nekCRF: A next generation high-order reactive low Mach flow solver for direct numerical simulations

Stefan Kerkemeier[a], Christos E. Frouzakis[b], Ananias G. Tomboulides[c],
Paul Fischer[d], Mathis Bode[e,*]

[a] *Micodat Computer GmbH, Germany*
[b] *Combustion, Acoustics & Flow Physics Laboratory, ETH Zürich, 8092 Zürich, Switzerland*
[c] *Department of Mechanical Engineering, Aristotle University of Thessaloniki, Greece*
[d] *University of Illinois Urbana-Champaign, U.S.A.*
[e] *Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, 52425 Jülich, Germany*
[*] *corresponding author, email: m.bode@fz-juelich.de*

Abstract

Exascale computing enables high-fidelity simulations of chemically reactive flows in practical geometries and conditions, and paves the way for valuable insights that can optimize combustion processes, ultimately reducing emissions and improving fuel combustion efficiency. However, this requires software that can fully leverage the capabilities of current high performance computing systems. The paper introduces *nekCRF*, a high-order reactive low Mach flow solver specifically designed for this purpose. Its capabilities and efficiency are showcased on the pre-exascale system JUWELS Booster, a GPU-based supercomputer at the Jülich Supercomputing Centre including a validation across diverse cases of varying complexity.

1 Introduction

The urgency to address energy sufficiency and environmental quality by reducing carbon emissions and transitioning to carbon-neutral fuels is a major challenge. Accurate and comprehensive data is required to improve our physical understanding for the combustion of novel fuels and support the development of predictive numerical models.

Recent advances in hardware and algorithms have made direct numerical simulations (DNS) a valuable tool in numerical turbulent combustion research [1], providing a "model-free" complete description resolved in space and time that cannot be obtained by other means. However, exploiting the full potential of DNS remains a daunting task. The wide range of time and length scales, coupled with the large number of tightly interconnected thermochemical scalars and the complex interactions between flow and chemistry, makes these simulations among the most demanding in computational fluid dynamics (CFD), and necessitate, fast solvers that can efficiently harness the capabilities of the latest supercomputers.

Unfortunately, most combustion research codes are not fully optimized for DNS on the latest supercomputer architectures. Even when they are, they frequently remain limited to simple geometries. Another issue is that most codes are not publicly available, limiting collaboration, transparency and benchmarking. Some of these concerns have been addressed with the release of Pele [2], developed as part of the US Exascale Computing Project. It is an open-source code suite designed for GPU-accelerated systems, employing a second-order finite volume method along with adaptive mesh refinement and an immersed boundary method to handle complex geometries and address real-world applications.

This paper introduces *nekCRF* the open-source successor of LAVp [3, 4, 5], specifically designed to align with the recent paradigm shift in HPC towards (GPU) accelerated computing. Drawing on the experience in the development and optimization of its predecessor, *nekCRF* shares the same key algorithmic features. The code supports finite rate chemistry, mixture-averaged transport, conjugate heat transfer, moving meshes and time-varying thermodynamic pressure.

It is built on the CFD code nekRS [6], which is based on a `MPI+X` programming model supporting CPUs and accelerators through the Open Concurrent Computing Abstraction (OCCA) [7]. nekRS implements the spectral element method (SEM) offering fast matrix-free operator evaluations with the geometric flexibility of the finite element method [8] enabling the accurate handling of various technical applications in complex geometries. Numerically, the method exhibits far lower levels of numerical diffusion and dissipation at higher polynomial orders at a competitive cost compared to low order method. This makes it

a particularly well-suited approximation choice in areas such as DNS of turbulent flows (see e.g. [9]), where the accurate time-advection of energetic structures such as vortices is a key concern. For optimal performance, the code supports only hexahedral elements, although geometric modeling can be challenging as the generation of accurate, boundary-conforming curvilinear hexahedral meshes remains an open problem. One approach to mitigate this issue is through the use of overset grids which are also supported. Nonconformal adaptive mesh refinement is another potential approach but it still relies on an initial coarse grid, which can be problematic for complex geometries. The highest geometric flexibility is offered by unstructured meshes of simplicial or hybrid elements implemented by solvers like Nektar++ [10]. However, it is evident and unavoidable that these element types will face challenges in achieving the same performance as the more naturally tensor-product structured hexahedral elements. Nonetheless, in some cases, the trade-off in performance may be justified. For a detailed description of nekRS development, including the discretization method, algorithmic considerations for the parallel GPU development of nekRS, as well as extensive performance studies on Summit the interested reader can consult [6].

The paper is structured as follows. First, the conservation equations and numerical method are presented. The code is validated in Section 3 against solutions obtained with well-established codes in setups of increasing complexity. The performance is subsequently studied in detail, including cross-code comparisons in Section 4.

2 Numerical method and code description

The system of equations for reacting flow considered here is based on the low Mach number formulation [11, 12]. Integrating the low Mach equations demands careful consideration to effectively couple advection with the diffusion and production terms operating on much faster time scales. The coupling strategy becomes a key determinant of the largest timestep size that preserves accuracy, stability, and computational efficiency. Different strategies have been proposed [13], each with its own advantages and disadvantages, but discussing them in detail is beyond the scope of this paper. In this work, the algorithm proposed in [14] is employed.

First, the *thermochemistry* subsystem

$$\frac{\partial Y_k}{\partial t} = -\mathbf{v} \cdot \nabla Y_k + \frac{1}{\rho}\left(-\nabla \cdot \rho Y_k \mathbf{V}_k + \dot{\omega}_k\right) \quad (1a)$$

$$\frac{\partial T}{\partial t} = -\mathbf{v} \cdot \nabla T + \frac{1}{\rho c_p}\left(\nabla \cdot \lambda \nabla T + \sum_{k=1}^{N} h_k^0 \dot{\omega}_k\right)$$
$$+ \frac{1}{\rho c_p}\left(-\nabla \cdot \rho T \sum_{k=1}^{N} c_{p,k} Y_k \mathbf{V}_k + \frac{dp_0}{dt}\right)$$
$$(1b)$$

$$\left(1 - \frac{\mathcal{R}}{c_p}\right)\frac{dp_0}{dt} = \frac{\mathcal{R}}{c_p}\left(\nabla \cdot \lambda \nabla T + \sum_{k=1}^{N} h_k^0 \dot{\omega}_k\right)$$
$$- (\nabla \cdot \mathbf{u})p_0 - \frac{\mathcal{R}}{c_p}\left(\nabla \cdot \rho T \sum_{k=1}^{N} c_{p,k} Y_k \mathbf{V}_k\right)$$
$$+ \sum_{i} \frac{\mathcal{R}T}{W_i}\left(-\nabla \cdot \rho Y_k \mathbf{V}_k + \dot{\omega}_k\right)$$
$$(1c)$$

$$p_0 = \rho \mathcal{R} T \quad (1d)$$

is solved, containing the gaseous species mass fractions Y_k and temperature T, where t is time, $\mathbf{V}_k, h_k^0, c_{p,k}, \dot{\omega}_k$ the diffusion velocity, enthalpy of formation, heat capacity at constant pressure and net production rate of species k, respectively, ρ, c_p, λ, p_0 the mixture density, heat capacity, thermal conductivity, and thermodynamic pressure, and $\mathcal{R}$ the ideal gas constant. The species diffusion velocities $\mathbf{V}_k$ are computed by a mixture-averaged transport model ignoring Soret and Dufour effects

$$\mathbf{V}_k = -\frac{D_k}{X_k}\nabla X_k + \mathbf{V}_c \quad (2)$$

with X_k being the mole fraction and D_k the mixture-average diffusivity of species k. A correction velocity $\mathbf{V}_c = -\sum_{k=1}^{N} Y_k \mathbf{V}_k$ needs to be introduced for mass conservation [15].

The system of equations (1) is integrated implicitly as a fully-coupled system of nonlinear differential equations using CVODE from the SUNDIALS package [16]. The velocity $\mathbf{v}$ in the advection terms is evaluated explicitly using a third-order polynomial extrapolation (EXT3) from previous timesteps. Within the outer (flow) timestep, CVODE selects its own timestep and may advance the thermochemical state using multiple substeps to satisfy the specified relative and absolute tolerance of the solution in time. To ensure good performance, it is important to set tolerances that are not stricter than the spatial discretization errors. This may require conducting some experiments to determine the appropriate levels. Each internal CVODE step involves Newton iterations with a generalized minimal residual (GMRES) linear solve, where the Jacobian-vector product is

approximated by difference quotients. As is typical for finite difference approximations, selecting the optimal perturbation factor can be challenging and significantly affects integrator performance. The approach outlined in [17] is adopted, which frequently performs better than the default strategy in CVODE. However, even this method is not fully robust and may require additional tuning. To accelerate the linear solve, a customized GMRES solver is used that stores the Krylov basis in reduced (FP32) precision [18]. In addition, an alternative classical Gram-Schmidt [19] with just one global synchronization per iteration is employed. Krylov-based methods, such as GMRES, often require an effective and fast-to-evaluate preconditioner to perform well, and constructing such a preconditioner is challenging. However, the time step size in a DNS is often sufficiently small, and a preconditioner is not necessary to achieve reasonable performance. The right-hand side (RHS) function, evaluated in each Newton step and linear iteration, employs highly optimized kernels for the SEM transport operators, species production rates, and thermodynamic and transport properties. For this purpose, a just-in-time code generator was developed to translate a combustion model expressed in Cantera format [20] into platform tuned source code. More details can be found in [21].

In reactive low Mach number compressible flows, flow and chemistry are tightly coupled through density and viscosity, and the non-zero divergence constraint Q_T is imposed on the velocity

$$\nabla \cdot \mathbf{v} = Q_T$$

$$Q_T := -\frac{1}{\rho}\frac{D\rho}{Dt} = \frac{1}{T}\frac{DT}{Dt} + \sum_{k=1}^{N}\frac{\overline{W}}{W_k}\frac{DY_k}{Dt} - \frac{1}{p_0}\frac{dp_0}{dt}$$

(3)

where D/Dt denotes the material derivative and $\overline{W}$ the mean molecular weight. The momentum equation for the velocity $\mathbf{v}$ is given by

$$\frac{\partial \mathbf{v}}{\partial t} = -\mathbf{v} \cdot \nabla \mathbf{v} - \frac{\nabla p_1}{\rho}$$
$$+ \frac{1}{\rho}\left[\nabla \cdot \mu \left(\underline{\nabla \mathbf{v}} + (\underline{\nabla \mathbf{v}})^T - \frac{2}{3}(\nabla \cdot \mathbf{v})\underline{\underline{\mathbf{I}}}\right)\right]$$

(4)

where μ is the dynamic viscosity, p_1 the hydrodynamic pressure, $\underline{\mathbf{I}}$ the unit tensor, and ^{T} denotes transposition. The *hydrodynamic* subsystem (Eqs. 3–4) is discretized in time using a third-order backward differentiation formula to approximate the time derivative. For the nonlinear convective term, over-integration (dealiasing) is applied in combination with the operator integrating factor
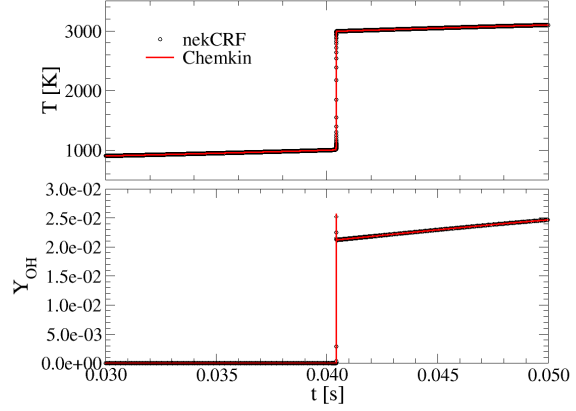

Figure 1: Comparison of the time histories of temperature and Y_{OH} (symbols) against the Chemkin solution (lines).

splitting method [22]. The resulting semi-discrete coupled system is solved in a three-step procedure [14]: First, an intermediate velocity is evaluated using the explicit contributions. Next, the hydrodynamic pressure is computed to enforce the divergence constraint (Eq. (3)). Finally, the velocity is advanced to the next (flow) timestep. Each step is efficiently treated by techniques tailored to the governing physics: classical fourth-order Runge-Kutta for the hyperbolic advection term, diagonally preconditioned conjugate gradient iteration for the viscous block coupled Helmholtz problem, and hybrid multigrid preconditioned GMRES with Chebyshev accelerated Schwarz (polynomial levels) and Jacobi (algebraic levels of coarse p=1 problem) smoothing for the variable coefficient Poisson solve [6].

3 Validation

nekRS has been validated through comparisons with nek5000 and the method of manufactured solutions. Consequently, this section focuses on the reactive low Mach number flow support introduced by *nekCRF*.

3.1 Homogeneous variable-volume autoignition

The first test case validates the integration of reaction source terms within a time-varying volume, where the thermodynamic pressure is also subject to variation. A homogeneous stoichiometric H_2-air mixture initially at 780 K and 1 atm is compressed in an adiabatic, closed domain mimicking an internal combustion engine of bore 92 mm,

stroke 86 mm operating at 500 rpm and compression ratio of 3. The piston kinematics are imposed in the movement of the lower surface in the *nekCRF* model, which tracks the mesh motion using the arbitrary Lagrangian-Eulerian formulation of [23]. The piston kinematics are also imposed on a variable-volume batch reactor model implemented and solved using Chemkin [24]. Chemical kinetics are described by the detailed reaction mechanism of Li et al. [25]. The temporal evolution of temperature and Y_{OH} of the two solutions are compared in Fig. 1, showing the autoignition resulting from the mixture compression. The relative error in the ignition delay time defined by the time of maximum Y_{OH} with respect to the Chemkin solution is 0.02%.

3.2 Laminar planar premixed flame

In contrast to the previous case, this test includes spatial gradients and incorporates both advection and diffusion terms in addition to the reactive source term. A planar premixed H_2-air flame is considered for a lean mixture (equivalence ratio $\phi = 0.6$) at $T_u = 298$ K and $p = 5$ atm. Using a Cantera-based [20] freely-propagating, premixed flat flame solver and the Li et al. [25] reaction mechanism, the laminar flame speed and thickness are found to be $S_L = 51.437$ cm/s and $\delta_f = (T_b - T_u)/\max(dT/dx) = 7.516 \times 10^{-2}$ mm, respectively. The 1-D Cantera solution is interpolated on the spectral element mesh for a domain with length equal to $L = 200\delta_f$ and height $H = \delta_f$, where the flame is placed at $x = 130\delta_f$. The velocity is set to $U_{in} = S_L$ at the inflow, where Dirichlet boundary conditions (BC) are imposed on temperature and species mass fractions. Zero-Neumann BC are considered at the outflow, while the remaining boundaries are periodic. The discretization uses a non-uniform grid with element of size $h = \delta_f$ around the flame increasing up to $h = 25\delta_f$ towards the in- and outflow; the solution on each element is approximated using $p = 7^{th}$-order polynomials. Figure 2 shows that the flame structure computed with the new solver is in very good agreement with the profiles obtained with Cantera and LAVp.

3.3 Early flame kernel development

Finally, the code is validated in a more complex application by considering a lean ($\phi = 0.4$) premixed hydrogen-air mixture initially at temperature $T_0 = 800$ K and pressure $p = 40$ bar. At these conditions, the laminar flame thickness and speed computed with Cantera and the Li et al. mechanism [25] are $\delta_f = 20$ μm and $S_L =$
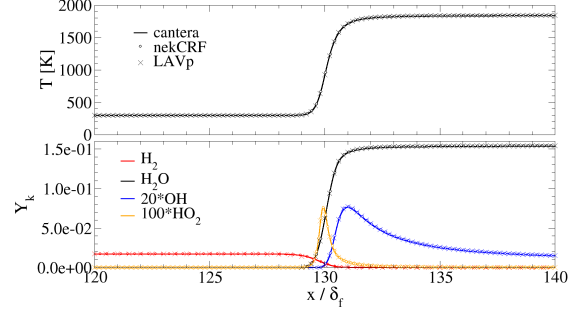

Figure 2: Comparison of temperature and selected species mass fraction profiles in the flame normal direction against Cantera (black line) and LAVp ($\times$ symbol) solutions.

44.5 cm/s, respectively, resulting in a flame time $t_f = \delta_f/S_L = 4.55 \times 10^{-5}$ s. The initial turbulent flow field is homogeneous and isotropic with integral length scale $l_I = 15.1\delta_f$ and turbulent intensity $u' = 6.6S_L$, and was constructed following the methodology proposed in [26], where the homogeneous and isotropic turbulent fields were generated with the controlled linear forcing method described in [27]. Ignition of the mixture is achieved with a centrally-located energy deposition source varying smoothly in space and time as discussed in [26]. A spherical mesh with a diameter of diameter $D = 144\delta_f$ is used (Fig. 3(a)) comprising $E = 0.5$ M spectral elements with a polynomial order $p = 7$, resulting in 2.23 billion degrees of freedom (DOFs), taking into account the 13 unknowns per grid point. The excess species N_2 is calculated as $Y_{N_2} = 1 - \sum_{i \neq N_2} Y_i$. In the central region of radius $R \leq 41\delta_f$, the elements are of size δ_f in all directions, while in the outer region they are of size $2.4\delta_f$ in the radial direction and increase from $2.8\delta_f$ to $4.8\delta_f$ in the other directions.

CVODE absolute tolerances were set to 10^{-4} of $\max\{Y_k\}$ of the planar flame, while the relative tolerance was 10^{-6} for all variables. For the linear solvers in the fluid solve, the absolute residual tolerance for pressure and velocity were 10^{-5} and 10^{-7}, respectively. The addition of heat during the initial $0.3t_f$ leads to the establishment of a flame kernel from which a propagating flame front is generated that interacts with the decaying turbulent flow field (Fig. 3(b)). Figure 4 shows good agreement in the evolution of the volumetric integral of the heat release rate, $iHRR = \int_V \sum_k h_k \dot{\omega}_k dV$, and the instantaneous isocontours of temperature with respect to LAVp for times up to $2t_f$ when the flame remains within the high resolution region.
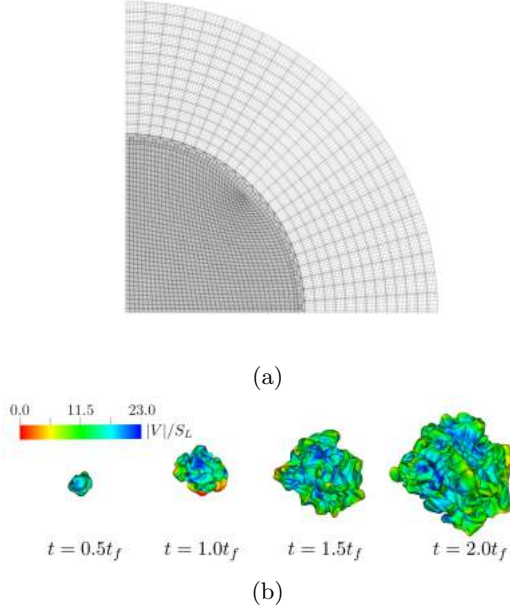
4

(a)

(b)

Figure 3: (a) Slice of a quarter of the mesh (b) flame kernels defined by the $T = 1600$ K isotherm and colored by the flow velocity magnitude at four time instants.

Figure 4: Comparison of the time histories of the integral heat release rate and instantaneous iso-contours of temperature. Insets: $T = 1600$K isotherms on an $x - z$ slice at four time instants.

4 Performance

In this section, the performance is initially assessed by comparing *nekCRF* with LAVp. Subsequently the strong scaling capabilities and compute kernels of *nekCRF* are explored. Unless specified otherwise, the performance experiments were conducted on JUWELS Booster at the Jülich Supercomputing Centre (JSC). The system consists of 936 compute nodes, each equipped with 4 NVIDIA A100 SXM4 GPUs.

It is noteworthy that the reported thermochemistry timings are approximately twice as fast compared to running CVODE with default settings and without the customized Jacobian-vector product and linear solver modifications discussed in Section 2. This speedup is primarily due to using a difference quotient perturbation factor about 100 times smaller than the default and a relaxed linear convergence safety factor of 0.5 instead of 0.05 for each Newton iteration.

4.1 Comparison to LAVp

To compare the performance, the case presented in Section 3.3 is evaluated using LAVp on the LUMI-C HPC system at the CSC IT Center for Science in Finland. It is important to recognize that the reported speedup factors are generally influenced by specific implementation details, hardware setup, and problem size. Initial microbench-
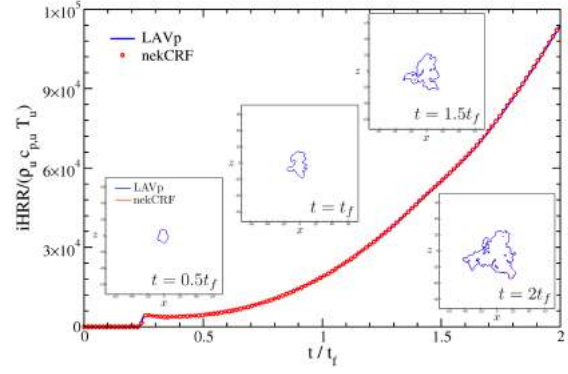
marks provide valuable insights into expected behavior, showing that reaction rates on GPUs can achieve speedups between 6x and 9x, depending on the size of the chemistry model. Other performance-relevant kernels, which are primarily limited by memory bandwidth, can achieve peak speedups of 9x. This performance difference aligns with the hardware capabilities. However, in typical production runs, the overall speedup tends to be lower because certain parts of the workload fit within the CPU's last-level cache, which provides more competitive bandwidth compared to the GPU's memory bandwidth.

Striving for a closely matching comparison, it is important to note that *nekCRF* has been more thoroughly optimized compared to LAVp, which may lead to an overestimation of performance improvements. To assess potential speedup ranges for the entire solver, multiple experiments were conducted and the results are summarized in Table 1.

use case	speedup
S_{max}	22
R_{max}	14
$R_{0.5}$	4

Table 1: Speedup factors compared to LAVp.

The maximum achievable speedup per compute unit (CU) was measured to be $S_{max} = 22$. In this scenario, both codes handle a large local problem size, which minimizes the impact of caching effects on the CPU. However, when considering the maximum throughput $R_{\max} = 1/(nCU \cdot t)$ of each solver, where nCU is the number of compute units (24 GPUs for *nekCRF* and 300 CPUs using all available cores for LAVp), the speedup

5

reduces to 14 due to caching effects. Notably, $R_{0.5}$, the speedup at 50% of $R_{\max}$, utilizing 480 GPUs on JUWELS Booster and 2000 CPUs on LUMI, offers a different perspective. This strong scaling use case demonstrates that while *nekCRF* remains faster on a GPU-based system, its competitive advantage diminishes as communication overhead becomes more significant.

4.2 Strong scaling

To stress test strong scaling, the investigation focuses on the configuration outlined in Section 3.3, employing a refined mesh with approximately doubled resolution in each spatial direction and a two times smaller outer timestep. The refinement leads to a case with $E = 4.1$ million spectral elements and 18 billion DOFs. Computations are conducted across a range of N=50 to 800 compute nodes (85% of the JSC system size), with the lower limit determined by the available GPU memory. However, the focus here is on up to 600 nodes, as the parallel efficiency η for larger node counts becomes smaller than 50%.

This scenario poses a challenge for strong scaling due to the relatively small size of the hydrogen reaction mechanism, which involves only 9 species participating in 21 reactions. Larger chemistry models typically require more local computations, leading to improved strong scaling of the thermochemistry solve. They also reduce the relative cost of the fluid solve, which is much harder to strong scale due to the smaller system size.

The solver statistics averages over the outer (fluid) timestep are reported in Table 2 including pressure iterations (pIter), velocity iterations (vIter), CVODE timesteps (cvSteps), Jacobian-vector evaluation per CVODE step (jtv/cvSteps), non-linear Newton iterations per CVODE step (nni/cvSteps), and ratio of linear to nonlinear iterations (nli/nni). The counters cvSteps and Jtv/cvSteps provide a rough measure of the overall cost of the thermochemistry solve. The ratio nni/cvSteps measures the performance of the nonlinear solver (typical values ranging from 1.1 to 1.8 according to [28]), while the ratio nli/nni measures the performance of the Krylov linear solver, and thus (indirectly) the condition number of the linear system.

The elapsed time for the integration of 200 timesteps with $dt = 10^{-3}t_f$ ($CFL \approx 3$) is shown in Fig. 5 together with its breakdown into the contributions of the velocity, pressure and thermochemistry parts. The solvers for velocity, pressure, and thermochemistry exhibit different scaling characteristics. Among them, the thermochemistry solver is the most compute-intensive,

pIter	11
vIter	34
cvSteps	4.5
jtv/cvSteps	7.4
nni/cvSteps	1.1
nli/nni	6.4

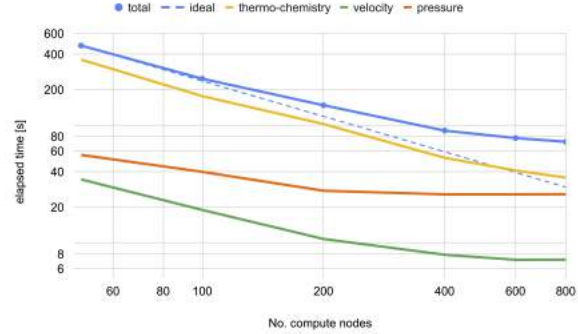Table 2: Solver statistics averaged over 200 timesteps.

Figure 5: Strong scalability of the different solver components.

with $\eta_{tc} = 0.72$ on 600 compute nodes. On the other hand, the fluid solver shows diminishing returns (poor scaling) beyond 200 nodes, specifically when the number of grid points per processing unit $n = 1.75M$ is considerably smaller than $n_{0.8}$ - the number of grid points per GPU required to achieve $\eta = 0.8$. According to [29], the value of $n_{0.8}$ was measured to be around 4-5M on ANL's Polaris, a system similar to JUWELS Booster. This results in an overall parallel efficiency of $\eta = 0.51$ on 600 nodes. It is noteworthy that in typical production runs employing a more complex chemistry model and/or higher number of grid points per GPU, the efficiency will be significantly higher.

To delve into the details of the thermochemistry solver, a time breakdown is presented in Table 3, revealing a significant influence of the embedded linear solve at each Newton step, consuming roughly 80% of the overall execution time. Besides the Jacobian-vector product evaluation (using a difference quotient of the RHS) in each GMRES iteration, the local Gram-Schmidt process accounts for up to 17% of total cost on average. The diffusion operator emerges as the most time-consuming kernel within the RHS. As expected for this small chemistry model, the computation of reaction rates and properties accounts for only a small fraction (8%) of the total time. The "other" category primarily consists of various Level 1 BLAS type operations.

To further analyze the observed parallel effi-

	N=50	%	N=600	%
total	339s		39.4s	
newton solve	64.4s	19	8.3s	21
linear solve	274.6s	81	31.1s	79
low-level components				
other	98.2s	29	10.0s	25
diffusion	95.8s	28	8.15s	21
gramSchmidt	57.2s	17	5.60s	14
MPI_Allreduce	2.60s	1	4.18s	11
MPI halo	12.8s	4	3.40s	9
localQQ^T	20.1s	6	3.30s	8
MPI pack/unpack	8.32s	2	2.31s	6
advection	25.4s	7	2.54s	6
rates	22.9s	7	2.23s	6
properties	6.32s	2	0.75s	2

Table 3: Time breakdown of the thermochemistry solve.

kernel	limiter	FOM
diffusion	GMEM bw	987 GB/s
gramSchmidt	GMEM bw	802 GB/s
rates	latency	2.4 TFLOPS
localQQ^T	GMEM bw	580 GB/s
advection	GMEM bw	935 GB/s

Table 4: Performance top 5 kernels.

ciency of the thermochemistry solve, it is useful to separate it into two factors: computational efficiency and communication efficiency.

Computational efficiency measures how well local compute kernels scale relative to peak performance and was evaluated at around 0.9 across all kernels. The efficiency of individual kernels varies depending on the kernel type. Kernels of the gather/scatter type (localQQ^T and MPI pack/unpack) exhibit less favorable scaling characteristics compared to all other kernels. These kernels tend to be short (for more details see Section 4.3), making latency a limiting factor. However, although the relative cost rises, it remains within a moderate range below 14%.

Communication efficiency (computation time to total runtime ratio) measures the overhead caused by off-device communication as the problem is scaled. The main limiter in the present case is associated with inner products (in the linear solver) necessitating global synchronization, which are known to scale poorly due to internode latency. To mitigate the impact, a Gram-Schmidt variant is employed that requires only one global synchronization step [19]. Yet, inner products to compute the initial and final residual norms in the Newton or linear solver are challenging to reduce or overlap, posing limitations on scalability. This is evident in the increasing relative MPI_Allreduce cost. The global C^0 assembly is imposed through a nearest-neighbor exchange locally (QQ^T) and across neighbor processes (MPI halo). On average, messages of size 212 kB are exchanged among 17 neighbors achieving an aggregated bidirectional bandwidth of 148 GB/s on 50 nodes (74% of theoretical node injection bandwidth 200 GB/s), which

decreases by 17% on 600 nodes. However, the off-device communication (MPI_Neighbor_alltoall) within QQ^T can be fully overlapped by local computation, allowing the effective cost to be hidden to a large extent. Consequently, the most significant factor contributing to the effective communication cost is the MPI_Allreduce operation. The measured communication efficiency was approximately 0.8.

4.3 Compute kernels

This section aims to assess the kernel performance, with a focus on the top five most time-consuming kernels of the thermochemistry solver, as summarized in Table 4. The evaluation includes identifying the dominant performance limiter and presenting a figure of merit (FOM).

The data was collected on 50 compute nodes (Tab. 3) with the local workload sufficiently high to achieve peak kernel throughputs. Kernels in the other category achieve an effective memory bandwidth close to the sustainable memory bandwidth of 1370GB/s [30] as expected for a simple streaming type. Most of the top 5 kernels are primarily limited by global memory bandwidth (GMEM bw), with other factors also constraining performance and preventing the attainment of ideal streaming-type performance. The measured effective memory bandwidth for these five most time-consuming kernels ranges from 42% to 73%. The gather/scatter type operations pose a challenge, as they exclusively operate on the element or partition surface, resulting in short kernels with irregular memory access patterns that waste memory bandwidth as the element data layout is lexicographical ordered to minimize indirect addressing in the local tensor contractions. SEM operators, such as advection and diffusion, as well as the Gram-Schmidt kernels, utilize shared resources (registers and scratch pad memory) to cache input data or intermediate results. This greatly reduces the amount of GMEM data transferred and improves the overall performance. However, it may lead to lower occupancy and, consequently, a decrease in achievable bandwidth. Optimizing the reaction rate kernel is challenging due to large

working sets, irregular execution orders, and complex data access patterns, with latency often becoming a significant constraint. Several GPU-optimized implementations have been developed, including those in [31, 32, 33, 2]. In the absence of a clear performance roofline, the floating-point operations per second (FLOPS) is used as a performance metric. Further details and performance comparisons are discussed in [21].

5 Conclusions and outlook

nekCRF was developed to enable efficient direct numerical simulations of reactive low Mach flows on accelerator-based supercomputers. The solver supports unstructured curvilinear boundary conforming meshes for accurate representation of complex geometries, finite rate chemistry, mixture-averaged transport, conjugate heat transfer, moving mesh capabilities and time-varying thermodynamic pressure variation. Validation was based on comparison with established numerical solvers and various reactive use cases. During development, the initial focus was on NVIDIA GPUs, for which computationally efficient chemistry kernels were implemented. The strong scalability of *nekCRF* was tested on up to 3600 GPUs on JUWELS Booster, showing that it scales efficiently. Additionally, cross-code performance comparisons demonstrated that *nekCRF* outperforms its predecessor, LAVp.

Ongoing efforts involve incorporating support for AMD and Intel GPUs, prioritizing performance optimization for these architectures, and ultimately testing our solver on real exascale use cases. This may prompt a reevaluation of the thermochemistry solver to enhance performance when dealing with more severe stiffness or incorporating additional multiphysics aspects like radiation, soot, or heterogeneous reactions. A viable approach could involve leveraging methodologies akin to those outlined in [34]. Moreover, mixed-precision techniques are being investigated to further accelerate the solver. Finally, evaluating against other well-established solvers will help to clarify the competitiveness of the performance.

6 Availability

At the time of writing, the code has not been made publicly available. However, after a testing phase, it will be released as open-source under the BSD license and can be accessed on GitHub: https://github.com/nekCRF.

7 CRediT authorship contribution statement

Stefan Kerkemeier: Software, Methodology, Investigation, Writing. **Christos E. Frouzakis**: Writing, Validation, Funding acquisition. **Ananias G. Tomboulides**: Review, Supervision. **Paul Fischer**: Review. **Mathis Bode**: Review, Funding acquisition, Supervision.

8 Acknowledgements

References

[1] P. Domingo and L. Vervisch. Recent developments in DNS of turbulent combustion. *Proc. Combust. Inst.*, 39(2):2055–2076, 2023.

[2] Pele Suite. https://amrex-combustion.github.io/.

[3] S.G. Kerkemeier. *Direct numerical simulation of combustion on petascale platforms. Applications to turbulent non-premixed hydrogen autoignition.* PhD thesis, ETH Zürich, 2010. Nr. 19162.

[4] A. Brambilla. *Direct numerical simulation of catalytic ignition.* PhD Thesis, ETH Zurich, 2014.

[5] B.O. Arani. *Three-dimensional DNS of turbulent flow hetero-/homogeneous combustion with detailed chemistry.* PhD thesis, ETH Zurich, 2018.

[6] P. Fischer, S. Kerkemeier, M. Min, Y.-H. Lan, M. Phillips, T. Rathnayake, E. Merzari, A. Tomboulides, A. Karakus, N. Chalmers, and T. Warburton. NekRS, a GPU-accelerated spectral element Navier–Stokes solver. *Parallel Comput.*, 114:102982, 2022.

[7] D.S. Medina, A. St-Cyr, and T. Warburton. OCCA: A unified approach to multithreading languages. *arXiv*, 1403.0968, 2014.

[8] M.O. Deville, P.F. Fischer, and E.H. Mund. *High-order methods for incompressible fluid flow*. University Press, Cambridge UK, 2002.

[9] S. Rezaeiravesh, R. Vinuesa, and P. Schlatter. On numerical uncertainties in scale-resolving simulations of canonical wall turbulence. *Comput. Fluids*, 227:105024, 2021.

[10] C.D. Cantwell, D. Moxey, A. Comerford, A. Bolis, G. Rocco, G. Mengaldo, D. De Grazia, S. Yakovlev, J.-E. Lombard, D. Ekelschot, B. Jordi, H. Xu, Y. Mohamied, C. Eskilsson, B. Nelson, P. Vos, C. Biotto, R.M. Kirby, and S.J. Sherwin. Nektar++: An open-source spectral/hp element framework. *Computer Physics Communications*, 192:205–219, 2015.

[11] B.T. Chu and L.S.G. Kovasznay. Non-linear interactions in a viscous heat-conducting compressible gas. *J. Fluid Mech.*, 3:494–514, 1958.

[12] G.R. Rehm and H.R. Baum. The equations of motion for thermally driven flows. *J. Res. National Bureau of Stanlards*, 83(3):494–514, 1978.

[13] M.S. Day A. Nonaka and J.B. Bell. A conservative, thermodynamically consistent numerical approach for low Mach number combustion. Part I: Single-level integration. *Combust. Theory Model.*, 22(1):156–184, 2018.

[14] A.G. Tomboulides, J.C.Y. Lee, and S.A. Orszag. Numerical simulation of low Mach number reactive flows. *J. Sci. Comput.*, 12:139–167, 1997.

[15] T.P. Coffee and J.M. Heimerl. Transport Algorithms for Premixed Laminar, Steady-State Flames. *Combust. Flame*, 43:273–289, 1982.

[16] A.C. Hindmarsh, P.N Brown, K.E. Grant, S.L. Lee, R. Serban, D.E. Shumaker, and C.S. Woodward. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Trans. Math. Soft. (TOMS)*, 31(3):363–396, 2005.

[17] M. Pernice and H.F. Walker. NITSOL: A Newton Iterative Solver for Nonlinear Systems. *SIAM J. Sci. Comput.*, 19(1):302–318, 1998.

[18] J.I. Aliaga, H. Anzt, T. Grützmacher, E.S. Quintana-Ortí, and A.E. Tomás. Compressed basis GMRES on high-performance graphics processing units. *Int. J. High Perf. Comput. Appl.*, 37(2):82–100, 2023.

[19] P. Ghysels, T.J. Ashby, K. Meerbergen, and W. Vanroose. Hiding Global Communication Latency in the GMRES Algorithm on Massively Parallel Machines. *SIAM Journal on Scientific Computing*, 35(1):C48–C71, 2013.

[20] D. G. Goodwin, H.K. Moffat, I. Schoegl, R.L. Speth, and B.W. Weber. Cantera: An Object-oriented Software Toolkit for Chemical Kinetics, Thermodynamics, and Transport Processes. https://www.cantera.org, 2023. Version 3.0.0.

[21] B. Danciu and C.E. Frouzakis. A code generator for reaction kinetics in combustion applications. *Comput. Phys. Comm.*, (in preparation).

[22] Y. Maday, A. T. Patera, and E. M. Rønquist. An operator-integration-factor splitting method for time-dependent problems: Application to incompressible fluid flow. *J. Sci. Comput.*, 5:263–292, 1990.

[23] L.W. Ho. *A Legendre spectral element method for simulation of incompressible unsteady viscous free-surface flows*. PhD thesis, Massachusetts Institute of Technology, 1989.

[24] R.J. Kee, F.M. Rupley, and J.A. Miller. Chemkin-II: A Fortran chemical kinetics package for the analysis of gas-phase chemical kinetics. *SAND-89-8009*, 1989.

[25] J. Li, Z. Zhao, A. Kazakov, and F.L. Dryer. An updated comprehensive kinetic model of hydrogen combustion. *Int. J. Chem. Kin.*, 36(10):566–575, 2004.

[26] T. Falkenstein, S. Kang, L. Cai, M. Bode, and H. Pitsch. DNS study of the global heat release rate during early flame kernel development under engine conditions. *Combust. Flame*, 213:455–466, 2020.

[27] M. Bassenne, J. Urzay, G. I. Park, and P. Moin. Constant-energetics physical-space forcing methods for improved convergence to homogeneous-isotropic turbulence with application to particle-laden flows. *Phys. Fluids*, 28(3):035114, 2016.

[28] A.C. Hindmarsh, R. Serban, C.J. Balos, D.J. Gardner, D.R. Reynolds, and C.S. Woodward. User Documentation for CVODE. https://sundials.readthedocs.io/en/latest/cvode, 2023. v6.6.2.

[29] M. Min, Y.-H. Lan, P. Fischer, T. Rathnayake, and J. Holmen. Nek5000/RS Performance on Advanced GPU Architectures. 2022.

[30] Yuhsiang Mike Tsai, Terry Cojean, and Hartwig Anzt. Evaluating the performance of nvidia's a100 ampere gpu for sparse linear algebra computations, 2020.

[31] M. Bauer, S. Treichler, and A. Aiken. Singe: Leveraging Warp Specialization for High Performance on GPUs. *SIGPLAN Not.*, 49(8):119–130, 2014.

[32] K. Kim, O.H. Díaz-Ibarra, H.N. Najm, J. Zádor, and C. Safta. TChem: A performance portable parallel software toolkit for complex kinetic mechanisms. *Comput. Phys. Comm.*, 285:108628, 2023.

[33] S. Barwey and V. Raman. A Neural Network-Inspired Matrix Formulation of Chemical Kinetics for Acceleration on GPUs. *Energies*, 14(9), 2021.

[34] M.S. Day A.Nonaka and J.B. Bell. A conservative, thermodynamically consistent numerical approach for low Mach number combustion. Part I: Single-level integration. *Combustion Theory and Modelling*, 22(1):156–184, 2018.