# Accelerating Batched Perfectly Stirred Reactor (PSR) Calculations Using General Purpose Graphics Processing Unit (GPGPU) Computing

[1] Sudip Adhikari, [2] Alan Sayre [3] Abhilash J. Chandy
[1]Department of Mechanical Engineering, University of Akron, Akron, Ohio 44325-3903, USA
[2]Babcok and Wilcox, Barberton, Ohio, USA

*Abstract:* Detailed analysis of efficiency and pollutant emission characteristics of practical turbulent combus- tion devices using complex combustion kinetics often depend on the interactions between the chemistry of both gaseous species and soot, and turbulent flow char- acteristics. Modeling of such combustion system often requires the use of chemical kinetic mechanisms with hundreds of species and thousands of reactions. Per- fectly stirred reactors (PSR) are idealized reactor en- vironments, where the reacting species have high rate of stirring, and the combustion products are uniformly distributed inside the reactor. The fundamental equa- tions describing a PSR constitute systems of highly nonlinear algebraic equations, due to the complex rela- tionship between the net production rate of the species and the species concentration, which ultimately makes the equations stiff, and the solution of such equations become highly compute-intensive leading to the need for a efficient and robust solution algorithms. Graph- ics processing units (GPUs) have widely been used in the past as an effective alternative to central process- ing units (CPUs), and highly parallel threads of GPUs can be used in a efficient manner to improve the al- gorithm performance for speeding up the calculations. A highly parallelized GPU implementation is presented for a batched calculation of PSR model.

## I. INTRODUCTION

Accurate prediction of pollutant and gaseous byproducts emanating from practical combustion de- vices due to chemical reactions has been an increasingly important research topic in the recent years. Unlike homogeneous thermodynamic systems, whose chemi- cal and physical properties are uniform throughout the system, and whose simulations are usually in the order of a few seconds, practical combustion devices in the industrial combustion systems are inhomogeneous ther- modynamic systems, and the simulations of such sys- tems becomes complicated, because of the strong inter- action between chemistry, turbulent flow, heat transfer and mass transfer. Detailed simulation of such combus- tion systems remains computationally expensive and al- most infeasible, despite the advancement in CPU hard- ware. So there has also been an enormous effort for the development of efficient solution strategies by ex- ploring the underlying theoretical and computational as- pects related to detailed chemical reaction mechanisms. Two of the approaches for reducing the central process- ing unit (CPU) time include the use of storage-retrieval techniques such as in-situ adaptive tabulation (ISAT) [1], and reduced kinetic mechanisms [2], but, both of these approaches incur CPU time savings at the expense of

accuracy. There is a need for a strategy that can save CPU time without losing the accuracy of the predicted solutions. The computational parallelization of the cal- culations can be a reasonable alternative, yet the paral- lelization algorithm and the choice of parallel comput ing hardware architecture has a significant impact on the effectiveness of the algorithm and speedup achieved.

Perfectly stirred reactor is considered as an ideal reactor with high rate of stirring [3] inside reactor con- trol volume, leading to the perfect mixing of reactants. Because of the intense recirculation inside the reactor, reactants and products rapidly mix to form a homoge- neous mixture. PSRs have been used in the past for the study of soot formation and growth [4, 5], and chemi- cal kinetics [6]. In the case of a PSR, the Damkohler number, Da, which is the ratio of the rate of chemical reaction to the rate of mixing, is 1, and the chemi- cal reactions occur over a timescale that is much larger than the mixing timescale. The properties inside the re- actor are assumed to be spatially uniform, because of the efficient mixing of the species, thereby providing a homogeneous composition of gas inside the reactor. In another words, a PSR can be described by a single re- actor temperature and concentration of the species. In the current study, the walls of the PSR are considered to be non-catalytic. There are no deposits on the reac- tor walls, and therefore the inlet mass flow and outlet mass flow rates are

469

equal. The efficiency of the reac- tor is dependent on its volume and the volume flow rate or residence time, and it can be related to the chemical processes occurring inside.

Similar to the concept of PSR, for the case inho- mogeneous turbulent reacting flows, the eddy dissipa- tion combustion model (EDM) [7], is based on the in- finitely fast chemistry hypothesis and assumes that the reaction rate is controlled by turbulent mixing. A gen- eralized formulation of the EDM was proposed in order to take into account finite rate chemistry effects. How- ever, this model was not suited to treat detailed reac- tion mechanisms and failed with number of reactions exceeding three or four [8]. On the other hand, the eddy dissipation concept (EDC), which is an extension of the eddy dissipation model, allows detailed kinetics to be included in the calculations [9]. The model assumes that chemical reactions occur within the smallest turbulent structures, called fine structures. In other words, these are treated as PSRs which exchange chemical species with the surrounding. The overall reaction rate in each PSR is controlled by chemical kinetics. Conservation of mass, energy, and species govern a well stirred re- actor yielding systems of nonlinear algebraic equations, the number of which, is equal to the number of species undergoing reactions inside. The larger the dimension of combustion system, the larger the number of PSRs, and hence the size of the system of nonlinear algebraic equations to be solved will also be larger. The meth- ods to solve such equations can be classified either as direct methods or iterative methods. Direct methods involving LU decomposition and back solving are the most widely used methods, because of its simplicity, stability and robustness [10]. Governing equations of the perfectly stirred reactor constitute a set of nonlinear- algebraic equations, which are solved by employing the Newton's iteration, and this in turn needs the calcula- tion of Jacobian matrix for the purpose of linearization of nonlinear system. The Jacobian matrix needed for the LU decomposition is obtained by applying finite differ- ence approximation, which requires the calculation of net production rate of species evaluated from the reac- tion rate constants.

While using EDC model, if the system being simulated has n computational cells, and the reaction mechanism being used has m reactions, a single iteration for such calculation consists of n × m evaluations of rate constant, and since the rate constant is only dependent on the temperature, the calculations for n cells can be performed as a set of n-batched PSR cal- culations with each of the batch evaluating m chemi- cal rate constant evaluations. Since the rate constant calculations involve multiplication and

exponentiation, the tasks could be carried out separately and the effi- ciency can be improved further with parallelizations on Central Processing Units (CPUs) or Graphics Process- ing Units (GPUs). GPUs are an alternative approach of accelerator technology, that can potentially improve the performance of various algorithms. GPU performance already exceeds that of modern CPUs by one or two orders of magnitude on selected applications [11] well suited to the GPU architecture. In addition to raw com- puting performance, GPUs offer similarly impressive metrics for performance per dollar and performance per watt. Also, massively-parallel, multi-core architecture of GPUs delivers higher performance, and allows re- searchers to tackle some of today's most demanding ap- plications - applications that previously could only be handled by massive supercomputers or clusters, thus re- lieving to a great extent the burden on the environment or the power and cooling budget, mainly due to a large reduction in electrical and thermal requirements.

GPUs have been used for a variety of studies in different areas in the past. Some of the them in- clude linear algebra [12], computational biology [13], weather prediction [14], molecular dynamics [15], and turbulence modeling [16]. GPU related works in the field of combustion include the development of GPU enhanced algorithm for integration of stiff ordinary differential equation[17, 18, 19]. Stone [17] implemented a fourth-order Runge-Kutta-Fehlberg ordinary differential equation solver on a GPU, and achieved a maximum speed up of 20.2x over CPU. The work of Niemeyer and Sung[18] presented computations of chemical ki- netics of a hydrogen oxidation mechanism using the ex- plicit fifth-order Runge-Kutta-Cash-Karp method. They reported GPU speedups of 126x and 25x over the single- and six-core CPU version, respectively. Sewerin and Rigopoulos [19] developed an implementation of a three-stage/fifth-order implicit Runge-Kutta method Radau5 [20] using OpenCL. They tested the implemen- tation in the context of a transient equilibrium scheme for the flamelet model.

The main objective of this paper is to develop a general purpose GPU (GPGPU) implementation of rate constant calculation for a single iteration in the context of batched calculations of PSRs using CUDA, to inves- tigate the soot distribution inside a PSR using a statis- tical state-of-the-art soot model such as the Method of Moment with Interpolative Closure (MOMIC) [21].

## II. GOVERNING EQUATIONS

The PSR is governed by three different conservation equations: conservation of mass, conservation of species and conservation of energy. The rate of change of total mass inside the PSR can be related to the mass inflow rate and mass outflow rate as:

$$\frac{d(\rho V)}{dt} = \dot{m}_{in} - \dot{m}_{out} \qquad (1)$$

The rate of change of concentration of mass fraction of a species inside the PSR can be related to its molar production rate and initial mass fraction as

$$\frac{dY_i}{dt} = \frac{\dot{m}_{in}}{\rho V}(Y_{i,in} - Y_i) + \frac{\dot{\omega}_i^g MW_i}{\rho} \qquad (2)$$

Finally, the energy equation is given by

$$C_{p,mix}\frac{dT}{dt} = \frac{\dot{m}}{\rho V}\sum_{i=1}^{n} Y_{i,in}(h_{i,in} - h_i) - \frac{\sum_{i=1}^{n} h_i \dot{\omega}_i^g MW_i}{\rho} + \frac{\dot{Q}}{\rho V} - \frac{1}{\rho}\frac{dP}{dt} \qquad (3)$$

## III. COMPUTATIONAL FORMULATION OF RATE EVALUATION

The species conservation equations governing a PSR are a set of nonlinear algebraic equations. Since these equations are nonlinear in nature, they have to be solved iteratively. For this purpose the Newton It- eration technique is employed . In this method, one of the first steps involves constructing an error vector by substituting the initial guesses into the nonlinear- algebraic equations. The next step is to construct a Ja- cobian matrix. The Jacobian matrix is decomposed into lower and upper triangular matrices, and then a back- solving operation is carried out to obtain the final so- lution. The PSR equations are solved by employing a hybrid/newton time integration approach as described and developed for a PSR with soot moment equations in one of the authors' previous studies [5]. The hybrid approach consists of two steps: a steady state Newton iteration and a pseudo-transient time integration. If con- vergence is not achieved using the Newton iteration, the solution estimate is conditioned using the time integra- tion. The process of switching between the Newton it- eration and time integration approaches continues until convergence is achieved, which is based on very spe- cific criteria related to absolute and relative tolerances. See [5] for more details.

The net production rate ω˙ g in the equation 2 needs to be calculated at every time instant, the Jaco- bian matrix is calculated, and is represented as the sum- mation of the rate of progress variable for all reactions involving the it h species:

$$\dot{\omega}_i^g = \sum_{k=1}^{I} v_{i,k} q_k \qquad (i = 1,2,3,...n) \qquad (4)$$

where

$$v_{i,k} = v_i'' - v_i'$$

and, I is the total number of elementary reactions. $v$ rr and $v$ r are the stoichiometric coefficient of the species, i, in the product and reactant side, respectively. The rate of progress variable qk for the reaction, k, is given by the difference of the forward rates and reverse rates as,

$$q_k = k_{f,k}\prod_{i=1}^{k}[X_i]^{v'_{i,k}} - k_{r,k}\prod_{i=1}^{k}[X_i]^{v''_{i,k}} \qquad (5)$$

where Xi is the molar concentration of the species, i and k f k and krk are the forward and reverse rate constants of the kt h reaction. The forward rate constant for the I reactions are generally assumed to have the following Arrhenius temperature dependence:

$$k_{f,k} = A_k T^{\beta_k}\exp\left(\frac{-E_k}{RT}\right) \qquad (6)$$

where Ak is pre-exponential factor, βk is temperature ex- ponent , and Ek is the activation energy for reaction k. The reverse rate constants, kr,k , are related to the for- ward rate constants through the equilibrium constants as follows:

$$k_{r,k} = \frac{k_{f,k}}{K_{c,k}} \qquad (7)$$

The constant, Kc,k , for reaction, k, given by;

$$K_{c,k} = K_{p,k}\left(\frac{P_{atm}}{RT}\right)^{\sum_{i=1}^{n} v_{i,k}} \qquad (8)$$

The equilibrium constants Kp,k are obtained with the relationship,

$$K_{p,k} = exp\left(\sum_{i=1}^{n} v_{i,k}\frac{S_i^o}{R} - \sum_{i=1}^{n} v_{ik}\frac{H_i^o}{RT}\right) \qquad (9)$$

For chemical mechanisms with a large number of reactions, calculation of the forward rate constants (in

Equation 6) is time consuming, specifically due to the exponential term. Since the values of the pre- exponent factor, temperature exponent and the activa- tion energy are different for the different reactions, the forward rate expression for different reactions can be evaluated separately and independently, and hence are better suited for GPU parallelization. From Equations 7, 8 and 9, it can be seen that the reverse rate constants are dependent on the entropy, enthalpy and the stoichio- metric coefficients of the species, both on the reactant and product sides, which clearly indicate that their cal- culations are not straightforward and involve a lot of operations. The first step in the calculation of reverse rate constants involves determining the equilibrium constant from Equation 9. The right hand side of the Equation 9 is an exponential term, where the exponent involves two different summations. The summation runs over the total number of species involved in a single reaction. The more the number of reactions, the more time consuming this particular operation becomes, and thereby is more suitable for GPU parallelization. The second step is the calculation of the constant, $K_{c,k}$, which involves the multiplication of equilibrium constants with $P_{atm}$ raised to the power of another summation term. Again, the summation is over the number of species in a reaction and so is computationally expensive for a large chemi- cal mechanisms, which in turn makes them suitable candidates for GPU parallelization as well. The third and final step is the evaluation of expression from Equation 7, which is a division operation, and can easily be set up for GPU acceleration

## IV. GPU IMPLEMENTATION

The pseudo code for the current GPU implementation is presented in this section. The CalcRateofProgress() function is the main function which calls three Cuda kernel functions for the calculation of reaction rate constants. Dynamic memory allocation and variable initialization is carried out in the beginning. Data required for the calculations is then transferred from the CPU to the GPU. After the calculation of rate constants, GPU data is transferred back to CPU. During the course of these calculations, the stoichiometric coefficients of the gaseous reactant and product species have a constant value, and hence, they are stored as a constant memory on the GPU.
function CalcRateofProgress()
{call CalcFwdRateConst(kf k; I;n; :::::)
call CalcKeqConst(kpk; I;n;So
i ;Hoi :::::)call CalcRevRateConst(krk;kf k;Kpk; I;n; :::::)
transfer kf k and krk from GPU to CPU
calculate the rate of progress variable qk}
end function CalcRateofProgress()

cuda kernel CalcFwdRateConst(kf k; I;n; :::::)
for each GPU thread in parallel
calculate forward rate constant kf k
end cuda kernel CalcFwdRateConst()
cuda kernel CalcKeqConst(kpk; I;n;So
i ;Ho
i :::::)
for each GPU thread in parallel
calculate equilibrium constant Kpk in parallel
end cuda kernel CalcKeqConst()
cuda kernel CalcRevRateConst(krk;kf k;Kpk; I;n; :::::)
for each GPU thread in parallel
calculate constant kck in parallel
calculate equilibrium constant krk in parallel
end cuda kernel CalcRevRateConst()

## V. VALIDATION OF PHYSICAL RESULTS

This section presents the results for ethylene/air combustion in a PSR for the purpose of validating the in-house CPU code and the results from GPU imple- mentation. Ethylene/air combustion in a PSR at specific operating conditions was previously numerically stud- ied and presented by Brown et al.[22], which serves as a reference here in the current study. Two sets of PSR simulations are carried out here: the first, with the re- actor temperature varying from 1500 K to 2000 K in steps of 50 K, at a pressure of 1 atm, with a residence time of 0.005 s, and the second, with the reactor resi- dence time varying from 5 ms to 95 ms in steps of 5 ms, again at a pressure of 1 atm, with a reactor tempera- ture of 1750 K. The fuel-air (F/A) equivalence ratio ($\varphi$) was 2.5 for all of the calculations presented in this sec- tion. Figure 1 shows the variation of soot volume frac- tion with reactor temperature and residence time using MOMIC as soot model. Results from Brown et. al [22] are also digitized and plotted alongside. The reaction mechanism used for the simulations was that of Wang and Frenklach [23]. Firstly from the figure, it is clear that, the soot volume fraction calculations from CPU code match perfectly with those calculated from GPU.
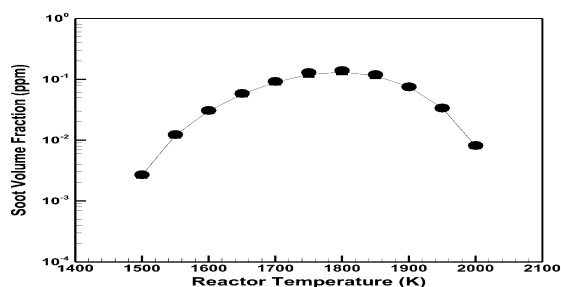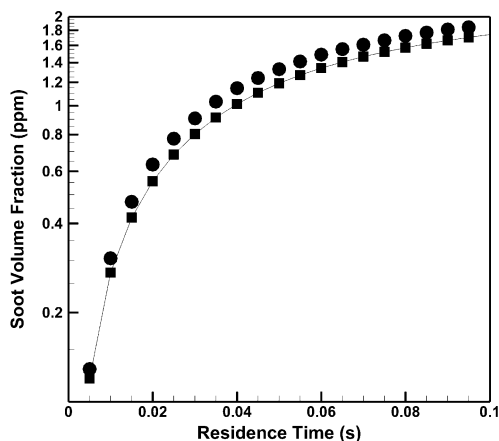
*Figure 1: Variation of soot volume fraction with reactor temperature (left) and residence time (right) for ethylene/air combus- tion from PSR simulations; Results from Brown et al. - symbols, CPU Re- sult with MOMIC - lines with closed square sym- bols, GPU Result with*

MOMIC - lines with open square symbols code, indicating the correctness of the GPU implemen- tation. Also, the soot volume fraction calculations at different residence times using the CPU code matches well with the results from Brown et. al [22], with a max- imum difference being 0.1 ppm from [22]. With regard to the reactor soot volume fraction versus reactor tem- perature, calculations from CPU code match perfectly with the predictions from Brown et. al [22] and GPU code. Overall, the predictions from CPU code and GPU code compare very well with those from Brown et. al [22], thereby validating the CPU code and GPU imple- mentation.

Figure 2 shows the variation of soot particle number density and soot particle diameter with reactor residence time, and also with reactor temperature. Soot particle number density decreases with residence time, due to the high coagulation rates at high resi- dence times. The particle diameter on the other hand, increases almost linearly with residence time. With regard to reactor temperature, particle number density first increases, attains a maximum, and then decreases. In this case, the soot particle diameter increases slowly for lower temperatures (< 1750K) and shows a much steeper increases until 1950 K, after which there is no change.

The next two figures show the behavior of ma- jor and minor gaseous species with respect to residence time and reactor temperature. Figure 3 shows the vari- ation of the mole fraction of $C_2H_4$, $O_2$, $CO_2$, and $CH_4$ with reactor

residence time. $C_2H_4$ and $O_2$ decrease with increasing residence time, which is expected because of the fact that higher the residence time higher will be the time available for the reactants to get consumed and be converted into products. The $CO_2$ concentration as a function of residence time exhibited a steep increase followed by almost constant value while $CH_4$ increase initially to a maximum at around 30 ms and then de- creases continuously. Figure 4 shows the variation of mole fraction of gaseous species with respect to reactor temperature. $C_2H_4$ is being consumed at higher rate for the temperature range of 1500 K to 1850 K as compared to the temperature range of 1850 K to 2000 K. As an- ticipated, consumption of $O_2$ is increasing with respect to reactor temperature. Concentration of $CO_2$ and $H_2O$ exhibited an increase to a maximum at 1750 K and then a continuous decrease.
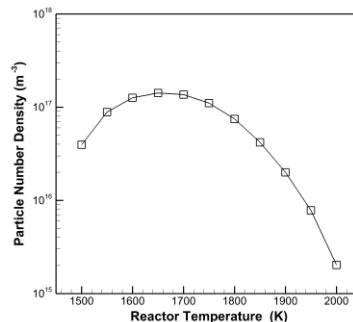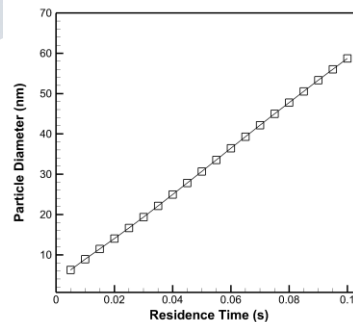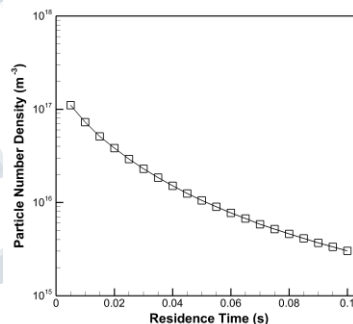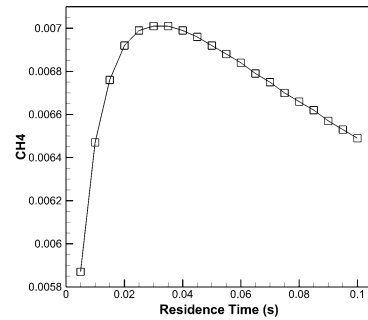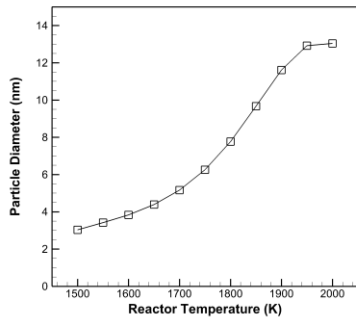
*Figure 2: Variation of particle number density (top left) and particle diameter (top right) with residence time, and vari- ation of particle number density (bottom left) and particle diameter (bottom right) with reactor tem- perature for ethylene/air combustion from PSR simulations; GPU Result with MOMIC - lines with open square symbols*

*Figure 3: Mole fraction of C2H4 (top left), Mole frac- tion of O2 (top right), Mole fraction of CO2 (bottom left), Mole fraction of CH4 (bottom right) with resi- dence time for ethylene/air combustion from PSR sim- ulations; GPU Result with MOMIC - lines with open square symbols*
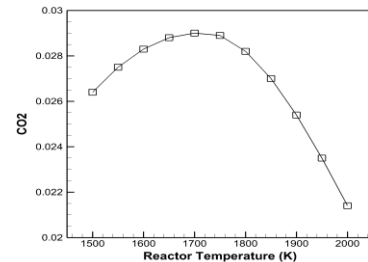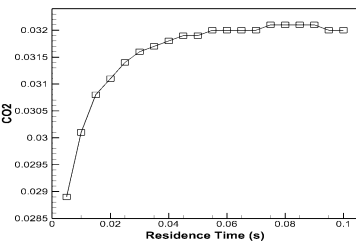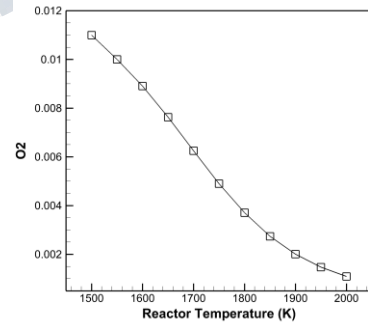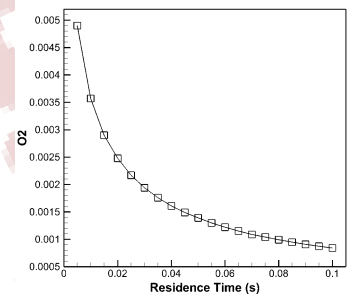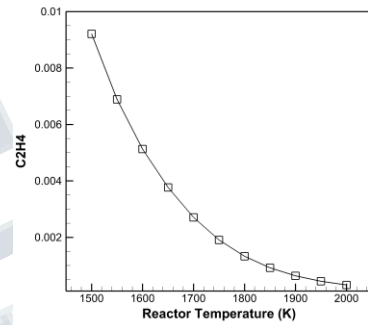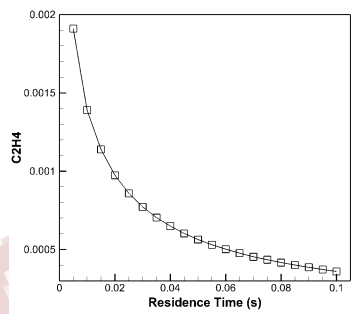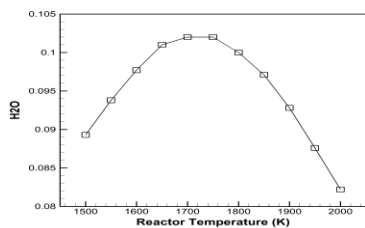
*Figure 4: Mole fraction of C2H4 (top left), Mole frac- tion of O2 (top right), Mole fraction of CO2 (bottom left), Mole fraction of H2O (bottom right) with reac- tor temperature for ethy- lene/air combustion from PSR simulations; GPU Re- sult with MOMIC – lines with open square symbols Speedup*

The GPU implementation of a batched PSR cal- culation presented here is integrated into our in-house PSR solver, which in turn is based on a hybrid/newton time integration approach [5]. A batched PSR calcula- tion (of say n PSRs) refers to a set of n PSRs that are solved simultaneously. The performance of the GPU implementation over the CPU is assessed by determin- ing the speedup, which is the ratio of the CPU computa- tion time over the GPU computation time. CPU compu- tations were performed on a workstation with Intel(R) Xeon(R) E5-2643 3.30 GHz processor while GPU com- putations were performed on NVIDIA Tesla K40. The speedup being shown here is for PSR batches for a sin- gle newton iteration only.

CPU and GPU calculations for ethylene combus- tion in PSR are carried out for a range of PSR batch sizes from 100 to 1 million, with each of the PSR having a constant temperature of 1800K and residence time of 50 ms. Inlet composition of each PSR in the batch is de- fined by the mole fraction of reacting species; 0.15 mole of $C_2H_4$ , 0.18 mole of $O_2$ , and 0.67 mole of $N_2$ . The re- action mechanism used in the PSR calculations consists of 101 species and 542 reactions [24]. Figure 5 shows the GPU speedup for the different batch sizes. While the GU and CPU timings are comparable for batch sizes below 104 , the maximum speedup obtained was 3 for a batch size of 1 million.
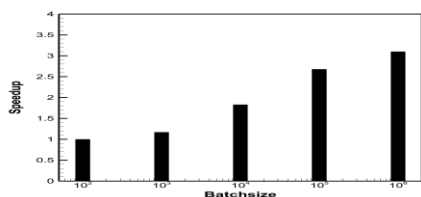


*Figure 5: Speedup versus size of batched PSR*

## VI. CONCLUSION

A highly-parallelized GPU algorithm for solving a batched PSR was developed, coupled to the moment- based soot modeling method, MOMIC, and success- fully tested for ethylene-combustion. The GPU algo- rithm is based on the GPU-enhanced algorithm for for- ward reaction rate calculation, equilibrium constant cal- culation, and reverse rate constant evaluation. This al- gorithm was first validated by comparing GPU and CPU results from the current PSR development to previous PSR calculations of ethylene combustion [22] for vari- ous reactor temperatures and residence times. Excellent agreement was observed in these comparisons, indicat- ing the correctness of the implementation. In addition, the GPU-enhanced algorithm was tested for 5 different batchsizes and GPU algorithm was found to be three times faster than the CPU algorithm for PSR batchsize of one million. The algorithm that has been presented here can be applied to other combustion simulations that make use of detailed kinetics.

## REFERENCES

[1] Pope, S. B., 1997. "Computationally efficient implementation of combustion chemistry using in situ adaptive tabulation". pp. 41–63.

[2] Bhattacharjee, B., Schwer, D. A., Barton, P. I., and Green, W. H., 2003. "Optimally-reduced kinetic models: reaction elimination in large-scale kinetic mechanisms". Combustion and Flame, 135(3), pp. 191–208.

[3] Glarborg, P., Kee, R., Grcar, J., and Miller, J., 1986. Psr: A fortran program for modeling well- stirred reactors; sandia national laboratories: Liv- ermore, ca; sand86-8209. Tech. rep.

[4] Adhikari, S., Sayre, A., and Chandy, A. J., 2015. "A gpu-parallelized perfectly stirred reactor (psr) model for soot modeling". Central States Section of the Combustion Institute.

[5] Adhikari, S., Sayre, A., and Chandy, A. J., 2016. "A hybrid newton/time integration approach coupled to soot moment methods for modeling soot formation and growth in perfectly stirred re- actors". Combustion Science and Technology, 188(8), pp. 1262–1282.

[6] Bowman, B., Pratt, D., and Crowe, C., 1973. "Ef- fects of turbulent mixing and chemical kinetic- son nitric oxide production in a jet-stirred reac- tor". In Symposium

(International) on Combus- tion, Vol. 14, Elsevier, pp. 819–830.

[7] Magnussen, B. F., and Hjertager, B. H., 1977. "On mathematical modeling of turbulent combustion with special emphasis on soot formation and com- bustion". In Symposium (International) on Com- bustion, Vol. 16, Elsevier, pp. 719–729.

[8] Galletti, C., Lenzi, M., Parente, A., and Tognotti, L., 2006. "Numerical modeling of hydrogen diffu- sion jet flame: the role of the combustion model". In 29th Combustion Meeting, Italian Section of Combustion Institute, pp. VIII3.1–VIII3.6.

[9] Magnussen, B. F., and Hjertager, B., 1981. "On the structure of turbulence and a generalized eddy dissipation concept for chemical reaction in turbu- lent flow". In 19th AIAA Aerospace Meeting, St. Louis, USA.

[10] Botsch, M., Kobbelt, L., Pauly, M., Alliez, P., and Lévy, B., 2010. Polygon mesh processing. CRC press.

[11] Barrachina, S., Castillo, M., Igual, F. D., Mayo, R., and Quintana-Ortí, E. S., 2008. "Solving dense linear systems on graphics processors". In Euro- pean Conference on Parallel Processing, Springer, pp. 739–748.

[12] Barrachina, S., Castillo, M., Igual, F. D., Mayo, R., and Quintana-Ortí, E. S., 2008. "Solving dense linear systems on graphics processors". In Euro- pean Conference on Parallel Processing, Springer, pp. 739–748.

[13] Schatz, M. C., Trapnell, C., Delcher, A. L., and Varshney, A., 2007. "High-throughput sequence alignment using graphics processing units". BMC bioinformatics, 8(1), p. 474.

[14] Michalakes, J., and Vachharajani, M., 2008. "Gpu acceleration of numerical weather prediction". Parallel Processing Letters, 18(04), pp. 531–548.

[15] Le Grand, S., Götz, A. W., and Walker, R. C., 2013. "Spfp: Speed without compromiseâĂŤa mixed precision model for gpu accelerated molec- ular dynamics simulations". Computer Physics Communications, 184(2), pp. 374–380.

[16] Rucki, C., and Chandy, A. J., 2014. "Towards fft-based direct numerical simulations of turbulent flows on a gpu".

International Journal of Model- ing, Simulation, and Scientific Computing, 5(01), p. 1350014.

[17] Stone, C. P., and Davis, R. L., 2013. "Tech- niques for solving stiff chemical kinetics on graph- ical processing units". Journal of Propulsion and Power, 29(4), pp. 764–773.

[18] Niemeyer, K. E., and Sung, C.-J., 2014. "Ac- celerating moderately stiff chemical kinetics in reactive-flow simulations using gpus". Journal of Computational Physics, 256, pp. 854–871.

[19] Sewerin, F., and Rigopoulos, S., 2015. "A method- ology for the integration of stiff chemical kinet- ics on gpus". Combustion and Flame, 162(4), pp. 1375–1394.

[20] Wanner, G., and Hairer, E., 1991. Solving ordinary differential equations II, Vol. 1. Springer-Verlag, Berlin.

[21] Frenklach, M., 2002. "Method of moments with interpolative closure". Chemical Engineering Sci- ence, 57(12), pp. 2229–2239.

[22] Brown, N. J., Revzan, K. L., and Frenklach, M., 1998. "Detailed kinetic modeling of soot forma- tion in ethylene/air mixtures reacting in a perfectly stirred reactor". In Symposium (International) on Combustion, Vol. 27, Elsevier, pp. 1573–1580.

[23] Wang, H., and Frenklach, M., 1997. "A detailed kinetic modeling study of aromatics formation in laminar premixed acetylene and ethylene flames". Combustion and flame, 110(1), pp. 173–221.

[24] Appel, J., Bockhorn, H., and Frenklach, M., 2000. "Kinetic modeling of soot formation with detailed chemistry and physics: laminar premixed flames of c 2 hydrocarbons". Combustion and Flame, 121(1), pp. 122–136.