Case C1.1: Transonic Ringleb Flow

Giorgio Giangaspero[*], Edwin van der Weide[†], Magnus Svärd[‡], Mark H. Carpenter[§] and Ken Mattsson[¶]

I. Discretization, iterative method and hardware

See the appendices.

II. Case summary

We submit two sets of results (in Tecplot format): one set obtained imposing an inviscid (slip) wall boundary condition on walls, and one set obtained imposing the exact solution on walls. This test case has been computed on four structured grids containing 129×65, 257×129, 513×257 and 1025×513 vertices respectively. The finest grid is obtained by applying elliptic smoothing to an algebraically created grid. Both a second and a fourth order discretization of the Laplace equation is used for this smoothing. It turned out that the resulting grids produced virtually indistinguishable results. The point distribution on the boundary is uniform and the coarse grids are obtained by deleting recursively every other grid line from the fine grid. The coarser grids are obtained by deleting every other grid line from the finer grid. The coarsest grid used is shown in figure 1.

When using the slip wall boundary condition, it was found that at least a 3^{rd}(on the coarsest grid at least a 4^{th}) order discretization must be used on all but the finest grid to obtain a stable solution. For the 2^{nd}order scheme the computation of the metric terms is not accurate enough to represent the boundary correctly. A typical convergence history for the 5^{th} order scheme on the (257×129) grid is shown in figure 2, which clearly shows the convergence to machine zero. As a restart is carried out from the exact solution, no grid sequencing needs to be used here. The total amount of CPU time required is 345 seconds, which corresponds to 61.7 times the cost of the Taubench.

As the exact solution for this case is isentropic, the deviation from the exact entropy can be used as a measure of the error, i.e.

$$\text{error} = \frac{\exp\left(\mathrm{s}\right) - \exp\left(\mathrm{s}^{\text{exact}}\right)}{\exp\left(\mathrm{s}^{\text{exact}}\right)}, \quad \exp\left(\mathrm{s}\right) = \frac{\mathrm{p}}{\rho^{\gamma}}. \tag{1}$$

The L^2 norm of this quantity is defined as

$$L^2_{\text{error}} = \left[\frac{\displaystyle\sum_{i=1}^{N} \text{error}_{\mathrm{i}}^2 \left|\mathrm{J_i}\right|}{\displaystyle\sum_{i=1}^{N} |J_i|}\right]^{1/2}, \tag{2}$$

$|J_i|$ being the determinant of the metric Jacobian. Contour plots of the Mach number and entropy error, defined in equation (1), for the fine grid are shown in figures 3 and 4 respectively. The entropy error as a function of grid size is plotted in figure 5, while in figure 6 the plot of the entropy error against the work units is shown.

[*]Department of Mechanical Engineering, University of Twente, the Netherlands, e-mail: g.giangaspero@utwente.nl
[†]Department of Mechanical Engineering, University of Twente, the Netherlands, e-mail: e.t.a.vanderweide@utwente.nl
[‡]Department of Mathematics, University of Bergen, Norway, e-mail: Magnus.Svard@math.uib.no
[§]NASA Langley Research Center, Hamption, VA, e-mail: mark.h.carpenter@nasa.gov
[¶]Uppsala University, Uppsala, Sweden, e-mail: ken.mattsson@it.uu.se

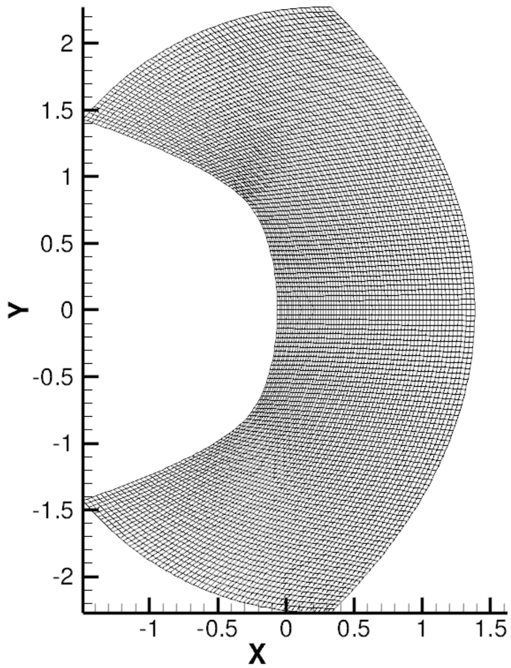


Figure 1. 129×65 **grid for test case C1.1. This grid is obtained from the finest grid by applying a regular coarsening three times.**

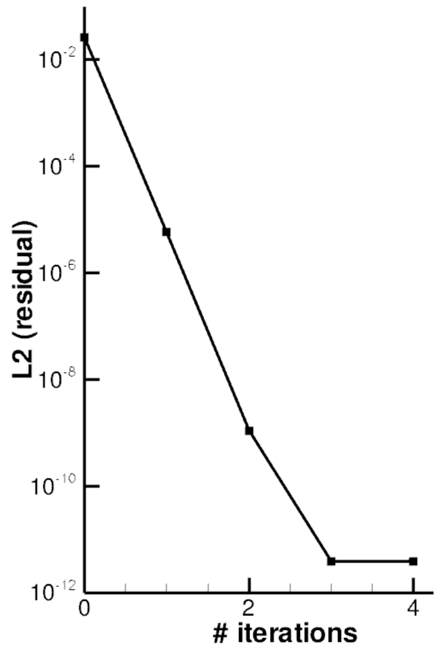


Figure 2. Convergence history for the 5^{th} **order scheme on the** (257×129) **grid for test case C1.1. The total amount of CPU time needed is 61.7 times the cost of the Taubench.**

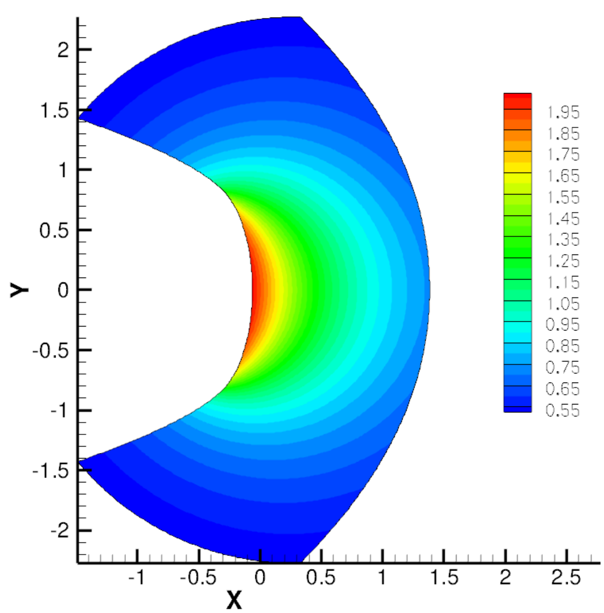


Figure 3. Contour plot of the Mach number for the 5^{th} **order scheme on the grid** (257×129) **grid for test case C1.1.**

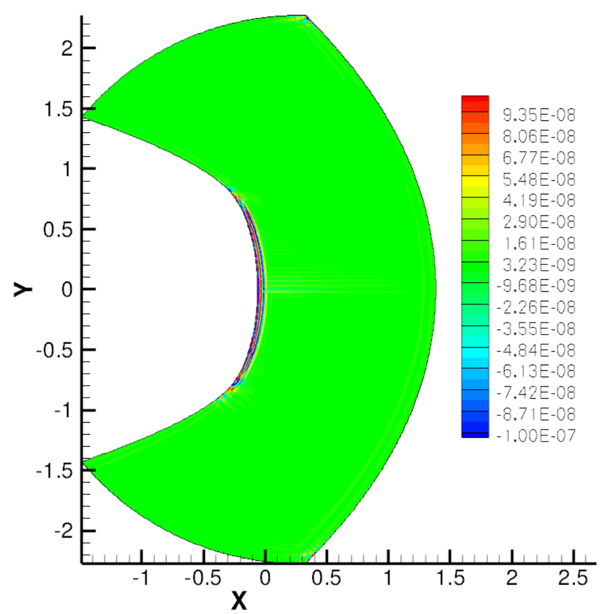
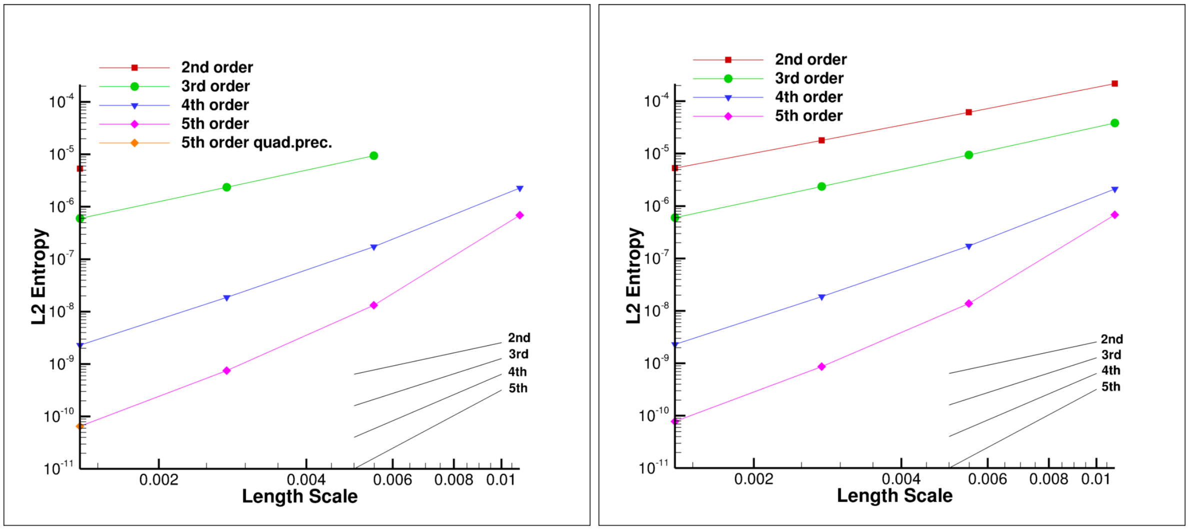


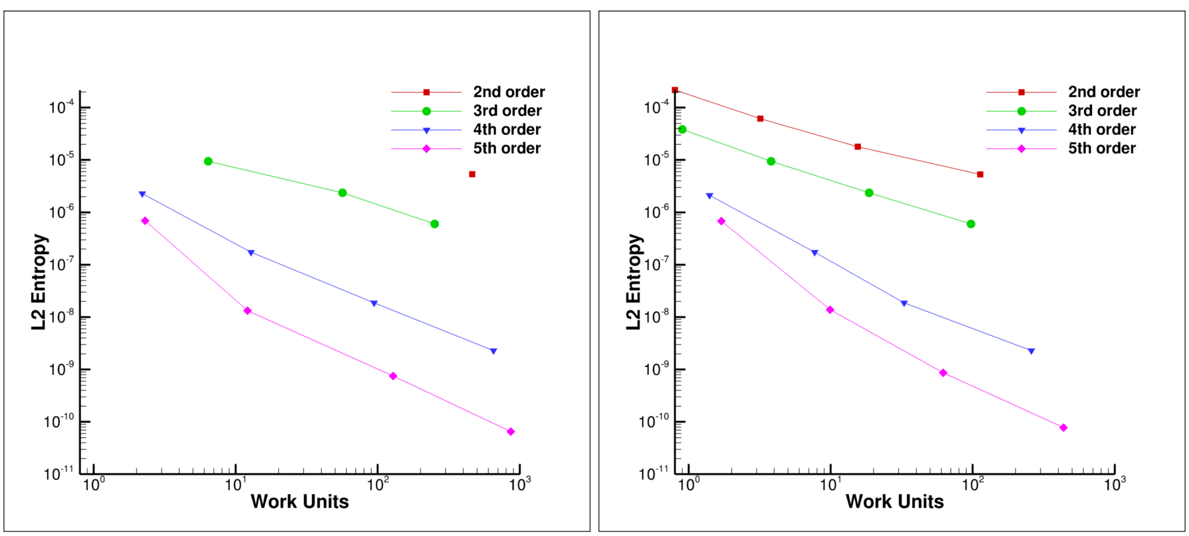
Figure 4. Contour plot of the entropy error for the 5^{th} **order scheme on the** (257×129) **grid for test case C1.1.**



(a) Inviscid wall boundary conditions

(b) Exact solution prescribed at walls

Figure 5. Ringleb flow: L^2 norm of the entropy error vs grid size. The slopes corresponding to 2^{nd} to 5^{th} order accuracy are also shown.



(a) Inviscid wall boundary conditions

(b) Exact solution prescribed at walls

Figure 6. Ringleb flow: L^2 entropy error vs work units.

Using inviscid wall boundary conditions does not lead to a decrease in accuracy compared to prescribing the exact solution on the walls. As a matter of fact the former results are slightly more accurate. However, when the exact solution at boundaries is imposed it requires less computational work to compute a solution than when inviscid wall boundary conditions are used. This is caused by the fact that the nonlinear system can be solved more easily when the exact solution is prescribed on all boundaries. Furthermore, for the finer grids the convergence rate decreases below the design accuracy of the scheme. In an attempt to assess whether or not this discrepancy is caused by finite arithmetic, we carried out a quadruple precision run for the Ringleb flow on the finest grid (1025×513), with the 5^{th}order scheme and using inviscid wall boundary condition at walls. It converged almost 18 orders of magnitude and the final residuals are $O(1e-24)$. Note that also the grid has been computed with quadruple precision. The quadruple precision computation did not bring any improvement in the entropy error and on the convergence rates. The cause of the drop in convergence rate is still under investigation.

A. Background information for the SBP-SAT scheme

As is well-known, stability of a numerical scheme is a key property for a robust and accurate numerical solution. Proving stability for high-order finite-difference schemes on bounded domains is a highly non-trivial task. One successful way to obtain stability proofs is to employ so-called Summation-by-Parts (SBP) schemes with Simultaneous Approximation Terms (SAT) for imposing boundary conditions. With a simple example, we will briefly describe how stability proofs can be obtained.

Consider the scalar advection equation,

$$\begin{aligned} &u_t + au_x = 0, \quad 0 < x < 1, \quad 0 < t \leq T \\ &a^+ u(0,t) = a^+ g_l(t) \\ &a^- u(1,t) = a^- g_r(t) \end{aligned} \tag{3}$$

where $a^+ = \max(a, 0)$ and $a^- = \min(a, 0)$. Furthermore, we augment the equation with initial data $u(x, 0) = f(x)$, bounded in L^2. To demonstrate well-posedness, we employ the energy method.

$$\begin{aligned} &\|u\|_t^2 + a \int_0^1 uu_x \, dx = 0 \\ &\|u\|_t^2 \leq au^2(0,t) - au^2(1,t) \leq a^+ g_l(t)^2 - a^- g_r(t)^2 \end{aligned} \tag{4}$$

Integrating in time gives the bound

$$\|u(\cdot, T)\| \leq \|f\| + a^+ \int_0^T g_l(t)^2 \, dt - a^- \int_0^T g_r(t)^2 \, dt. \tag{5}$$

For linear PDEs, such a bound is sufficient to prove well-posedness.

Next, we turn to the SBP-SAT semi-discretization of (3). To this end, we introduce the computational grid, $x_i = ih$, $i \in \{0, 1, 2, ..., N\}$ and $h > 0$ is the grid spacing. For the moment, we keep time continuous. With each grid point x_i, we associate a value $v_i(t)$, and define a grid function $v(t) = (v_0, v_1, v_2, ...)^T$. The SBP difference operator, D is a matrix with the following properties: $D = P^{-1}Q$ where P and Q are two matrices; $P = P^T > 0$ and $Q + Q^T = B = \text{diag}(-1, 0, ..., 0, 1)$. The matrix P can be used to define a weighted l^2 equivalent norm as $\|v\|^2 = v^T P v$. We will also need the vectors $e_0 = (1, 0, 0..., 0)^T$ and $e_N = (0, ..., 0, 1)^T$.

Let w denote a smooth function and define a grid function $\bar{w} = (w(x_0), ..., w(x_N))^T$ and $\bar{w}_x = (w_x(x_0), ..., w(x_N))^T$. It turns out that the SBP property precludes the accuracy of D to be uniform in space. We have

$$D\bar{w} = \bar{w}_x + \bar{T}$$

where $\bar{T}$ is the truncation error. In general, it takes the form,

$$\bar{T}^T = (\mathcal{O}(h^s), ..., \mathcal{O}(h^s), \mathcal{O}(h^p), ..., \mathcal{O}(h^p), \mathcal{O}(h^s), ..., \mathcal{O}(h^s)). \tag{6}$$

where $s < p$ and the lower accuracy is confined to a few (finite) number of points close to the boundary. SBP operators exist with various orders of accuracy, [1]. In particular, if P is a diagonal matrix, there are

SBP operators with p even and $p \leq 8$, and $s = p/2$. If P is allowed to have off-diagonal elements for a few points near the boundary $s = p - 1$ can be achieved.

Using the SBP operators, we now define a semi-discrete scheme for (3).

$$v_t + aDv = \sigma_l a^+ P^{-1} e_0 (v_0 - g_l(t)) + a^- \sigma_r P^{-1} e_N (v_N - g_r(t))$$

The right-hand side are the SAT:s, which impose the boundary conditions weakly. (Originally proposed in [2].) $\sigma_{l,r}$ are two scalar parameters, to be determined by the stability analysis. Multiplying by $2v^T P$, we obtain

$$\|v\|_t^2 - a(v_0^2 - v_N^2) = 2\sigma_l a^+ v_0 (v_0 - g_l(t)) + 2a^- \sigma_r v_N (v_N - g_r(t)) \tag{7}$$

For stability, it is sufficient to obtain a bound with $g_{l,r} = 0$. In that case, it is easy to see that we must require $\sigma_l \leq -1/2$ and $\sigma_r \geq 1/2$ to obtain a bounded growth of $\|v\|$. More generally, allowing boundary data to be inhomogeneous when deriving a bound leads to *strong stability.* (See [3]. The benefit of proving strong stability as opposed to stability is that less regularity in the boundary data is required.) For strong stability, it can be shown that $\sigma_{l,r}$ must satisfy $\sigma_l < -1/2$ and $\sigma_r < 1/2$, i.e., strict inequalities. As an example, the choice $\sigma_l = -1, \sigma_r = 1$ leads to

$$\|v\|_t^2 - a(v_0^2 - v_N^2) = -2a^+ v_0 (v_0 - g_l(t)) + 2a^- v_N (v_N - g_r(t))$$

or

$$\|v\|_t^2 \leq -a^+ (v_0 - g)^2 + a^+ g_l(t)^2 + a^- (v_N - g_r(t)^2) - a^- g^2 \tag{8}$$

If $v_0 = g_l, v_N = g_r$, (8) is the same as (4), but this is not the case and the additional terms add a small damping to the boundary. Upon integration of (8) in time, an estimate corresponding to (5) is obtained. We also remark that the SAT terms are accurate as they do not contribute to a truncation error in the scheme. Furthermore, semi-discrete stability guarantees stability of the fully discrete problem obtained by employing Runge-Kutta schemes in time, [4].

The above example, demonstrates the general procedure for obtaining energy estimates for an SBP-SAT scheme. Naturally, for systems of PDEs, in 3-D with stretched and curvilinear multi-block grids, and with additional parabolic terms, the algebra for proving stability becomes more involved. However, the resulting schemes are still fairly straightforward to use. For the linearized Euler and Navier-Stokes equations, semi-discrete energy estimates have been derived. (See [5–7] and references therein.) Different boundary types, including far-field, walls and grid block interfaces are included in the theory. For flows with smooth solutions, linear stability implies convergence as the grid size vanishes. (See [8].)

B. Code description

Both a general code and specialized codes for some of the test cases (used in the 1$^{\text{st}}$ and 2$^{\text{nd}}$ high order workshop, see [9]) are available. The general code is a 3D code that can handle multiblock grids and can run on (massively) parallel platforms. For load balancing reasons the blocks are split during runtime in an arbitrary number of sub-blocks with a halo treatment of the newly created interfaces, such that the results are identical to the sequential algorithm.

The specialized codes assume a single block 2D grid and do not have parallel capabilities, hence they are relatively easy to modify for testing purposes. Due to the fact that these codes can only be used for one specific test case and the fact that the general purpose code can only handle 3D problems, the efficiency of the specialized codes is quite a bit higher than the general purpose code.

The discretization schemes used are finite difference SBP-SAT schemes, see section A, of order 2 to 5. Thanks to the energy stability property of these schemes no or a significantly reduced amount of artificial dissipation is needed compared to schemes which do not posses this (or a similar) property. This leads to a higher accuracy of the numerical solutions.

For the steady test cases the set of nonlinear algebraic equations is solved using the nonlinear solver library of PETSc [10]. This library requires the Jacobian matrix of the spatial residual, which is computed via dual numbers [11] and appropriate coloring of the vertices of the grid, for which the PETSc routines are used. Initial guesses are obtained via grid sequencing, where appropriate. The solution of the linear systems needed by PETSc's nonlinear solution algorithm is obtained by Block ILU preconditioned GMRES.

Implicit time integration schemes of the ESDIRK type [12] are available, for which the resulting nonlinear systems are solved using a slightly adapted version of the steady state algorithm explained above. However, for the unsteady test cases considered, the Euler vortex and the Taylor-Green vortex, the time steps needed for accuracy are relatively small compared to the stability limit of explicit time integration schemes and therefore the explicit schemes are better suited for these cases. The available explicit schemes are the classical 4^{th} order Runge Kutta scheme (RK4, [13]) and TVD Runge Kutta schemes up till 3^{rd} order [14]. As the maximum CFL number of the RK4 scheme is significantly higher than the CFL number of the TVD Runge Kutta schemes, the RK4 scheme is used for the unsteady test cases mentioned above.

For the post processing standard commercially available software, such as Tecplot, and open-source software, such as Gnuplot, are used. Grid adaption has not been carried out.

C. Machines description

The results for the easy test cases have been obtained on a Linux work station running Ubuntu 10.04 with an Intel i7-2600 CPU running at 3.4 GHz, with 8 Mb of cache. The machine contains 16 Gb of RAM memory with an equivalent amount of swap. Running the Taubench on this machine led to a CPU time of 5.59 seconds (average over 4 runs).

The difficult test cases were run on up to 512 processors on the LISA machine of SARA, the Dutch Supercomputing Center and Hexagon, the Cray XE6 machine of the University of Bergen. Running the Taubench on these machines led to a CPU time of 10.3 and 10.8 seconds respectively (average over 4 runs).

References

[1] Strand, B., "Summation by Parts for Finite Difference Approximations for d/d x," *J. Comput. Phys.*, Vol. 110, 1994.

[2] Carpenter, M. H., Gottlieb, D., and Abarbanel, S., "Time-stable boundary conditions for finite-difference schemes solving hyperbolic systems: Methodology and application to high-order compact schemes," *J. Comput. Phys.*, Vol. 111(2), 1994.

[3] Gustafsson, B., Kreiss, H.-O., and Oliger, J., *Time dependent problems and difference methods*, John Wiley & Sons, Inc., 1995.

[4] Kreiss, H.-O. and Wu, L., "On the stability definition of difference approximations for the initial boundary value problem," *Applied Numerical Mathematics*, Vol. 12, 1993, pp. 213–227.

[5] Svärd, M., Carpenter, M., and Nordström, J., "A stable high-order finite difference scheme for the compressible Navier-Stokes equations, far-field boundary conditions," *Journal of Computational Physics*, Vol. 225, 2007, pp. 1020–1038.

[6] Svärd, M. and Nordström, J., "A stable high-order finite difference scheme for the compressible Navier-Stokes equations, no-slip wall boundary conditions," *J. Comput. Phys.*, Vol. 227, 2008, pp. 4805–4824.

[7] Nordström, J., Gong, J., van der Weide, E., and Svärd, M., "A stable and conservative high order multi-block method for the compressible Navier-Stokes equations," *J. Comput. Phys.*, Vol. 228, 2009, pp. 9020–9035.

[8] Strang, G., "Accurate partial difference methods II. Non-linear problems," *Num. Math.*, Vol. 6, 1964, pp. 37–46.

[9] van der Weide, E., Giangaspero, G., and Svärd, M., "Efficiency Benchmarking of an Energy Stable High-Order Finite Difference Discretization," *AIAA Journal*, 2015, doi: 10.2514/1.J053500.

[10] Balay, S., Brown, J., Buschelman, K., Eijkhout, V., Gropp, W., Kaushik, D., Knepley, M., McInnes, L. C., Smith, B., and H.Zhang, "PETSc Users Manual, Revision 3.2," Tech. rep., Argonne National Laboratory, 2011.

[11] Fike, J. A., Jongsma, S., Alonso, J., and v.d. Weide, E., "Optimization with Gradient and Hessian Information Calculated Using Hyper-Dual Numbers," *AIAA paper 2011-3807*, 2011.

[12] Kennedy, C. and Carpenter, M. H., "Additive Runge-Kutta schemes for convection-diffusion-reactio equations," *Applied Numerical Mathematics*, Vol. 44, 2003, pp. 139–181.

[13] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, 3rd ed., 2007.

[14] Gottlieb, S., Shu, C. W., and Tadmor, E., "High order time discretizations with strong stability property," *SIAM Review*, Vol. 43, 2001, pp. 89 – 112.