GPU-based explicit integration algorithms for accelerating chemical kinetics in CFD simulations

Kyle Niemeyer^{1,2} and Chih-Jen Sung²

¹Case Western Reserve University, Cleveland, OH, USA; ²University of Connecticut, Storrs, CT, USA

Introduction

Detailed chemistry introduces prohibitive computational demands into reactive-flow simulations, due to the stiffness of even moderately sized detailed reaction mechanisms. Integrating stiff mechanisms requires expensive implicit methods.

Alternative algorithms, paired with the maturing processing power of massively parallel graphics processing units (GPUs), offer an approach to significantly reduce the cost of high-fidelity reactive-flow simulations.

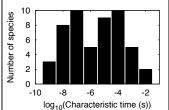


Figure 1. The species creation timescales for a methane mechanism with 53 species range over eight orders of magnitude.

Methods

In our strategy for GPU-acceleration of reactive-flow simulations, GPU threads independently integrate the chemistry terms for each CFD cell/control volume [1,2]. While implicit algorithms involve complex logical flow that may induce poor performance if implemented in this manner on GPUs, explicit algorithms follow more similar instruction pathways and are therefore better-suited for GPU acceleration.

Mechanisms with little-to-no stiffness may be integrated using highly-efficient explicit integration algorithms; we implemented the fifth-order adaptive Runge-Kutta-Cash-Karp (RKCK) algorithm [3].

Standard explicit algorithms become unstable in the presence of stiffness, so for mechanisms with moderate levels we implemented the second-order adaptive Runge-Kutta-Chebyshev (RKC) algorithm [4], a stabilized explicit method.

Results

We studied the performance of the RKCK and RKC algorithms by comparing the average wall-clock time of ten integration steps between equivalent CPU and GPU versions for a wide range of independent systems of ODEs (corresponding to CFD grid points/cells). Constant global time steps were specified, such that the integration algorithms performed adaptive substepping as needed. We considered three mechanisms:

- Nonstiff hydrogen mechanism (9 species & 38 irreversible reactions)
- Moderately stiff methane mechanism (53 species & 325 irrev. reactions)
- Stiff ethylene mechanism (111 species & 1566 irrev. reactions)

Figure 2 shows the performance comparison for the CPU and GPU versions of RKCK with the nonstiff hydrogen mechanism, for global time steps of 10 ns. This case is classified as little-to-no stiffness because the RKCK algorithm performed more than twice as fast as RKC. The GPU-based RKCK performed faster than the single- and six-core CPU versions for problem sizes of more than 128 and 512 ODEs, respectively. At most, RKCK-GPU ran 126 and 25 times faster than the single- and six-core CPU versions.

Moving to the moderately stiff methane mechanism, where the RKC algorithm now runs more than 15 times as fast as RKCK, Figure 3 shows the performance comparison between the six-core CPU and GPU versions of RKC, as well as the standard implicit VODE algorithm on six CPU cores. In this case, we used a global time step size of 1 μs . Since this problem involves only moderate levels of stiffness, the CPU version of RKC actually outperformed VODE by three to six times. As with the hydrogen mechanism, at smaller problem sizes (less than 512 ODEs) the six-core RKC-CPU ran faster than RKC-GPU. At larger numbers of ODEs, RKC-GPU performed better, running up to 13 times faster than six-core RKC-CPU and 57 times faster than VODE (on six CPU cores).

Finally, Figure 4 shows the performance comparison between VODE on six CPU cores and RKC-GPU for a case with more severe stiffness, using the ethylene mechanism with a global time step size of 1×10^{-4} s. At all numbers of ODEs, VODE outperformed RKC-GPU, with RKC-GPU running at best 2.5 times slower than VODE.

Do explicit methods perform faster on GPUs?

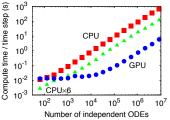


Figure 2. Comparison of performance between single- and six-core CPU and GPU versions of RKCK with nonstiff hydrogen mechanism.

With some stiffness, are GPU-based stabilized explicit methods faster?

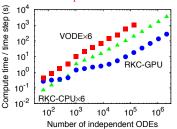


Figure 3. Comparison of performance between six-core RKC-CPU and VODE and RKC-GPU with moderately stiff methane mechanism.

What happens with greater levels of stiffness?

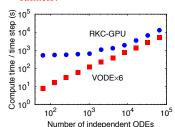


Figure 4. Comparison of performance between six-core VODE and RKC-GPU with stiff ethylene mechanism.

Conclusions

Detailed chemistry can account for 90% or more of the total computational time in a reactive-flow simulation; reducing the cost of detailed chemistry is necessary in order to:

- Study larger and more complex real-world problems
- Add higher levels of detail for greater accuracy
- Reduce simulation times to allow more cases and variation of parameters for design studies

In this study, we showed that by pairing explicit integration algorithms with GPU acceleration for reaction mechanisms exhibiting little-to-moderate levels of stiffness, the computational time needed to integrate many independent systems of ODEs—corresponding to CFD grid points/ volumes—can be reduced more than an order of magnitude compared to the same algorithm running on six CPU cores. This work builds and improves upon the studies of Spafford et al. [1], Niemeyer et al. [2], Shi et al. [5], and Stone et al. [6].

With no stiffness, the explicit Runge–Kutta–Cash–Karp algorithm ran up to 25 times faster on a GPU than on six CPU cores. In the presence of moderate stiffness, the stabilized Runge–Kutta–Chebyshev algorithm, implemented on a GPU, outperformed its six-core CPU equivalent by a factor of 13. Compared against the VODE implicit integration algorithm—commonly used in reactive-flow simulations—running on six CPU cores, the GPU-based RKC ran nearly 60 times faster for large problem sizes.

However, when more severe levels of stiffness were present as in the case of the ethylene mechanism, the performance of RKC (on both CPU and GPU) degraded. At best, the GPU implementation of RKC performed 2.5 times slower than VODE on six CPU cores. This suggests that an alternative algorithm is needed for GPU acceleration of stiff chemistry, which we are currently developing.

Literature cited

- [1] K Spafford, J Meredith, J Vetter, JH Chen, R Grout, R Sankaran. *LNCS* 6043 (2010) 122– 131.
- [2] KE Niemeyer, CJ Sung, CG Fotache, JC Lee. 7th FTM ESSCI, 2011.
- [3] JR Cash, AH Karp. ACM Trans. Math. Software. 16 (1990) 201–222.
- [4] BP Sommeijer, LF Shampine, JG Verwer. *J. Comput. Appl. Math.* 88 (1997) 315–326.
- (1997) 313–320.[5] Y Shi, WH Green, H Wong, OO Oluwole. Combust. Flame 159 (2012) 2388–2397.
- [6] CP Stone, RL Davis, B Sekar. 51st AIAA ASM, 2013

Acknowledgments

This work was supported by the NSF under Grant No. 0932559, the U.S. Department of Defense through the National Defense Science and Engineering Graduate Fellowship program, and the NSF Graduate Research Fellowship under Grant No. DGE-0951783.

Further information

Kyle Niemeyer

kyle.niemeyer@case.edu, www.kyleniemeyer.com

Chih-Jen Sung

cjsung@engr.uconn.edu, http://combdiaglab.engr.uconn.edu/



