Proceedings of the 16th Minisymposium Verfahrenstechnik and 7th Partikelforum (TU Wien, Sept. 21/22, 2020)

Title:
Performance optimisation approaches for NOx Postprocessor with detailed chemistry model using OpenFOAM

Corresponding author:
Senthilathiban Swaminathan (K1-MET GmbH / MU Leoben), senthilathiban.swaminathan@k1-met.com

Date of submission:
28.02.20

Date of revision:
01.09.20

Date of acceptance:
09.09.20

Chapter ID:
DiV2-(04)

Length:
4 pages

Download available from (online, open-access):
http://www.chemical-engineering.at/minisymposium

ISBN (full book):
978-3-903337-01-5

Performance optimisation approaches for NOx Postprocessor with detailed chemistry model using OpenFOAM

Senthilathiban Swaminathan[1,2], Christoph Spijker[2], Harald Raupenstrauch[2], Irmela Kofler[3]
1: K1-MET GmbH, Leoben, Austria. senthilathiban.swaminathan@k1-met.com
2: Chair of Thermal Processing Technology, Montanuniversität Leoben, Austria
3: K1-MET GmbH, Linz, Austria.

Keywords: NOx, Sandia Flame D, OpenFOAM, Combustion Chemistry

Abstract

High temperature combustion processes produce NOx emissions (mainly NO and NO_2), which cause various health and environmental effects. Emission of NOx results in the formation of acid rain, ozone depletion and reacts with chemicals in the air to form particulate matter thereby resulting in air pollution [1]. To predict the formation of NOx in turbulent combustion, several models such as fluid dynamics, heat transfer and chemical kinetics need to be modelled. In recent years various research work have been done to reduce the NOx emissions. CFD has been proved to be an important tool in predicting the NOx emissions. But using CFD with a detailed chemistry model for modelling combustion in industrial burners requires a lot of computational effort and hence, the present study concentrates on further reduction in the computational time of the existing NOx postprocessor [2]. In this study, two optimisation approaches were investigated to further increase the performance. The existing postprocessor uses a constant temperature field, so that an optimisation approach was implemented to modify the reaction rate constant calculation method in OpenFOAM. Another optimisation approach was investigated by predicting the initial values for the postprocessor with Zeldovich mechanism [3]. Both optimisation approaches were analysed with the benchmark test case of Sandia Flame D [4]. The approaches are currently being investigated and the achieved results indicate that, even though a minimum increase in performance was achieved, further optimisation needs to be done for increasing the performance of the postprocessor.

Introduction

CFD is a valuable tool to determine NOx emissions more precisely which requires immense computational effort. The decrease in computational effort causes less accurate prediction of emissions. For determining realistic NOx emissions from combustion processes in a rapid approach, a postprocessor was developed by W. R. Pollhammer [2]. As a preliminary step, a flamelet model was used to describe the flame and then the data from this flamelet model was transformed to the postprocessor to estimate the NO mass fraction.

The existing NOx postprocessor revolves around the idea that a detailed chemistry approach could be used only at the final stage of simulation and a computationally cheap approach could be used to model the flame. The discrete ordinate radiation model was implemented along with the flamelet model in OpenFOAM [5] [6] and was used to describe the flame temperature, pressure, flow field, turbulence and radiation.

Subsequently the data from the flamelet model was transferred to the postprocessor in which only the species transport equations are solved, with chemical reaction rate as the source term, to estimate the NOx using PaSR (Partially Stirred Reactor) combustion model. The chemical kinetics were solved using the GRI 3.0 mechanism. The chemical reaction rate calculation was further optimized, so that the chemistry is solved only when the species concentration in a cell changes more than 5% and for other cells, where the change is less than 5%, values from the previous iterations are used. To avoid nonphysical values, i.e. if a concentration of certain species falls below zero, then the entire chemistry is computed again. As a final step, to avoid divergence, the entire chemistry is computed every 200 Iterations.

This paper presents further optimisation strategies to reduce the calculation time of the existing postprocessor and evaluates the performance and accuracy of the proposed approaches. The first approach modifies the reaction rate constant calculation method in OpenFOAM, and the second approach starts the postprocessor by generating initial values using the Zeldovich mechanism.

Performance optimisation approach

The current implementation of the reaction rate constants calculation in OpenFOAM is based on the concept that the temperature field is changing every iteration in the chemistry calculation.

Based on the implemented chemistry model in OpenFOAM, the reaction progress rate Ω_i , as in equation (1), is calculated for every iteration, for every cell, for every chemical time step and for every reaction.

$$\Omega_i = k_f \prod_{K=1}^{N} [M_k]^{\nu'_k} \; - \; k_r \prod_{K=1}^{N} [M_k]^{\nu''_k} \tag{1}$$

where, $[M_k]$ is the concentration of the species k and ν_k is the stoichiometric coefficient of the species k. The forward rate constant k_f is calculated based on the Arrhenius equation as in Eq. (2), which is mainly dependent on the temperature field. Whereas the reverse rate constant k_r is calculated from the forward rate constant k_f and the equilibrium constant k_p.

$$k_f = A\, T^{\beta} \exp\left(\frac{-T_a}{T}\right) \qquad k_r = \frac{k_f}{k_p}$$

$$k_p = exp\left(\sum_k \nu_k \left(\frac{S_k}{R} - \frac{H_k}{RT}\right)\right)\left(\frac{p_{atm}}{RT}\right)^{\sum_k \nu_k} \tag{2}$$

where, A is the pre-exponential constant, T is the temperature and β is the temperature exponent and T_a is the ratio of the activation energy E_a and the gas constant R. H_k and S_k are the molar enthalpy and entropy of the species k. The above shows an example calculation of reaction progress rate in reversible Arrhenius reaction. A similar procedure is followed for irreversible Arrhenius reaction, third-body reaction, and other reactions as in the standard OpenFOAM code.

For a single calculation of k_f, approximately 30 floating point operations per second, would be used. This calculation is repeated N number of times (number of iterations times number of reactions times number of cells times number of chemical time steps) which is unnecessary in case of the existing postprocessor, where the temperature field remains constant. Hence calculating the rate constants only once would tremendously reduce the simulation time.

Accordingly, the chemistry model in OpenFOAM was adapted to reduce the calculation time as the first optimisation approach. The rate constants are different for every reaction and every cell but remain constant throughout the simulation. Therefore, volScalarFields are created for k_f and k_r for every reaction and the rate constants are calculated and stored in the first iteration, thereby instead of calculating the rate constant N number of times, they are calculated only once for each cell and each reaction. In further iterations, the rate constants are looked up from the stored data.

To analyse the computational effort of this approach initially the forward reaction rate calculation was modified (hereafter called 'code 1'), and the efficiency of the code was compared with the existing postprocessor and then both forward and reverse reaction rate constant calculations were modified (hereafter called 'code 2') and compared.

As a second optimisation approach, initializing the NO concentration with more proximate and realistic values to the experimental data was analysed. According to W.R. Pollhammer [7], the flamelet model overpredicts the NOx mass fraction ([8] and [9]) and hence transferring the species concentration data from the flamelet model to the postprocessor results in increased simulation time. Instead, starting the postprocessor by initializing some of the species, which take part in the formation of NOx with zero concentration results in reduced simulation time.

To initialize the postprocessor with more proximate values, the NO mass fraction was predicted by computing the source term, the rate of NO formation, from Zeldovich mechanism and then solving the species transport equation. The Zeldovich mechanism is formulated by the elementary reactions with rate coefficients k_1, k_2 and k_3 as follows.

$$O + N_2 \xrightarrow{k_1} NO + N$$
$$N + O_2 \xrightarrow{k_2} NO + O$$
$$N + OH \xrightarrow{k_3} NO + H$$

The rate of NO formation [10] from the above reactions is

$$\frac{d[NO]}{dt} = 2k_1[O][N_2] \qquad (3)$$

The N_2 concentration is obtained from the flamelet model whereas the concentration of oxygen atoms is determined by the partial equilibrium approach as in Eq. (4) of hydrogen combustion mechanism [10], by using the concentration of O_2, H_2 and H_2O from the flamelet model and the appropriate rate coefficients.

$$[O] = \frac{k_{H+O_2}\, k_{OH+H_2}[O_2]\,[H_2]}{k_{OH+O}\, k_{H+H_2O}\,[H_2O]} \qquad (4)$$

Eq. (3) was further extended by using the equilibrium oxygen atom concentration as in Eq. (5) to predict the NO formation rate as in Eq. (6) [11]. The temperature, concentration of O_2 and N_2 were used from the flamelet model.

$$[O] = \frac{k_{p(O)}[O_2]^{0.5}}{(RT)^{0.5}} \qquad (5)$$

where $k_{p(o)}$ is the equilibrium rate constant of the oxygen atom.

$$\frac{d[NO]}{dt} = 6.10^{16}\,(T^{-0.5})\,exp\left(\frac{-69090}{T}\right)\,[O_2]^{0.5}\,[N_2] \quad (6)$$

The NO concentration was predicted by both methods, one by using Eqs. (3) and (4) (hereafter called 'code 3') and another by using Eq. (6) (hereafter called 'code 4'). After predicting the NO concentration by both methods, the data was transferred to the postprocessor and the simulation was carried out to evaluate the performance.

Performance Evaluation

To evaluate the performance of the optimisation approaches, numerical simulations were performed, and the results were compared with the existing postprocessor on AMD Ryzen 7 3800x, 8 core processor. As a benchmark test, numerical simulations were performed and compared with experimental data [4]. The optimised postprocessor was evaluated with a mesh size of 204,800 cells on a 2D-axisymmetric mesh and chemical kinetics were solved using the GRI 3.0 mechanism with 325 reactions and 53 species. After calculating the flame properties using the flamelet model, the data was transferred to the optimised postprocessor and simulated.

For code 1, the postprocessor completes 0.3 s of real time simulation in 32,459 s of computation time and the existing postprocessor takes 49,983 s as could be seen in Fig.1. Whereas in case of code 2, it takes 43,969 s of computation time. This certainly proves that the optimized approach reduces the calculation time. The discrepancies between both codes is because of the memory consumption during the simulation. Creating a volScalarField for every reaction result in additional 325 data fields for the used GRI 3.0 mechanism. When both k_f and k_r calculation methods were modified, it results in additional 650 data fields. Hence code 2 consumes more memory and thereby slows down the simulation.

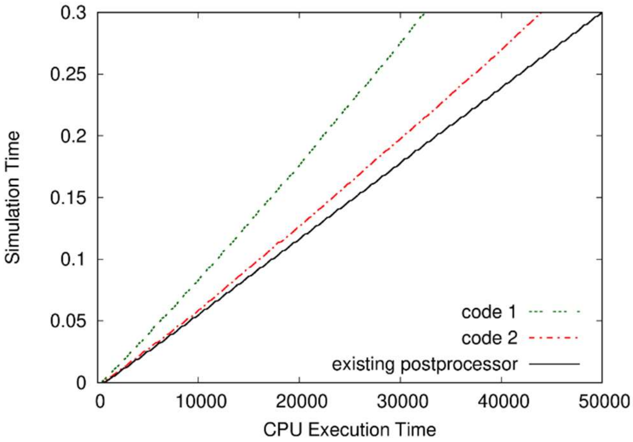

Fig. 1 Comparison of performance between the existing and optimization approaches (code 1 and 2)

By doing further studies with increased mesh size, which increases the size of the data fields, thereby consuming more memory and results in slower calculation time. Also, creating a volScalarField in OpenFOAM results in additional fields for storing the values of previous iteration, which is unnecessary for the proposed optimised approach. The reaction constants do not change throughout the entire simulation and they are only dependent on the temperature field on each cell, which is also constant throughout the simulation. Hence reducing the memory consumption and optimizing the code further will reduce the simulation time. To verify that the code did not influence the results, a comparison with experimental data was done and the code produces exact results as the existing post processor, which could be seen in Fig.2. Since, only the calculation method was modified, there is no change in the results between the existing postprocessor and the optimisation approach.

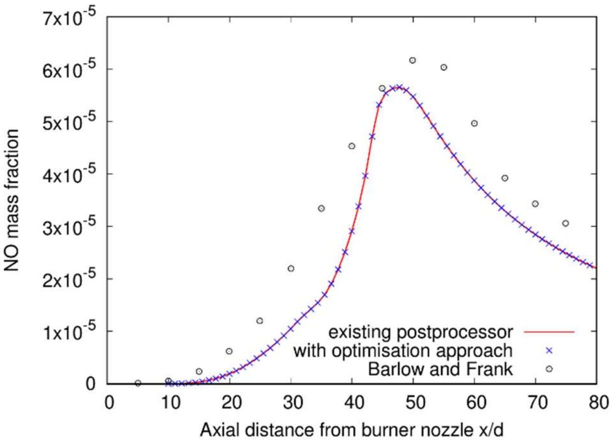

Fig. 2 Comparison of accuracy between the existing and optimisation approaches (code 1 and 2) along the centreline with experimental data

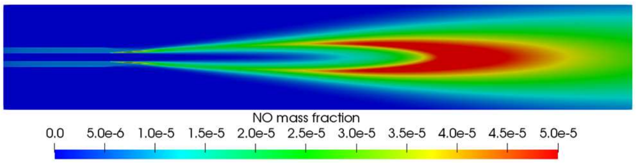

Fig. 3 NO mass fraction computed from existing post processor.

The computations show that the optimisation approach using Zeldovich mechanism modifies the results, since the NO mass fraction values are initialized. In the existing postprocessor, the NO concentration is initialized with zero, whereas in the optimization approach the NO concentration is initialized as shown in Fig. 4 and Fig. 5. When using the equilibrium concentration of the oxygen atoms with code 4, the NO mass fraction is overpredicted throughout the flame and when using the partial equilibrium approach with code 3, even though NO mass fraction is slightly higher, the predictions are closer to the experimental values than the code 4. The equilibrium approach always underestimates the concentration of oxygen atoms and the better approximation for estimating the concentration of oxygen atoms is the partial equilibrium approach [12]. Due to the data transfer from the flamelet model, which overpredicts the NO concentration, Zeldovich mechanism also overestimates the NO concentration. Initializing the postprocessor with NO mass fraction values from code 3 and code 4, reduces the accuracy of the results.

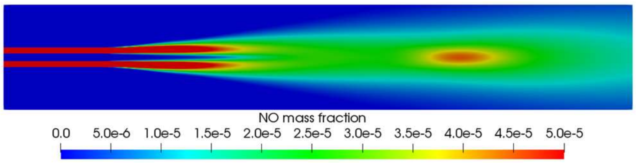

Fig. 4 Initialized NO mass fraction values using code 3

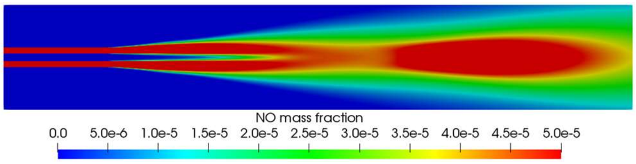

Fig. 5 Initialized NO mass fraction values using code 4

This optimisation approach using Zeldovich mechanism modifies the results and requires longer time for convergence. The convergence of the postprocessor was evaluated by calculating the integrated reaction rate of NO. The existing postprocessor requires around 0.004 s for convergence while the postprocessor started with NO mass fraction values obtained from code 3 and code 4, requires more time for convergence as could be seen in the following figure.

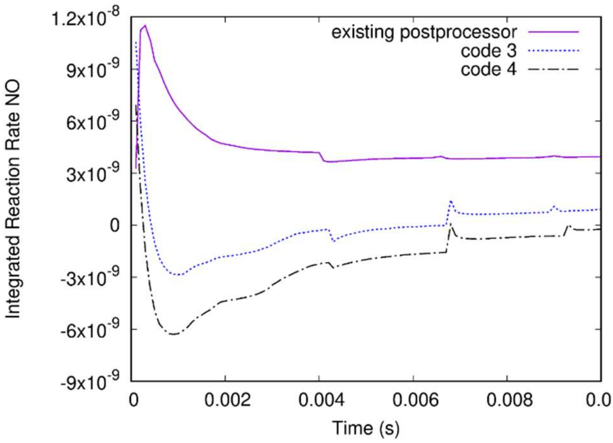

Fig. 6 Comparison of Integrated Reaction Rates between existing post processor and the optimised approaches.

Also, starting the postprocessor with initial values produces a hump shaped prediction of the NO mass fraction as shown in Fig.7 which is due to the initialization of the values in the same pattern as in Fig. 4 and 5. Even though both code 3 and 4 are initialized with different values, solving the steady state transport equation with the converged data produce almost similar results. To achieve the exact results as the existing postprocessor, the simulation needs to be continued for a longer time. Fig.7 shows the results obtained at 0.01 s from the optimized code 3 and 4. The simulation was stopped at this point, since the existing postprocessor was able to produce better results at 0.004 s. As the accuracy of the results is not as expected, predicting more precise initial values will optimize the performance of the postprocessor.

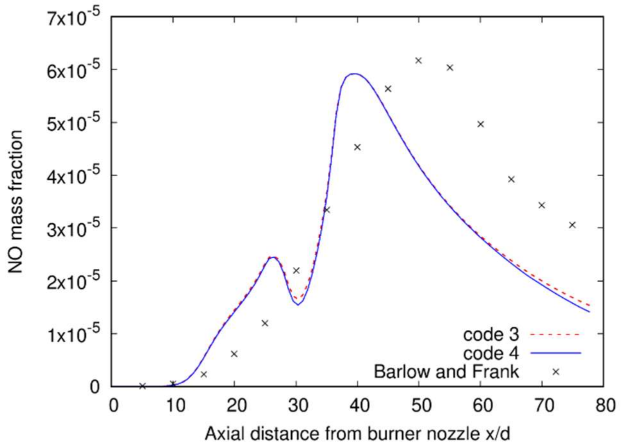

Fig. 7 Comparison of experimental data with the optimisation approaches along the centreline at 0.01 s – code 3 and code 4.

Conclusion

Performance optimisation approaches were implemented on the existing NOx postprocessor and were investigated with the benchmark case of Sandia Flame D. The reaction rate calculation method reduces the computation time as expected for some cases. Increasing the mesh size, number of reactions or modifying both forward and reverse rate constant calculation method results in increased memory consumption which slows down the simulation and hence the developed code needs to be further optimised. Predicting the initial values with Zeldovich mechanism requires more time for convergence and influences the results of the postprocessor. Therefore, more precise prediction of the initial values would increase the efficiency of the postprocessor.

Outlook

The investigations from the proposed optimisation approaches provide inputs for further optimisation. The next step is to develop a more efficient code for the reaction rate calculation method and as a third optimisation strategy, dynamic mesh refinement would be used. After the flamelet calculation, the mesh could be coarsened in far regions from the flame and only the necessary regions near the flame would be solved with a fine mesh. All these optimisation strategies will be further developed, and benchmark simulations will be performed to produce a more efficient and accurate NOx postprocessor.

Acknowledgements

The authors gratefully acknowledge the funding support of K1-MET GmbH, metallurgical competence center. The research program of the competence center K1-MET is supported by COMET (Competence Center for Excellent Technologies), the Austrian program for competence centers. COMET is funded by the Federal Ministry for Transport, Innovation and Technology, the Federal Ministry for Science, Research and Economy, the province of Upper Austria, Tyrol, and Styria and the Styrian Business Promotion Agency.

References

[1] C. E. Baukal Jr, Oxygen enhanced combustion. CRC press, 2013. doi: 10.1201/b13974

[2] W. R. Pollhammer, C. Spijker, H. Raupenstrauch, and M. Koller, “Numerical modelling of industrial burners for reduction of no x emissions using flamelet methods in combination with a newly developed postprocessor for fast and accurate emission prediction,” AIChE Annual Meeting, AIChE, 2017. ISBN: 978-0-8169-1102-8

[3] Y. B. Zeldovich, “The oxidation of nitrogen in combustion and explosions.,” ActaPhysiochem USSR 21:577, 1946. doi: 10.1515/9781400862979

[4] R. Barlow and J. Frank, Piloted ch4/air flames c, d, e, and f–release 2.1 15-jun-2007, 2007. Available: http://www.sandia.gov/TNF/DataArch/FlameD/SandiaPilotDoc21.pdf [Accessed Sep 1, 2020]

[5] A. Cuoci, A. Frassoldati, T. Faravelli, and E. Ranzi, “Formation of soot and nitrogen oxides in unsteady counterflow diffusion flames,” Combustion and Flame, vol. 156, no. 10, pp. 2010–2022, 2009. doi: 10.1016/j.combustflame.2009.06.023

[6] A. Cuoci, A. Frassoldati, T. Faravelli, and E. Ranzi, “Kinetic modeling of soot formation in turbulent non premixed flames,” Environmental Engineering Science, vol. 25, no. 10, pp. 1407–1422, 2008. doi: 10.1089/ees.2007.0193

[7] W. R. Pollhammer, “A cfd-dem model for nitrogen oxide prediction in shaft furnaces using openfoam,” PhD thesis, Montan Universität Leoben, Feb. 2019.

[8] L. Cutrone, P. de Palma, G. Pascazio, and M. Napolitano, “A rans flamelet-progress-variable method for computing reacting flows of real-gas mixtures.” Computers and Fluids, 39(3):485-498, 2010. doi: 10.1016/j.compfluid.2009.10.001

[9] M. Ihme and H. Pitsch. Modeling of radiation and nitric oxide formation in turbulent nonpremixed flames using a flamelet/progress variable formulation. Physics of Fluids, 20(5):055110, 2008. doi: 10.1063/1.2911047

[10] J. Warnatz, “Concentration, pressure, and temperature dependence of the flame velocity in hydrogen oxygen nitrogen mixtures,” Combustion Science and Technology, vol. 26, no. 5-6, pp. 203–213, 1981. doi: 10.1080/00102208108946961

[11] J. B. Heywood, “Internal combustion engine fundamentals,” McGraw-Hill, Inc., 1988. Available: https://www.accessengineeringlibrary.com/content/book/9781260116106 [Accessed: Sep 1, 2020]

[12] J. Warnatz, U. Maas and R. W. Dibble, Combustion. Springer, 1996, vol. 2 doi: 10.1007/978-3-540-45363-5