EVALUATION OF STATE-SPECIFIC TRANSPORT PROPERTIES USING MACHINE LEARNING METHODS

Vladimir A. Istomin
Saint Petersburg State University
Saint Petersburg
Russian Federation
v.istomin@spbu.ru

Elena V. Kustova
Saint Petersburg State University
Saint Petersburg
Russian Federation
e.kustova@spbu.ru

Semen M. Lagutin
Saint Petersburg State University
Saint Petersburg
Russian Federation
lasimondep@gmail.com

Ivan Y. Shalamov
Saint Petersburg State University
Saint Petersburg
Russian Federation
i.y.shalamov@gmail.com

Article history:
Received 17.04.2023, Accepted 04.06.2023

Abstract

In this study, machine learning algorithms are employed to calculate state-to-state transport coefficients in nonequilibrium reacting gas flows. The focus is on the evaluation of thermal conductivity, shear viscosity, and bulk viscosity coefficients under conditions of strong coupling between vibrational-chemical kinetics and gas dynamics. In order to solve a regression problem for evaluating state-to-state transport coefficients, a specific software application with user interface is developed, which allows loading, processing, and saving of data arrays; configuring model architecture; training and evaluating models with various optimizers, loss functions, and metrics; making predictions using trained models. Using the developed software the multi-layer perceptron regression model is constructed and trained. The model is assessed in a binary mixture of molecular and atomic nitrogen taking into account 48 vibrational states; the coefficients are computed in the wide temperature range for the varying mixture composition. Good agreement of the results with the original transport coefficients calculated using rigorous but computationally expensive kinetic theory algorithms is shown. Applying machine learning techniques yields a significant speedup of about two orders of magnitude in the computation of transport coefficients. It is concluded that implementation of machine learning methods may considerably reduce the computational efforts required for nonequilibrium flow simulations.

Key words

State-to-state Approach, Transport Coefficients, Machine Learning (ML), Neural Networks (NN)

1 Introduction

In modern aerospace applications, low-temperature plasma technologies, environmental problems gas flows are usually far from thermal and chemical equilibrium, which makes fluid-dynamic simulations rather challenging. Under nonequilibrium conditions, it is necessary to take into account excitation of internal degrees of freedom (translational, rotational, vibrational, electronic), chemical reactions, and ionization. Depending on the flow conditions, characteristic times of internal energy transitions and chemical reactions may vary by several orders of magnitude, and sometimes become comparable to the mean time of variation of fluid-dynamic parameters such as velocity, pressure, and temperature. In this case, there is a strong coupling between physical–chemical processes and gas dynamics, which alters the set of governing equations and its closure.

In nonequilibrium gas dynamics, physical models of various complexity can be developed on the basis of kinetic scaling corresponding to specific deviations from equilibrium [Nagnibeda and Kustova, 2009]. Among different approaches, the most detailed is the state-to-state (STS) model based on the assumption of fast translational and rotational relaxation and slow vibrational-chemical kinetics fully coupled to fluid dynamics [Giordano et al., 1997; Armenise et al., 1999; Armenise and Kustova, 2014; Bonelli et al., 2021]. The set of STS governing equations includes conservation equations for mass, momentum, and total energy coupled with balance equations for the vibrational level populations and the mass fractions of chemical species. The number of equations, depending on the mixture composition, ranges from several tens to several thousands. For inviscid gas flow simulations, some coarse-graining techniques aimed at reducing computational costs of the STS model can be applied [Munafò et al., 2014; Sahai et al., 2017]. Although these methods show high efficiency in modelling inviscid flows, their application to viscous flow simulations is limited since the transport kinetic theory has not yet been developed for such approaches.

For the closure of extended fluid-dynamic equations in the case of viscous flows, constitutive relations are required for the stress tensor, diffusion velocities, and heat flux. In the STS approach, these constitutive relations involve state-specific transport coefficients (thermal conductivity, shear viscosity, bulk viscosity, diffusion, and thermal diffusion) that have to be evaluated using accurate kinetic theory transport algorithms based on the generalized Chapman–Enskog method [Nagnibeda and Kustova, 2009]. To implement these algorithms for gas mixtures with vibrational and electronic excitation, chemical reactions, and ionization, software packages $KAPPA$ [Campoli et al., 2019] and $PAINeT$ [Istomin and Oblapenko, 2018; Istomin, 2018; Istomin, 2019] were designed earlier by the authors. These packages can be coupled with computational fluid dynamic (CFD) solvers to provide the transport properties of nonequilibrium reacting gas mixtures, but their computational cost remains prohibitively high. The reason is that the rigorous transport algorithms require solving high-order systems of linear algebraic equations in each cell of the computational grid. Therefore, development of new techniques for faster calculation of state-specific transport coefficients in the STS approach is of vital importance for the nonequilibrium viscous reacting gas flow simulations.

Recent studies have shown that machine learning methods, including neural networks, have potential for simulating nonequilibrium gas-dynamic problems [Stokes et al., 2020; Schmidt et al., 2019; Brunton et al., 2020]. The use of machine learning methods for modeling of physical systems has grown sharply, see [Fradkov, 2022; Plotnikov et al., 2019; Fradkov and Shepeljavyi, 2022]. Machine learning methods help to accurately predict physical quantities by processing large amounts of available data, which significantly reduces computational effort and allows for implementing detailed models of physical-chemical kinetics and transport processes [Istomin and Kustova, 2021; Campoli et al., 2022; Bushmakova and Kustova, 2022]. In our preliminary studies, we used neural networks to speed up the evaluation of specific heats, thermal conductivity, and shear viscosity for single-component gases and simple mixtures, which resulted in a speed-up ratio of up to 10^3 [Istomin and Kustova, 2021]. Similarly, using a neural network approach for modeling carbon dioxide vibrational kinetics significantly improved the efficiency of the relaxation rate computations [Gorikhovskii and Kustova, 2022].

Although machine learning techniques are promising for nonequilibrium fluid dynamics, their implementation is limited because most practical work occurs through the high-level libraries, such as *TensorFlow* [Abadi et al., 2015] and *Scikit-learn* [Pedregosa et al., 2011]. The use of these libraries considerably slows down the solution of the problem, since it is necessary to study the libraries' application programming interface (API) and the intricacies of the programming language. To overcome these limitations, in the present study we develop an open-source software package [Istomin et al., 2022] with a user interface for processing raw data, building, training, and analyzing the model, and presenting the results in an interactive format. This package allows users to focus on solving the problem itself and quickly change the necessary model settings during the analysis.

The objectives of this study are: 1) to calculate state-specific transport coefficients (thermal conductivity, shear and bulk viscosity) for a binary gas mixture using both accurate kinetic theory methods and the developed software package with machine learning (ML) techniques; 2) to assess the accuracy and efficiency of the developed ML approach; 3) to recommend a suitable ML approach and possible neural network architecture for the evaluation of the transport coefficients.

2 State-to-state fluid-dynamic equations

The energy ε of each molecule in the gas includes contributions from the translational energy associated with its motion as a whole, rotational energy connected to the rotation of atoms in a molecule about its center of mass, vibrational energy related to the oscillations of atoms about their equilibrium position, and electronic energy specified by the location of electrons in the electronic shells. Atomic species only have electronic internal energies. While translational motion is usually described classically, internal modes are treated in the framework of quantum mechanics, allowing for discrete energy states in each internal mode. The number of internal states is determined by the energy of dissociation and ionization.

When the gas undergoes sudden heating or cooling, it goes out of equilibrium. A new equilibrium state is achieved through the relaxation processes involving energy transitions between various internal degrees of freedom and chemical reactions. As mentioned previously, the characteristic times of different collisional processes can vary considerably depending on the flow.

In the present study, we consider nonequilibrium flows of reacting gas mixtures under the following relation between the characteristic times of elementary processes:

$$\tau_{tr} \leq \tau_{rot} \ll \tau_{vibr} \sim \tau_{react} \sim \theta, \qquad (1)$$

here τ_{tr} and τ_{rot} are the characteristic times for translational and rotational relaxation, respectively; τ_{vibr} is the characteristic time of vibrational relaxation including both vibrational-vibrational and vibrational-translational energy transitions, τ_{react} is the characteristic time of chemical reactions, and θ is the mean time of the variation of gas-dynamic parameters. Under such an assumption, relaxation of vibrational energy and chemical reactions are fully coupled with fluid dynamics [Nagnibeda and Kustova, 2009].

To derive the set of extended fluid-dynamic equations, we start with the dimensionless form of the Boltzmann equation for the distribution function. Based on kinetic scaling (1), the collision operator is split to the operators of rapid processes proceeding in the microscopic scale (τ_{tr}, τ_{rot}) and slow processes (τ_{vibr}, τ_{react}) occurring at the fluid-dynamic scale; the small parameter $\epsilon = \tau_{tr}/\theta$ is introduced according to (1). In the framework of the generalized Chapman–Enskog method we expand the distribution function into the series in ϵ and derive the set of governing equations for the macroscopic flow parameters [Nagnibeda and Kustova, 2009]. The closure of governing equations depends on the number of terms kept in the expansion of the distribution function.

In the case of the absence of external forces and magnetic fields the governing equations corresponding to kinetic scaling (1) are obtained in the following form:

$$\begin{aligned} \frac{dn_{c_i}}{dt} + n_{c_i}\nabla\cdot\mathbf{v} + \nabla\cdot(n_{c_i}\mathbf{V}_{c_i}) = \\ R_{c_i}^{vibr} + R_{c_i}^{react}, \qquad (2) \\ c = 1,..,L, \;\; i = 0..L_c, \end{aligned}$$

$$\rho\frac{d\mathbf{v}}{dt} + \nabla\cdot\boldsymbol{P} = 0, \qquad (3)$$

$$\rho\frac{dU}{dt} + \nabla\cdot\mathbf{q} + \boldsymbol{P} : \nabla\mathbf{v} = 0, \qquad (4)$$

here n_{c_i} is the number density of chemical species c on the vibrational level i (in case of molecular species) with the corresponding energy ε_{c_i}; $\varepsilon_{c_i} = \varepsilon_{vibr,c}(i) + \varepsilon_{f,c}$, where $\varepsilon_{vibr,c}(i)$ and $\varepsilon_{f,c}$ are vibrational and formation energy per particle; the number of species in a mixture is L, and the number of vibrational levels for the species c is L_c; $\mathbf{v}$ is the gas velocity, U is the total specific energy, $\rho U = \frac{3}{2}nkT + \rho U_{rot} + \sum_{c,i}\varepsilon_i n_{c_i}$, $n = \sum_{c,i} n_{c_i}$, k is the Boltzmann constant, T is the temperature, U_{rot} is the specific rotational energy, $\mathbf{V}_{c_i}$ is the diffusion velocity of c-species for each vibrational state i, ρ is the mixture density, $\boldsymbol{P}$ is the pressure tensor, $\mathbf{q}$ is the heat flux; $R_{c_i}^{react}$, $R_{c_i}^{vibr}$ are production terms due to chemical reactions and vibrational energy transitions. Note that in the case of atomic species, the right hand side of (2) includes only $R_{c_i}^{react}$, while for molecular species all production terms have to be taken into account.

The number of master equations (2) depends on the mixture composition. Thus, in polyatomic gases like carbon dioxide, there are several thousands of coupled vibrational states below the dissociation threshold [Kunova et al., 2020], which yields the corresponding number of equations. In the present study, we consider a simple case of a binary mixture (N_2, N) including two chemical species ($L = 2$) and 48 vibrational levels of the nitrogen molecule ($L_{N_2} = 48$).

In the first-order approximation of the generalized Chapman–Enskog method, the expressions for the pressure tensor, diffusion velocity, and heat flux in a viscous flow are derived in the form

$$\boldsymbol{P} = (p - p_{rel} - \zeta\nabla\cdot\mathbf{v})\boldsymbol{I} - 2\eta\boldsymbol{S}, \qquad (5)$$

$$\mathbf{V}_{c_i} = -\sum_{d,j} D_{c_i d_j}\,\mathbf{d}_{c_i} - D_{Tc_i}\nabla\ln T, \qquad (6)$$

$$\begin{aligned} \mathbf{q} = -\lambda'\nabla T - p\sum_{c,i} D_{Tc_i}\,\mathbf{d}_{c_i} + \\ \sum_{c,i}\left(\frac{5}{2}kT + <\varepsilon_c>_{rot} + \varepsilon_{c_i}\right) n_{c_i}\mathbf{V}_{c_i}. \qquad (7) \end{aligned}$$

Here $\boldsymbol{S}$ and $\boldsymbol{I}$ are the traceless deformation rate tensor and the unit tensor, $\mathbf{d}_{c_i}$ is the diffusive driving force for each vibrational state i, $<\varepsilon_c>_{rot}$ is the averaged rotational energy of molecular species c; p is the pressure, η is the shear viscosity coefficient, ζ is the bulk viscosity coefficient, p_{rel} is the relaxation pressure, $D_{c_i d_j}$ and D_{Tc_i} are the multi-component diffusion and thermal diffusion coefficients for different chemical species and vibrational states, λ' is the partial thermal conductivity coefficient. It is worth noting that in the state-specific approach, the partial thermal conductivity coefficient λ' is specified only by the translational and rotational (for molecules) degrees of freedom, whereas the transport of vibrational energy is governed by diffusion processes and can be strongly affected by nonequilibrium kinetics of vibrational states. Bulk viscosity, which occurs due to the finite rate of internal energy relaxation in the rapid processes, is specified in the STS approach by the rotational energy transitions; this phenomenon is important in compressible flows with high velocity divergence, such as shock waves. Relaxation pressure describes the effect of slow processes (vibrational relaxation and chemical reactions) on the normal stress.

For numerical simulations of viscous gas flows in the STS approach, it is necessary to solve equations (2)–(4); this includes calculation of all state-specific transport coefficients as well as relaxation terms R_{ci}^{vibr}, R_{ci}^{react} in each cell of the computational grid. The number of cells reaches hundreds of millions and even more, depending on the flow geometry. Direct implementation of machine learning techniques for solving partial differential equations (2)–(4) is hardly possible; nevertheless, one may attempt to speed up evaluation of transport and relaxation terms.

3 Transport coefficients

3.1 Exact algorithm

Accurate kinetic theory algorithm for the transport coefficients evaluation consists of the following steps [Nagnibeda and Kustova, 2009]. First, the first-order distribution functions is derived in terms of the gradients of macroscopic flow parameters (velocity, temperature, and species number densities); coefficients at the gradients are unknown functions of molecular velocity; for these unknown functions the integral equations are derived. Next, the transport coefficients are expressed in terms of the bracket integrals with respect to the above-mentioned unknown functions. In the kinetic theory, a bilinear form $[A, B]$ (where A and B are the arbitrary functions of molecular velocities) is referred to as a bracket integral [Nagnibeda and Kustova, 2009]:

$$[A, B] = \sum_{cij} \int B_{cij} I_{cij}(A) d\mathbf{u}_c. \qquad (8)$$

The bracket integrals are introduced on the basis of the linearized integral operator I_{cij} of rapid processes [Nagnibeda and Kustova, 2009], $\mathbf{u}_c$ is the velocity of species c. In the state-to-state approach, the integral operator I_{cij} is determined by the cross sections of elastic collisions and collisions leading to the change of rotational state j.

In the next step, the unknown functions in the expression for the first-order distribution function are expanded into the series of orthogonal polynomials, which allows one to reduce the initial integral equations to the systems of linear algebraic equations and express the transport coefficients in terms of the expansion coefficients. Coefficients in these systems depend on the collision integrals obtained by simplifying the bracket integrals. The order of linear systems depends on the number of terms retained in the expansions; for the state-to-state model, the order ranges from several tens to several thousands. Finally, the linear transport systems are solved numerically.

To assess our approach we choose evaluation of state-to-state thermal conductivity, shear viscosity, and bulk viscosity coefficients in a (N_2, N) mixture. In this simple case, the mixture comprises 48 vibrational levels of molecules and atomic nitrogen, resulting in 49 species in total. Calculation of thermal conductivity, shear viscosity, and bulk viscosity coefficients (λ, η, ζ) requires solving 147, 49, and 98 linear algebraic equations, respectively. Note that the STS transport coefficients are functions of temperature, pressure and number densities of all species, $\lambda = \lambda(T, p, n_1, ..., n_{49})$, $\eta = \eta(T, p, n_1, ..., n_{49})$, $\zeta = \zeta(T, p, n_1, ..., n_{49})$. This rigorous but computationally expensive algorithm has been implemented in the $KAPPA$ library [Campoli et al., 2019] and $PAINeT$ software package [Istomin and Kustova, 2021].

3.2 ML algorithm

To assess the machine learning algorithms, at the first step the transport coefficients of the 49-component mixture are calculated using the $KAPPA$ software package. The input vector for the model includes temperature, pressure, vibrational level populations and atomic molar fractions, resulting in a total of 51 features for each target variable. The dataset contains 13, 200 samples of these features for each coefficient, where values below $1e - 7$ are set to zero. The reliability of data is confirmed by the validation of the kinetic theory algorithms against experimental data carried out in our previous studies [Campoli et al., 2019]. Subsequently, all values are scaled using the *StandardScaler* from the *Scikit-learn* library. Furthermore, the dataset is split into training and test sets, with proportions of 0.9 and 0.1, respectively.

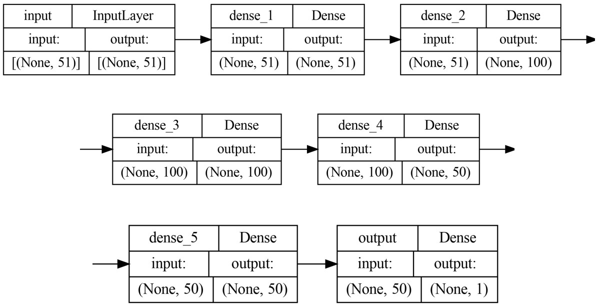

Figure 1. Model architecture graph.

For the construction of multi-layer perceptron regression model, the software application developed for the present study is used (description is given in the next section). The graph of the regression model applied to calculate all coefficients is shown in Fig. 1. The Input layer contains 51 values of the considered features, followed by 5 Dense layers containing 51, 100, 100, 50, and 50 nodes, respectively. Each of these layers uses the Rectified Linear Unit activation function (*ReLU*):

$$f(x) = max(0, x).$$

The last Dense layer contains one node that represents the predicted value by the regression process and has a Linear activation function. Overall, the model contains 25, 603 trainable parameters.

The model is trained using the Adam optimizer with a learning rate of $5e - 5$ for 1000 epochs and a batch size of 512. The validation split, representing the subset of data not used during training, is set to 0.15 (15% of the training set). Despite the model's relative simplicity, satisfactory results are achieved. As shown in Fig. 2, the Mean Squared Error (MSE) loss function for each coefficient on the training and validation sets quickly reaches error values of the order of $1e - 5 \sim 1e - 6$ due to the

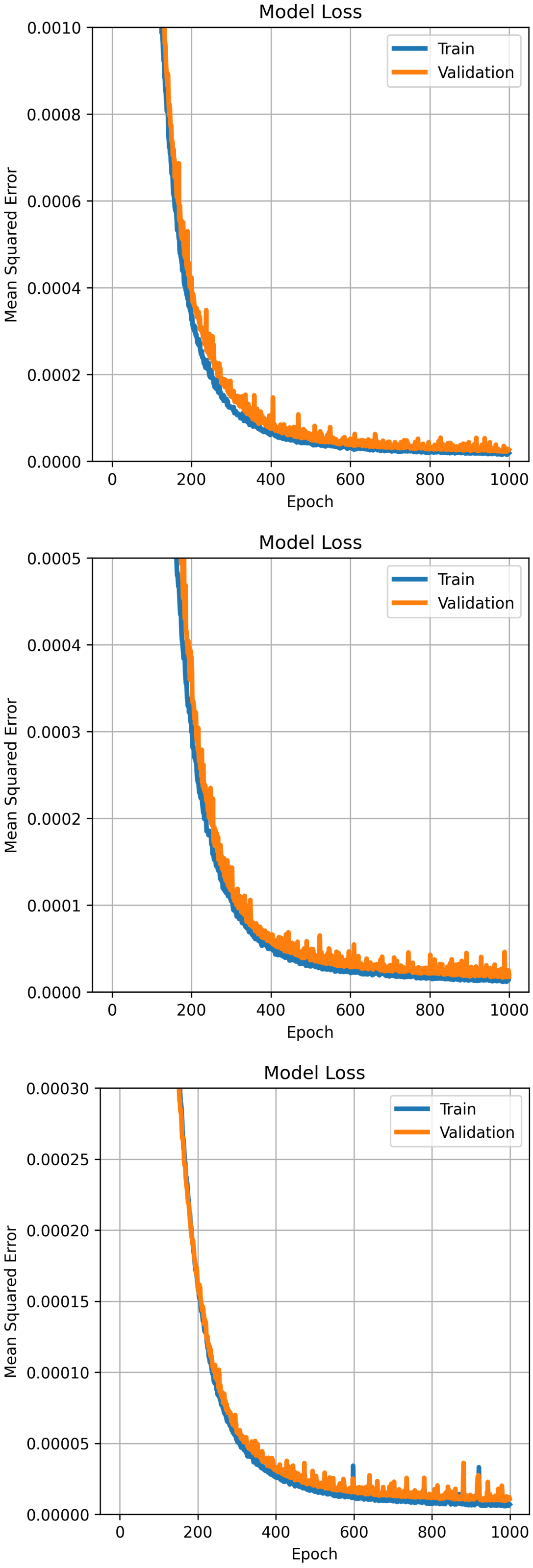

Figure 2. Model's Mean Square Error. From top to bottom: thermal conductivity, shear viscosity, bulk viscosity.

initial data scaling. Mean Absolute Error (MAE) metric, tracked during the model fitting for each coefficient on the training and validation sets, also reaches small values after around 200 epochs of training.

3.3 Discussion

The model is evaluated on the scaled test set after training. Table 1 shows the values for the loss function (MSE) and the tracked metric (MAE), which are considered as satisfactory. It is evident that for the current task, there is no difference in the convergence of both metrics. This is obvious because all the predicted values are positive and scaled. However, for other problems, such as predicting diffusion coefficients where values can be positive or negative in the output vector, the MSE metric is preferable. Additionally, the MSE metric's convex and smooth nature makes it advantageous when using other optimizers, such as stochastic gradient descent. Further experimentation with different configurations of model architecture and training hyperparameters could potentially lead to further improvements in the results.

The model predictions on a test set are also retrieved. An inverse transformation is applied to the predicted values to obtain the original scale. Additionally, the values below $1e-7$ are set to 0. Fig. 3 presents the obtained results. All the graphs are plotted for different mixture compositions, depending on the atomic nitrogen molar fraction $x_N = n_N/n$ in the mixture (ranging from 0.1 to 0.9). The Boltzmann distribution over vibrational states is chosen for the model assessment, although arbitrary vibrational distributions can be used in general. For the Boltzmann distribution, all coefficients increase monotonically with the temperature. Under the considered conditions for the mixture (N_2, N) the thermal conductivity coefficient increases with rising atomic molar fraction, while both viscosity coefficients decrease. It is worth mentioning that the bulk viscosity coefficient is comparable to the shear viscosity coefficient, and therefore, its contribution to the normal stress may be important in nonequilibrium compressible flows.

Contrary to the conventional statistical methods, the use of machine learning algorithms can handle a larger number of variables such as the manifold of molar fractions of different internal states. This enables us to effectively handle complex interactions within the dataset, resulting in more accurate predictions, and consolidates the choice of machine learning (ML) over conventional statistical methods for our problem. However, this also poses a need for a greater number of input-output pairs to learn the model, which can be considered a limitation of the ML algorithm. Considering the accuracy of the regression model for transport coefficients, it is evident that the model fits very well to the original transport coefficients obtained by exact kinetic theory methods for every mixture composition. The final accuracy of the trained model, defined in terms of relative error (as shown in Table 1), is found to be around 99.5% on the predicted values after transformation, indicating a high level of agreement between the predicted and actual values.

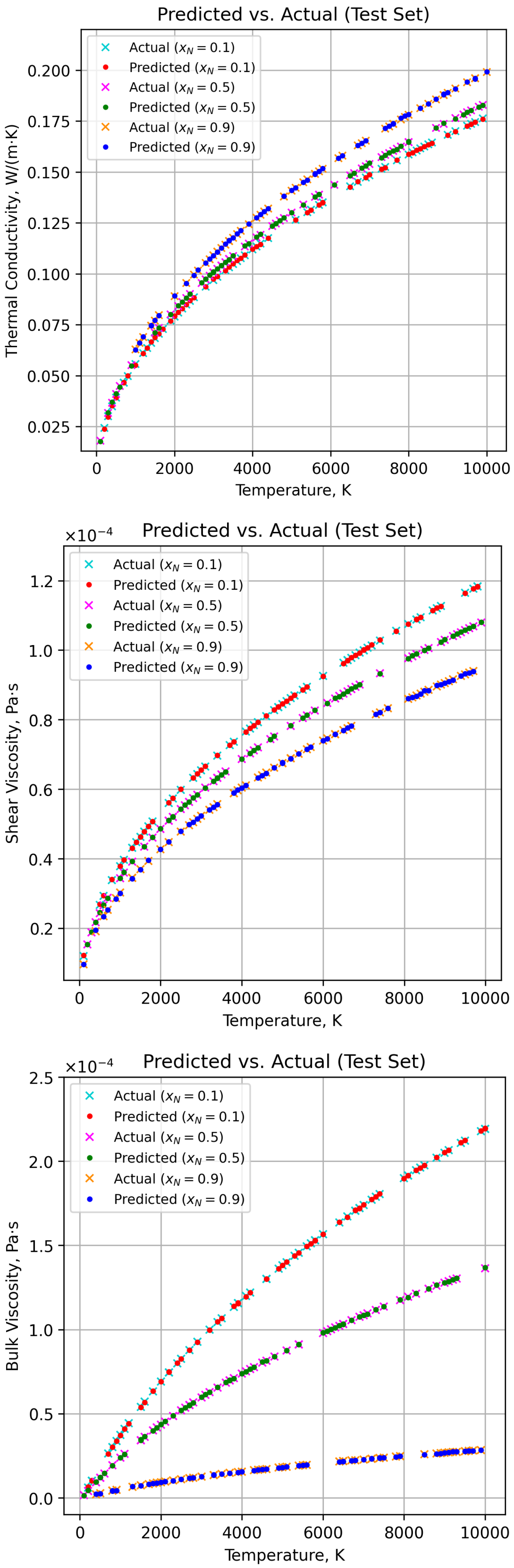

Figure 3. Comparison between the predicted and actual values on a test set. From top to bottom: thermal conductivity, shear viscosity, bulk viscosity as functions of T.

Table 1. Model evaluation results.

Metrics \ TC	Thermal conductivity	Shear viscosity	Bulk viscosity
Loss (MSE)	$2.4255e-05$	$1.5613e-05$	$9.6022e-06$
MAE	0.0038	0.003	0.0024
Average relative error	0.182%	0.159%	0.506%

The most important achievement is a significant speed-up in the calculation time. The computational efficiency of the accurate transport coefficients calculation using $PAINeT$ (where calculations are already parallelized on the processors and therefore are more computationally efficient than in $KAPPA$) can be compared with the efficiency of the machine learning methods proposed in this paper. The comparison is performed on a personal computer (Intel Core i5-6500 CPU 3.20 GHz). For a given output vector (λ, η, ζ), the neural network constructed with the new software reduced the computation time by up to 100 times, from 5.5 seconds using $PAINeT$ to 0.057 seconds. It is worth mentioning that this comparison includes the data preparation time (about an hour) and model training time (nearly 2 minutes). Despite this, the decrease in the computational time for one calculation step is significant. This is especially useful for 2D and 3D fluid-dynamic problems, where the number of such calculation steps may be very high depending on the mesh.

Once the model is configured and trained using the developed software, it can be stored in the *.h5* format and loaded directly as a surrogate model-based engineering model [Jiang et al., 2020] into external CFD solvers. The obtained results show promising opportunities for implementing the developed software in CFD solvers for simulations of viscous nonequilibrium reacting flows. Moreover, based on the given approach, the developed software can be implemented in other research fields.

4 Software Application Description

The software application developed for this study is written in *Python 3.10* [Python Software Foundation, 2021] using the *Jupyter Notebook* [Kluyver et al., 2016] interactive environment as well as the *TensorFlow 2.10 Functional API*. It is worth mentioning that the notebook can be opened and used in *Binder* [Project Jupyter et al., 2018] environment. The notebook is organized into separate cells for distinct modules, including data preparation, model configuration, model training, model evaluation, and model predictions. Gradual compilation of each sell leads for project step-by-step build-up procedure.

As the first step, the training data must be prepared: To ensure compatibility with *TensorFlow*, the data must first be converted into a *NumPy* format [Harris et al., 2020]. The data file containing all the feature headers can be placed inside a separate repository folder and then loaded from the notebook. It is important that data should be loaded in either *.csv* or *.tsv* format initially. Next, columns can be selected as model inputs and outputs, and the data can be split into train and test portions. Finally, the data will be automatically converted into a *.npy* format for further training of the model.

The next step is to build the model. For this purpose, the *TensorFlow 2.10 API* is used, which allows one to build quite flexible models with a different number of layers. In the application it is possible to either upload an existing model and reconfigure it or create a new one. It is possible to use different activation functions in each layer: Linear, Hyperbolic Tangent (*tanh*) or Rectified Linear Unit activation function (*ReLU*). After construction of the model, the summary and graph representation can be displayed in separate tiles. Once the configuration is complete, the model can be saved in the *.h5* format for future use.

Once all the settings have been specified, the model can be compiled. This involves selecting an optimizer along with appropriate hyperparameters and a suitable loss function. After the model compilation, the next step is to train the model. This involves choosing hyperparameters such as the batch size, number of epochs, etc. Once the training process begins, the corresponding cell output will interactively display the progress of the model.

Finally, the user can view the history plots, which display the loss functions and tracked metrics during the training and validation stages. The trained model can also be saved for future use. The last two cells of the notebook are responsible for evaluating the model on a test set and making predictions.

The software documentation, source code, and the link to the Binder environment can be found in the original repository, which is accessible at [Istomin et al., 2022].

5 Conclusion

Machine learning methods for modelling state-to-state transport coefficients are considered. A regression model capable of calculating state-specific thermal conductivity, shear viscosity, and bulk viscosity transport coefficients is built using the software application with a user-friendly interface that utilizes modern machine learning libraries; the software is designed in the frame of this study. The model is fitted and evaluated on training and test sets, and its predictions are compared with actual values of transport coefficients obtained in the framework of accurate kinetic-theory methods. The analysis of the considered metrics and visual illustrations confirm that the developed neural network architecture is suitable for calculating the STS transport coefficients.

The architecture comprises a few Dense layers with ReLU activation functions. Despite its relative simplicity, the model achieves a good convergence to the original values, demonstrating great predictive ability. Depending on the application, different data preprocessing techniques as well as deeper architectures may be considered to obtain even better results. Additionally, the model simplicity provides a fast inference time, significantly accelerating the calculation of resource-intensive quantities.

In the present study, a speed-up of about two orders of magnitude is achieved for a single computation of transport coefficients; for real fluid-dynamic simulations, the gain in computational efficiency will be considerably higher since the complete set of transport coefficients is calculated in each cell of the grid.

This approach presents a novel way to convert accurate research computational models into computationally fast and precise machine learning models, which can be conveniently built using the provided software package.

Acknowledgements

The research leading to these results has received funding from Saint Petersburg State University, projects 93022273.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Armenise, I., Capitelli, M., Kustova, E., and Nagnibeda, E. (1999). The influence of nonequilibrium kinetics on the heat transfer and diffusion near re-entering body. *J. Thermophys. Heat Transfer*, **13** (2), pp. 210–218.

Armenise, I. and Kustova, E. (2014). On different contributions to the heat flux and diffusion in nonequilibrium flows. *Chem. Phys.*, **428**, pp. 90–104.

Bonelli, F., Pascazio, G., and Colonna, G. (2021). Effect of finite-rate catalysis on wall heat flux prediction in hypersonic flow. *Phys. Rev. Fluids*, **6**, pp. 033201.

Brunton, S. L., Noack, B. R., and Koumoutsakos, P. (2020). Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, **52** (1), pp. 477–508.

Bushmakova, M. and Kustova, E. (2022). Modeling the vibrational relaxation rate using machine-learning methods. *Vestnik St.Petersb. Univ.Math.*, **55**, pp. 87–95.

Campoli, L., Kustova, E., and Maltseva, P. (2022). Assessment of machine learning methods for state-to-state approach in nonequilibrium flow simulations. *Mathematics*, **10** (6), pp. 1–21.

Campoli, L., Oblapenko, G., and Kustova, E. (2019). Kappa: Kinetic approach to physical processes in atmospheres library in c++. *Computer Physics Communications*, **236**, pp. 244–267.

Fradkov, A. L. (2022). Cybernetics and physics: Ten years online together. *Cybernetics and Physics*, **11**, pp. 4–5.

Fradkov, A. L. and Shepeljavyi, A. I. (2022). The history of cybernetics and artificial intelligence: a view from Saint Petersburg. *Cybernetics and Physics*, **11**, pp. 253–263.

Giordano, D., Bellucci, V., Colonna, G., Capitelli, M., Armenise, I., and Bruno, C. (1997). Vibrationally relaxing flow of N_2 past an infinite cylinder. *J. Thermophys. Heat Transfer*, **11** (1), pp. 27–35.

Gorikhovskii, V. I. and Kustova, E. V. (2022). Neural-network-based approach to the description of vibrational kinetics of carbon dioxide. *Vestnik St. Petersb. Univ. Math.*, **55**, pp. 434–442.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, **585** (7825), pp. 357–362.

Istomin, V. and Kustova, E. (2021). PAINeT: Implementation of neural networks for transport coefficients calculation. *Journal of Physics: Conference Series*, **1959** (1), pp. 1–8.

Istomin, V., Kustova, E., Lagutin, S., Pavlov, S., Shalamov, I., and Tan, L. (2022). Machine Learning UI. `https://github.com/quantum-entangled/machine-learning-ui`.

Istomin, V. and Oblapenko, G. (2018). Transport coefficients in high-temperature ionized air flows with electronic excitation. *Physics of Plasmas*, **25** (1), pp. 013514.

Istomin, V. A. (2018). Painet: An object-oriented software package for simulations of flow-field, transport coefficients and flux terms in non-equilibrium gas mixture flows. *AIP Conference Proceedings*, **1959** (1), pp. 060006.

Istomin, V. A. (2019). Painet: Similarity criteria and different approaches of kinetic theory. *AIP Conference Proceedings*, **2132**, pp. 130005.

Jiang, P., Zhou, Q., and Shao, X. (2020). *Surrogate Model-Based Engineering Design and Optimization*. Springer-Verlag, Berlin, Heidelberg.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C., and development team, J. (2016). Jupyter notebooks – a publishing format for reproducible computational workflows. In Loizides, F. and Scmidt, B., editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, IOS Press, pp. 87–90.

Kunova, O., Kosareva, A., Kustova, E., and Nagnibeda, E. (2020). Vibrational relaxation of carbon dioxide in state-to-state and multi-temperature approaches. *Physical Review Fluids*, **5** (12), pp. 123401.

Munafò, A., Panesi, M., and Magin, T. E. (2014). Boltzmann rovibrational collisional coarse-grained model for internal energy excitation and dissociation in hypersonic flows. *Phys. Rev. E*, **89** (2), pp. 023001.

Nagnibeda, E. and Kustova, E. (2009). *Nonequilibrium Reacting Gas Flows. Kinetic Theory of Transport and Relaxation Processes*. Springer-Verlag, Berlin, Heidelberg.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, **12**, pp. 2825–2830.

Plotnikov, S. A., Lipkovich, M., Semenov, D. M., and Fradkov, A. L. (2019). Artificial intelligence based neurofeedback. *Cybernetics and Physics*, **8**, pp. 287–291.

Project Jupyter, Matthias Bussonnier, Jessica Forde, Jeremy Freeman, Brian Granger, Tim Head, Chris Holdgraf, Kyle Kelley, Gladys Nalvarte, Andrew Osheroff, Pacer, M., Yuvi Panda, Fernando Perez, Benjamin Ragan Kelley, and Carol Willing (2018). Binder 2.0 - Reproducible, interactive, sharable environments for science at scale. In Fatih Akici, David Lippa, Dillon Niederhut, and Pacer, M., editors, *Proceedings of the 17th Python in Science Conference*, pp. 113 – 120.

Python Software Foundation (2021). Python 3.10. `https://www.python.org/downloads/release/python-3100/`.

Sahai, A., Lopez, B., Johnston, C., and Panesi, M. (2017). Adaptive coarse graining method for energy transfer and dissociation kinetics of polyatomic species. *J. Chem. Phys.*, **147**, pp. 054107.

Schmidt, J., Marques, M. R. G., Botti, S., and Marques, M. A. L. (2019). Recent advances and applications of machine learning in solid-state materials science. *npj Computational Materials*, **5** (1), pp. 1–36.

Stokes, P. W., Casey, M. J. E., Cocks, D. G., de Urquijo, J., García, G., Brunger, M. J., and White, R. D. (2020). Self-consistent electron–THF cross sections derived using data-driven swarm analysis with a neural network model. *Plasma Sources Science and Technology*, **29** (10), pp. 1–10.