Quasi-a priori truncation error estimation in the DGSEM

**Gonzalo Rubio · François Fraysse · David A. Kopriva ·
Eusebio Valero**

Abstract In this paper we show how to accurately perform a quasi-*a priori* estimation of the truncation error of steady-state solutions computed by a discontinuous Galerkin spectral element method (DGSEM). We estimate the spatial truncation error using the τ-estimation procedure. While most works in the literature rely on fully time-converged solutions on grids with different spacing to perform the estimation, we use non time-converged solutions on one grid with different polynomial orders. The quasi-*a priori* approach estimates the error while the residual of the time-iterative method is not negligible. Furthermore, the method permits one to decouple the surface and the volume contributions of the truncation error, and provides information about the anisotropy of the solution as well as its rate of convergence in polynomial order. First, we focus on the analysis of one dimensional scalar conservation laws to examine the accuracy of the estimate. Then, we extend the analysis to two dimensional problems. We demonstrate that this quasi-*a priori* approach yields a spectrally accurate estimate of the truncation error.

1 Introduction

Discontinuous Galerkin (DG) methods were first introduced by Reed and Hill [37] to solve the neutron transport equation. They have emerged in recent years as an efficient and flexible method to solve convection dominated problems [11, 12]. The discontinuous Galerkin Spectral Element Method (DGSEM) [4, 24, 27, 28, 31] can be seen as a Spectral Element Method (SEM) [7] where the continuity requirement across element boundaries is relaxed, or as a high order Finite Volume (FV) [15] method with a compact stencil. As in a usual FV method, the Riemann Solver [46] stabilizes the solution. However in this case higher accuracy may be achieved by increasing the orde r of the approximation, N, as well as by reducing the size of the elements, h. The DGSEM is used in a wide range of applications such as compressible flows [5, 35], electromagnetics and optics [1, 13, 14, 29], heat transfer [32], aeroacoustics [9, 36, 42, 43], meteorology [22, 23, 38], and geophysics [16, 17].

This work estimates numerical errors in the solutions of the DGSEM. Error estimates have been extensively used in numerical simulation [40] as they provide valuable information about the quality of the solution [33, 39], and are an integral part of every mesh adaptation algorithm [30, 49]. The largest and most difficult error to estimate is the spatial discretization error [34], which is the difference between the exact and the numerical solution. Moreover, the spatial truncation error is defined as the difference between the discrete partial differential operator and the exact partial differential operator both applied to the steady, exact solution of the problem, as follows

---

G. Rubio (✉), F. Fraysse, and E. Valero
E.T.S.I.Aeronáuticos, Universidad Politécnica de Madrid, Ciudad Universitaria, E-28040 Madrid, Spain
E-mail: g.rubio@upm.es

D.A. Kopriva
Department of Mathematics, The Florida State University, Tallahassee, FL 32312, USA

$$\tau^N = \mathcal{R}^N\left(u\right) - \mathcal{R}\left(u\right). \tag{1.1}$$

Both the discretization and the truncation error are related through the Discretization Error Transport Equation (DETE) [40], where the truncation error acts as a local source for the discretization error. This relationship makes the truncation error especially well suited to act as a sensor for a mesh adaptation algorithm [44], [10], showing the regions where the mesh should be adapted to reduce the error. Moreover, an accurate estimate of the truncation error also allows for an increase in the order of the scheme via a procedure known as τ-extrapolation [19].

The truncation error may be estimated by means of the τ-estimation method of Brandt [6]. This method relies on the evaluation of the discrete PDE operator on a hierarchy (fine to coarse) of meshes. The original works of Berger [2], Bernert [3] and Fulton [20] posed the fundamentals of the method and studied the conditions on the restriction operators for transfers from fine to coarse and coarse to fine grids, mainly for finite difference uniform meshes. Syrakos and Goulas [45] successfully implemented τ-estimation for finite volume (FV) discretizations of the incompressible Navier-Stokes equations. Fraysse *et al.* [18] have extended those previous analyses to FV discretizations on any kind of meshes, with an interesting extension to non-converged solutions [19]. Recently, Rubio *et al.* [41] extended this methodology to high-order methods using a spectral collocation method. It was shown in [41] that some of the fundamental assumptions about error tendency well established for low-order methods are no longer valid in high-order schemes, making necessary a complete revision of the error behavior before redefining the algorithm. To facilitate this task, pure spectral Chebyshev collocation schemes were considered in that analysis, limiting their application to simple geometries.

In this paper, we extend the application of the τ-estimation method to the DGSEM. Weak formulation, multidomain discretization and the discontinuous formulation will introduce additional features to the accurate definition and estimation of the error. From a practical point of view, we consider the development of a method to perform a quasi-*a priori* estimation of the truncation error using the quasi-*a priori* idea exposed by Fraysse *et al.*, namely by employing a solution where the difference from steady state is not negligible. Iterative errors appear in standard iterative procedures used to obtain the solution of systems of Partial Differential Equations (PDE), e.g. Navier Stokes equations, like Richardson iteration or Newton's method. The solution is considered converged or steady when the iterative errors are below a certain threshold, normally close to machine roundoff. Obtaining the truncation error in advance, or before the solution is totally converged, would give computational savings while providing valuable information to perform adaptation in advance.

The rest of the paper is organized as follows: in Sect. 2 and for completeness, the mathematical formulation of the DGSEM and the definition of the errors are revisited. In Sect. 3, the truncation error in the DGSEM is analyzed. This analysis results in the separation of the truncation error into volume and surface contributions, which is a particularity of the extension to the DGSEM and could be interesting for applications of mesh adaptation algorithms. Furthermore, the quasi-*a priori* τ-estimation method is extended to the DGSEM. This extension includes the decoupling between the volume and the surface contributions to the error. In Sect. 4, three algorithms to efficiently estimate the truncation error are developed. The algorithms permit the extraction of information about the anisotropy and rate of convergence of the problem. Furthermore, the cost of the algorithms is analyzed. Finally, in Sect. 5, numerical experiments are carried out to support the estimation procedure. These experiments include one and two dimensional examples of the linear and nonlinear advection equation.

2 Mathematical foundations

In Sect. 2.1 we review the mathematical fundamentals of the DGSEM. Then, in Sect. 2.2, we define the errors that play a role in the definition of the truncation error or in its estimation by means of τ-estimation.

2.1 DGSEM

Discontinuous Galerkin (DG) methods were originally developed to solve conservation laws,

$$u_t + \nabla \cdot \mathbf{f} = 0. \tag{2.1}$$

The DGSEM [4,28] solves (2.1) in general three-dimensional geometries. In this paper, the approximation is restricted to at most two dimensional problems. The region under consideration is divided into non-overlapping quadrilateral elements, denoted by Ω_k. Each element in the subdivided domain is mapped individually onto the unit square by an isoparametric transformation. The mapping of each element onto the unit square is described by $\mathbf{x} = \mathbf{r}(\xi, \eta)$. Inside each element the solution is approximated by a series of orthogonal polynomials of degree N (P_N) [7]. A basis for this space is the set of Lagrange interpolating polynomials, $\ell_j(\xi)$, $j = 0, \ldots, N$, which can be written as

$$\ell_j(\xi) = \prod_{\substack{i=0 \\ i \neq j}}^{N} \frac{(\xi - \xi_i)}{(\xi_j - \xi_i)}. \tag{2.2}$$

The nodal points, ξ_i, which will become the grid points of the scheme, are chosen to be the nodes of the Legendre-Gauss quadrature. Multiple space dimensions are spanned by tensor products of these polynomials, so that we write $P_{N,N} = P_N \times P_N$. For simplicity of exposition only, we will take the same polynomial order in each direction, though this is not required in practice. As a matter of fact, we will use different polynomial orders in each direction in other sections of this paper. Finally, we simplify the derivation by considering that no mapping is done, i.e. the elements are already reference elements. For a complete derivation see [28].

In two space dimensions, the spectral element method approximates the solution and the fluxes element-by-element by the polynomials

$$u^N(\xi, \eta) = \sum_{\mu, \nu=0}^{N} u_{\mu,\nu}^N \phi_{\mu,\nu}, \quad \mathbf{f}^N(\xi, \eta) = \sum_{\mu, \nu=0}^{N} \mathbf{f}_{\mu,\nu}^N \phi_{\mu,\nu}, \tag{2.3}$$

where $\phi_{\mu,\nu} = \ell_\mu(\xi)\ell_\nu(\eta)$. The nodal (grid point) values of the fluxes are computed from the grid point values of the solution, i.e. $\mathbf{f}_{\mu,\nu}^N = \mathbf{f}\left(u_{\mu,\nu}^N\right)$. Note that $u_{\mu,\nu}^N$ is not the nodal value of $u(\xi, \eta)$, but the result of solving the discretized PDE. Therefore we distinguish

$$u^N(\xi, \eta) = \sum_{\mu, \nu=0}^{N} u_{\mu,\nu}^N \phi_{\mu,\nu}, \quad \text{and} \quad I_N u(\xi, \eta) = \sum_{\mu, \nu=0}^{N} u_{\mu,\nu} \phi_{\mu,\nu}. \tag{2.4}$$

The former is the solution of the discretized PDE while the latter is the spectral interpolation of the exact solution of the PDE. The same applies to the fluxes

$$\mathbf{f}^N(\xi, \eta) = \sum_{\mu, \nu=0}^{N} \mathbf{f}_{\mu,\nu}^N \phi_{\mu,\nu}, \quad \text{and} \quad I_N \mathbf{f}(\xi, \eta) = \sum_{\mu, \nu=0}^{N} \mathbf{f}_{\mu,\nu} \phi_{\mu,\nu}. \tag{2.5}$$

As the method is Galerkin, the residual is required to be orthogonal to the approximation space locally within an element. Thus,

$$\left(u_t^N, \phi_{i,j}\right) + \left(\nabla \cdot \mathbf{f}^N, \phi_{i,j}\right) = 0 \quad i, j = 0, 1, \ldots, N \tag{2.6}$$

where $(u, v) = \int_{-1}^{1} uv \, dx$ represents the usual L^2 inner product. Integration of (2.6) by parts gives

$$\left(u_t^N, \phi_{i,j}\right) + \sum_{e \in \partial\Omega} \int_e \mathbf{f}^N \cdot \mathbf{n}\phi_{i,j} dS - \left(\mathbf{f}^N, \nabla\phi_{i,j}\right) = 0 \quad i, j = 0, 1, \ldots, N, \tag{2.7}$$

where $\partial\Omega$ represents the boundary of the element and the summation is extended over the edges e of $\partial\Omega$.

To solve these equations, the integrals are replaced by Legendre-Gauss quadratures, which in two dimensional rectangular domains have the property

$$\int_{-1}^{1} \int_{-1}^{1} v(\xi, \eta) \, d\xi d\eta = \sum_{i,j=0}^{N} v(\xi_i, \eta_j) w_i w_j \quad \forall v \in P_{2N+1, 2N+1}. \tag{2.8}$$

This replacement is exact provided that the element sides are straight. If the sides are curved a quadrature error is incurred in the evaluation of $\left(u_t^N, \phi_{i,j}\right)$ and in the evaluation of the fluxes. Finally, substituting

(2.3) into (2.7), taking into account (2.8) and the discrete orthogonality of the Lagrange interpolating polynomials yields [7]

$$u_{t_{i,j}}^N w_i w_j + \sum_{e \in \partial \Omega} \int_e^N \mathbf{f}^N \cdot \mathbf{n} \phi_{i,j} dS - \sum_{\mu,\nu=0}^N \mathbf{f}_{\mu,\nu}^N \cdot \nabla \phi_{i,j} w_\mu w_\nu = 0 \quad i,j = 0,1,\ldots,N. \tag{2.9}$$

In (2.9), $\sum_{e \in \partial \Omega} \int_e^N \mathbf{f}^N \cdot \mathbf{n} \phi_{i,j} dS$ is the sum of all the integrals over all the edges of the element approximated by quadrature. The boundary term can be written as follows,

$$\sum_{e \in \partial \Omega} \int_e^N \mathbf{f}^N \cdot \mathbf{n} \phi_{i,j} dS = \mathbf{f}^N(1,\eta_j) \phi_{i,j}(1,\eta_j) w_j - \mathbf{f}^N(-1,\eta_j) \phi_{i,j}(-1,\eta_j) w_j +$$
$$+ \mathbf{f}^N(\xi_i,1) \phi_{i,j}(\xi_i,1) w_i - \mathbf{f}^N(\xi_i,-1) \phi_{i,j}(\xi_i,-1) w_i. \tag{2.10}$$

It can be seen that due to the discrete orthogonality of the Lagrange interpolating polynomials, the mass matrix is diagonal, so no inversion is necessary.

The discretization requires the evaluation of the fluxes along the element edges. Since the grid point values of the solutions are not defined on the edges, the edge values come from evaluating the polynomial representation of the solution. The interpolants from each side differ at the element faces. The difference is resolved by solving a Riemann problem for the flux. Thus,

$$u_{t_{i,j}}^N w_i w_j + \sum_{e \in \partial \Omega} \int_e^N \mathbf{f}^{N*} \cdot \mathbf{n} \phi_{i,j} dS - \sum_{\mu,\nu=0}^N \mathbf{f}_{\mu,\nu}^N \cdot \nabla \phi_{i,j} w_\mu w_\nu = 0 \quad i,j = 0,1,\ldots,N, \tag{2.11}$$

where $\mathbf{f}^{N*}$ is the result of solving the Riemann problem.

Finally, we introduce a couple of definitions. First we define the discrete partial differential operator for each element,

$$\mathcal{R}^N(u) = \sum_{e \in \partial \Omega} \int_e^N I_N \mathbf{f}^{N*} \cdot \mathbf{n} \phi_{i,j} dS - \sum_{\mu,\nu=0}^N \mathbf{f}_{\mu,\nu} \cdot \nabla \phi_{i,j} w_\mu w_\nu \quad i,j = 0,1,\ldots,N. \tag{2.12}$$

Secondly we define the *isolated* discrete partial differential operator as,

$$\hat{\mathcal{R}}^N(u) = \sum_{e \in \partial \Omega} \int_e^N I_N \mathbf{f}^N \cdot \mathbf{n} \phi_{i,j} dS - \sum_{\mu,\nu=0}^N \mathbf{f}_{\mu,\nu} \cdot \nabla \phi_{i,j} w_\mu w_\nu \quad i,j = 0,1,\ldots,N. \tag{2.13}$$

As can be seen, the difference between (2.12) and (2.13) is the use of the solution of the Riemann problem, $I_N \mathbf{f}^{N*}$, as numerical flux. While in the former the values at the interfaces of the element are substituted by the Riemann flux, $I_N \mathbf{f}^{N*}$, in the latter the flux is calculated using the interior values $I_N \mathbf{f}^N$. From a practical point of view, this change deactivates the communication between the elements and also between the elements and the boundaries.

A remark should be made on the definitions of the discrete partial differential operator, $\mathcal{R}^N(u)$, and the isolated discrete partial differential operator, $\hat{\mathcal{R}}^N(u)$. Both operators can be decomposed in a sum of an *homogeneous* operator that is a function of u, and an independent term that accounts for the source terms and the value of the boundary conditions,

$$\mathcal{R}^N(u) = \tilde{\mathcal{R}}^N(u) + S^N,$$
$$\hat{\mathcal{R}}^N(u) = \hat{\tilde{\mathcal{R}}}^N(u) + \hat{S}^N. \tag{2.14}$$

As a result of (2.14), we have

$$\mathcal{R}^N(u+v) = \tilde{\mathcal{R}}^N(u) + \tilde{\mathcal{R}}^N(v) + S^N = \mathcal{R}^N(u) + \tilde{\mathcal{R}}^N(v), \tag{2.15}$$

for linear operators and,

$$\mathcal{R}^N(u+v) = \tilde{\mathcal{R}}^N(u) + \left.\frac{\partial \tilde{\mathcal{R}}^N}{\partial u}\right|_u v + S^N + \mathcal{O}\left(v^2\right) = \mathcal{R}^N(u) + \left.\frac{\partial \mathcal{R}^N}{\partial u}\right|_u v + \mathcal{O}\left(v^2\right), \tag{2.16}$$

4

for nonlinear operators. A similar result can be obtained for the isolated discrete partial differential operator. In the following, we will assume that $\tilde{\mathcal{R}}^N$ in (2.15) and $\left.\frac{\partial \mathcal{R}^N}{\partial u}\right|_u$ in (2.16) are invertible. This assumption is reasonable as they are the spatial discretization and the linearization of the spatial discretization of a well posed PDE problem (2.1).

The τ-estimation procedure interpolates the solution from a fine grid to a coarse grid. Since the DGSEM works with the values of polynomial expansions from a set of nodes, the interpolant from order P to N is

$$I_P^N f^P (\xi_i, \eta_j) = \sum_{\mu,\nu=0}^{P} f(\xi_\mu, \eta_\nu) \phi_{\mu,\nu}(\xi_i, \eta_j), \quad i,j = 0, \ldots, N, \tag{2.17}$$

where (ξ_i, η_j) are the $(N+1) \times (N+1)$ Gauss-Legendre nodal points of order N and (ξ_μ, η_ν) the $(P+1) \times (P+1)$ Gauss-Legendre nodal points of order P. $I_P^N f^P$ is the polynomial of order N whose values in the Gauss-Legendre nodes of order N match f^P. To apply the discrete operator $\mathcal{R}^N$ to a solution of different order u^P it is necessary to evaluate this solution at the Gauss-Legendre nodes of order N, i.e. to interpolate to the coarse grid. For compactness, the notation in this work omits the interpolant

$$\mathcal{R}^N u^P = \mathcal{R}^N I_P^N u^P. \tag{2.18}$$

2.2 Error definitions

In this section we define the errors that play a role in the definition of the truncation error or in its estimation by means of τ-estimation, specifically, discretization, interpolation, iteration error and the truncation error itself.

– *Discretization error*
  The discretization (solution) error is the difference between the exact solution of the problem, $u(x)$, and the approximate solution, $u^N(x)$,

  (Discretization error) $$\epsilon^N = u - u^N. \tag{2.19}$$

  In the asymptotic range, for sufficiently smooth functions, we assume that the convergence of this error is spectral (or exponential). This means that for a fixed size h_k of the elements, we assume that the behavior in each element Ω_k with the polynomial order N_k is

  $$||\epsilon_k^N|| \leq \sum_{k=1}^{K} C_k \exp(-\eta_k N_k), \tag{2.20}$$

  where C_k and η_k are constants that depend on the smoothness of the functions [8,25] and K the total number of elements.

  Remark 1 In general, the discretization error in one element is influenced by the discretization error in the others. To get a good solution, not only should the element be well resolved, but also its neighbors.

  Furthermore, for problems with anisotropic solutions, we assume a "tensor-product"-type error bound

  $$||\epsilon_k^N|| \leq \sum_{k=1}^{K} \sum_{i=1}^{N_{dim}} C_{ik} \exp(-\eta_{ik} N_{ik}), \tag{2.21}$$

  where N_{dim} is the number of spatial dimensions of the problem. For a detailed explanation of anisotropic error bounds in DG see [21].

  Remark 2 Eq. (2.21) forms the basis for anisotropic refinement. For problems with solutions whose complexity differ according to the spatial direction of measurement, e.g. boundary layers, an increment in the number of degrees of freedom (DOF) in only one direction is more efficient than isotropic refinement.

– *Interpolation error*
  We define the interpolation error as the difference between a function, (in particular, this function can be the solution of the problem) $u(x)$, and its Lagrangian interpolant, $I_N u(x)$,

(Interpolation error)
$$\varepsilon_u^N = u - I_N u. \tag{2.22}$$

If the Gauss-Legendre nodes are used as interpolation nodes, the interpolation error behaves asymptotically as the best possible approximation for the set of nodes (the truncated Legendre series) [7]. For sufficiently smooth functions, in the asymptotic range, the convergence of this error is spectral (or exponential). This means that for a fixed size h_k of the elements, the behavior in each element Ω_k with the polynomial order N_k is approximated by

$$||\varepsilon_k^N|| \leq C_k \exp(-\eta_k N_k), \tag{2.23}$$

where, again, C_k and η_k are constants which depend on the smoothness of the functions.

Remark 3 The interpolation error measures the suitability of the element but not its neighbors to approximate the solution locally.

Furthermore, for anisotropic functions we assume a "tensor-product"-type error bound

$$||\varepsilon_k^N|| \leq \sum_{i=1}^{N_{dim}} C_{ik} \exp(-\eta_{ik} N_{ik}), \tag{2.24}$$

where N_{dim} is the number of spatial dimensions of the problem. This result is obtained in [7] or in [21].

– *Iteration error*
  We define the iteration error as the difference between the steady, time-converged approximate solution, u^N, and the current approximation of the solution (not time-converged), $\tilde{u}^N$,

(Iteration error)
$$\epsilon_{it}^N = u^N - \tilde{u}^N. \tag{2.25}$$

The iteration error is directly related to the residual of the time-iterative method, e.g. Runge-Kutta, used to find the solution of the discrete partial differential set of equations, because

$$\mathcal{R}^N\left(\tilde{u}^N\right) = \mathcal{R}^N\left(u^N - \epsilon_{it}^N\right) = \mathcal{R}^N\left(\cancel{u^N}\right) - \left.\frac{\partial \mathcal{R}^N}{\partial u^N}\right|_{u^N} \epsilon_{it}^N + \mathcal{O}\left(\epsilon_{it}^N\right)^2. \tag{2.26}$$

– *Truncation error*
  The truncation error is defined as the difference between the discrete operator applied to the exact solution and the exact operator applied to the exact solution,

$$\tau^N = \mathcal{R}^N(u) - \mathcal{R}(u). \tag{2.27}$$

When $u(x)$ is the steady exact solution, it exactly solves the exact operator $\mathcal{R}(u) = 0$, so

(Truncation error)
$$\tau^N = \mathcal{R}^N(u). \tag{2.28}$$

Remark 4 The assumption of $u(x)$ being the steady exact solution of the problem $\mathcal{R}(u) = 0$ means (2.28) is only valid for steady exact solutions. The truncation error defined as (2.28) gives a measure of the suitability of the spatial discretization to solve the steady problem. Any temporal evolution is considered as an iteration, and is only a mathematical artifice to get the steady solution of the problem.

3 Truncation error

This section studies the truncation error. First, in Sect. 3.1, we analyze the truncation error in the DGSEM, which results in the decoupling of the surface and volume contributions of the truncation error. The latter will be defined as the isolated truncation error. Second, in Sect. 3.2, we extend the quasi-*a priori* error estimation of Fraysse *et al.* [18] to the DGSEM. This extension includes the estimation of both the truncation and the isolated truncation error.

3.1 Truncation error analysis in the DGSEM

In this section we analyze the truncation error in the DGSEM. First, in Sect. 3.1.1, we show that the truncation error includes a volume and a surface contribution; furthermore, we show how to decouple both contributions, which results in the definition of the isolated truncation error. Second, in Sect. 3.1.2, we establish the connections of the truncation error to the errors defined in Sect. 2.2, i.e. discretization and interpolation error. Third, in Sect. 3.1.3, we study the behavior of the truncation error for problems with anisotropic solutions.

3.1.1 Isolated truncation error

The DGSEM allows a straightforward separation between volume and surface contributions to the truncation error, as

$$\tau^N = \mathcal{R}^N_{\partial\Omega}(u) + \mathcal{R}^N_{\Omega}(u), \tag{3.1}$$

where $\mathcal{R}^N_{\partial\Omega}(u)$ is the surface discretization and $\mathcal{R}^N_{\Omega}(u)$ is the volume discretization. Furthermore, the volume contribution can be calculated using the isolated discrete partial differential operator (2.13),

$$\hat{\mathcal{R}}^N(u) = \mathcal{R}^N_{\Omega}(u). \tag{3.2}$$

From (3.2) we can define the isolated truncation error as the result of the isolated discrete partial differential operator applied to the exact solution,

$$\hat{\tau}^N = \hat{\mathcal{R}}^N(u). \tag{3.3}$$

The difference between the truncation error and the isolated truncation error is the surface contribution to the truncation error,

$$\tau^N - \hat{\tau}^N = \mathcal{R}^N_{\partial\Omega}(u). \tag{3.4}$$

   Proof:

For simplicity of exposition only, we prove this result in 1D. In a one dimensional problem, (2.11) is

$$u_t^N \frac{\Delta x}{2} \omega_j + \mathbf{f}^{N*}(u) \phi_i \Big|_{-1}^{1} - \left(\mathbf{f}^N(u), \phi_\xi\right)_N = 0, \tag{3.5}$$

where $\mathbf{f}^{N*}(u)$ is the value of the flux calculated with the contributions of the interior values and the neighbor values, i.e. the Riemann flux. The definition of the truncation error (2.28) for the DGSEM operator in 1D, (3.5), reads

$$\tau^N = \mathcal{R}^N(u) = I_N \mathbf{f}^* \phi_i \Big|_{-1}^{1} - (I_N \mathbf{f}, \phi_\xi)_N. \tag{3.6}$$

If we add and subtract $I_N \mathbf{f} \phi_i \big|_{-1}^{1}$ we get,

$$\tau^N = \left(I_N \mathbf{f}^* - I_N \mathbf{f}\right) \phi_i \Big|_{-1}^{1} + I_N \mathbf{f} \phi_i \Big|_{-1}^{1} - (I_N \mathbf{f}, \phi_\xi)_N, \tag{3.7}$$

where $I_N \mathbf{f} \phi_i \big|_{-1}^{1}$ represents the flux calculated using only the interior values and, as a result, does not depend on the neighbors. Finally, we identify the two last terms in (3.7) as the volumetric integral before the integration by parts is performed

$$\begin{aligned} \tau^N &= \left(I_N \mathbf{f}^* - I_N \mathbf{f}\right) \phi_i \Big|_{-1}^{1} + (I_N \mathbf{f}_\xi, \phi)_N \\ &= \mathcal{R}^N_{\partial\Omega}(u) + \mathcal{R}^N_{\Omega}(u). \end{aligned} \tag{3.8}$$

It can be seen that, by definition, the operator $\hat{\mathcal{R}}^N(u)$ fulfills the condition $\mathbf{f}^* = \mathbf{f}$, therefore

$$\hat{\tau}^N = \hat{\mathcal{R}}^N(u) = \mathcal{R}^N_{\Omega}(u). \tag{3.9}$$

3.1.2 Truncation error dependence on discretization and interpolation errors

The truncation error, τ^N, acts as the local source for the discretization error

$$\tau^N = \tilde{\mathcal{R}}^N\left(\epsilon^N\right) \tag{3.10}$$

for linear operators and

$$\tau^N = \left.\frac{\partial \mathcal{R}^N}{\partial u^N}\right|_{u^N} \epsilon^N + \mathcal{O}\left(\epsilon^N\right)^2 \tag{3.11}$$

for nonlinear ones. The isolated truncation error, $\hat{\tau}^N$, (the volumetric contribution to the truncation error) is related to the interpolation error by

$$\hat{\tau}^N = -\left(\nabla \cdot \varepsilon_{\mathbf{f}}^N, \phi_{i,j}\right)_N. \tag{3.12}$$

   Proof:

Equations (3.10-3.11) are the direct application of the DETE equation [40]. For linear operators, from (2.15), (2.19) and (2.28),

$$\tau^N = \mathcal{R}^N\left(u^N + \epsilon^N\right) = \tilde{\mathcal{R}}^N\left(u^N + \epsilon^N\right) + S^N = \mathcal{R}^N\left(\cancel{u^N}\right) + \tilde{\mathcal{R}}^N\left(\epsilon^N\right) \tag{3.13}$$

and for linear operators, from (2.16), (2.19) and (2.28),

$$\tau^N = \mathcal{R}^N\left(u^N + \epsilon^N\right) = \mathcal{R}^N\left(\cancel{u^N}\right) + \left.\frac{\partial \mathcal{R}^N}{\partial u^N}\right|_{u^N} \epsilon^N + \mathcal{O}\left(\epsilon^N\right)^2. \tag{3.14}$$

Equation (3.12) is a representation of the isolated truncation error (3.3). Remember that we define the isolated truncation error as the residual obtained by substituting the exact solution onto the isolated discretized partial operator,

$$\hat{\tau}^N = \hat{\mathcal{R}}^N\left(u\right) \tag{3.15}$$

and, in the DGSEM discretization

$$\hat{\tau}^N = \hat{\mathcal{R}}^N\left(u\right) = \sum_{e \in \partial \Omega} \int_e^N I_N \mathbf{f}^N \cdot \mathbf{n} \phi_{i,j} dS - \sum_{i,j=0}^N \mathbf{f}_{i,j} \cdot \nabla \phi_{i,j} w_i w_j. \tag{3.16}$$

As can be seen, $\hat{\tau}^N$ is the discretized projection of the divergence of the function $\nabla \cdot \mathbf{f}$ onto the basis

$$\hat{\mathcal{R}}^N\left(u\right) = \left(\nabla \cdot I_N \mathbf{f}, \phi_{i,j}\right)_N. \tag{3.17}$$

We then recall the definition of the interpolation error $\nabla \cdot \left(I_N \mathbf{f}\right) = \nabla \cdot \mathbf{f} - \nabla \cdot \varepsilon_{\mathbf{f}}$. By that definition, for a steady state solution $\nabla \cdot \mathbf{f} = 0$, the isolated truncation error is

$$\hat{\tau}^N = \hat{\mathcal{R}}^N\left(u\right) = -\left(\nabla \cdot \varepsilon_{\mathbf{f}}^N, \phi_{i,j}\right)_N. \tag{3.18}$$

Remark 5 Both the homogeneous discrete operator for linear problems, $\tilde{\mathcal{R}}^N$, and the discrete Jacobian for nonlinear problems, $\left.\frac{\partial \mathcal{R}^N}{\partial u^N}\right|_{u^N}$, establish connections between the elements through the Riemann problem. Equations (3.10-3.11) show that the truncation error in each element is a function of the discretization error in it and, through the Riemann problem, on the discretization error of its neighbors.

Remark 6 The isolated discrete partial differential operator, $\hat{\mathcal{R}}^N$, does not include connections between the elements. Therefore by (3.12) the isolated truncation error is a function of the interpolation error inside the element.

Remark 7 Although by (3.10-3.12) the truncation error and the isolated truncation error are functions of the discretization and interpolation error, it does not mean that the rate of convergence is the same. For discontinuous Galerkin methods, the truncation error predicts a lower order of accuracy than the actual rate, as a consequence of superconvergence [48]. A detailed analysis was performed in [41] showing that, for the Chebyshev spectral collocation method, even though the rate of convergence is not the same, all of them show exponential convergence. Furthermore, it was shown that the differences depend on the order of the PDE.

3.1.3 Anisotropic behavior of the truncation error

We are interested in the behavior of the truncation and the isolated truncation error for anisotropic solutions, e.g. boundary layers. In these problems, there is a difference between the complexity (in the sense of nonlinearity) of the solution in each of the spatial dimensions. Without loss of generality, we will suppose a two dimensional problem where x is the direction of high complexity. Following (2.21) and (2.24) it is reasonable to describe the discretization, ϵ^N, and the interpolation error, ε^N, as,

$$
\begin{aligned}
\epsilon^N &= \epsilon_x^N + \epsilon_y^N \\
\varepsilon^N &= \varepsilon_x^N + \varepsilon_y^N ,
\end{aligned}
\tag{3.19}
$$

with $\epsilon_x^N >> \epsilon_y^N$ and $\varepsilon_x^N >> \varepsilon_y^N$. These components represent the contribution to the whole error, which can be decreased by increasing the number of degrees of freedom (DOF) in each spatial direction.
Under these assumptions, the following results are obtained for the truncation error for linear problems

$$
\tau^N = \tilde{\mathcal{R}}^N \left(\epsilon_x^N \right) + \mathcal{O} \left(\left. \frac{\partial \mathcal{R}^N}{\partial u^N} \right|_{u^N} \epsilon_y^N \right),
\tag{3.20}
$$

and for the truncation error for nonlinear problems

$$
\tau^N = \left. \frac{\partial \mathcal{R}^N}{\partial u^N} \right|_{u^N} \epsilon_x^N + \mathcal{O} \left(\max \left(\left(\epsilon^N \right)^2 , \left. \frac{\partial \mathcal{R}^N}{\partial u^N} \right|_{u^N} \epsilon_y^N \right) \right).
\tag{3.21}
$$

Similarly, the following result is obtained for the isolated truncation error

$$
\hat{\tau}^N = - \left(\nabla \cdot \varepsilon_{x\mathbf{f}}^N, \phi_{i,j} \right)_N + \mathcal{O} \left(\varepsilon_{y\mathbf{f}}^N \right).
\tag{3.22}
$$

Remark 8 Both the truncation error and the isolated truncation error preserve the anisotropy of the discretization and interpolation errors.

> *Proof:*

Under the assumption of (3.19) and using the nonlinear DETE equation [40] we have

$$
\tau^N = \left. \frac{\partial \mathcal{R}^N}{\partial u^N} \right|_{u^N} \left(\epsilon_x^N + \epsilon_y^N \right) + \mathcal{O} \left(\epsilon^N \right)^2.
\tag{3.23}
$$

Since we assume $\epsilon_x^N >> \epsilon_y^N$,

$$
\tau^N = \left. \frac{\partial \mathcal{R}^N}{\partial u^N} \right|_{u^N} \epsilon_x^N + \mathcal{O} \left(\max \left(\left(\epsilon^N \right)^2 , \left. \frac{\partial \mathcal{R}^N}{\partial u^N} \right|_{u^N} \epsilon_y^N \right) \right).
\tag{3.24}
$$

This result is also valid for linear equations with minor modifications.

A similar approach is followed for the isolated truncation error. Under the assumption of (3.19), the isolated truncation error reads

$$
\hat{\tau}^N = \hat{\mathcal{R}}^N (u) = - \left(\nabla \cdot \varepsilon_{x\mathbf{f}}^N, \phi_{i,j} \right)_N - \left(\nabla \cdot \varepsilon_{y\mathbf{f}}^N, \phi_{i,j} \right)_N.
\tag{3.25}
$$

Since $\varepsilon_x^N >> \varepsilon_y^N$,

$$
\hat{\tau}^N = \hat{\mathcal{R}}^N (u) = - \left(\nabla \cdot \varepsilon_{x\mathbf{f}}^N, \phi_{i,j} \right)_N + \mathcal{O} \left(\varepsilon_{y\mathbf{f}}^N \right).
\tag{3.26}
$$

This result is valid for linear and nonlinear equations.

3.2 Quasi-*a priori* truncation error estimation in the DGSEM

In this section, we explain how to accurately estimate the truncation error and the isolated truncation error using the τ-estimation method. The idea underlying the τ-estimation method is to use an approximate solution instead of the exact solution that appears in the definitions of the truncation error (2.28) and the isolated truncation error (3.3). The approximate solution is obtained by solving the same problem with a higher polynomial order in each element. The estimation is quasi-*a priori*, in the sense that the approximate solution may not be time-converged.

The truncation error estimation procedure permits accurate estimations of the truncation error, τ^N, of a lower order approximation using a not necessarily converged higher order approximation, $\tilde{u}^P$. Therefore it is assumed that $(P > N)$. We introduce the next expression to estimate the truncation error based on the low order relative truncation error first used by Fraysse *et al.* in [18] and extended later to the pure spectral Chebyshev collocation method by Rubio *et al.* in [41]:

$$\tau_P^N \equiv \mathcal{R}^N\left(\tilde{u}^P\right) - \bar{I}_P^N \mathcal{R}^P\left(\tilde{u}^P\right). \tag{3.27}$$

Here $\bar{I}_P^N$, the transfer operator of the residual from order P to N, is

$$\bar{I}_P^N \equiv \tilde{\mathcal{R}}^N I_P^N \left(\tilde{\mathcal{R}}^P\right)^{-1} \tag{3.28}$$

for linear operators and

$$\bar{I}_P^N \equiv \left.\frac{\partial \mathcal{R}^N}{\partial u^N}\right|_{\tilde{u}^P} I_P^N \left(\left.\frac{\partial \mathcal{R}^P}{\partial u^P}\right|_{\tilde{u}^P}\right)^{-1} \tag{3.29}$$

for nonlinear operators. Note that for time-converged solutions, $u^P = \tilde{u}^P$ and, by definition, $\mathcal{R}^P\left(u^P\right) = 0$. This means that the second term on the RHS of (3.27) is zero for time-converged solutions.

Following the previous definition for the estimate of the truncation error (3.27), we define the next estimate for the isolated truncation error,

$$\hat{\tau}_P^N \equiv \hat{\mathcal{R}}^N\left(\tilde{u}^P\right) - \hat{I}_P^N \mathcal{R}^P\left(\tilde{u}^P\right). \tag{3.30}$$

The main difference from (3.27) is the substitution of the discrete partial differential operator, $\mathcal{R}^N$, by the isolated discrete partial differential operator, $\hat{\mathcal{R}}^N$. Besides $\hat{I}_P^N$, the transfer operator of the residual from order P to N for the isolated truncation error is,

$$\hat{I}_P^N \equiv \hat{\tilde{\mathcal{R}}}^N I_P^N \left(\tilde{\mathcal{R}}^P\right)^{-1} \tag{3.31}$$

for linear operators and

$$\hat{I}_P^N \equiv \left.\frac{\partial \hat{\mathcal{R}}^N}{\partial u^N}\right|_{\tilde{u}^P} I_P^N \left(\left.\frac{\partial \mathcal{R}^P}{\partial u^P}\right|_{\tilde{u}^P}\right)^{-1} \tag{3.32}$$

for nonlinear operators. As previously mentioned, $u^P = \tilde{u}^P$ for time-converged solutions and by definition $\mathcal{R}^P\left(u^P\right) = 0$. Therefore the second term on the RHS of (3.30) is zero for time-converged solutions.

Remark 9 For time-converged solutions, u^P, (*a posteriori* τ-estimation), the second term on the RHS of (3.27) and (3.30) is zero. For non time-converged solutions, $\tilde{u}^P$, (quasi-*a priori* τ-estimation), the second term on the RHS of (3.27) and (3.30) acts as a "correction" term for the iteration error, ϵ_{it}^P.

The difference between the exact truncation error (2.28) and the approximate truncation error (3.27) is

$$\tau_P^N = \tau^N - \tilde{\mathcal{R}}^N\left(\epsilon^P\right) \tag{3.33}$$

for linear operators and

$$\tau_P^N = \tau^N - \left.\frac{\partial \mathcal{R}^N}{\partial u^N}\right|_u \epsilon^P + \mathcal{O}\left(\epsilon_{it}^P\right)^2 \tag{3.34}$$

for nonlinear operators.

Similarly, the difference between the exact isolated truncation error (3.3) and the approximate isolated truncation error (3.30) is

$$\hat{\tau}_P^N = \hat{\tau}^N - \hat{\tilde{\mathcal{R}}}^N \left(\epsilon^P \right) \tag{3.35}$$

for linear operators and

$$\hat{\tau}_P^N = \hat{\tau}^N - \left. \frac{\partial \hat{\mathcal{R}}^N}{\partial u^N} \right|_u \epsilon^P + \mathcal{O} \left(\epsilon_{it}^P \right)^2 \tag{3.36}$$

for nonlinear operators.

Proof:

Due to the similarities in the definitions of the approximations of the truncation error (3.27) and the isolated truncation error (3.30), the proofs of (3.33-3.34) and (3.35-3.36) are similar, so we will only prove (3.33-3.34) here. This proof can be extended to (3.35-3.36) by taking into account that for the isolated truncation error estimate, the transfer operator of the residual from order P to N, $\hat{I}_P^N$, includes the homogeneous isolated discrete partial differential operator, $\hat{\tilde{\mathcal{R}}}^N$ as well as the discrete partial differential operator, $\mathcal{R}^N$.

Substituting the definitions of the iteration error, $\tilde{u}^P = u^P - \epsilon_{it}^P$, and the discretization error, $u^P = u - \epsilon^P$, onto the estimate of the truncation error (3.27) and using (2.16),

$$\tau_P^N = \mathcal{R}^N (u) - \left. \frac{\partial \mathcal{R}^N}{\partial u^N} \right|_u \epsilon^P - \left. \frac{\partial \mathcal{R}^N}{\partial u^N} \right|_u \epsilon_{it}^P - \bar{I}_P^N \mathcal{R}^P \left(\tilde{u}^P \right) + \mathcal{O} \left(\epsilon^P \right)^2 + \mathcal{O} \left(\epsilon_{it}^P \right)^2. \tag{3.37}$$

Then

$$\tau_P^N = \tau^N - \left. \frac{\partial \mathcal{R}^N}{\partial u^N} \right|_u \epsilon^P - \left. \frac{\partial \mathcal{R}^N}{\partial u^N} \right|_u \epsilon_{it}^P - \bar{I}_P^N \mathcal{R}^P \left(\tilde{u}^P \right) + \mathcal{O} \left(\epsilon^P \right)^2 + \mathcal{O} \left(\epsilon_{it}^P \right)^2 \tag{3.38}$$

and by (2.26),

$$\tau_P^N = \tau^N - \left. \frac{\partial \mathcal{R}^N}{\partial u^N} \right|_u \epsilon^P - \left. \frac{\partial \mathcal{R}^N}{\partial u^N} \right|_u \epsilon_{it}^P + \bar{I}_P^N \left. \frac{\partial \mathcal{R}^P}{\partial u^P} \right|_{u^P} \epsilon_{it}^P + \mathcal{O} \left(\epsilon^P \right)^2 + \mathcal{O} \left(\epsilon_{it}^P \right)^2. \tag{3.39}$$

Taking into account that, by definition $u - \tilde{u}^P = \epsilon^P + \epsilon_{it}^P$, it can be seen that for $\bar{I}_P^N = \left. \frac{\partial \mathcal{R}^N}{\partial u^N} \right|_{\tilde{u}^P} I_P^N \left(\left. \frac{\partial \mathcal{R}^P}{\partial u^P} \right|_{\tilde{u}^P} \right)^{-1}$ we have

$$\tau_P^N = \tau^N - \left. \frac{\partial \mathcal{R}^N}{\partial u^N} \right|_u \epsilon^P + \mathcal{O} \left(\epsilon^P \right)^2 + \mathcal{O} \left(\epsilon_{it}^P \right)^2. \tag{3.40}$$

Equation (3.40) is for nonlinear equations. However, it is also valid for linear ones taking into account that, for linear equations the Jacobian $\left. \frac{\partial \mathcal{R}^N}{\partial u^N} \right|_u ( )$ is substituted by the homogeneous discrete partial differential operator $\tilde{\mathcal{R}}^N ( )$ and that the Taylor expansions are exact taking only one term (so $\left(\epsilon^P \right)^2 = 0$ and $\left(\epsilon_{it}^P \right)^2 = 0$).

Remark 10 For linear problems, the accuracy of the estimates of the truncation error and the isolated truncation error, (3.33) and (3.35), is a function of the discretization error on the fine mesh ϵ^P. For nonlinear problems, the accuracy of the estimates, (3.34) and (3.36), is a function of the discretization error on the fine mesh, ϵ^P, and of the square of the iteration error on the fine mesh, $\left(\epsilon_{it}^P \right)^2$.

Remark 11 For linear problems, or for nonlinear problems with time-converged solutions, the relative accuracy of the estimate of the truncation error is,

$$\frac{\tau_P^N - \tau^N}{\tau^N} = \mathcal{O} \left(\epsilon^P / \epsilon^N \right) \tag{3.41}$$

and the relative accuracy of the estimate of the isolated truncation error is,

$$\frac{\hat{\tau}_P^N - \hat{\tau}^N}{\hat{\tau}^N} = \mathcal{O} \left(\epsilon^P / \varepsilon^N \right). \tag{3.42}$$

Proof:

Due to the similarities in the procedure we will only prove (3.41) for nonlinear problems. From (3.11) we know that the order of magnitude of the truncation error is

$$\tau^N = \left.\frac{\partial \mathcal{R}^N}{\partial u^N}\right|_{u^N} \epsilon^N + \mathcal{O}\left(\epsilon^N\right)^2 = \mathcal{O}\left(\epsilon^N\right). \tag{3.43}$$

From (3.34) the accuracy of the estimate of the truncation error for nonlinear problems with time-converged solutions is

$$\tau_P^N - \tau^N = -\left.\frac{\partial \mathcal{R}^N}{\partial u^N}\right|_u \epsilon^P = \mathcal{O}\left(\epsilon^P\right). \tag{3.44}$$

Therefore, from (3.43-3.44) the relative accuracy of the estimate of the truncation error for nonlinear problems with time-converged solutions is

$$\frac{\tau_P^N - \tau^N}{\tau^N} = \mathcal{O}\left(\epsilon^P/\epsilon^N\right). \tag{3.45}$$

Equation (3.41) is obtained from (3.10) and (3.33) for linear problems. Equation (3.42) is obtained from (3.12) and (3.35) for linear problems and from (3.12) and (3.36) for nonlinear problems.

Furthermore, it should be noticed that in (3.33-3.34) the discretization error, ϵ^P, is multiplied by the homogeneous discrete partial differential operator, $\tilde{\mathcal{R}}^N$, for linear problems or its Jacobian, $\left.\frac{\partial \mathcal{R}^N}{\partial u^N}\right|_u$, for nonlinear problems. Since the discrete partial differential operator couples the elements, an accurate solution in an element and in its neighbors is required for an accurate estimate of the truncation error. On the contrary, in (3.35-3.36) the discretization error, ϵ^P, is multiplied by the homogeneous isolated discrete partial differential operator, $\hat{\tilde{\mathcal{R}}}^N$, for linear problems or its Jacobian, $\left.\frac{\partial \hat{\mathcal{R}}^N}{\partial u^N}\right|_u$, for nonlinear problems. Since the isolated discrete partial differential operator does not couple the elements, an accurate solution in an element is sufficient for an accurate estimate of the isolated truncation error.

Finally, it should be noticed that a similar argument as the one followed in Sect. 3.1.3 may be used in (3.33-3.36) to decouple the rate of convergence of the estimates for problems with anisotropic solutions.

Remark 12 For nonlinear problems with non time-converged solutions the iteration error, ϵ_{it}^P, acts on the accuracy of the estimates of the truncation and the isolated truncation error (3.34) and (3.36). The effect of the iteration error can be estimated with a low cost, as it is the same order of magnitude as the residual of the time-iterative method on the fine mesh,

$$\left.\frac{\partial \mathcal{R}^N}{\partial u^N}\right|_{u^N} \epsilon_{it}^P \approx \mathcal{R}^P\left(\tilde{u}^P\right) = \mathcal{R}^P\left(u^P - \epsilon_{it}^P\right) = \mathcal{R}^P\left(\cancel{u^P}\right) - \left.\frac{\partial \mathcal{R}^P}{\partial u^P}\right|_{u^P} \epsilon_{it}^P + \mathcal{O}\left(\epsilon_{it}^P\right)^2. \tag{3.46}$$

Equation (3.46) can be derived from (3.39) and (2.26). It should be noticed that the "correction" term, second term on the RHS of (3.27) and (3.30), reduces the effect of the iteration error, ϵ_{it}^P, in the accuracy of the estimates, (3.34) and (3.36), from first order, ϵ_{it}^P, to second order, $\left(\epsilon_{it}^P\right)^2$. It should only be calculated if the first order effect of the iteration error is larger than the required accuracy in the estimates.

4 Implementation

In Sect. 3.2 we derived two formulas to estimate the truncation error (3.27) and the isolated truncation error (3.30). The formulas use an approximate solution on a fine grid, P, to estimate the error on a coarse grid, N. The estimates are quasi-*a priori*, in the sense that the approximate solution on the fine grid, $\tilde{u}^P$, may be not time-converged.

In this section we show how to implement the quasi-*a priori* estimates of the truncation error (3.27) and the isolated truncation error (3.30). First, in Sect. 4.1 we derive the computational cost to evaluate them. Second, in Sect. 4.2 we develop three algorithms to estimate the truncation error and the isolated truncation error with a minimum cost. The three algorithms follow different approaches for the treatment of the iteration error.

4.1 Computational cost

In this section we analyze the computational cost of the estimates of the truncation error (3.27) and of the isolated truncation error (3.30). First, in Sect. 4.1.1 we estimate the required number of operations to evaluate the estimates. Second, in Sect. 4.1.2 we estimate the memory requirements. In Sect. 4.1.3 we show that for a fixed fine mesh, P, the computation of the estimates on all the coarser meshes, $N < P$, cost the same as the computation of the truncation error on one coarse mesh. We will assume a nonlinear operator, however the results may be used for linear operators with minor modifications.

4.1.1 Operation count

The operation count (only multiplications) per degree of freedom (DOF) to update the solution in the DGSEM, i.e. to evaluate the discrete partial differential operator (2.12) or the isolated discrete partial differential operator (2.13), scales with the polynomial order $\mathcal{O}\left(P\right)$ [26].

The estimates (3.27) and (3.30) can be divided into four steps:

1. $\mathcal{R}^N\left(\tilde{u}^P\right)$ or $\hat{\mathcal{R}}^N\left(\tilde{u}^P\right)$. First term on the RHS (*a posteriori τ-estimation*):

    a) $\tilde{u}^P$. Approximate non time-converged solution on a fine mesh.

    b) $\mathcal{R}^N\left(\tilde{u}^P\right)$ or $\hat{\mathcal{R}}^N\left(\tilde{u}^P\right)$. Evaluation of the approximate solution, $\tilde{u}^P$, on the discrete partial differential operator on the coarse mesh.

2. $\bar{I}_P^N\mathcal{R}^P\left(\tilde{u}^P\right)$ or $\hat{I}_P^N\mathcal{R}^P\left(\tilde{u}^P\right)$. Second term on the RHS ("correction" term for quasi-*a priori τ-estimation*):

    a) $\epsilon_{it}^P \approx \left(\left.\frac{\partial\mathcal{R}^P}{\partial u^P}\right|_{\tilde{u}^P}\right)^{-1}\mathcal{R}^P\left(\tilde{u}^P\right)$. Approximate iteration error on the fine mesh (2.26). See the definition of the transfer operators (3.29) and (3.32).

    b) $\left.\frac{\partial\mathcal{R}^N}{\partial u^N}\right|_{\tilde{u}^P} I_P^N$ or $\left.\frac{\partial\hat{\mathcal{R}}^N}{\partial u^N}\right|_{\tilde{u}^P} I_P^N$. Transfer of the iteration error from the fine to the coarse mesh.

The number of operations to obtain the approximate solution on a fine mesh, for an explicit Euler time-iterative method is

$$\text{Step 1a} \approx \text{N}_{it} \times \text{DOF} \times P, \tag{4.1}$$

where N_{it} is the number of iterations and DOF is the number of degrees of freedom on the fine mesh (polynomial order P).

The evaluation of the approximate solution, $\tilde{u}^P$, on the discrete partial differential operator on the coarse grid, $\mathcal{R}^N\left(\tilde{u}^P\right)$ or $\hat{\mathcal{R}}^N\left(\tilde{u}^P\right)$, requires the interpolation of the fine grid solution to the coarse grid and the evaluation on the coarse operator. From (2.12), (2.13) and (2.17) we can see that the interpolation requires fewer operations per DOF than the evaluation on the coarse grid operators. Therefore,

$$\text{Step 1b} \leq 2 \times \text{DOF} \times P. \tag{4.2}$$

Both operations are at most twice the cost of the evaluation itself.

The computation of the estimate of the iteration error on the fine mesh requires the resolution of a linear system of algebraic equations of dimension DOF of the problem. The inversion of the Jacobian on the fine mesh by Gauss-Jordan elimination algorithm is DOF^3. Therefore the resolution of the system is

$$\text{Step 2a} \leq (\text{DOF})^3. \tag{4.3}$$

The exact number of operations of Step 2a depends on the efficiency of the algorithm used to solve the linear system.

Finally the transfer of the iteration error to the coarse mesh requires another interpolation from the fine to the coarse grid and a matrix vector multiplication. The dimension of the Jacobian on the coarse grid is the number of DOF on the coarse grid so,

$$\text{Step 2b} \approx (\text{DOF}_{\text{coarse}})^2, \tag{4.4}$$

where $\text{DOF}_{\text{coarse}}$ is the number of DOF on the coarse grid (polynomial order N).

13

We compare Step 1a with Step 1b and find that

$$\frac{\text{Step 1b}}{\text{Step 1a}} \approx \frac{2 \times \text{DOF} \times P}{\text{N}_{it} \times \text{DOF} \times P} = \frac{2}{\text{N}_{it}}. \tag{4.5}$$

Since the number of iterations to convergence is $\text{N}_{it} >> 1$, Step 1a $>>$ Step 1b. The cost to evaluate the approximate solution, $\tilde{u}^P$, with the discrete partial differential operator on the coarse grid, $\mathcal{R}^N(\tilde{u}^P)$ or $\hat{\mathcal{R}}^N(\tilde{u}^P)$, is negligible compared to the cost to obtain an approximate non time-converged solution on a fine mesh.

We compare Step 2a with Step 2b and see that

$$\frac{\text{Step 2b}}{\text{Step 2a}} \approx \frac{(\text{DOF}_{\text{coarse}})^2}{(\text{DOF})^3} \lessapprox \frac{1}{\text{DOF}}. \tag{4.6}$$

Since DOF $>> 1$, Step 2a $>>$ Step 2b. The cost to transfer the iteration error from the fine to the coarse mesh is negligible compared to the cost to approximate the iteration error on the fine mesh.

Finally, we compare Step 2a with Step 1a

$$\frac{\text{Step 1a}}{\text{Step 2a}} \approx \frac{\text{N}_{it} \times \text{DOF} \times P}{(\text{DOF})^3} = \frac{\text{N}_{it} \times P}{(\text{DOF})^2}. \tag{4.7}$$

The ratio of Step 1a and Step 2a will be different for each problem.

4.1.2 Memory requirements

The memory requirements for the estimate of the truncation error (3.27) and the estimate of the isolated truncation error (3.30) are, at most, twice the memory requirements of the DGSEM on the fine mesh, P, if the "correction" term is not calculated. From [26] we know that the DGSEM memory requirements are low (≈ 0.5 Gb for the the Navier-Stokes equations discretized with 1 million DOF).

The computation of the "correction" term requires the storage of the Jacobian on the fine and on the coarse grids. The memory requirements to store a square matrix of dimension DOF $= 10^6$ is ≈ 8000 Gb. However due to the compactness of the scheme, the memory requirements decrease drastically if the Jacobian is stored in sparse form. For each DOF the Jacobian only includes information about the element and its neighbors so, for a two dimensional problem,

$$\frac{\text{Jacobian non zeros}}{\text{Jacobian total}} \approx \frac{5 \times \text{DOF} \times (N+1)^2}{\text{DOF}^2} = \frac{5(N+1)^2}{\text{N}_{el}(N+1)^2} = \frac{5}{\text{N}_{el}}. \tag{4.8}$$

An estimate for the memory requirements of the sparse Jacobian of a 2D problem with DOF $= 10^6$ and polynomial order $N = 4$ is ≈ 1 Gb.

4.1.3 Multiple estimates for different coarse grids

For a fixed fine mesh, P, the computation of the estimates of the truncation error (3.27) and of the isolated truncation error (3.30) on all the coarser meshes, $N < P$, cost about the same as the computation of the truncation error (3.27) on one coarse mesh.

From (3.27) and (3.30), we can see that both estimates use the non time-converged approximate solution on the fine mesh, $\tilde{u}^P$. Furthermore, the "correction" term in the truncation error estimate (3.27) is

$$\bar{I}_P^N \mathcal{R}^P(\tilde{u}^P) = \left.\frac{\partial \mathcal{R}^N}{\partial u^N}\right|_{\tilde{u}^P} I_P^N \left(\left.\frac{\partial \mathcal{R}^P}{\partial u^P}\right|_{\tilde{u}^P}\right)^{-1} \mathcal{R}^P(\tilde{u}^P), \tag{4.9}$$

while in the isolated truncation error estimate (3.30) is

$$\hat{I}_P^N \mathcal{R}^P(\tilde{u}^P) = \left.\frac{\partial \hat{\mathcal{R}}^N}{\partial u^N}\right|_{\tilde{u}^P} I_P^N \left(\left.\frac{\partial \mathcal{R}^P}{\partial u^P}\right|_{\tilde{u}^P}\right)^{-1} \mathcal{R}^P(\tilde{u}^P). \tag{4.10}$$

As can be seen, (4.9-4.10) share the two last terms. The non time-converged approximate solution on the fine mesh, $\tilde{u}^P$, and the two last terms in (4.9-4.10) are also shared by any estimate on a coarse grid, since they only depend on the polynomial order on the fine mesh, P.

Finally, (4.5-4.6) show that to find the non time-converged approximate solution, $\tilde{u}^P$, on the fine mesh and to approximate the iteration error on the fine mesh, the two last terms in (4.9-4.10) account for most of the cost to evaluate (3.27) and (3.30).

4.2 Algorithms

In this section we develop three algorithms for the efficient computation of the estimates of the truncation and the isolated truncation errors. The three algorithms compute the estimates, using an approximate solution on a fine mesh, P, on all the coarser meshes, $N < P$. The computation of the estimates on coarser meshes permits analysis of the rate of convergence of the problem with polynomial order as well as the anisotropy of the solution. The algorithms are efficient since, from Sect. 4.1.3, the estimation of the truncation error and the isolated truncation error on all the coarser meshes cost about the same as the estimation of the truncation error on one coarse mesh. It should be noticed that the possibility of the computation of the estimates on all the coarser meshes is a big difference from the τ-estimation based methods for low order schemes, where one fine mesh is usually generated by refinement of a coarser mesh. The construction method of the fine mesh permits injection, i.e. direct substitution of the fine mesh approximation onto the coarse mesh.

The three algorithms deal differently with the iteration error. The first uses the time-converged solution of the problem, therefore the "correction" term is zero. The second uses a non converged solution and computes the "correction" term. The third also uses a non converged solution, but does not compute the "correction", although it is different from zero.

We assume a nonlinear problem in the development of the algorithms, however the results may be extrapolated to linear problems by taking into account that the Taylor expansions are exact with only one term for the linear problems.

A posteriori τ-estimation

The *a posteriori* τ-estimation, Algorithm 1, estimates the truncation and the isolated truncation error with an approximate time-converged solution on a fine mesh, u^P, on coarser meshes, $N < P$. As the approximate solution is time-converged, the iteration error, ϵ_{it}^P, and the "correction" term, the second term on the RHS of (3.27) and (3.30), are zero.

Data: Polynomial order on the fine mesh P
Result: τ_P^N, $\hat{\tau}_P^N$
Integrate in time on the fine mesh P to steady state;
for $N < P$ **do**
 Estimate the truncation error τ_P^N; (3.27) with no "correction" term ;
 Estimate the isolated truncation error $\hat{\tau}_P^N$; (3.30) with no "correction" term ;
end
Algorithm 1: *A posteriori* τ-estimation

Quasi-a priori τ-estimation

The quasi-*a priori* τ-estimation, Algorithm 2, estimates the truncation and the isolated truncation error with an approximate non converged solution on a fine mesh, $\tilde{u}^P$, on coarser meshes, $N < P$. As the approximate solution is non converged, the iteration error, ϵ_{it}^P, and the "correction" term, second term on the RHS of (3.27) and (3.30), are not zero.

The definition of the non fully-converged solution is important as it is related to the accuracy of the estimate through the iteration error on the fine mesh, ϵ_{it}^P. Remark 12 assumes that the iteration error may be estimated from the residual on the fine mesh, $\mathcal{R}^P(\tilde{u}^P)$. Therefore the solution is converged until the residual of the relaxation scheme fulfills $\max\left(\mathcal{R}^P(\tilde{u}^P)\right)^2 <$ tolerance. It should be noticed that this tolerance only applies to the iteration error, ϵ_{it}^P, of the estimation . The error associated with the spatial discretization on the fine mesh, ϵ^P, cannot be bounded *a priori* as the discretization error is an unknown of the problem. A Richardson iteration is used to approximate the iteration error on the fine mesh, ϵ_{it}^P.

It is important to remark that by (3.33) and (3.35) the tolerance requirement in Algorithm 2 is fulfilled at the first iteration for linear problems.

Data: Polynomial order on the fine mesh P, tolerance
Result: τ_P^N, $\hat{\tau}_P^N$
while $\max\left(\mathcal{R}^P\left(\tilde{u}^P\right)\right)^2 > $ *tolerance* **do**
  | Integrate in time on the fine mesh;
end

Calculate (Richardson iteration) $\epsilon_{it}^P \approx \left(\left.\frac{\partial \mathcal{R}^P}{\partial u^P}\right|_{\tilde{u}^P}\right)^{-1} \mathcal{R}^P\left(\tilde{u}^P\right)$;

for $N < P$ **do**
  | Estimate the non converged truncation error τ_P^N; (3.27) with "correction" term ;
  | Estimate the non converged isolated truncation error $\hat{\tau}_P^N$; (3.30) with "correction" term ;
end

<div align="center">

Algorithm 2: Quasi-*a priori* τ-estimation

</div>

Quasi-a priori τ-estimation with no "correction" term

The quasi-*a priori* τ-estimation with no "correction" term, Algorithm 3, estimates the truncation error (3.27) and the isolated truncation error (3.30) with an approximate non converged solution on a fine mesh, $\tilde{u}^P$, on coarser meshes, $N < P$. As the approximate solution is non converged, the iteration error, ϵ_{it}^P, and the "correction" term, second term on the RHS of (3.27) and (3.30), are not zero. However Algorithm 3 does not compute the "correction" term resulting in, by Remark 12, an effect of first order of the iteration error in the accuracy of the estimates.

Following the same argument explained in Algorithm 2, the solution is converged until the residual of the relaxation scheme fulfills $\max\left(\mathcal{R}^P\left(\tilde{u}^P\right)\right) < $ tolerance.

Data: Polynomial order on the fine mesh P, tolerance
Result: τ_P^N, $\hat{\tau}_P^N$
while $\left(\mathcal{R}^P\left(\tilde{u}^P\right)\right) > $ *tolerance* **do**
  | Integrate in time on the fine mesh;
end
for $N < P$ **do**
  | Estimate the non converged truncation error τ_P^N; (3.27) with no "correction" term;
  | Estimate the non converged isolated truncation error $\hat{\tau}_P^N$; (3.30) with no "correction" term;
end

<div align="center">

Algorithm 3: Quasi-*a priori* τ-estimation. No "correction" term

</div>

Algorithms 1-3 computational cost

From Sect. 4.1.3, we assume that the estimation of the truncation error and the isolated truncation error on all the coarser meshes cost about the same as the estimation of the truncation error in one coarse mesh. Then, from (4.1-4.4), the number of operations of Algorithm 1 is

$$\text{Algorithm 1} = \text{N}_{it1} \times \text{DOF} \times P, \tag{4.11}$$

the number of operations of Algorithm 2 is

$$\text{Algorithm 2} = \text{N}_{it2} \times \text{DOF} \times P + (\text{DOF})^3, \tag{4.12}$$

and the number of operations of Algorithm 3 is

$$\text{Algorithm 3} = \text{N}_{it3} \times \text{DOF} \times P. \tag{4.13}$$

From the time-iterative residual requirements of Algorithms 1-3, the number of iterations to converge the fine mesh solution, $\tilde{u}^P$, is $N_{it2} < N_{it3} < N_{it1}$. Algorithm 3 is cheaper than Algorithm 1 for any problem. Algorithm 2 is cheaper than Algorithm 3 if

$$N_{it3} - N_{it2} > \frac{(\text{DOF})^2}{P} \tag{4.14}$$

It should be noticed that the estimation for the number of operations to solve the linear system of equations to compute the "correction" term, $(\text{DOF})^3$, assumes that the inverse of the Jacobian is explicitly computed.

Table 1: Summary of the features of the three algorithms

	Algorithm 1	Algorithm 2	Algorithm 3
Residual	Machine error	$(\text{tolerance})^{1/2}$	tolerance
"Correction" term	No	Yes	No
Operation count	$N_{it1} \times \text{DOF} \times P$	$N_{it2} \times \text{DOF} \times P + (\text{DOF})^3$	$N_{it3} \times \text{DOF} \times P$
Memory requirements	Low	High	Low

The difference between $N_{it3} - N_{it2}$ may be reduced with more efficient algorithms that bypass the actual computation of the inverse of the Jacobian. Furthermore, it should be noticed that for stiff problems, which require small time steps, the difference $N_{it3} - N_{it2}$ increases while the cost to estimate the iteration error remains constant.

From Sect. 4.1.2, the memory requirements for Algorithm 1 and 3 are similar to the DGSEM. The computation of the "correction" term in Algorithm 2 requires sparse storage.

5 Detailed analysis on reference problems

In this section we test the analysis made in Sects. 3 and 4. The test is performed on one and two dimensional linear and nonlinear problems. In Sect. 5.1, the one dimensional numerical experiments are shown. These experiments support the theory developed in Sect. 3 with special focus on the behavior of the errors and the accuracy of the estimates. Two dimensional numerical experiments are shown in Sect. 5.2. The first objective of Sect. 5.2 is to support the anisotropic analysis of Sect. 3.1.3. The second is to apply the algorithms shown in Sect. 4 and support the cost analysis.

5.1 One dimensional problems. Linear and nonlinear advection equation

5.1.1 Propagation of errors

This section supports the analysis made in Sect. 3.1.1. We show how the discretization error and both the truncation error and the isolated truncation error propagate in the computational domain. Error propagation is an important issue in hyperbolic problems, as under-resolution errors in narrow zones of the computational domain propagate downstream.

The test problem chosen is

$$
\begin{aligned}
u_t + u_x &= 20 \left(\cosh \left(20 \left(0.75 + x \right) \right) \right)^{-2}, \quad x \in (-1, 1), \\
u(-1, t) &= \tanh \left(20 \left(-1 + 0.75 \right) \right), \quad t > 0,
\end{aligned}
\tag{5.1}
$$

whose exact solution for steady state, $u_t = 0$, is $u = \tanh \left(20 \left(x + 0.75 \right) \right)$. The test problem is approximated using four elements and a polynomial degree, $N = 7$, Fig. 1a. As can be seen, the solution is only difficult to approximate in the first element. However, the discretization error is high in all the computational domain. Due to the hyperbolic nature of the problem, the under-resolution in the "shock" (Element 1) has been transmitted downstream.

In Fig. 1b we represent the discretization error, the truncation error and the isolated truncation error. Several remarks should be made

- The truncation error, τ^N, is high where the discretization error, ϵ^N, is high (and non-constant). Furthermore, the truncation error is also high if the discretization error of its upstream neighbor is high (see Remark 5).
- The isolated truncation error, $\hat{\tau}^N$, measures the difficulty of approximation of the solution and its derivatives only inside the element (see Remark 6). This is why it is high only in Element 1 (where the "shock" is located).

This example illustrates why discretization error is not a good sensor for adaptation algorithms in hyperbolic problems. Furthermore, it can be seen that the isolated truncation error is superior to the truncation error as a sensor for adaptation algorithms because it is not contaminated by neighbors' errors.
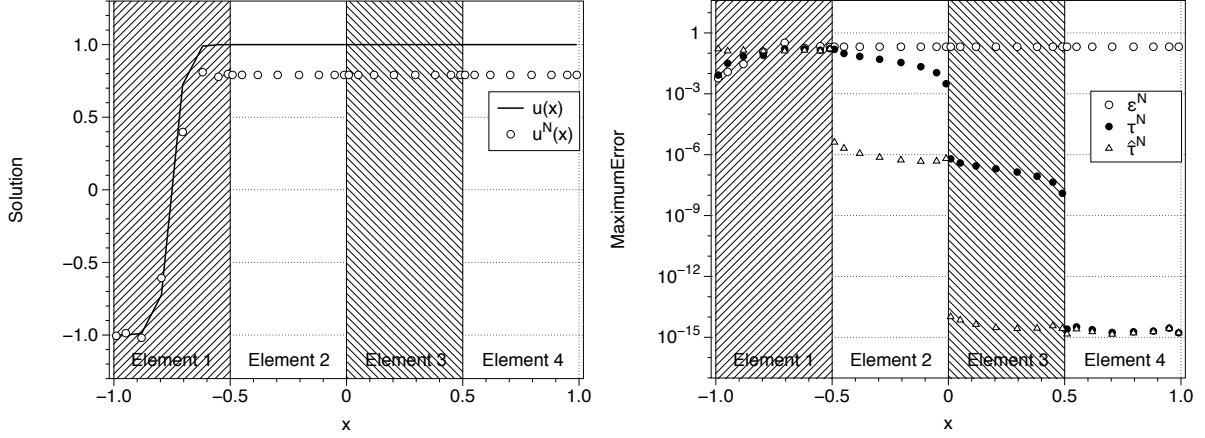
(a) Steady exact and approximate solution
$u = \tanh\left(20\left(x + 0.75\right)\right)$

(b) Discretization ϵ^N, truncation τ^N and isolated truncation error $\hat{\tau}^N$

Fig. 1: Test Problem 5.1

5.1.2 Convergence of errors

This section tests the analysis made in Sect. 3.1.2. We show convergence of the different errors with the polynomial order for the linear equation

$$u_t + u_x = \frac{-12\sin\left(x\right)}{\left(5 - 4\cos\left(x\right)\right)^2}, \quad x \in \left(-1, 1\right),$$

$$u\left(-1, t\right) = \frac{3}{5 - 4\cos\left(-1\right)}, \quad t > 0 \tag{5.2}$$

whose exact solution for steady state, $u_t = 0$, is $u = \frac{3}{5 - 4\cos(x)}$, and the nonlinear advection equation

$$u_t + u u_x = \left(-2 + \sin\left(x\right)\right)\cos\left(x\right), \quad x \in \left(-1, 1\right),$$

$$u\left(-1, t\right) = 2 - \sin\left(-1\right), \quad t > 0 \tag{5.3}$$

whose exact solution for steady state, $u_t = 0$, is $u = 2 - \sin\left(x\right)$. The physical domain is divided into four elements, and polynomials degrees, N, from one to 14 were used. In Fig. 2 we show, both for the linear and the nonlinear equations, that the convergence of the discretization error, ϵ^N, is spectral and, as shown in Sect. 3.1.2, the truncation, τ^N, and the isolated truncation error, $\hat{\tau}^N$, exhibit the same convergence rate.

5.1.3 A posteriori estimates

In this section we show convergence with the polynomial order of the accuracy of the estimates of the truncation error and the isolated truncation error for the same test cases used in Sect. 5.1.2. We use (3.27) and (3.30) to estimate the truncation error, τ_P^N, and the isolated truncation error, $\hat{\tau}_P^N$, in a fixed coarse mesh. To test the convergence, we vary the accuracy of the estimate of the exact solution through the polynomial order on the fine mesh, P.

We estimate the truncation error and the isolated truncation error in a fixed coarse mesh, $N = 4$. The estimates are computed using the fine meshes, $P = 5, \ldots, 14$, solutions. As before, the physical domain is divided into four elements. In Fig. 3 we show the maximum error in the estimate of the truncation error and the isolated truncation error. We also plot the discretization error for each polynomial order. Both errors in the estimates are the same order of magnitude as the discretization error on the fine mesh, ϵ^P, as established in (3.33-3.36).

5.1.4 Quasi-a priori estimates

In this section we show the convergence of the quasi-*a priori* estimate for the truncation error and the isolated truncation error with the number of iterations of the time-iterative method, for the same test cases used in Sect. 5.1.2.
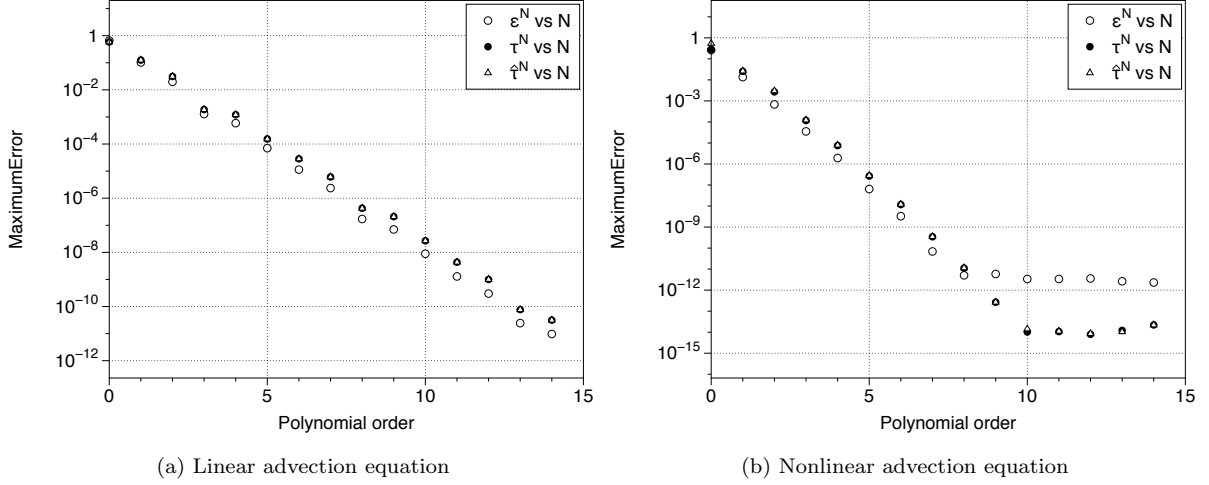
(a) Linear advection equation

(b) Nonlinear advection equation

Fig. 2: Discretization ϵ^N, truncation τ^N and isolated truncation $\hat{\tau}^N$ error convergence

(a) Linear advection equation

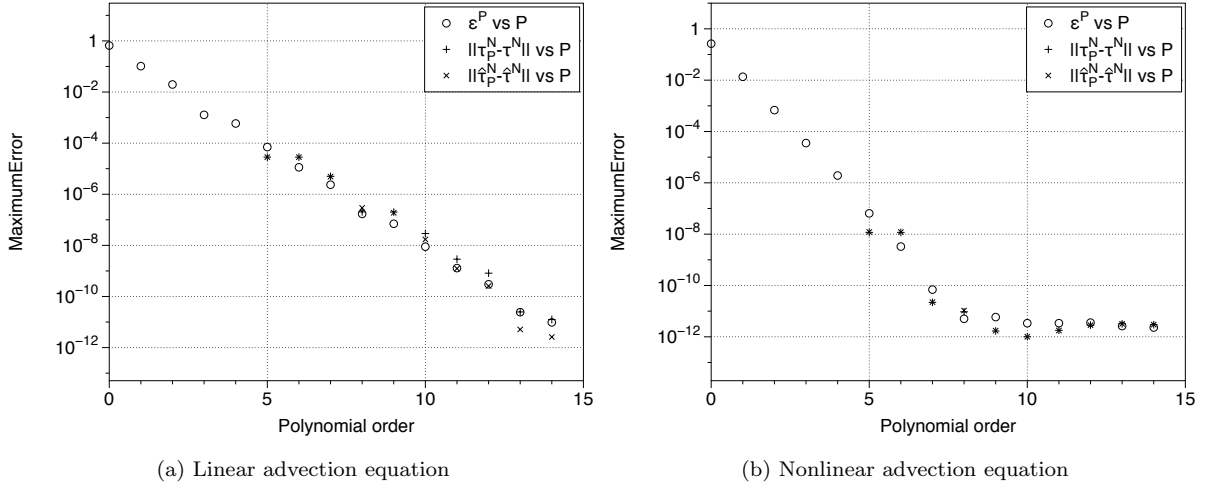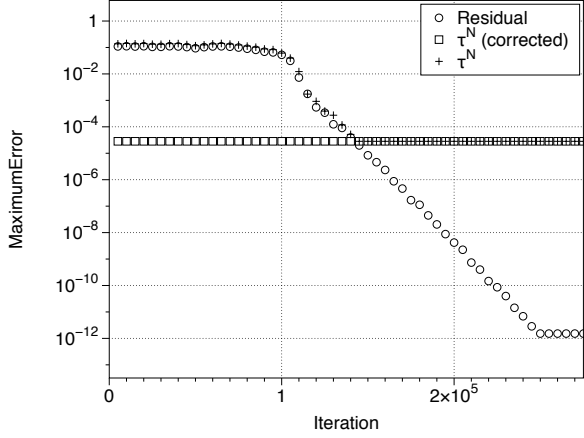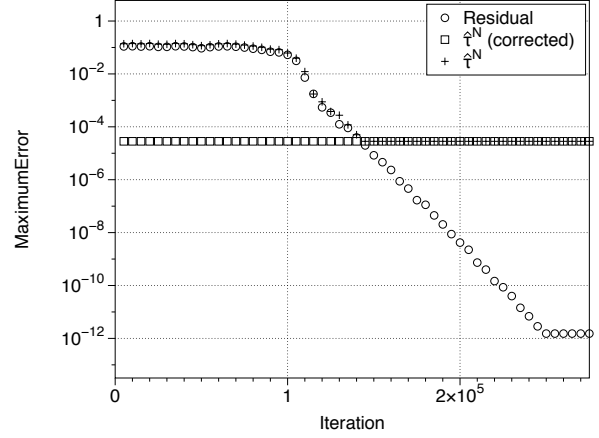(b) Nonlinear advection equation

Fig. 3: Accuracy of the estimates of the truncation error τ_P^N and the isolated truncation error $\hat{\tau}_P^N$. The coarse mesh is fixed $N = 4$ while the fine mesh varies $P = 5, \ldots, 14$

Now, since we are interested in the convergence of the accuracy of the estimates with the number of iterations, we fix the polynomial order for the coarse mesh, $N = 4$, and for the fine mesh, $P = 6$. Approximations of the solution on the fine mesh with several tolerances (iteration error, ϵ_{it}^P) are used to estimate the truncation error (3.27) and the isolated truncation error (3.30) on the coarse mesh. We show the results for the truncation error and the isolated truncation error, respectively, in Figs. 4 and 5.

For the linear equation, without correction, the error in the estimate is the same order of magnitude as the residual on the fine mesh. On the other hand, with the correction the problem is solved in the first iteration. For the nonlinear problem without correction the error is, again, the same order of magnitude as the residual. However, the error after the correction is applied is $\mathcal{O}\left(\epsilon_{it}^2\right)$. These results support (3.33-3.36). It should be remarked that, without applying the correction term, the true values of both the truncation and the isolated truncation error are masked by the iteration error. If the desired accuracy in the estimation of the error is known, it is possible to use the residual on the fine mesh to know how many iterations are required. These results support Remark 12.
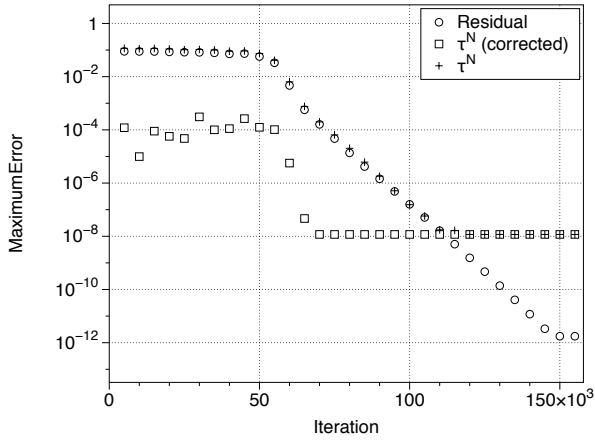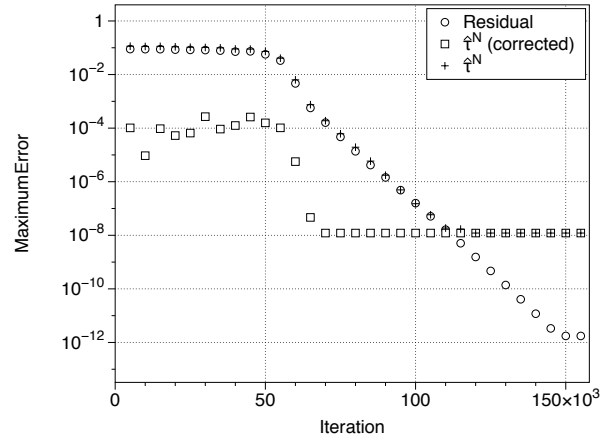
(a) Truncation error estimate   (b) Isolated truncation error estimate

Fig. 4: Linear advection equation. Truncation error τ_P^N and isolated truncation error $\hat{\tau}_P^N$ estimates with and without correction. The residual here is defined as $\mathcal{R}^P\left(\tilde{u}^P\right)$

(a) Truncation error estimate   (b) Isolated truncation error estimate

Fig. 5: Nonlinear advection equation. Truncation error τ_P^N and isolated truncation error $\hat{\tau}_P^N$ estimates with and without correction. The residual here is defined as $\mathcal{R}^P\left(\tilde{u}^P\right)$

5.2 Two dimensional problems. Nonlinear advection equation

In this section we support the multidimensional analysis made in Sect. 3.1.3. Furthermore, we present an application of the algorithms shown in Sect. 4 that supports the computational cost analysis.

    We solve the two dimensional nonlinear advection equation

$$u_t + uu_x + uu_y = f(x,y) \quad (x,y) \in (0,1) \times (0,1), \tag{5.4}$$

where the source term $f(x,y)$ is

$$f(x,y) = \left(\frac{50}{\cosh^2(50(y-0.5))} + \cos(x-0.5)\right)(\tanh(50(y-0.5)) + 2 + \sin(x-0.5)), \tag{5.5}$$

and appropriate Dirichlet boundary conditions so the exact solution for steady state $u_t = 0$ is

$$u(x,y) = \tanh(50(y-0.5)) + 2 + \sin(x-0.5). \tag{5.6}$$

    The exact solution of this problem, Fig. 6, is anisotropic (it has larger gradients in the y direction than in the x).
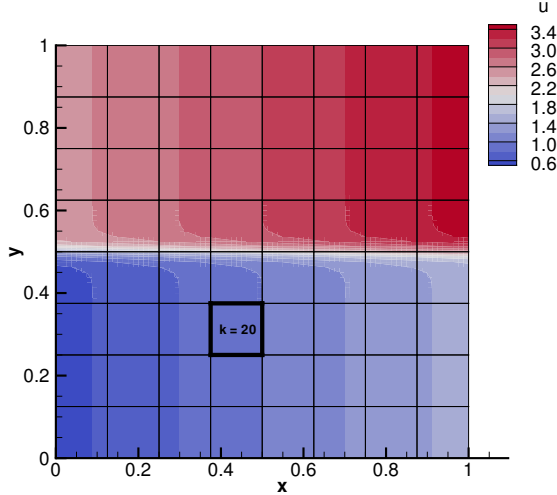
Fig. 6: Exact solution for the nonlinear advection problem, (5.4)

5.2.1 Discretization, truncation and isolated truncation error

In this section we analyze the discretization, truncation and isolated truncation error for problem (5.4-5.6). In Fig. 7 we show maps for the three errors for a polynomial order of $[3 \times 3]$ in each of the elements. In Fig. 8 the same result is presented for a polynomial order of $[3 \times 8]$.

First it should be noticed that, as in Sect. 5.1.1, the discretization error is advected, whereas the truncation and the isolated truncation error are contained in a narrow region where the function is more difficult to approximate.

As far as the rate of convergence is concerned, we exposed in Sect. 3.1.3 that for anisotropic problems the truncation error, τ^N, and the isolated truncation error, $\hat{\tau}^N$, should retain the anisotropy. Therefore in Fig. 9 we show the maximum value of the different errors in element $k = 20$, Fig. 6, for polynomial orders $N = [1, 10]$. As can be seen, the rates of convergence are decoupled and are exponential in both directions. In Fig. 10, we show sections of these plots for constant values of N_x or N_y where the exponential convergence can be clearly seen.

5.2.2 A posteriori truncation error and isolated truncation error estimates

Here we show the convergence of the accuracy of the estimates of the truncation error and the isolated truncation error with the polynomial order for problem (5.4-5.6). Remark 11 assumed that the errors in the estimates are decoupled in each direction.

We estimate the truncation error and the isolated truncation error on coarse meshes $N = [1\text{-}7 \times 1\text{-}7]$ using an approximate solution on a fine mesh $P = [8 \times 8]$.

First we solve the problem using a polynomial order of $P = [8 \times 8]$ in each element. Then we use this solution to calculate estimates of the errors on each of the coarser meshes. We show the results, again for element $k = 20$, in Fig. 11. It should be noticed that the error made in the estimates is almost constant, as it is only a function of the discretization error on the fine mesh, which is constant. However, the relative accuracy increases with $P - N$, as assumed in Remark 11. It is important to recall that the accuracy is decoupled in each of the spatial dimensions.

5.2.3 Computational cost of Algorithms 1-3

In this section we test the performance of Algorithms 1, 2 and 3 developed in Sect. 4. The test problem is the scalar nonlinear advection problem (5.4). We use (5.7) as initial condition,

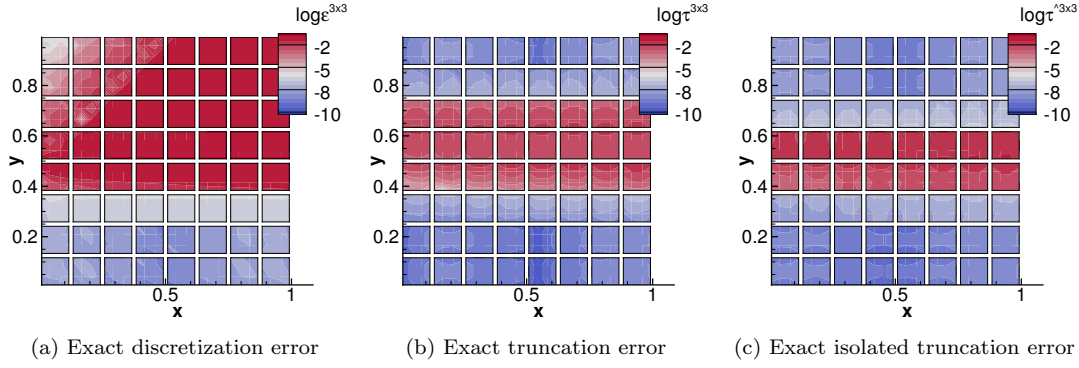$$f(x, y) = \tanh(50(y - 0.5))e^{-100x} + \sin(x - 0.5)e^{-100y} + 2. \tag{5.7}$$

(a) Exact discretization error   (b) Exact truncation error   (c) Exact isolated truncation error

Fig. 7: Error maps for the 2D nonlinear advection equation with a polynomial order of $[3 \times 3]$

(a) Exact discretization error   (b) Exact truncation error   (c) Exact isolated truncation error

Fig. 8: Error maps for the 2D nonlinear advection equation with a polynomial order of $[3 \times 8]$

(a) Exact discretization error   (b) Exact truncation error   (c) Exact isolated truncation error
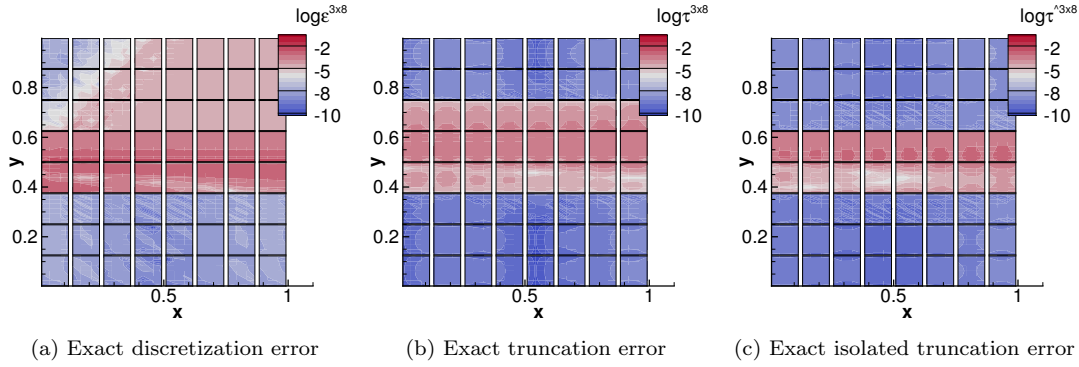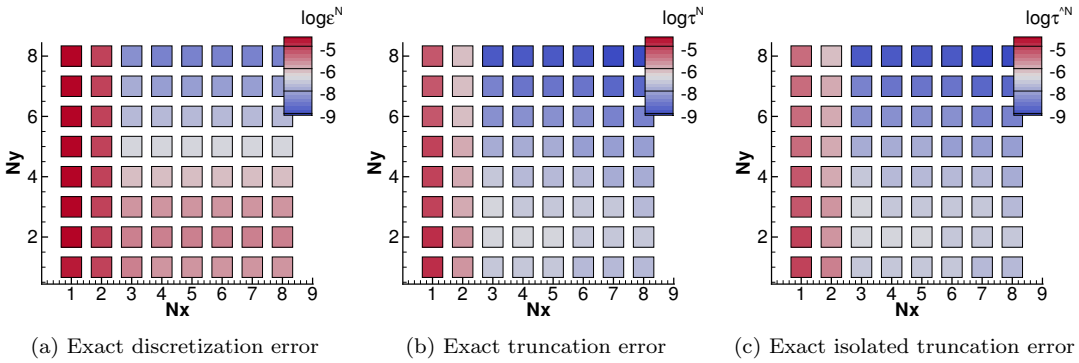
Fig. 9: Convergence of the errors for the 2D Nonlinear advection equation with different polynomial orders for element $k = 20$

The initial condition satisfies the equation at the bottom and left boundaries, so that there are no discontinuities between the initial condition and the inflow boundary condition.

The three algorithms require the fine mesh solution $P = [4 \times 4]$ as an input. Furthermore, Algorithms 2 and 3 require a tolerance for the iteration error, which is set to 10^{-4}. The tolerance is chosen depending on the accuracy requirements of the estimates.

The first step of the three algorithms is to converge a solution on the fine mesh; however, different levels of time-convergence are required. The first algorithm supposes that the solution is fully converged to steady state, u^P. The second and the third require that, for a tolerance of 10^{-4}, the residual is $\mathcal{R}^P\left(\tilde{u}^P\right) = 10^{-2}$ and the residual is $\mathcal{R}^P\left(\tilde{u}^P\right) = 10^{-4}$ respectively. In Fig. 12 we show the maximum value of the residual on the fine mesh, $\mathcal{R}^P\left(\tilde{u}^P\right)$. It can be seen that $14000, 3000$ and 8000 time steps are necessary for Algorithms 1, 2 and 3 (respectively).
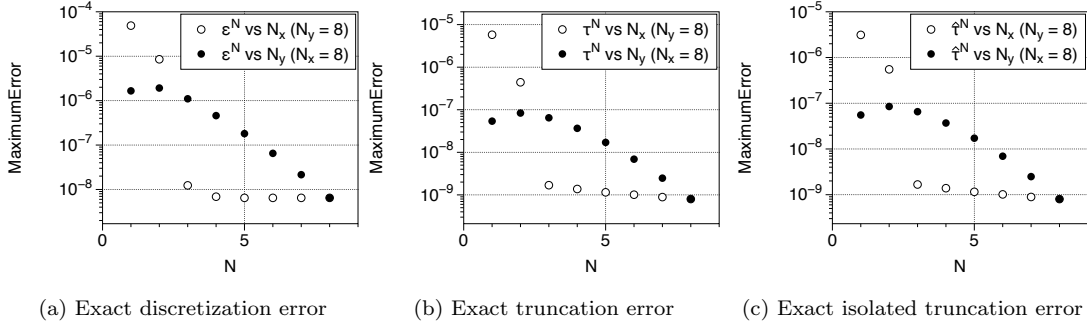
(a) Exact discretization error   (b) Exact truncation error   (c) Exact isolated truncation error

Fig. 10: Sections of Fig. 9 for constant values of $N_x = 8$ and $N_y = 8$

(a) Approximate truncation error   (b) Approximate isolated truncation error
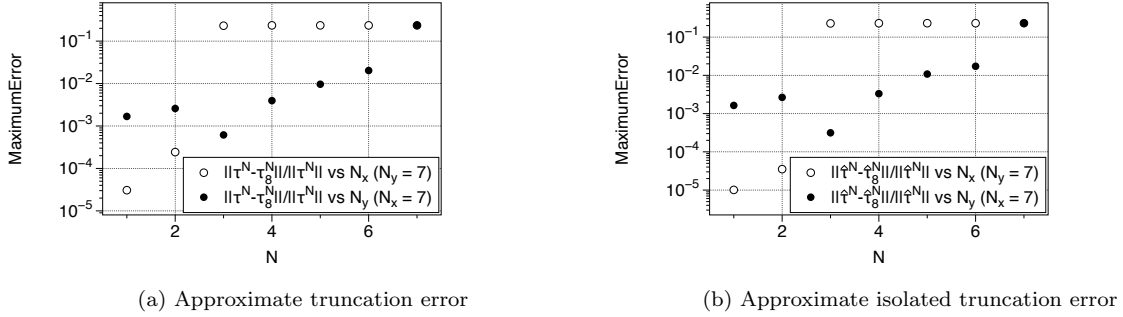
Fig. 11: Convergence of the approximated errors for the 2D Nonlinear advection equation with different polynomial orders for element $k = 20$. The errors are estimated using a fine mesh of $P = [8 \times 8]$
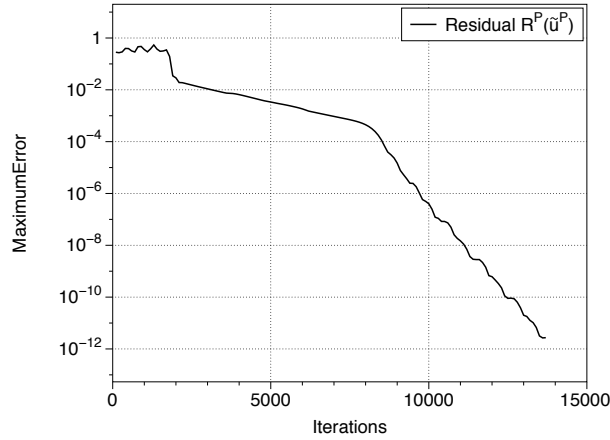

Fig. 12: Maximum value of the residual $\mathcal{R}^P\left(\tilde{u}^P\right)$ with the number of iterations of the RK3 low storage method [47]

In Fig. 13 we show the difference between the estimated truncation errors on a $N = [3 \times 3]$ mesh following Algorithm 2 ($\mathcal{R}^P\left(\tilde{u}^P\right) = 10^{-2}$) and Algorithm 1 (with the completely converged solution). In Fig. 14 we show the difference between the estimated truncation errors in a $N = [3 \times 3]$ mesh following Algorithm 3 ($\mathcal{R}^P\left(\tilde{u}^P\right) = 10^{-4}$) and the estimated errors using Algorithm 1 (again, with the completely converged solution). The results support Remark 12, as the differences between the three estimates are, at most, $\mathcal{O}\left(10^{-4}\right)$.

In Tab. 2 we show the time required to obtain the approximate solution on the fine mesh, $\tilde{u}^P$, the evaluation of the first term on the RHS of the estimates (3.27) and (3.30) (*a posteriori* estimation), the evaluation of the second term on the RHS of the estimates (3.27) and (3.30) ("correction" term) and the total time to run Algorithms 1-3. Algorithm 2 is the cheapest while Algorithm 1 is the least. Algorithm 3
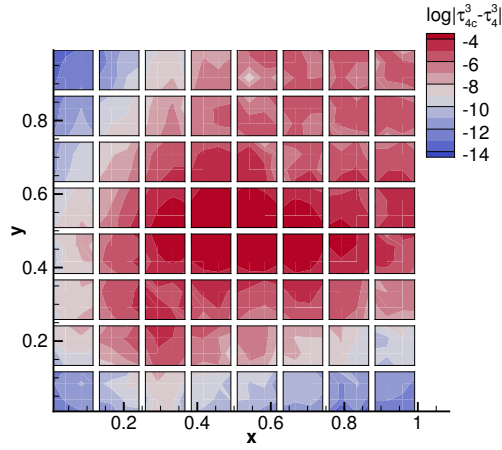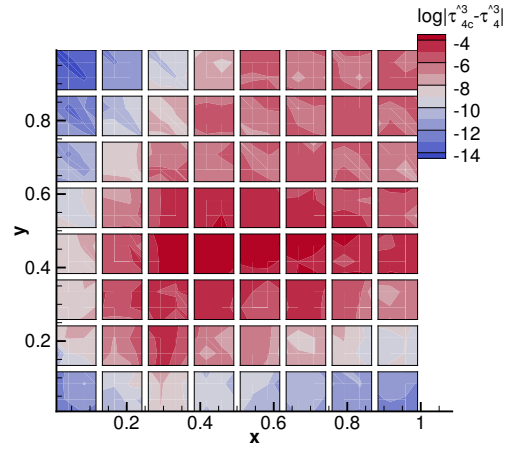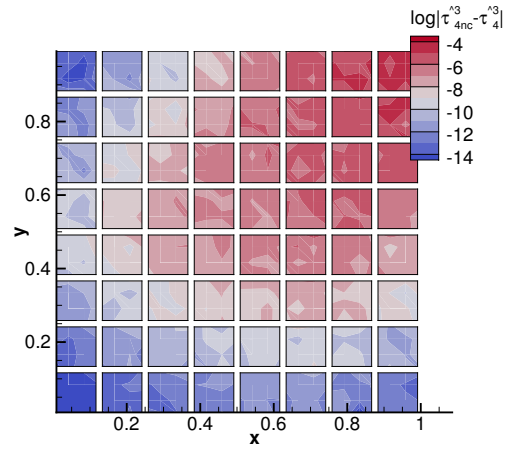
(a) Truncation error estimate   (b) Isolated truncation error estimate

Fig. 13: Accuracy of the estimated error maps for a residual of 10^{-2} with correction term
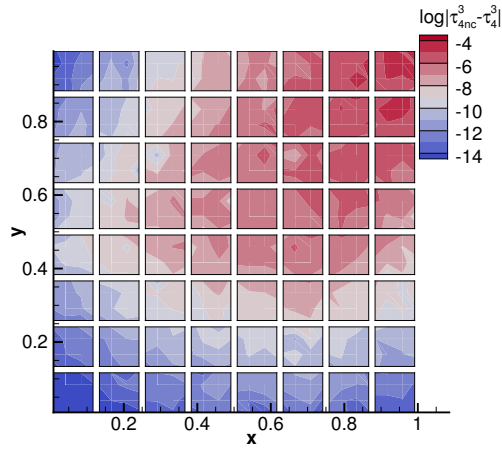
(a) Truncation error estimate   (b) Isolated truncation error estimate

Fig. 14: Accuracy of the estimated error maps for a residual of 10^{-4} without correction term

Table 2: Computational cost in time (s) of the three algorithms

	Approximate solution	*A posteriori*	"Correction" term	Total
Algorithm 1	42.00	0.02	0.00	42.02
Algorithm 2	9.00	0.02	0.38	9.40
Algorithm 3	24.00	0.02	0.00	24.02

Table 3: Computational cost in time (s) of the estimates

Step	Time	Algorithm
$\tilde{u}^P = u^P$	42.0	1
$\tilde{u}^P$	9.0	2
$\tilde{u}^P$	24.0	3
$\mathcal{R}^P\left(\tilde{u}^P\right)$	1.0×10^{-3}	-
$\mathcal{R}^N\left(\tilde{u}^P\right)$	2.0×10^{-3}	1,2,3
$\epsilon_{it}^P \approx \frac{\partial \mathcal{R}^P}{\partial u^P}^{-1} \mathcal{R}^P\left(\tilde{u}^P\right)$	3.6×10^{-1}	2
$\frac{\partial \mathcal{R}^N}{\partial u^N} I_P^N \epsilon_{it}^P$	3.0×10^{-3}	2

Table 4: Computational cost in time (s) of multiple estimates with "correction" term

	Time per 1 estimate	Time per $(P-1)^{N_{dim}}$ estimates
τ_P^N	3.65×10^{-1}	4.05×10^{-1}
τ_P^N and $\hat{\tau}_P^N$	3.70×10^{-1}	4.50×10^{-1}

is cheaper than Algorithm 1 too. It should be noticed that the computation of the estimates, Algorithms 1-3, cost about the same as it costs to converge to the approximate solution on the fine mesh. These results support (4.11-4.13).

In Tab. 3 we show a more detailed analysis of the computational cost of each step of Algorithms 1-3. The results support (4.1-4.7).

In Tab. 4 we show the time required to estimate the the truncation error or the truncation error and the isolated truncation error on any coarser mesh. Additionally we show the time required to estimate the truncation error or the truncation error and the isolated truncation error on all the coarser meshes. As assumed in Sect. 4.1.3, the cost to get one estimate is the same order of magnitude as the cost to estimate the truncation error and the isolated truncation error on all the coarser meshes, $N < P$.

6 Conclusions

The τ-estimation method of Brandt [6] has been successfully extended to the DGSEM. The extension exploits the characteristics of the method. One of the features of the DGSEM is the ability to estimate the isolated truncation error. The isolated error is better for determining the error within an element, which will be important for adaptation. Three algorithms to estimate the truncation error and the isolated truncation error have been developed. Two of them are quasi-*a priori* in the sense that the fully time-converged solution on the fine mesh is not required for the estimation of the error. The algorithms estimate both errors on all the coarse meshes, which provides information about the anisotropy of the solution as well as its rate of convergence in polynomial order. The truncation error estimates retain the spectral accuracy of the DGSEM and cost about the same as calculating the solution.

Acknowledgments

References

1. Acosta, C., Kopriva, D.A.: Discontinuous Galerkin spectral element approximations on moving meshes. Journal of Computational Physics **230**(5), 1876–1902 (2011)
2. Berger, M.J.: Adaptive finite difference methods in fluid dynamics. Tech. rep., New York: Courant Institute of Mathematical Sciences, New York University (1987)
3. Bernert, K.: τ-extrapolation-theoretical foundation, numerical experiment, and application to Navier-Stokes equations. SIAM Journal on Scientific Computing **18**, 460–478 (1997)
4. Black, K.: A conservative spectral element method for the approximation of compressible fluid flow. Kybernetika **35**(1), 133–146 (1999)
5. Black, K.: Spectral element approximation of convection–diffusion type problems. Applied Numerical Mathematics **33**(1-4), 373–379 (2000)
6. Brandt, A., Livne, O.: Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics. SIAM (1984)
7. Canuto, C., Hussaini, M.Y., Quarteroni, A., Zang, T.A.: Spectral Methods. Fundamentals in Single Domains. Springer (2006)
8. Canuto, C., Hussaini, M.Y., Quarteroni, A., Zang, T.A.: Spectral Methods. Evolution to Complex Geometries and Applications to Fluid Dynamics. Springer (2007)
9. Castel, N., Cohen, G., Durufle, M.: Application of discontinuous Galerkin spectral method on hexahedral elements for aeroacoustic. Journal of Computational Acoustics **17**(02), 175 (2009)
10. Choudhary, A., Roy, C.J.: Structured Mesh r-Refinement using Truncation Error Equidistribution for 1D and 2D Euler Problems. In: 21st AIAA Computational Fluid Dynamics Conference. American Institute of Aeronautics and Astronautics, Reston, Virginia (2013)
11. Cockburn, B., Karniadakis, G.E., Shu, C.W.: The development of discontinuous Galerkin methods. Springer, Berlin, Heidelberg (2000)
12. Cockburn, B., Shu, C.W.: The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. Journal of Computational Physics **141**(2), 199–224 (1998)

13. Deng, S.: Numerical simulation of optical coupling and light propagation in coupled optical resonators with size disorder. Applied Numerical Mathematics **57**(5-7), 475–485 (2007)

14. Deng, S., Cai, W., Astratov, V.: Numerical study of light propagation via whispering gallery modes in microcylinder coupled resonator optical waveguides. Optics express **12**(26), 6468–6480 (2004)

15. Eymard, R., Gallouët, T., Herbin, R.: Finite volume methods. Handbook of numerical analysis (2000)

16. Fagherazzi, S., Furbish, D.J., Rasetarinera, P., Hussaini, M.Y.: Application of the discontinuous spectral Galerkin method to groundwater flow. Advances in Water Resources **27**(2), 129–140 (2004)

17. Fagherazzi, S., Rasetarinera, P., Hussaini, M.Y.: Numerical solution of the dam-break problem with a discontinuous Galerkin method. Journal of Hydraulic Engineering **130**(6), 532–539 (2004)

18. Fraysse, F., De Vicente, J., Valero, E.: The estimation of truncation error by τ-estimation revisited. Journal of Computational Physics **231**, pp. 3457–3482 (2012)

19. Fraysse, F., Valero, E., Rubio, G.: Quasi-*a priori* truncation error estimation and higher order extrapolation for nonlinear partial differential equations. Journal of Computational Physics **253**, 389–404 (2013)

20. Fulton, S.R.: On the accuracy of multigrid truncation error estimates. Electronic transactions on numerical analysis **15**, 29–37 (2003)

21. Georgoulis, E.H.: Discontinuous Galerkin methods on shape-regular and anisotropic meshes. Ph.D. thesis (2003)

22. Giraldo, F.X., Hesthaven, J.S., Warburton, T.: Nodal High-Order Discontinuous Galerkin Methods for the Spherical Shallow Water Equations. Journal of Computational Physics **181**(2), 499–525 (2002)

23. Giraldo, F.X., Restelli, M.: A study of spectral element and discontinuous Galerkin methods for the Navier–Stokes equations in nonhydrostatic mesoscale atmospheric modeling: Equation sets and test cases. Journal of Computational Physics **227**(8), 3849–3877 (2008)

24. Hesthaven, J.S.: High-order accurate methods for time-domain electromagnetics. Advances in imaging and electron physics **127**, 59–125 (2003)

25. Hesthaven, J.S., Warburton, T.: Nodal discontinuous Galerkin methods: algorithms, analysis, and applications. Springer (2008)

26. Hindenlang, F., Gassner, G.J., Altmann, C., Beck, A., Staudenmaier, M., Munz, C.D.: Explicit discontinuous Galerkin methods for unsteady problems. Computers and Fluids **61**(C), 86–93 (2012)

27. Kopriva, D.A.: Metric Identities and the Discontinuous Spectral Element Method on Curvilinear Meshes. Journal of Scientific Computing **26**(3), 301–327 (2006)

28. Kopriva, D.A.: Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers. Scientific Computation. Springer Science+Business Media B.V. (2009)

29. Kopriva, D.A., Woodruff, S.L., Hussaini, M.Y.: Computation of electromagnetic scattering with a non-conforming discontinuous spectral element method. International Journal for Numerical Methods in Engineering **53**(1), 105–122 (2002)

30. Löhner, R.: Mesh adaptation in fluid mechanics. Engineering Fracture Mechanics **50**(56), 819 – 847 (1995). DOI 10.1016/0013-7944(94)E0062-L. URL http://www.sciencedirect.com/science/article/pii/0013794494E0062L

31. Lu, T., Zhang, P., Cai, W.: Discontinuous Galerkin methods for dispersive and lossy Maxwell's equations and PML boundary conditions. Journal of Computational Physics **200**(2), 549–580 (2004)

32. Mao, S., Luongo, C.A., Kopriva, D.A.: Discontinuous Galerkin spectral element simulation of quench propagation in superconducting magnets. Applied Superconductivity, IEEE Transactions on **15**(2), 1675–1678 (2005)

33. Oberkampf, W.L., Roy, C.J.: Verification and Validation in Scientific Computing. Cambridge University Press (2010)

34. Phillips, T., Roy, C.: Residual Methods for Discretization Error Estimation. In: 20th AIAA Computational Fluid Dynamics Conference. American Institute of Aeronautics and Astronautics, Reston, Viriginia (2012)

35. Rasetarinera, P., Hussaini, M.Y.: An Efficient Implicit Discontinuous Spectral Galerkin Method. Journal of Computational Physics **172**(2), 718–738 (2001)

36. Rasetarinera, P., Kopriva, D.A., Hussaini, M.Y.: Discontinuous Spectral Element Solution of Acoustic Radiation from Thin Airfoils. AIAA Journal **39**(11), 2070–2075 (2001)

37. Reed, W.H., Hill, T.R.: Triangular mesh methods for the neutron transport equation. Los Alamos Report LA-UR-73-479 (1973)

38. Restelli, M., Giraldo, F.X.: A conservative discontinuous Galerkin semi-implicit formulation for the Navier-Stokes equations in nonhydrostatic mesoscale modeling. SIAM Journal on Scientific Computing **31**(3), 2231–2257 (2009)

39. Roache, P.J.: Verification and validation in computational science and engineering. Hermosa Publishers (1998)

40. Roy, C.J.: Review of discretization error estimators in scientific computing. AIAA Paper **126**, 2010–0126 (2010)

41. Rubio, G., Fraysse, F., de Vicente, J., Valero, E.: The Estimation of Truncation Error by τ-estimation for Chebyshev Spectral Collocation Method. Journal of Scientific Computing **57**(1), 146–173 (2013)

42. Stanescu, D., Hussaini, M., Farassat, F.: Aircraft engine noise scattering - A discontinuous spectral element approach. In: 40th AIAA Aerospace Sciences Meeting & Exhibit. American Institute of Aeronautics and Astronautics, Reston, Viriginia (2012)

43. Stanescu, D., Xu, J., Hussaini, M.Y., Farassat, F.: Computation of engine noise propagation and scattering off an aircraft. International Journal of Aeroacoustics **1**(4), 403–420 (2009)

44. Syrakos, A., Efthimiou, G., Bartzis, J.G., Goulas, A.: Numerical experiments on the efficiency of local grid refinement based on truncation error estimates. Journal of Computational Physics **231**, 6725–6753 (2012)

45. Syrakos, A., Goulas, A.: Finite volume adaptive solutions using SIMPLE as smoother. International Journal for Numerical Methods in Fluids **52**, 1215–1245 (2006)

46. Toro, E.F.: Riemann Solvers and Numerical Methods for Fluid Dynamics. A Practical Introduction. Springer (2009)

47. Williamson, J.H.: Low-storage Runge-Kutta schemes. Journal of Computational Physics **35**(1), 48–56 (1980)

48. Zhang, M., Shu, C.W.: An analysis of and a comparison between the discontinuous Galerkin and the spectral finite volume methods. Computers and Fluids **34**, 581–592 (2005)

49. Zienkiewicz, O.C.: The background of error estimation and adaptivity in finite element computations. Computer Methods in Applied Mechanics and Engineering **195**(4-6), 207–213 (2006)