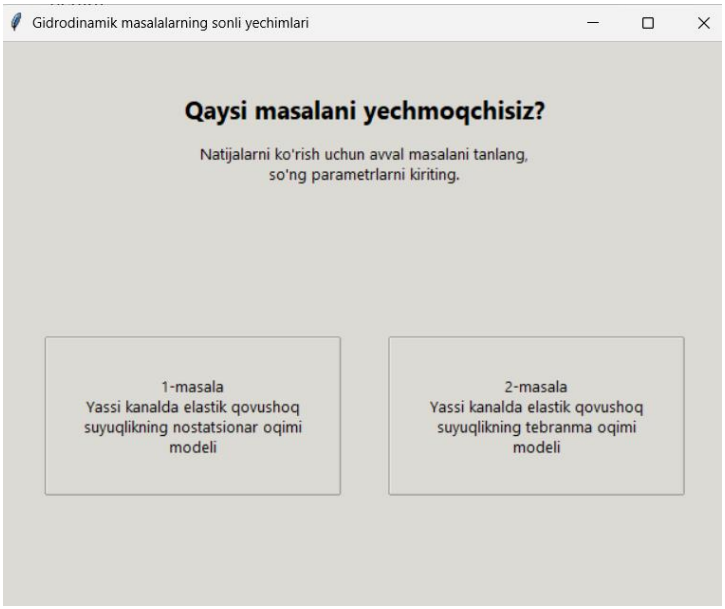


Fig. 1. Initial interface of the program at launch.

After a problem is selected, the field for entering the parameters of that problem is activated (Figs. 2 and 3):

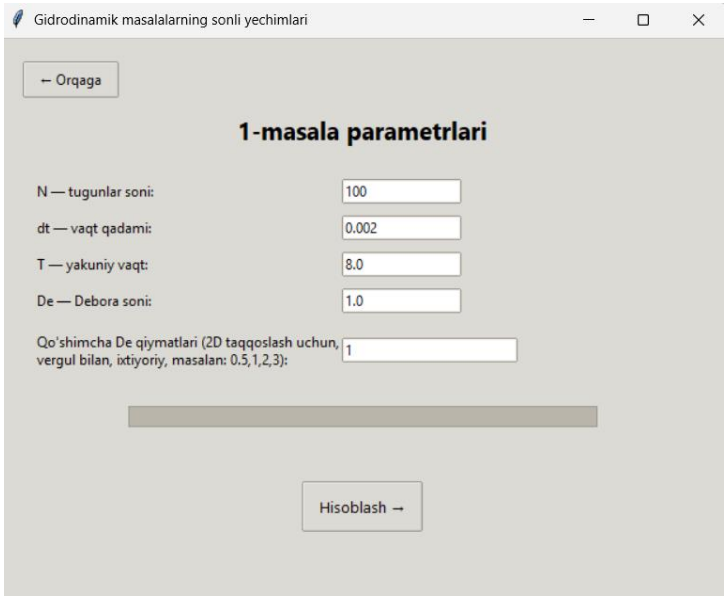


Fig. 2. Interface when problem 1 is selected.

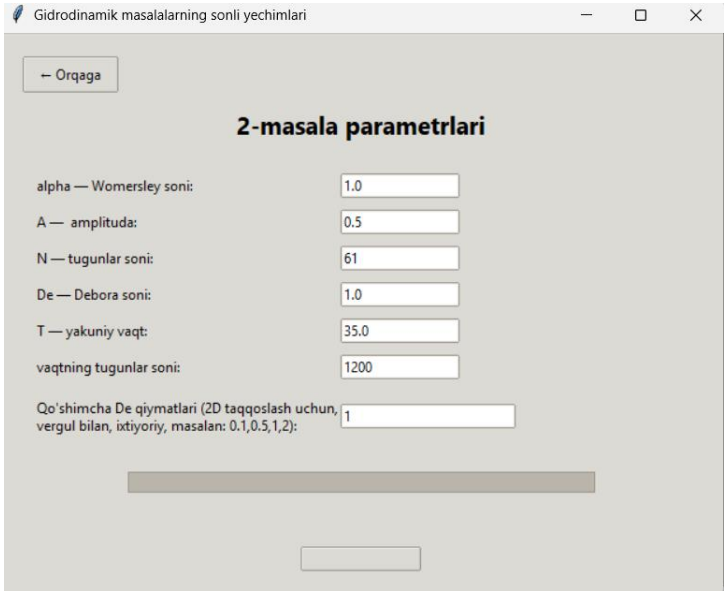


Fig. 3. Interface when problem 2 is selected.

Here N is the number of nodes across the channel width, dt is the number of nodes in time, T is the length of time over which the experiment is to be studied, Dye is the Deborah number, A is the amplitude, and $alpha$ is the Womersley number. After the necessary parameters have been entered, the "Calculate" button is pressed to obtain the results, upon which the following interface window appears (Fig. 4):

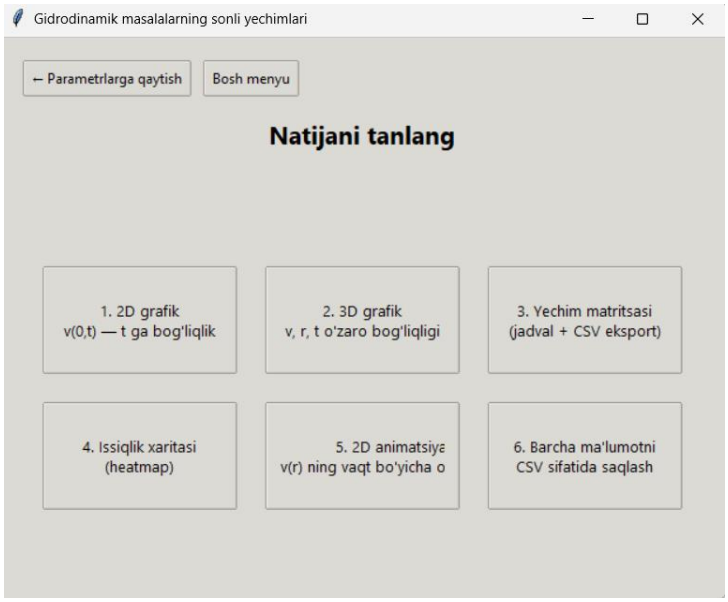


Fig. 4. Available results.

The results interface is the same for both models, as shown above.

Results for model 1

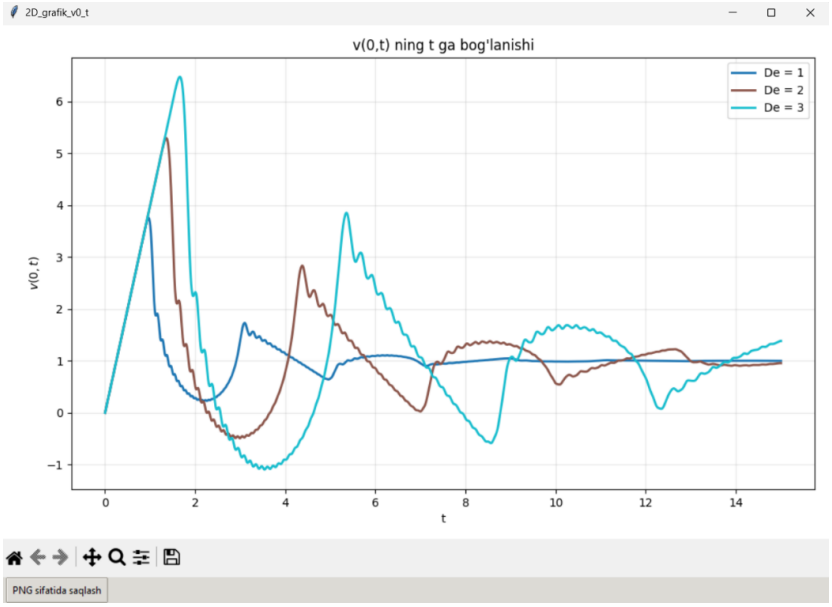


Fig. 5. Graph of the velocity at the center of the pipe versus time for model 1.

Unlike the previous program, in this program it is not necessary to choose a Deborah number in order to view the 3D graphs. The graph is a subplot, that is, the graphs for all entered values are displayed side by side in a single figure, which also makes it convenient to compare the results (Fig. 6):

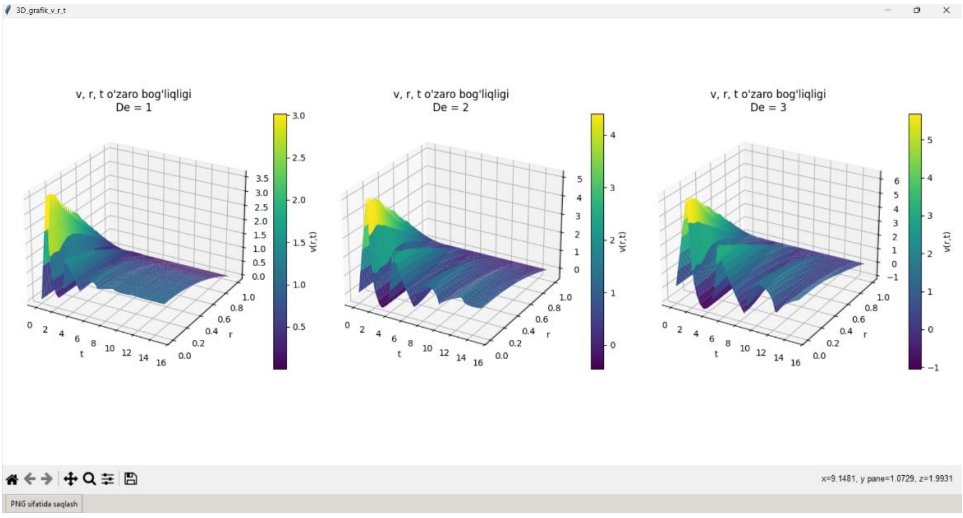


Fig. 6. Graph of the velocity versus time at the corresponding coordinates of the pipe for model 1.

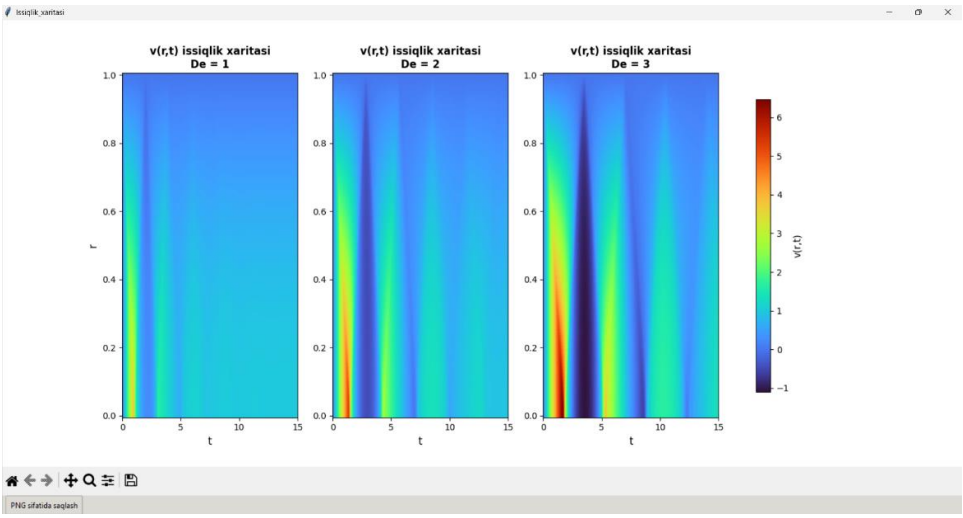


Fig. 7. Distribution of the velocity function over the radial coordinate.

This graph shows the distribution of the velocity function $v(r,t)$ in time and over the radial coordinate for different Deborah numbers (De = 1, 2, 3), and shows how the oscillation and distribution characteristics of the flow velocity in time change as the Deborah number increases.

Animation, solution matrix and availability

When the 2D animation button is pressed, all the entered Deborah values are displayed in one window (Fig. 8):

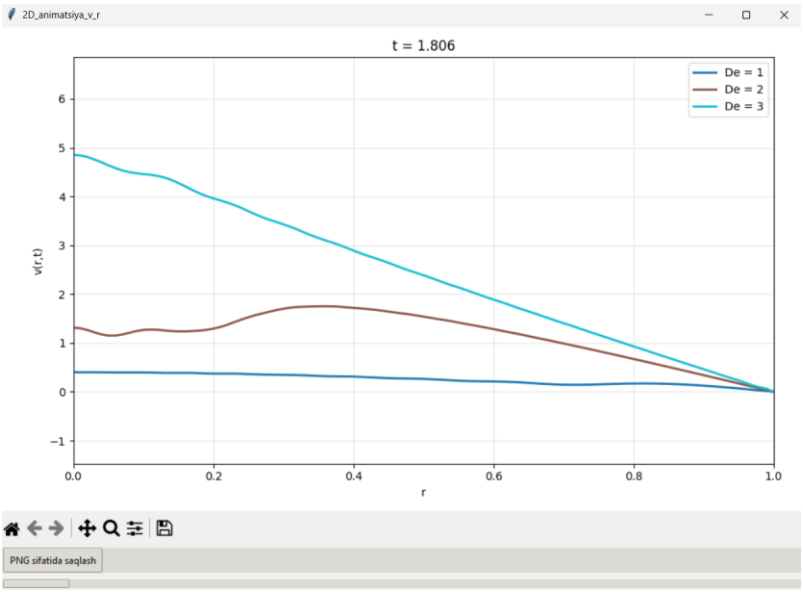
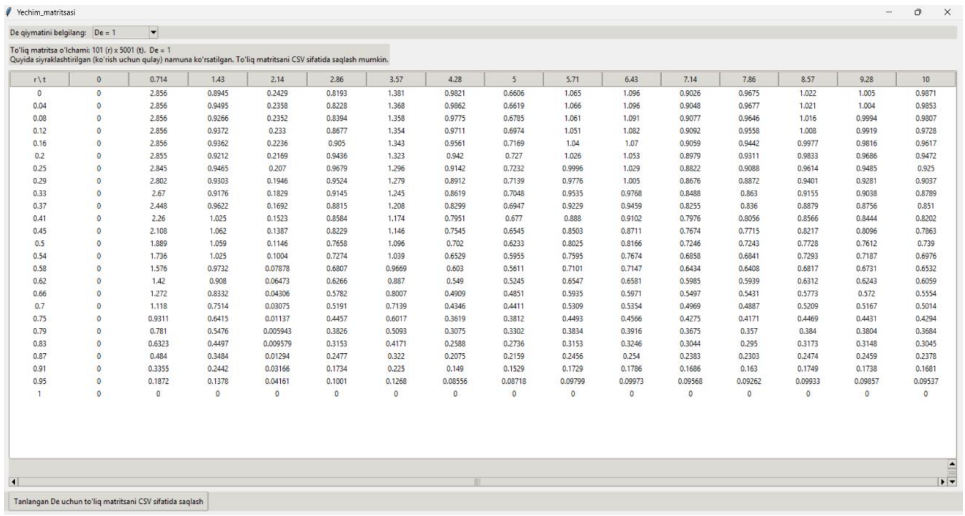


Fig. 8. Animation of the dependence of the velocity function on the pipe radius.

When the solution matrix button is pressed, the solution appears in the form of a table (Fig. 9):



r\t	0	0.714	1.43	2.14	2.86	3.57	4.28	5	5.71	6.43	7.14	7.86	8.57	9.28	10
0	0	2.856	0.8945	0.2429	0.8193	1.381	0.9821	0.6606	1.065	1.096	0.9026	0.9675	1.022	1.005	0.9871
0.04	0	2.856	0.9495	0.2358	0.8228	1.368	0.9862	0.6619	1.066	1.096	0.9048	0.9677	1.021	1.004	0.9853
0.08	0	2.856	0.9266	0.2352	0.8394	1.358	0.9775	0.6785	1.061	1.091	0.9077	0.9646	1.016	0.9994	0.9807
0.12	0	2.856	0.9372	0.233	0.8677	1.354	0.9711	0.6974	1.051	1.082	0.9092	0.9558	1.008	0.9919	0.9728
0.16	0	2.856	0.9362	0.2236	0.905	1.343	0.9561	0.7169	1.04	1.07	0.9059	0.9442	0.9977	0.9816	0.9617
0.2	0	2.855	0.9212	0.2169	0.9436	1.323	0.942	0.727	1.026	1.053	0.8979	0.9311	0.9833	0.9686	0.9472
0.25	0	2.845	0.9465	0.207	0.9679	1.296	0.9142	0.7232	0.9996	1.029	0.8822	0.9088	0.9614	0.9485	0.925
0.29	0	2.802	0.9303	0.1946	0.9524	1.279	0.8912	0.7139	0.9776	1.005	0.8676	0.8872	0.9401	0.9281	0.9037
0.33	0	2.67	0.9176	0.1829	0.9145	1.245	0.8619	0.7048	0.9535	0.9768	0.8488	0.863	0.9155	0.9038	0.8789
0.37	0	2.448	0.9622	0.1692	0.8815	1.208	0.8299	0.6947	0.9229	0.9459	0.8255	0.836	0.8879	0.8756	0.851
0.41	0	2.26	1.025	0.1523	0.8584	1.174	0.7951	0.677	0.888	0.9102	0.7976	0.8056	0.8566	0.8444	0.8202
0.45	0	2.108	1.062	0.1387	0.8229	1.146	0.7545	0.6545	0.8503	0.8711	0.7674	0.7715	0.8217	0.8096	0.7863
0.5	0	1.889	1.059	0.1146	0.7658	1.096	0.702	0.6233	0.8025	0.8166	0.7246	0.7243	0.7728	0.7612	0.739
0.54	0	1.736	1.025	0.1004	0.7274	1.039	0.6529	0.5955	0.7595	0.7674	0.6858	0.6841	0.7293	0.7187	0.6976
0.58	0	1.576	0.9732	0.07878	0.6807	0.9669	0.603	0.5611	0.7101	0.7147	0.6434	0.6408	0.6817	0.6731	0.6532
0.62	0	1.42	0.908	0.06473	0.6266	0.887	0.549	0.5245	0.6547	0.6581	0.5985	0.5939	0.6312	0.6243	0.6059
0.66	0	1.272	0.8332	0.04306	0.5782	0.8007	0.4909	0.4851	0.5935	0.5971	0.5497	0.5431	0.5773	0.572	0.5594
0.7	0	1.118	0.7514	0.03075	0.5191	0.7139	0.4346	0.4411	0.5309	0.5354	0.4969	0.4887	0.5209	0.5167	0.5014
0.75	0	0.9311	0.6415	0.01137	0.4457	0.6017	0.3619	0.3812	0.4493	0.4566	0.4275	0.4171	0.4469	0.4431	0.4294
0.79	0	0.781	0.5476	0.005943	0.3826	0.5093	0.3075	0.3302	0.3834	0.3916	0.3675	0.357	0.384	0.3804	0.3684
0.83	0	0.6323	0.4497	0.009579	0.3153	0.4171	0.2588	0.2736	0.3153	0.3246	0.3044	0.295	0.3173	0.3148	0.3045
0.87	0	0.484	0.3484	0.01294	0.2477	0.322	0.2075	0.2159	0.2456	0.254	0.2383	0.2303	0.2474	0.2459	0.2378
0.91	0	0.3355	0.2442	0.03166	0.1734	0.225	0.149	0.1529	0.1729	0.1786	0.1686	0.163	0.1749	0.1738	0.1681
0.95	0	0.1872	0.1378	0.04161	0.1001	0.1268	0.08556	0.08718	0.09799	0.09973	0.09568	0.09262	0.09933	0.09857	0.09537
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 9. Solution matrix.

This program has also been developed as an .exe file for computers running the Windows operating system; it can be downloaded and used directly, and it can be used on any internet-connected device by scanning the following QR code:

DISCUSSION

The package combines two problems in a single environment with a common results interface, so that the same set of outputs (2D graph, 3D graph, heat map, animation, table and CSV/PNG export) is available for both models. In the second program, the 3D graphs and the animation display all the entered Deborah numbers together, which makes it convenient to compare the oscillation and distribution characteristics of the velocity for different values of the Deborah number, as in Fig. 7 for De = 1, 2 and 3. Although the second program is designed to compare the numerical solution with the analytical solution, of the kind available for oscillatory flows of viscoelastic fluids [17], [18], [19], quantitative results of such a comparison are not presented in this article; they remain a task for further work.

CONCLUSION

Two independent programs have been developed in Python for the modeling of unsteady and oscillatory flows of an elastic-viscous fluid: the first computes the unsteady flow in a flat channel by the fourth-order Runge–Kutta method, and the second computes the flow in the radial coordinate and offers 2D and 3D graphs, a heat map, animation, a table and CSV/PNG export for one or several Deborah numbers; the second program is available as a Windows .exe file and, through a QR code, on internet-connected devices.

REFERENCES

[1] M. Reiner, "The Deborah number," *Phys. Today*, vol. 17, no. 1, p. 62, 1964, doi: 10.1063/1.3051374.

[2] R. J. Poole, "The Deborah and Weissenberg numbers," *Rheol. Bull.*, vol. 53, no. 2, pp. 32–39, 2012.

[3] J. G. Oldroyd, "On the formulation of rheological equations of state," *Proc. Roy. Soc. London A*, vol. 200, pp. 523–541, 1950, doi: 10.1098/rspa.1950.0035.

[4] R. B. Bird, R. C. Armstrong, and O. Hassager, *Dynamics of Polymeric Liquids, Vol. 1: Fluid Mechanics*, 2nd ed. New York, NY, USA: Wiley, 1987.

[5] R. G. Larson, *The Structure and Rheology of Complex Fluids*. New York, NY, USA: Oxford Univ. Press, 1999.

[6] D. D. Joseph, *Fluid Dynamics of Viscoelastic Liquids*. New York, NY, USA: Springer, 1990.

[7] J. R. Womersley, "Method for the calculation of velocity, rate of flow and viscous drag in arteries when the pressure gradient is known," *J. Physiol.*, vol. 127, no. 3, pp. 553–563, 1955, doi: 10.1113/jphysiol.1955.sp005276.

[8] M. Zamir, *The Physics of Pulsatile Flow*. New York, NY, USA: Springer, 2000.

[9] T. J. Pedley, *The Fluid Mechanics of Large Blood Vessels*. Cambridge, U.K.: Cambridge Univ. Press, 1980.

[10] N. D. Waters and M. J. King, "Unsteady flow of an elastico-viscous liquid," *Rheol. Acta*, vol. 9, pp. 345–355, 1970, doi: 10.1007/BF01975401.

[11] L. Casanellas and J. Ortín, "Laminar oscillatory flow of Maxwell and Oldroyd-B fluids: Theoretical analysis," *J. Non-Newton. Fluid Mech.*, vol. 166, no. 23–24, pp. 1315–1326, 2011.

[12] D. Tsiklauri and I. Beresnev, "Enhancement in the dynamic response of a viscoelastic fluid flowing through a longitudinally vibrating tube," *Phys. Rev. E*, vol. 63, Art. no. 046304, 2001, doi: 10.1103/PhysRevE.63.046304.

[13] C. Runge, "Über die numerische Auflösung von Differentialgleichungen," *Math. Ann.*, vol. 46, pp. 167–178, 1895, doi: 10.1007/BF01446807.

[14] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, 2nd rev. ed. Berlin, Germany: Springer, 1993.

[15] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*, 3rd ed. Chichester, U.K.: Wiley, 2016.

[16] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[17] C. R. Harris *et al.*, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, 2020, doi: 10.1038/s41586-020-2649-2.

[18] P. Virtanen *et al.*, "SciPy 1.0: Fundamental algorithms for scientific computing in Python," *Nat. Methods*, vol. 17, pp. 261–272, 2020, doi: 10.1038/s41592-019-0686-2.

[19] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/MCSE.2007.55.