Journal of Computational Physics 256 (2014) 854–871

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Accelerating moderately stiff chemical kinetics in reactive-flow simulations using GPUs

Kyle E. Niemeyer [a,b,*], Chih-Jen Sung [b]

[a] *Department of Mechanical and Aerospace Engineering, Case Western Reserve University, Cleveland, OH 44106, USA*
[b] *Department of Mechanical Engineering, University of Connecticut, Storrs, CT 06269, USA*

A R T I C L E   I N F O

Article history:
Received 11 May 2013
Received in revised form 22 August 2013
Accepted 13 September 2013
Available online 25 September 2013

Keywords:
Reactive-flow modeling
GPU
Chemical kinetics
Stiff chemistry
CUDA

A B S T R A C T

The chemical kinetics ODEs arising from operator-split reactive-flow simulations were solved on GPUs using explicit integration algorithms. Nonstiff chemical kinetics of a hydrogen oxidation mechanism (9 species and 38 irreversible reactions) were computed using the explicit fifth-order Runge–Kutta–Cash–Karp method, and the GPU-accelerated version performed faster than single- and six-core CPU versions by factors of 126 and 25, respectively, for 524,288 ODEs. Moderately stiff kinetics, represented with mechanisms for hydrogen/carbon-monoxide (13 species and 54 irreversible reactions) and methane (53 species and 634 irreversible reactions) oxidation, were computed using the stabilized explicit second-order Runge–Kutta–Chebyshev (RKC) algorithm. The GPU-based RKC implementation demonstrated an increase in performance of nearly 59 and 10 times, for problem sizes consisting of 262,144 ODEs and larger, than the single- and six-core CPU-based RKC algorithms using the hydrogen/carbon-monoxide mechanism. With the methane mechanism, RKC-GPU performed more than 65 and 11 times faster, for problem sizes consisting of 131,072 ODEs and larger, than the single- and six-core RKC-CPU versions, and up to 57 times faster than the six-core CPU-based implicit VODE algorithm on 65,536 ODEs. In the presence of more severe stiffness, such as ethylene oxidation (111 species and 1566 irreversible reactions), RKC-GPU performed more than 17 times faster than RKC-CPU on six cores for 32,768 ODEs and larger, and at best 4.5 times faster than VODE on six CPU cores for 65,536 ODEs. With a larger time step size, RKC-GPU performed at best 2.5 times slower than six-core VODE for 8192 ODEs and larger. Therefore, the need for developing new strategies for integrating stiff chemistry on GPUs was discussed.

1. Introduction

The heavy computational demands of high-fidelity computational fluid dynamics (CFD) simulations, caused by fine grid resolutions and time step sizes in addition to complex physical models, are the primary bottleneck preventing most industrial and academic researchers from performing and using such studies. Reactive-flow simulations considering detailed chemistry in particular pose prohibitive computational demands due to (1) chemical stiffness, caused by rapidly depleting species and/or fast reversible reactions, and (2) the large and ever-increasing size of detailed reaction mechanisms. While reaction mechanisms for fuels relevant to hypersonic engines, such as hydrogen or ethylene, may contain 10–70 species [1,2], a recent surrogate mechanism for gasoline consists of about 1550 species and 6000 reactions [3]; a surrogate mechanism for biodiesel contains almost 3300 species and over 10,000 reactions [4]. Strategies for incorporating such large, realistic

[*] Corresponding author. Current address: School of Mechanical, Industrial, and Manufacturing Engineering, Oregon State University.
E-mail addresses: Kyle.Niemeyer@oregonstate.edu (K.E. Niemeyer), cjsung@engr.uconn.edu (C.-J. Sung).

0021-9991/$ – see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2013.09.025

reaction mechanisms in reactive-flow simulations are beyond the scope of this paper; for example, Lu and Law [5] recently reviewed strategies for mechanism reduction.

Even compact mechanisms pose challenges due to stiffness. In the presence of stiffness, explicit integration algorithms generally require time step sizes on the same order as the fastest chemical time scales, which can be many orders of magnitude smaller than the flow time scale [5]. Due to the resulting computational inefficiency, most reactive-flow simulations rely on specialized integration algorithms such as high-order implicit solvers based on backward differentiation formulas (BDFs) [6,7]. However, these implicit solvers involve expensive linear algebra operations, so techniques for removing stiffness via reduced chemistry have also been developed [5].

Exploiting graphics processing unit (GPU) acceleration offers another avenue for enabling the use of accurate, detailed reaction mechanisms in high-fidelity reactive-flow simulations. Most reactive-flow codes rely on the operator-splitting or fractional-step method [8–15], where the large system of governing partial differential equations (PDEs) is separated such that different physical processes are evaluated separately. For the chemistry—typically the most time-consuming portion of the simulation, accounting for 90% or more of the total simulation time in some cases—this results in a system of independent ordinary differential equations (ODEs) for the conservation of species mass in each spatial location (i.e., at each grid point or volume).

Due to the independent nature of the integration for the systems of ODEs governing chemistry in all locations, the entire set can be integrated simultaneously. One option is to parallelize the chemistry integration on multiple central processing unit (CPU) cores or processors using the Message Passing Interface (MPI) [16] or OpenMP [17–19], but the massive parallelism and increasing performance of GPUs—as well as the potential to reduce capital costs through improved energy efficiency—make them an attractive option for accelerating reactive-flow codes. General CFD applications also benefit from GPU acceleration due to the inherent data parallelism of most calculations for both finite difference and finite volume methods. Vanka et al. [20] surveyed some of the literature on using GPUs to accelerate general CFD simulations; more recently, Niemeyer and Sung [21] comprehensively reviewed advances in this area for both nonreactive and reactive flows. In the following, we will summarize important results related to GPU-based reactive-flow simulations.

The first effort in this area came from Spafford et al. [22], who accelerated the species rate evaluations in the direct numerical simulation (DNS) code S3D [23,24] on the GPU. In their approach, the CPU handles the time integration of the chemical source terms using an explicit fourth-order Runge–Kutta method. Each integration step requires four species rate evaluations, and for each evaluation the CPU invokes the GPU to evaluate the species rates of change for all grid points simultaneously. Using an ethylene reaction mechanism with 22 species, Spafford et al. [22] achieved performance speedups of around 15$\times$ and 9$\times$ for single- and double-precision calculations, respectively.

Most recent efforts follow the spatially-independent acceleration paradigm introduced by Spafford et al. [22], beginning with Niemeyer et al. [25], who developed a GPU-based explicit integration algorithm for nonstiff chemistry. Using a compact hydrogen mechanism with 9 species and 38 irreversible reactions [26], Niemeyer et al. [25] demonstrated a computational speedup of up to 75$\times$ compared to a single-core CPU over a wide range of independent ODE systems. Shi et al. [27] presented a hybrid CPU/GPU chemistry integration strategy where the GPU simultaneously integrates nonstiff chemistry in grid cells using an explicit algorithm and the CPU handles spatial locations with stiff chemistry using a standard implicit integrator. This combined approach, paired with a reactive-flow code, achieved an overall performance speedup of 11–46$\times$ over the algorithms executed on a single CPU core.

Le et al. [28] developed the first reactive-flow solver where the GPU evaluates both the fluid transport and chemical kinetics terms. As with most other approaches, they used operator splitting to decouple and independently solve the fluid transport and chemistry terms. They handled the stiff chemical kinetics terms in parallel on the GPU using a first-order implicit method (the backward Euler method), employing a direct Gaussian elimination to solve the resulting linear system of equations. Compared against an equivalent CPU version executed on a single processor core, their combined GPU solver performed up to 40 times faster using a reaction mechanism for methane with 36 species and 308 reversible reactions, on a grid with greater than 10^4 cells. However, the low order of the chemistry solver—first order—should be noted.

Stone et al. [29] implemented two chemistry integrators on the GPU: (1) a fourth-order adaptive Runge–Kutta–Fehlberg (RKF45) method and (2) the standard fifth-order accurate implicit CVODE method. Applied to a reduced mechanism for ethylene with 19 species and 15 global reaction steps [30] and compared against equivalent single-core CPU versions over a range of ODE numbers, the RKF45 and CVODE methods achieved up to 28.6$\times$ and 7.7$\times$ speedup, respectively. The GPU-based RKF45 method performed 20.2$\times$ faster than the CPU-based DVODE solver operating on a single core. It should be noted that the reduced mechanism used by Stone et al. [29] may not exhibit much stiffness, since it was developed by applying the quasi-steady-state approximation to certain radical species and eliminating fast elementary reactions [30].

Alternative approaches for GPU acceleration of chemical kinetics have also been presented that exploit other areas of data independence. Shi et al. [31] used the GPU to (1) simultaneously calculate all the reaction rates for a single kinetic system (i.e., a single computational volume/grid point) and (2) accelerate the matrix inversion step of the implicit time integration using a commercial GPU linear algebra library, CULA [32]. They found this approach beneficial for large reaction mechanisms (e.g., more than 1000 species), accelerating homogeneous autoignition simulations up to 22$\times$, but for moderate-size mechanisms (e.g., less than 100 species) their GPU-based implementation performed slower than the original CPU version. More recently, Sankaran [33] presented a new approach for accelerating the chemistry in turbulent combustion simulations where the GPU solves the unsteady laminar flamelet equations; the controlling CPU handles the main flow solver. This method involves three levels of concurrency on the GPU: (1) the solution of species reaction rates, thermochemical properties, and