

# A scalable high-order mesh-free framework for direct numerical simulations of combustion

J. R. C. King\*

*Department of Mechanical, Aerospace and Civil Engineering,  
The University of Manchester, Manchester, M13 9PL, UK*

(Dated: October 30, 2023)

This document details the governing equations and numerical methods used in the **Scalable Unstructured Node-SET** code (aka. SUNSET code).

## I. INTRODUCTION

This document describes the **Scalable Unstructured Node-SET** code (aka. SUNSET code). The code is designed for direct numerical simulations of compressible turbulent reacting flows in complex geometries. It builds on the work in [1, 2], using the Local Anisotropic Basis Function Method (LABFM) for spatial discretisation, combined with high-order finite differences in the third dimension, and a third-order Runge-Kutta time integration.

Key features:

- Nearly arbitrary geometries in the  $x_1 - x_2$  plane. Homogeneous in the  $x_3$  direction.
- High-order ( $8^{th}$ ) discretisation in space dropping to  $4^{th}$  order at boundaries.
- $3^{rd}$  order time integration.
- NSCBC formulation for (non-reflecting) inflows and outflows, and isothermal or adiabatic walls.
- Periodic/symmetry boundaries also available.
- Spatially variable resolution, including on boundaries.
- Parallelised for MPI+OpenMP, scales well up to  $> 1000$  cores.
- Can handle any desired level of chemistry (e.g. single-step, 21 step  $H_2$ , 25 step  $CH_4$ )

### A. A comment on notation

Subscripts  $i, j$ , and  $k$  are vector indices for Einstein notation. Subscripts  $\alpha$  and  $\beta$  indicate chemical species. Subscripts  $a$  and  $b$  indicate collocation point indices. Subscripts  $l$  and  $L$  are used for polynomial coefficient indices. Subscript  $m$  is for chemical reaction number (regular  $m$  is polynomial order).

## II. GOVERNING EQUATIONS

The code solves the compressible Navier-Stokes equations in conservative form

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_k}{\partial x_k} = 0 \quad (1a)$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_k}{\partial x_k} = - \frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ki}}{\partial x_k} + \rho f_i \quad (1b)$$

$$\frac{\partial \rho E}{\partial t} + \frac{\partial \rho u_k E}{\partial x_k} = - \frac{\partial p u_k}{\partial x_k} - \frac{\partial q_k}{\partial x_k} + \frac{\partial \tau_{ki} u_i}{\partial x_k} + \rho \sum_{\alpha} Y_{\alpha} f_k (u_k - V_{\alpha,k}) \quad (1c)$$

$$\frac{\partial \rho Y_{\alpha}}{\partial t} + \frac{\partial \rho u_k Y_{\alpha}}{\partial x_k} = \dot{\omega}_{\alpha} - \frac{\partial \rho V_{\alpha,k} Y_{\alpha}}{\partial x_k} \quad \forall \alpha \in [1, N_S] \quad (1d)$$

---

\* jack.king@manchester.ac.uk

where  $\rho$  is the density,  $u_i$  the  $i$ -th component velocity,  $p$  the pressure,  $E$  the total energy,  $\tau$  the viscous stress,  $f$  a body force,  $q$  the heat flux,  $Y_\alpha$  the concentration of species  $\alpha \in [1, N_S]$ , and  $V_{\alpha,k}$  is the  $k$ -th component of the molecular diffusion velocity of species  $\alpha$ .  $\dot{\omega}_\alpha$  is the production rate of species  $\alpha$ .

The viscous stress is defined as

$$\tau_{ki} = \mu \left( \frac{\partial u_k}{\partial x_i} + \frac{\partial u_i}{\partial x_k} - \frac{2}{3} \frac{\partial u_j}{\partial x_j} \delta_{ik} \right), \quad (2)$$

where  $\mu$  is the dynamic viscosity and  $\delta_{ik}$  is the Kronecker delta. The energy is

$$E = \sum_{\alpha} h_{\alpha} Y_{\alpha} - \frac{p}{\rho} + \frac{1}{2} u_i u_i \quad (3)$$

where the sum over  $\alpha$  is over all  $\alpha \in [1, N_S]$ , and  $h_{\alpha}$  is the enthalpy of species  $\alpha$ . The molecular diffusion term in (1d) is modelled as a Fickian diffusion process with

$$-\rho V_{\alpha,k} Y_{\alpha} = \rho \hat{V}_{\alpha,k} Y_{\alpha} - Y_{\alpha} \sum_{\beta \in [1, N_S]} \rho \hat{V}_{\beta,k} Y_{\beta}, \quad (4)$$

in which  $\rho \hat{V}_{\alpha,k} Y_{\alpha}$  is obtained from the diffusion model (either constant Lewis numbers or mixture-averaged), and the final term is a correction to ensure that

$$\sum_{\alpha} Y_{\alpha} = 1. \quad (5)$$

The heat flux vector is then given by

$$q_k = -\lambda \frac{\partial T}{\partial x_k} + \sum_{\alpha} \rho V_{\alpha,k} Y_{\alpha} h_{\alpha}, \quad (6)$$

where  $\lambda$  is the thermal conductivity of the mixture. The system is closed with an ideal gas equation of state

$$\frac{p}{\rho} = R_{mix} T = \frac{R_0 T}{W_{mix}} = R_0 T \sum_{\alpha} \frac{Y_{\alpha}}{W_{\alpha}} \quad (7)$$

where  $R_{mix}$  is the gas constant for the mixture,  $R_0$  is the universal gas constant, and  $W_{\alpha}$  is the molar mass of species  $\alpha$ , and  $W_{mix}$  is the molar mass of the mixture.

Reaction rates  $\dot{\omega}_{\alpha}$  are obtained through finite rate chemical kinetics mechanisms - details/refsXX. XX Including Lindemann steps, gibbs backwards rates and third bodies. Specific heat capacities, enthalpies and gibbs functions are obtained through CHEMKIN/NASA polynomials XXrefXX.

### A. Evaluation of temperature and heat capacity

The temperature dependence of the specific heat capacity of species  $\alpha$  is given by a polynomial of order  $L$

$$c_{p,\alpha} = \sum_{l=1}^L a_{\alpha,l} T^{l-1}, \quad (8)$$

which is valid over a specified temperature range  $T \in [T_{low}, T_{high}]$ , and in which the coefficients  $a_{\alpha,l}$  are taken from CHEMKIN tables. Integration of this yields an expression for the enthalpy

$$h_{\alpha} = \sum_{l=1}^L \frac{a_{\alpha,l}}{l} T^l + a_{\alpha,L+1}, \quad (9)$$

whilst the temperature derivative of  $c_{p,\alpha}$  is

$$\frac{dc_{p,\alpha}}{dT} = \sum_{l=1}^L (l-1) a_{\alpha,l} T^{l-2}. \quad (10)$$

Noting that the specific heat capacity of species  $\alpha$  is  $R_0/W_\alpha$ , where  $R_0$  is the universal gas constant and  $W_\alpha$  is the molar mass of species  $\alpha$ , the energy can be written as

$$E = \sum_{\alpha} Y_{\alpha} \left( h_{\alpha} - R_0 \frac{T}{W_{\alpha}} \right) + \frac{1}{2} \mathbf{u} \cdot \mathbf{u}. \quad (11)$$

Substituting the polynomial for  $h_{\alpha}$  and collecting terms in powers of  $T$ , we can write

$$f(T) = C_0 + \sum_{l=1}^L C_l T^l = 0, \quad (12)$$

where the coefficients  $C$  are given by

$$C_0 = \frac{1}{2} \mathbf{u} \cdot \mathbf{u} - E + \sum_{\alpha} Y_{\alpha} a_{\alpha, L+1} \quad (13a)$$

$$C_1 = \sum_{\alpha} Y_{\alpha} a_{\alpha, 1} - R_{mix} \quad (13b)$$

$$C_l = \sum_{\alpha} Y_{\alpha} \frac{a_{\alpha, l}}{l} \quad \forall l \in [2, L]. \quad (13c)$$

The derivative of  $f$  is

$$f'(T) = \sum_{l=1}^L l C_l T^{l-1}. \quad (14)$$

Having calculated the coefficients  $C$ , we solve the nonlinear equation (12) numerically via a Newton-Raphson method to obtain the temperature  $T$ .

Note that in the case where  $a_{\alpha, 1} \neq 0$  and  $a_{\alpha, l \in [2, L]} = 0$  (i.e.  $c_{p, \alpha}$  is independent of temperature),  $f(T)$  is linear in  $T$ , and the Newton-Raphson method will converge (exactly) after a single iteration.

## B. Transport properties

Two models are implemented for transport properties. The constant Lewis number approximation, and a mixture averaged transport model.

### 1. Constant Lewis numbers

In this model, the diffusive flux is given by

$$\rho \hat{V}_{\alpha, k} Y_{\alpha} = -\rho D_{\alpha} \frac{\partial Y_{\alpha}}{\partial x_k}, \quad (15)$$

in which  $D_{\alpha}$  is the molecular diffusivity of species  $\alpha$ . The temperature dependence of transport properties is modelled by the relationship

$$\mu = \mu_{ref} \left( \frac{T}{T_{ref}} \right)^{r_T}, \quad (16)$$

where  $\mu_{ref}$  is the viscosity at the reference temperature  $T_{ref}$ , and  $r_T = 0.7$  is a constant exponent. The mixture thermal conductivity is given by

$$\lambda = \frac{\mu c_p}{Pr}, \quad (17)$$

in which  $c_p$  is the heat capacity of the mixture, and  $Pr$  is the Prandtl number, taken as constant. The molecular diffusivity of species  $\alpha$  is given by

$$D_{\alpha} = \frac{\lambda}{\rho c_p Le_{\alpha}}, \quad (18)$$

where  $Le_{\alpha}$  is the Lewis number for species  $\alpha$ , which is assumed constant for each species.

## 2. Mixture-averaged transport

In this model the diffusive flux is given by

$$\rho \hat{V}_{\alpha,k} Y_\alpha = -\rho D_\alpha \frac{\partial Y_\alpha}{\partial x_k} - \rho D_\alpha Y_\alpha \left( \frac{\partial \ln T}{\partial x_k} + \frac{\partial \ln \rho}{\partial x_k} - \frac{W_\alpha}{\rho R_0 T} \frac{\partial p}{\partial x_k} \right), \quad (19)$$

Chemkin polynomial fits for transport data [3] Review of transport models in [4]. Mixture-averaged transport for diffusion using Hirschfelder-Curtiss approximation [5]. Combination rule for viscosity and thermal conductivity following [6, 7].

Here the transport properties for individual species and species pairs are evaluated using polynomials [xx ref xx], where

$$\ln \mu_\alpha = \sum_{l=1}^L c_{\mu,\alpha,l} \left( \ln \left( \frac{T}{T_{ref}} \right) \right)^{l-1} \quad (20a)$$

$$\ln \lambda_\alpha = \sum_{l=1}^L c_{\lambda,\alpha,l} \left( \ln \left( \frac{T}{T_{ref}} \right) \right)^{l-1} \quad (20b)$$

$$\ln D_{\alpha\beta} = \ln \left( \frac{p_{ref}}{p} \right) + \sum_{lj=1}^L c_{D,\alpha\beta,l} \left( \ln \left( \frac{T}{T_{ref}} \right) \right)^{l-1}, \quad (20c)$$

in which the  $c_{...,l}$  are constants, and  $L = 4$ . [xx more details xx].

The transport properties are then obtained from combination rules as

$$\mu = \left[ \sum_{\alpha} X_{\alpha} \mu_{\alpha}^{n_{\mu}} \right]^{\frac{1}{n_{\mu}}}, \quad (21)$$

with  $n_{\mu} = 6$ ,

$$\lambda = \left[ \sum_{\alpha} X_{\alpha} \lambda_{\alpha}^{n_{\lambda}} \right]^{\frac{1}{n_{\lambda}}}, \quad (22)$$

with  $n_{\lambda} = 1/4$ , and

$$D_{\alpha} = \frac{\sum_{\beta \neq \alpha} X_{\beta} W_{\beta}}{W_{mix} \sum_{\beta \neq \alpha} \frac{X_{\beta}}{D_{\alpha\beta}}}. \quad (23)$$

Note that the molar concentration  $X_{\alpha} = Y_{\alpha} W_{mix} / W_{\alpha}$ . Soret and Dufour effects are neglected.

### C. Evaluation of reaction rates

The reaction mechanism is given by  $M$  equations written in the form

$$\sum_{\alpha} \nu'_{\alpha,m} \mathcal{M}_{\alpha} \rightarrow \sum_{\alpha} \nu''_{\alpha,m} \mathcal{M}_{\alpha} \quad \forall m \in [1, M], \quad (24)$$

where  $\mathcal{M}_{\alpha}$  symbolises the chemical symbol of species  $\alpha$ , and  $\nu'_{\alpha,m}$  and  $\nu''_{\alpha,m}$  are the molar stoichiometric coefficients of species  $\alpha$  in equation  $m$ .

The reaction rate of species  $\alpha$  is given by

$$\omega_{\alpha} = W_{\alpha} \sum_{m=1}^M \hat{\omega}_{\alpha,m} \quad (25)$$

where  $\hat{\omega}_{\alpha,m}$  is the molar production rate of species  $\alpha$  due to the  $m$ -th reaction, and  $M$  is the total number of reactions. The molar production rates are given by

$$\hat{\omega}_{\alpha,m} = (\nu''_{\alpha,m} - \nu'_{\alpha,m}) \left[ k_{f,m}(T) \prod_{\beta=1}^{N_S} \left( \frac{\rho Y_{\beta}}{W_{\beta}} \right)^{\nu'_{\beta,m}} - k_{b,m}(T) \prod_{\beta=1}^{N_S} \left( \frac{\rho Y_{\beta}}{W_{\beta}} \right)^{\nu''_{\beta,m}} \right], \quad (26)$$

in which  $k_{f,m}$  and  $k_{b,m}$  are the forward and backward (respectively) rate constants for reaction  $m$ . The forward rate constant is given by the Arrhenius expression

$$k_{f,m}(T) = A_m T^{n_m} \exp \left( -\frac{E_{a,m}}{R_0 T} \right), \quad (27)$$

with  $A_m$ ,  $n_m$  and  $E_{a,m}$  being specified constants representing the pre-exponential factor, the temperature dependence of the pre-exponential factor, and the activation energy.

### 1. Backwards rate constant

The backwards rate constant is related to the forward rate constant via

$$\ln k_{b,m} = \ln k_{f,m} + \sum_{\alpha} (\nu''_{\alpha,m} - \nu'_{\alpha,m}) \left( \frac{\hat{g}_{\alpha}}{R_0 T} + \ln \left( \frac{P_0}{R_0} \right) - \ln T \right), \quad (28)$$

in which  $P_0$  is a reference pressure, and  $\hat{g}_{\alpha}$  is the Gibbs function for species  $\alpha$ , which is given by

$$\frac{\hat{g}_{\alpha}}{W_{\alpha} T} = \frac{a_{\alpha,J+1}}{T} - a_{\alpha,1} \ln T + (a_{\alpha,1} - a_{\alpha,J+2}) - \sum_{j=2}^J \frac{a_{\alpha,j}}{j(j-1)} T^{j-1}. \quad (29)$$

### 2. Third bodies

For a step  $m$  including third bodies  $\mathcal{M}$ , reaction rates are scaled by the third-body concentration  $c_{M,m}$ , which is given by

$$c_{M,m} = \sum_{\alpha} \eta_{\alpha,m} \frac{\rho Y_{\alpha}}{W_{\alpha}}, \quad (30)$$

and  $\eta_{\alpha,m}$  are the third-body efficiencies for species  $\alpha$  for step  $m$ .

### 3. Lindemann forms

Where the Arrhenius form is insufficient to describe the reaction rate, a pressure (partial pressure/concentration) dependency is introduced, and the forward reaction rate takes the Lindemann form:

$$k_{f,m} = k_{\infty,m} \left( \frac{P_r}{1 + P_r} \right) F_m, \quad (31)$$

where  $F_m$  is the Troe fall-off rate, and a constant for steps with Lindemann form.  $P_r$  is a reduced pressure given by

$$P_r = \frac{k_{0,m}}{k_{\infty,m}}, \quad (32)$$

$k_{0,m}$  is the Arrhenius rate of the reaction, and  $k_{\infty,m}$  is a second Arrhenius rate. Hence for Lindemann steps, in addition to the three Arrhenius coefficients, the constant  $F_m$  is prescribed, along with the pre-exponential factor, the temperature dependence of the pre-exponential factor and the activation energy for  $k_{\infty,m}$ . Note that after evaluation of the rate constants for Lindemann steps, the third-body concentration of the step is reset to unity.

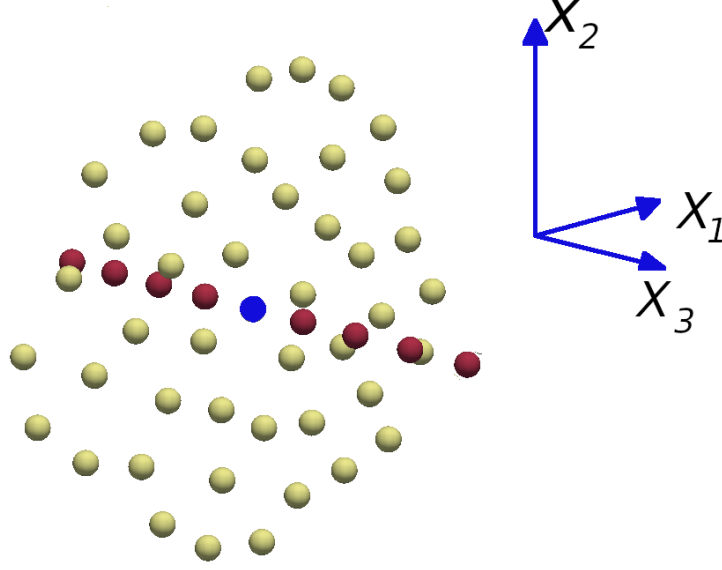


FIG. 1. A typical stencil for a three-dimensional simulation. The nodes shown are those in the stencil of the central node in blue. The unstructured nodes in yellow lie in the same plane (same value of  $x_3$ ) as the central node. The nodes in red are those in the finite difference stencil for the  $x_3$  direction.

### III. NUMERICAL IMPLEMENTATION

The spatial discretisation is based on the Local Anisotropic Basis Function Method (LABFM) in the first two dimensions, and high-order finite differences in the third dimension. LABFM has been detailed and extensively analysed in [1, 2], and we refer the reader to these works for a complete description.

The domain is discretised in the  $x_1 - x_2$  plane with an unstructured point cloud of  $N_{12}$  nodes. The discretisation in the third dimension is uniform: there are  $N_3$  copies of each node in the  $x_1 - x_2$  plane, equispaced along the  $x_3$  axis. Hence, there are a total of  $N = N_{12}N_3$  nodes in the domain. Each node  $a \in [1, N]$  has position  $x_{a,i}$ , a distribution lengthscale  $s_a$ , and a computational stencil lengthscale  $h_a$ .  $s_a$  is the average node spacing around node  $a$ , and is analogous to the grid spacing in a finite difference scheme. The spacing in  $x_3$  is denoted  $s_3$ , and is uniform throughout the domain, whilst  $s_a$  can vary as a function of  $x_1$  and  $x_2$ . Each node holds the conservative variables  $\rho_a$ ,  $\rho u_{i,a}$ ,  $\rho E_a$  and  $\rho Y_{\alpha,a}$ . The governing equations are solved on the set of  $N$  nodes.

The difference between properties at two nodes is denoted  $(\cdot)_{ba} = (\cdot)_b - (\cdot)_a$ . The computational stencil for each node  $a$  is denoted  $\mathcal{N}_a$ , and is constructed to contain all nodes  $b$  in the same  $x_1 - x_2$  plane such that  $|x_{i,ba}x_{i,ba}| \leq 4h_a^2$ , (with the repeated subscript  $i$  implying summation), alongside all the nodes with  $x_{1,ba} = x_{2,ba}$  required to build a finite difference stencil in the  $x_3$  dimension. Figure 1 shows a typical computational stencil.

Figure 2 shows an example discretisation in the present framework for an aerofoil. Wall, inflow and outflow boundaries are discretised with a locally structured node distribution, whilst the remaining space is filled with a quasi-uniform point cloud. XX. The resolution is non-uniform, and in the present example  $s$  varies by a factor of 10 through the domain. The unstructured nodes are created to fill the domain using a variation of the propagating front algorithm of [8], following [2].

#### A. Evaluation of spatial derivatives

Spatial derivative operators take the form

$$L_a^d(\cdot) = \sum_b (\cdot)_{ba} w_{ba}^d, \quad (33)$$

where  $d$  identifies the derivative of interest (e.g.  $d = x_1$  indicates the operator  $L^d$  approximates  $\partial(\cdot)/\partial x_1$ , and  $d = x_i x_i$  (summation implied) indicates the operator approximates the Laplacian), and  $w_{ba}^d$  are a set of inter-node weights for the operator. Note that the construction of (33) takes the same form for both LABFM-based and finite

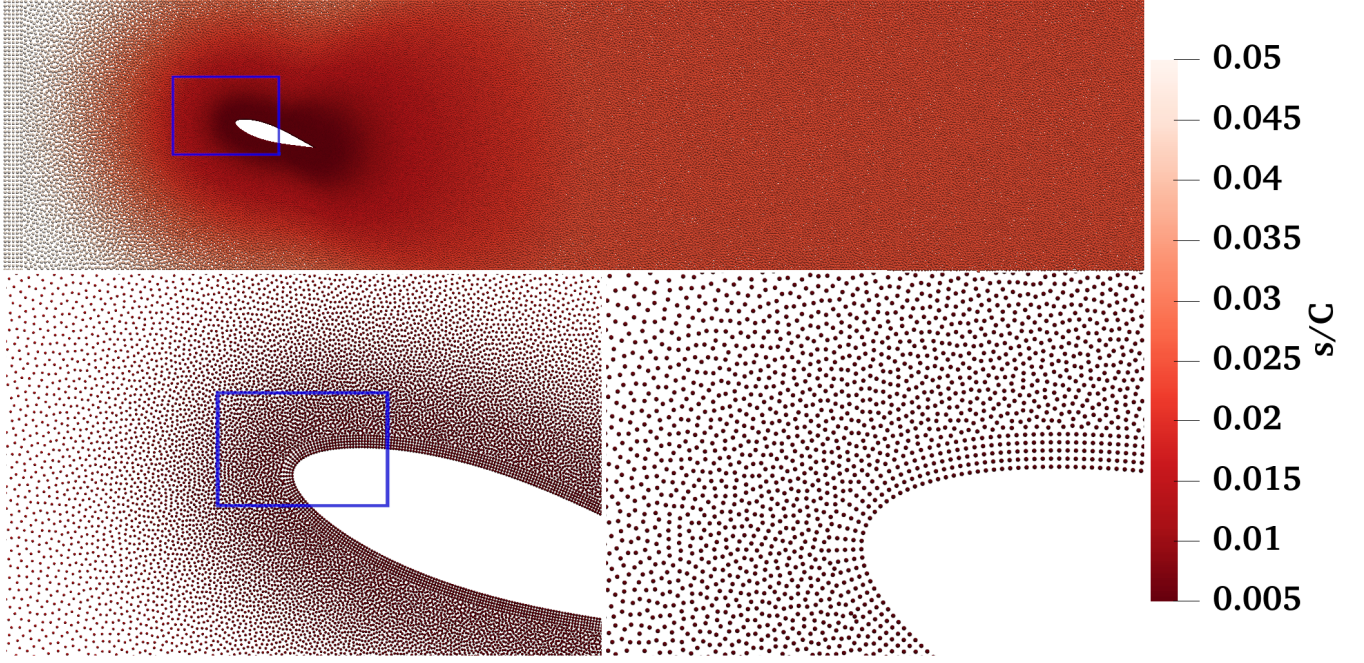


FIG. 2. Example of an unstructured node distribution around an aerofoil. The colour represents the local resolution length-scale  $s$  normalised by the aerofoil chord length  $C$ .

difference-based derivative approximations. For derivatives in the  $x_3$  dimension, the weights  $w_{ba}^d$  are simply central finite difference weights of order  $m$ . For derivatives in the  $x_1 - x_2$  plane, LABFM is used to set the  $w_{ba}^d$ , which is constructed from a weighted sum of anisotropic basis functions centred on node  $a$ . The basis functions used herein are formed from bi-variate Hermite polynomials multiplied by a radial basis function, and the basis function weights are set such that the operator achieves a specified polynomial consistency  $m$ . Calculation of these weights is done as a pre-processing step, and described in detail in [2].

The order of the spatial discretisation can be specified between  $m = 4$  and  $m = 10$ , and although there is capability for this to be spatially (or temporally) varying, in this work we set  $m = 8$  uniformly away from boundaries (except where explicitly stated in our investigations of the effects of changing  $m$ ). This provides a good compromise between accuracy and computational cost. At non-periodic boundaries, the consistency of the LABFM reconstruction is smoothly reduced to  $m = 4$ . The stencil scale  $h_a$  is initialised to  $h_a = 2.7s_a$  in the bulk of the domain, and  $h_a = 2.4s_a$  near boundaries (choices informed through experience as being large enough to ensure stability). In bulk of the domain, the stencil scale is then reduced following the optimisation procedure described in [2]. This has the effect of both reducing computational costs, and increasing the resolving power of LABFM, yielding the smallest stencil size  $h_a$  for which the discretisation remains stable.

The evaluation of derivatives using (33) constitutes the bulk of the computational cost, and it is hence desirable only use it where essential, and where possible to use analytic expressions to relate derivatives of secondary properties to those evaluated via (33). First derivatives of  $\rho$ ,  $u_i$ ,  $\rho E$  and  $Y_\alpha$ , alongside the temperature  $T$  and pressure  $p$ , are evaluated directly using (33). Convective terms are then constructed from combinations thereof: for example

$$\frac{\partial \rho u_i E}{\partial x_i} = u_i \frac{\partial \rho E}{\partial x_i} + \rho E \frac{\partial u_i}{\partial x_i}. \quad (34)$$

Laplacians and direct second (i.e.  $\partial^2(\cdot)/\partial x_i \partial x_i$ ) derivatives are also evaluated with (33). To avoid the explicit evaluation of cross-second derivatives of velocity, the viscous stress divergence is evaluated as

$$\frac{\partial \tau_{ki}}{\partial x_k} = \frac{1}{\mu} \frac{\partial \mu}{\partial x_k} \tau_{ki} + \mu \left( \frac{\partial^2 u_i}{\partial x_k \partial x_k} + \frac{1}{3} \frac{\partial^2 u_k}{\partial x_i \partial x_k} \right), \quad (35)$$

where the final term is the gradient of the velocity divergence, which is calculated through two iterations of first derivative evaluations with (33). The benefit of this approach is that it reduces the stencil size, and provides a significant reduction in the complexity associated with parallelising the scheme. Gradients of secondary properties

are evaluated from analytic expressions in terms of gradients of primary properties. The gradient of the enthalpy of species  $\alpha$  is given by

$$\frac{\partial h_\alpha}{\partial x_k} = \frac{dh_\alpha}{dT} \frac{\partial T}{\partial x_k} = c_{p,\alpha} \frac{\partial T}{\partial x_k}, \quad (36)$$

and the gradient of the mixture specific heat capacity is

$$\frac{\partial c_p}{\partial x_k} = \sum_\alpha \left[ Y_\alpha \frac{dc_{p,\alpha}}{dT} \frac{\partial T}{\partial x_k} + c_{p,\alpha} \frac{\partial Y_\alpha}{\partial x_k} \right]. \quad (37)$$

When constant Lewis numbers are assumed, additional relations are available to evaluate the spatial derivatives of transport properties:

$$\frac{\partial \mu}{\partial x_k} = \frac{r_T \mu}{T} \frac{\partial T}{\partial x_k} \quad (38)$$

$$\frac{\partial \lambda}{\partial x_k} = \frac{r_T \lambda}{T} \frac{\partial T}{\partial x_k} + \frac{\lambda}{c_p} \frac{\partial c_p}{\partial x_k} \quad (39)$$

$$\frac{\partial \rho D_\alpha}{\partial x_k} = \frac{r_T D_\alpha}{T} \frac{\partial T}{\partial x_k}. \quad (40)$$

For the mixture averaged model, gradients of  $\mu$ ,  $\lambda$  and  $\rho D_\alpha$  are obtained directly using (33).

### B. Temporal integration and filtering

The governing equations are integrated in time using a four-step third-order, low-storage explicit Runge-Kutta scheme, with embedded second order error estimation, denoted RK3(2)4[2R+]C in the classification system of Kennedy et al. [9]. The value of the time step is set adaptively using a PID controller to keep time-integration errors below a specified threshold (set to  $10^{-4}$  in throughout the present work).

The solution is de-aliased after each time-step by high-order spatial filtering using the filters described in [2] combined with a finite difference filter following [10]. The combined filter is defined by

$$F(\phi) = -(-1)^{m/2} \left(\frac{s_3}{2}\right)^m \frac{\partial^m}{\partial x_3^m} \left[ (1 + \kappa_m \nabla_{x_1 x_2}^m) \phi \right], \quad (41)$$

where  $\kappa_m$  is the filter coefficient, specific to each node, calculated as in [2], and  $\nabla_{x_1 x_2}^m$  is the two-dimensional Laplacian raised to the power of  $m/2$  (as used in the filtering in [2]), defined in index notation as

$$\nabla_{x_1 x_2}^m = \sum_{n=0}^m \binom{m}{n} \frac{\partial^m}{\partial x_1^{m-n} \partial x_2^n}. \quad (42)$$

The term within the square brackets in (41) is the two-dimensional filter in the  $x_1 - x_2$  plane, whilst the operator outside the square brackets is the filter in the  $x_3$  dimension. The filters are applied sequentially in this manner because the filter does not include any cross-components in  $x_1 x_3$  or  $x_2 x_3$ . This approach follows those used in finite difference codes, where the dimensions are filtered sequentially. With the LABFM filter, the  $x_1$  and  $x_2$  dimensions are filtered simultaneously, because the filter includes cross-derivative terms.

For density, momenta and energy, we simply apply the filter as defined by (41), such that the filtered variables passed to the next time step are

$$\widetilde{\phi} = F(\phi) \quad \text{for } \phi = \rho, \rho u_i, \rho E \quad (43)$$

As with the molecular diffusion terms, the filtering procedure does not guarantee satisfaction of the compatibility criteria (5). Hence, we introduce a new correction to the filtering procedure for the species mass fractions, which takes inspiration from the the diffusion correction velocity. We modify the procedure with the addition of a filter correction term  $F_c$  such that

$$\widetilde{\rho Y_\alpha} = F(\rho Y_\alpha) + F_c, \quad (44)$$



TABLE I. Schemes for derivatives near boundaries.

Node	Boundary normal derivative	Boundary tangential derivative (in $x_1 - x_2$ plane)	Second derivatives	Filtering operator
Boundary	One-sided FD	1D-LABFM	FD + 1D-LABFM	FD + 1D-LABFM
Row 1	Asymmetric FD	1D-LABFM	LABFM ( $m = 4$ )	LABFM ( $m = 4$ )
Row 2	LABFM ( $m = 4$ )	LABFM ( $m = 4$ )	LABFM ( $m = 4$ )	LABFM ( $m = 4$ )
Row 3	LABFM ( $m = 6$ )	LABFM ( $m = 6$ )	LABFM ( $m = 6$ )	LABFM ( $m = 6$ )
Row 4	LABFM ( $m = 6$ )	LABFM ( $m = 6$ )	LABFM ( $m = 6$ )	LABFM ( $m = 6$ )

where

$$F_c = \frac{1}{N_S} \left[ \tilde{\rho} - \sum_{\alpha} F(\rho Y_{\alpha}) \right] \quad (45)$$

This approach ensures that

$$\sum_{\alpha} \tilde{Y}_{\alpha} = 1 \quad (46)$$

by construction. This method results in the maximum error in (46) remaining below machine precision. In all our simulations, the maximum error in (46) is below  $3 \times 10^{-18}$ .

### C. Boundaries

The boundary framework largely follows that described in detail for isothermal flows in [2], extended to accommodate thermal and reacting flows. The discretisations on boundaries are detailed in Table I. At wall, inflow and outflow boundaries, the node-set is structured, with a band of 5 rows of uniformly spaced nodes. Wall boundaries may be curved, and the resolution along boundaries need not be uniform. On boundaries, normal derivatives are calculated using one-sided 5-point finite difference stencils, whilst tangential (in the  $x_1 - x_2$  plane) derivatives are obtained with a one-dimensional formulation of LABFM detailed in [2] with  $m = 6$ . For the first row in from the boundaries (Row 1), first derivatives are evaluated using asymmetric finite differences (normal) and one-dimensional LABFM (tangential), whilst second derivatives and filters are obtained from two-dimensional LABFM with  $m = 4$ . For nodes in row 2, all operators use two-dimensional LABFM with  $m = 4$ . For rows 3 and 4, all operators use two-dimensional LABFM with  $m = 6$ . Note that at all nodes including boundary nodes, all operators in the third dimension use central finite differences.

Symmetry and periodic boundary conditions are available, alongside no-slip wall boundaries (isothermal or adiabatic), inflows (both partially non-reflecting and “hard”), non-reflecting outflows. Walls, inflows and outflows are imposed through the Navier Stokes characteristic boundary condition (NSCBC) formalism, following [11]. The improved outflow conditions of [12] has been implemented, but was found to be less stable to the passage of flames than the method of [11].

### D. Parallelisation

The code is accelerated with a shared-distributed framework, using both OpenMP and MPI. A non-uniform block-structured three-dimensional domain decomposition scheme is used, with the physical sizes of subdomains set such that each MPI rank contains the same number of nodes. The decomposition takes as input the desired number of processors in each dimension  $N_{P1}$ ,  $N_{P2}$ , and  $N_{P3}$ , giving a total of  $N_P = N_{P1}N_{P2}N_{P3}$  processors. The allocation of nodes to processors then forms a preprocessing step, following the algorithm:

1. Generate the node discretisation in the  $x_1 - x_2$  plane.
2. Re-arrange the nodes in order of increasing  $x_1$ .
3. Split the discretisation into  $N_{P1}$  slice each containing the same number of nodes.

4. For each slice

- (a) Re-arrange the nodes in order of increasing  $x_2$ .
- (b) Split the slice into  $N_{P2}$  blocks each containing the same number of nodes.
- (c) For each slice, create the  $x_3$  discretisation by creating  $N_3$  copies equispaced along the  $x_3$  axis, and split into  $N_{P3}$  processor domains.

This approach results in each processor containing the same number of nodes, and load balancing is typically achieved to within 5%.

### Appendix A: Equations written out in full, in Einstein notation

The following is valid for constant Lewis numbers. For mixtured-averaged diffusion, there are even more terms in the energy and species equations.

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0 \quad (\text{A1})$$

$$\frac{\partial \rho u_i}{\partial t} + \rho u_j \frac{\partial u_i}{\partial x_j} + u_i \frac{\partial \rho u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \mu \left[ \frac{\partial^2 u_i}{\partial x_j \partial x_j} - \frac{1}{3} \frac{\partial}{\partial x_i} \left( \frac{\partial u_j}{\partial x_j} \right) \right] + \left[ \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \right] \frac{\partial \mu}{\partial x_j} + \rho f_i \quad (\text{A2})$$

$$\frac{\partial \rho E}{\partial t} + u_i \frac{\partial \rho E}{\partial x_i} = -\rho E \frac{\partial u_i}{\partial x_i} - u_i \frac{\partial p}{\partial x_i} - p \frac{\partial u_i}{\partial x_i} \quad (\text{A3})$$

$$+ u_i \left\{ \frac{\mu}{\rho} \left[ \frac{\partial^2 u_i}{\partial x_j \partial x_j} - \frac{1}{3} \frac{\partial}{\partial x_i} \left( \frac{\partial u_j}{\partial x_j} \right) \right] + \frac{1}{\rho} \left[ \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \right] \frac{\partial \mu}{\partial x_j} \right\} \quad (\text{A4})$$

$$+ \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \right) \frac{\partial u_i}{\partial x_j} \quad (\text{A5})$$

$$+ \lambda \frac{\partial^2 T}{\partial x_k \partial x_k} + \frac{\partial \lambda}{\partial x_k} \frac{\partial T}{\partial x_k} \quad (\text{A6})$$

$$+ \sum_{\alpha} \left\{ \rho D_{\alpha} \frac{\partial Y_{\alpha}}{\partial x_i} \frac{\partial h_{\alpha}}{\partial x_i} + h_{\alpha} \left[ \rho D_{\alpha} \frac{\partial^2 Y_{\alpha}}{\partial x_i \partial x_i} + \frac{\partial \rho D_{\alpha}}{\partial x_i} \frac{\partial Y_{\alpha}}{\partial x_i} \right] \right\} \quad (\text{A7})$$

$$- \sum_{\alpha} Y_{\alpha} h_{\alpha} \sum_{\beta} \left[ \rho D_{\beta} \frac{\partial^2 Y_{\beta}}{\partial x_i \partial x_i} + \frac{\partial \rho D_{\beta}}{\partial x_i} \frac{\partial Y_{\beta}}{\partial x_i} \right] \quad (\text{A8})$$

$$- \sum_{\alpha} \left[ h_{\alpha} \frac{\partial Y_{\alpha}}{\partial x_i} + Y_{\alpha} \frac{\partial h_{\alpha}}{\partial x_i} \right] \sum_{\beta} \rho D_{\beta} \frac{\partial Y_{\beta}}{\partial x_i} \quad (\text{A9})$$

$$+ \rho \sum_{\alpha} Y_{\alpha} f_i (u_i - V_{\alpha,i}) \quad (\text{A10})$$

and

$$\frac{\partial \rho Y_{\alpha}}{\partial t} + \rho u_i \frac{\partial Y_{\alpha}}{\partial x_i} + Y_{\alpha} \frac{\partial \rho u_i}{\partial x_i} = \dot{\omega}_{\alpha} + \left[ \rho D_{\alpha} \frac{\partial^2 Y_{\alpha}}{\partial x_i \partial x_i} + \frac{\partial \rho D_{\alpha}}{\partial x_i} \frac{\partial Y_{\alpha}}{\partial x_i} \right] \quad (\text{A11})$$

$$- Y_{\alpha} \sum_{\beta} \left[ \rho D_{\beta} \frac{\partial^2 Y_{\beta}}{\partial x_i \partial x_i} + \frac{\partial \rho D_{\beta}}{\partial x_i} \frac{\partial Y_{\beta}}{\partial x_i} \right] - \frac{\partial Y_{\alpha}}{\partial x_i} \sum_{\beta} \rho D_{\beta} \frac{\partial Y_{\beta}}{\partial x_i} \quad (\text{A12})$$

Note, I haven't expanded out  $V_{\alpha,i}$  in the body force term in the energy equation in full.

- [2] J. King, S. Lind, High-order simulations of isothermal flows using the local anisotropic basis function method (labfm), *Journal of Computational Physics* 449 (2022) 110760. doi:doi:<https://doi.org/10.1016/j.jcp.2021.110760>.
- [3] R. J. Kee, G. Dixon-Lewis, J. Warnatz, M. E. Coltrin, J. A. Miller, A Fortran computer code package for the evaluation of gas-phase multicomponent transport properties, Technical Report SAND86-8246, Sandia National Laboratories, 1986.
- [4] E. Giacomazzi, F. R. Picchia, N. Arcidiacono, A review of chemical diffusion: Criticism and limits of simplified methods for diffusion coefficient calculation, *Combustion Theory and Modelling* 12 (2008) 135–158. doi:doi:10.1080/13647830701550370.
- [5] J. O. Hirschfelder, C. F. Curtiss, R. B. Bird, *Molecular theory of gases and liquids*, 1964.
- [6] A. Ern, V. Giovangigli, *Multicomponent transport algorithms*, Springer Science & Business Media, 1994.
- [7] A. Ern, V. Giovangigli, Fast and accurate multicomponent transport property evaluation, *Journal of Computational Physics* 120 (1995) 105–116. doi:doi:10.1006/jcph.1995.1151.
- [8] B. Fornberg, N. Flyer, Fast generation of 2-D node distributions for mesh-free PDE discretizations, *Computers & Mathematics with Applications* 69 (2015) 531 – 544. doi:doi:10.1016/j.camwa.2015.01.009.
- [9] C. A. Kennedy, M. H. Carpenter, R. Lewis, Low-storage, explicit Runge–Kutta schemes for the compressible Navier–Stokes equations, *Applied Numerical Mathematics* 35 (2000) 177 – 219. doi:doi:10.1016/S0168-9274(99)00141-5.
- [10] C. A. Kennedy, M. H. Carpenter, Several new numerical methods for compressible shear-layer simulations, *Applied Numerical Mathematics* 14 (1994) 397 – 433. doi:doi:10.1016/0168-9274(94)00004-2.
- [11] J. C. Sutherland, C. A. Kennedy, Improved boundary conditions for viscous, reacting, compressible flows, *Journal of Computational Physics* 191 (2003) 502 – 524. doi:doi:10.1016/S0021-9991(03)00328-0.
- [12] C. S. Yoo, H. G. Im, Characteristic boundary conditions for simulations of compressible reacting flows with multi-dimensional, viscous and reaction effects, *Combustion Theory and Modelling* 11 (2007) 259–286. doi:doi:10.1080/13647830600898995.