PhD Thesis

High order finite difference schemes for hyperbolic conservation laws on complex domains: extrapolation and time integration techniques

David Zorío Ventura

Advisor: Antonio Baeza Manzanares
Co-advisor: Pep Mulet Mestre

Universitat de València
Programa de Doctorat en Matemàtiques
València, Setembre 2016.

HIGH ORDER FINITE DIFFERENCE SCHEMES FOR HYPERBOLIC CONSERVATION LAWS ON COMPLEX DOMAINS: EXTRAPOLATION AND TIME INTEGRATION TECHNIQUES

Memòria presentada per David Zorío Ventura, Llicenciat en Matemàtiques; realitzada al departament de Matemàtiques de la Universitat de València sota la direcció d'Antonio Baeza Manzanares, Ajudant Doctor d'aquest departament, i codirecció de Pep Mulet Mestre, Catedràtic d'aquest departament, amb l'objectiu d'aspirar al Grau de Doctor en Matemàtiques.

València, 2 de setembre de 2016

Antonio Baeza Manzanares Director de la Memòria	Pep Mulet Mestre Codirector de la Memòria	David Zorío Ventura Aspirant al grau de Doctor

DEPARTAMENT DE MATEMÀTIQUES
FACULTAT DE MATEMÀTIQUES
UNIVERSITAT DE VALÈNCIA

Contents

Agraïments

En primer lloc, vull donar les gràcies a Toni Baeza i Pep Mulet, directors i codirectors de la meua tesi, respectivament, per la seua paciència i per haver realitzat un gran treball de direcció durant aquests anys. També agraïsc la resta de companys, personal investigador i administratiu de l'antic Departament de Matemàtica Aplicada, pel seu tracte excepcional, així com als doctorands amb qui he tingut el gust de compartir despatx: M^a Carmen Martí, Likhi Rubio, Sergio López i Birte Schmidtmann.

A Gabriel Gatica i Raimund Bürger, director i subdirector respectivament del Centro de Investigación en Ingeniería Matemática (CI^{2}MA, Concepción, Xile), per haver-me acollit temporalment, així com a la resta de personal del centre, que en tot moment m'han fet sentir com en casa.

A Valentín Navalón, amic i company de llicenciatura, per la riquesa matemàtica i personal que la seua companyia m'ha transmés durant tota l'etapa d'estudiant universitari.

A José A. Jiménez, per tots aquests anys d'amistat, la seua paciència, amabilitat, tracte immillorable i per haver-me acollit a la seua llar en aquest període de precarietat econòmica i baix poder adquisitu; en definitiva, per haver actuat en tot moment com un gran amic.

A tots els amics i companys d'afició amb qui he tingut i continue tenint el gust de compartir activitats i experiències, especialment als membres de la Colla de Campaners d'Ontinyent i de Campaners de l'Horta.

Finalment, vull donar les gràcies a les persones de les quals sempre he rebut recolzament incondicional: els meus pares, els meus germans Judith i Xavi, l'oncle Ricard i els meus avis, amb especial menció al iaio Ricardo Zorío, qui des de sempre ha sigut el meu model de referència.

València, 2016 *David*

Resum

Introducció

Els sistemes de lleis de conservació hiperbòliques i les equacions que es deriven d'aquestes han estat el tema central de moltes línies de recerca en les darreres quatre dècades, per exemple a l'hora de modelitzar el flux d'aire al voltant d'un vehicle, meteorologia i prediccions de l'oratge, o models de flux d'aigua sobre un canal o sedimentació de partícules sòlides petites dispersades en un fluid viscós.

Com que no es coneix la solució analítica de gran part d'aquestes equacions, s'han anat desenvolupant al llarg del temps diferents tècniques per tal d'abordar aquests problemes des d'una perspectiva numèrica, amb mètodes que han anat evolucionant i millorant al llarg d'aquests anys. El nostre interés se centra en l'obtenció de resultats el més ràpidament possible amb la major precisió possible, però la resolució numèrica de problemes físics modelats per sistemes de lleis de conservació és un assumpte delicat, degut a la presència de discontinuïtats a la solució. Aquestes discontinuïtats es desenvolupen fins i tot quan les dades inicials són suaus. Si calculem solucions discontínues de lleis de conservació emprant mètodes estàndard desenvolupats sota l'assumpció que les solucions són suaus, llavors típicament s'obtenen resultats numèrics que no són suficientment acurats.

Així doncs, es necessita fer ús d'esquemes *shock-capturing*, desenvolupats per tal d'obtenir aproximacions fines de solucions discontínues automàticament, sense una detecció explícita o condicions de salt, per tal d'assegurar un tractament adequat de discontinuïtats a les simulacions numèriques.

Els mètodes d'ordre petit són més ràpids i fàcils d'implementar, però proporcionen resultats menys precisos que els mètodes d'alta resolució.

Els mètodes *High-Resolution Shock-Capturing* (HRSC) són l'estat de l'art de simulacions numèriques per a problemes físics. L'objectiu de dits mètodes és el d'obtenir alta resolució on la solució és suau, al mateix temps que les discontinuïtats queden ben capturades, tot evitant al mateix temps la formació d'oscil·lacions espúries al seu voltant.

Com que l'inconvenient de les reconstruccions d'alt ordre són les oscil·lacions que aquestes poden crear, s'han anat suggerint una sèrie de tècniques per a combinar el marc d'*upwinding*, en el qual la discretització de les equacions en una malla es realitza d'acord amb la direcció de propagació de la informació en eixa malla, amb un mecanisme per a prevenir la creació i evolució d'aquestes d'oscil·lacions espúries numèriques. Per tant, la major part d'aquests esquemes emergeixen d'una combinació d'*upwinding* i interpolació d'alt ordre.

Els esquemes HRSC robustos i acurats sovint tenen un cost computacional alt, que està relacionat amb la incorporació d'*upwinding* a través d'informació característica requerida a la frontera de cadascuna de les cel·les del domini computacional i procediments de reconstrucció d'alt ordre.

Per tal de resoldre equacions en derivades parcials (EDPs), substituïm el problema continu representat per les EDPs per un conjunt finit de valors discrets. Aquests s'obtenen discretitzant primer el domini de les EDPs en un conjunt finit de punts o volums mitjançant una malla. Típicament el domini computacional es divideix en cel·les i les equacions contínues se substitueixen per una aproximació discreta a cada cel·la.

Els esquemes *weighted essentially non-oscillatory* (WENO), basats en una discretització espacial per diferències finites, s'han convertit en un dels mètodes més populars per a aproximar les solucions d'equacions hiperbòliques; és per això que aquests s'han anat desenvolupant de manera considerable. Aquests mètodes tenen un ingredient bàsic: les reconstruccions WENO, ço és, "interpoladors de mitjanes en cel·la", amb una precisió d'alt ordre i un control d'oscil·lacions.

Aquests esquemes van ser desenvolupats per Liu, Osher i Chan en [35], com una millora dels esquemes ENO (*essentially non-oscillatory*), originalment introduïts i desenvolupats en [16, 18]. L'única diferència entre aquests esquemes i la versió estàndard de mitjanes en cel·la de l'ENO és la definició del procediment de reconstrucció que produeix una aproximació global amb alt ordre de precisió de la solució a partir de les seues mitjanes en cel·la, que venen donades.

En [25], Jiang i Shu milloraren els esquemes de diferències finites d'alt ordre WENO definint una nova manera de mesurar la suavitat de la solució numèrica, que resulta en un esquema WENO de cinquè ordre per a stencils de cinc punts, en lloc de l'esquema de quart ordre obtingut amb la mesura de suavitat original proposada per Liu et al. [35].

Pel que fa a les condicions de frontera d'alt ordre, alguns autors han abordat aquest problema des de perspectives diferents. En [43] els autors desenvolupen una tècnica basada en interpolació de Lagrange amb un limitador el qual queda restringit a mètodes de segon ordre i una sola cel·la fantasma. També relacionats amb el nostre procediment són els treballs de Shu i col·laboradors [45, 46], on l'equació que es resol s'empra per extrapolar valors de la derivada de la solució numèrica als punts de la frontera on hi ha condicions inflow, els quals s'aproximen a través d'un desenvolupament de Taylor. Pel que fa a les fronteres outflow, s'empra una tècnica d'extrapolació basada en el mètode WENO, tot assolint alt ordre quan les dades són suaus en ambdós casos. Els inconvenients d'aquesta manera de procedir és que hi ha una dependència del problema (veure [23, 52] per a un mètode similar aplicat a altres equacions), que requereix un tractament diferent en funció del tipus de frontera i que té un cost computacional relativament alt.

Quant a la discretització temporal, la més típicament emprada, amb propietats excel·lents d'estabilitat, eficiència i baix emmagatzemament, i que ha estat emprada molt freqüentment en molts treballs, és el mètode de tercer ordre anomenat Runge-Kutta 3 TVD (*total variation diminishing*) [15]. Com que apareixen problemes d'estabilitat a partir de mètodes de Runge-Kutta de quart ordre i superior, en un intent de desenvolupar una família d'esquemes amb ordre temporal arbitràriament alt, Qiu i Shu [39] desenvoluparen en 2003 un esquema basat en el procediment de Lax-Wendroff, també conegut com la tècnica de Cauchy-Kowalewski. L'inconvenient en aquest cas és que novament la implementació depèn fortament de l'equació i les derivades corresponents del flux, així com un cost d'implementació i computacional alt.

En aquest treball desenvolupem un seguit de tècniques amb l'objectiu d'obtenir un esquema d'ordre arbitràriament alt, tant espacial com temporal, a partir de la consecució de dos objectius principals:

- Desenvolupar un esquema d'alt ordre per aplicar condicions de frontera numèriques d'alt ordre i guardar la informació a les cel·les fantasma a cada pas temporal. Aquest procediment ha de tenir en compte la possible geometria complexa de la frontera, de manera que el procediment siga adientment acurat, i que tinga en compte la possible presència eventual de discontinuïtats prop de la frontera, amb disseny de pesos contenint la corresponent informació de suavitat.

- Dissenyar un esquema d'alt ordre temporal que siga competitiu amb els esquemes Runge-Kutta TVD, amb menys dificultats d'implementació i menor cost computacional que els proposats per Qiu i Shu. La implementació d'aquests esquemes no hauria de ser més difícil que la implementació dels esquemes de Runge-Kutta i el cost computacional corresponent al càlcul de les derivades d'alt ordre no hauria de ser excessivament alt. Això, juntament amb el fet que solament es requereix una descomposició espectral per pas temporal, hauria de donar lloc a un esquema més eficient que la família dels mètodes de Runge-Kutta. En aquest sentit, també desenvolupem un mecanisme que evita la propagació de termes grans a les aproximacions de les derivades d'alt ordre al voltant de discontinuïtats.

El contingut del text s'organitza com segueix:

En el Capítol 2 es recorden els conceptes bàsics i idees relacionades amb les lleis de conservació hiperbòliques i els mètodes numèrics per a la seua resolució, essent el procediment de diferències finites de Shu-Osher i el procediment de reconstrucció WENO el focus central.

Al Capítol 3 descrivim un procediment per a mallar automàticament amb una malla cartesiana la frontera d'un conjunt de dues dimensions descrit per una corba tancada. El procediment garanteix el càlcul de totes les interseccions de les rectes de la malla amb la frontera, així com el càlcul de totes les rectes normals a la frontera que passen per cada cel·la fantasma, ambdós calculades per a la precisió desitjada.

En el Capítol 4 introduïm algunes tècniques per a efectuar les extrapolacions associades a condicions de frontera numèriques amb precisió d'ordre arbitràriament alt, amb un procediment que té en compte la possible formació o aproximació de discontinuïtats a la frontera. Per a fer-ho, es desenvolupen un seguit de dissenys de paràmetres de tolerància, *thresholds*, i pesos, en ambdós casos independents de l'escala i adimensionals, que permeten realitzar les extrapolacions sota les consideracions anteriorment esmentades.

El Capítol 5 tracta el desenvolupament d'un esquema d'alt ordre temporal, basat en el procediment de Lax-Wendroff proposat per Qiu i Shu, amb algunes millores d'implementació, eficiència i resolució. Més específicament, desenvolupem un esquema en què no és necessari el càlcul de cap derivada del flux, més ràpid que l'esquema original sota circumstàncies comuns, degut a una simplificació considerable del càlcul dels termes d'alt ordre, i capaç de capturar millor les discontinuïtats.

Finalment, es presenten algunes conclusions i treball futur al Capítol 6.

Lleis de conservació hiperbòliques

Una llei de conservació hiperbòlica és un sistema d'equacions en derivades parcials de la forma

$$\frac{\partial u}{\partial t}+\sum_{i=1}^{d}\frac{\partial f^i(u)}{\partial x_i}=0,\quad x\in\mathbb{R}^d,\quad t\in\mathbb{R}^+, \tag{1}$$

on $u=(u_1,\ldots,u_m)^T:\mathbb{R}^d\times\mathbb{R}^+\longrightarrow\mathbb{R}^m$ és el vector de les variables conservades i les funcions $f^i:\mathbb{R}^m\longrightarrow\mathbb{R}^m$ reben el nom de fluxos, $i=1,\ldots,d$.

L'equació (1) ve suplementada amb condicions inicials

$$u(x,0)=u_0(x),\quad x\in\mathbb{R}^d,$$

per tal de resoldre un problema de Cauchy, és a dir, trobar l'estat del sistema després d'un cert temps $t=T$, donat l'estat al temps $t=0$. El sistema (1) rep el nom d'hiperbòlic si qualsevol combinació lineal de les matrius jacobianes de f^i, $\sum_{i=1}^{d}\alpha_i(f^i)'(v)$ és diagonalitzable amb valors propis reals $\forall v\in\mathbb{R}^m$. Aquesta condició garanteix l'estabilitat dels problemes de Cauchy per a sistemes linealitzats sobre estats constants.

També s'han d'especificar condicions de frontera quan es considera un domini fitat, $\Omega\subseteq\mathbb{R}^d$. Una part d'aquesta tesi se centra en l'abordament de condicions de frontera numèriques sobre dominis complexos en múltiples dimensions.

El sistema (1) pot escriure's en forma quasi-lineal com

$$\frac{\partial u}{\partial t}+\sum_{i=1}^{d}(f^i)'(u)\frac{\partial u}{\partial x_i}=\frac{\partial u}{\partial t}+\sum_{i=1}^{d}\sum_{j=1}^{m}\frac{\partial f^i(u)}{\partial u_j}\frac{\partial u_j}{\partial x_i}=0.$$

En el cas particular $m=1$ l'equació (1) rep el nom de llei de conservació escalar, que per al cas $d=1$ es pot escriure com

$$u_t+f(u)_x=0,\quad x\in\mathbb{R},\quad t\in\mathbb{R}^+.$$

Com a exemples canònics, tenim l'equació d'advecció lineal

$$u_t+au_x=0,\quad a\in\mathbb{R},$$

i l'equació de Burgers

$$u_t+(\frac{u^2}{2})_x=0.$$

Un altre exemple corresponent a $m = 4$ i $d = 2$ són les equacions d'Euler en dues dimensions, que venen donades per

$$u_t + f(u)_x + g(u)_y = 0 \tag{2}$$

amb

$$u = \begin{bmatrix} \rho \\ \rho v^x \\ \rho v^y \\ E \end{bmatrix}, \quad f(u) = \begin{bmatrix} \rho v^x \\ \rho (v^x)^2 + p \\ \rho v^x v^y \\ v^x(E+p) \end{bmatrix}, \quad g(u) = \begin{bmatrix} \rho v^y \\ \rho v^y v^x \\ \rho (v^y)^2 + p \\ v^y(E+p) \end{bmatrix}, \tag{3}$$

on ρ denota la densitat, v^x i v^y són les components cartesianes del vector velocitat v, E és l'energia i p és la pressió, on l'energia (densitat) E està definida com la suma de l'energia cinètica i l'energia interna ρe

$$E = \frac{1}{2}\rho((v^x)^2 + (v^y)^2) + \rho e, \tag{4}$$

on e denota l'energia interna específica, unida amb la pressió i densitat a través d'una equació d'estat termodinàmica, $e = e(p, \rho)$. Emprarem l'equació d'estat dels gasos perfectes.

$$e = \frac{p}{(\gamma - 1)\rho},$$

on

$$\gamma = \frac{c_p}{c_v} \tag{5}$$

és el quocient dels calors específics a pressió constant, c_p, i a volum constant, c_v, i depèn del gas. Per a l'aire pren el valor $\gamma \approx 1.4$.

La versió en una dimensió ($d = 1$, $m = 3$) de les equacions s'obté postulant que totes les quantitats depenen solament de x i v^y és constant, de manera que s'obté

$$\begin{bmatrix} \rho \\ \rho v^x \\ E \end{bmatrix}_t + \begin{bmatrix} \rho v^x \\ \rho (v^x)^2 + p \\ v^x(E+p) \end{bmatrix}_x = 0. \tag{6}$$

Les lleis de conservació provenen habitualment de relacions integrals que representen la conservació d'una certa quantitat u. Per conservació s'entén que la quantitat continguda en un cert volum únicament pot canviar degut al fet que el flux d'eixa quantitat creua les interfícies d'un volum determinat. En una dimensió espacial pot escriure's com:

$$\int_{x_1}^{x_2} (u(x,t_2) - u(x,t_1))dx = \int_{t_1}^{t_2} f(u(x_1,t))dt - \int_{t_1}^{t_2} f(u(x_2,t))dt, \tag{7}$$

on el volum de control en el pla $x - t$ és $V = [x_1, x_2] \times [t_1, t_2] \subseteq \mathbb{R} \times \mathbb{R}$.

Estructura característica

S'entén per estructura característica d'una llei de conservació hiperbòlica l'estructura de valors i vectors propis de la matriu jacobiana dels fluxos, on les velocitats característiques es corresponen amb els valors propis λ_k^i de les matrius jacobianes $(f^i)'$, $i = 1, \ldots, d$, $k = 1, \ldots, m$. En el cas unidimensional, les característiques per a una funció u són corbes $(t, x(t))$ verificant $x'(t) = \lambda_k(u(x(t), t))$. Per a equacions escalars, aquesta condició es redueix a $x'(t) = f'(u(x(t), t) = f'(u(x(0), 0))$, de manera que les característiques són rectes de pendent $f'(u_0(x(0)))$, sobre les quals la informació roman constant.

Solucions febles i condicions de Rankine-Hugoniot

Entenem per solució clàssica de (1) una funció suau $u : \mathbb{R}^d \times \mathbb{R}^+ \longrightarrow \mathbb{R}^m$ que satisfà les equacions puntualment. No obstant això, relaxant les condicions imposades per (1) es poden considerar solucions des d'un context més general i físicament rellevant.

Definició 1. *Una funció $u(x, t)$ és solució feble de (1) donades unes certes dades inicials $u_0(x)$ si es compleix*

$$\int_{\mathbb{R}^+} \int_{\mathbb{R}^d} \left[u(x,t) \frac{\partial \phi}{\partial t}(x,t) + \sum_{j=1}^{d} f^j(u) \frac{\partial \phi}{\partial x_j} \right] dxdt = - \int_{\mathbb{R}^d} \phi(x,0) u_0(x) dx$$

per a tot $\phi \in C_0^1(\mathbb{R}^d \times \mathbb{R}^+)$, on $C_0^1(\mathbb{R}^d \times \mathbb{R}^+)$ és l'espai de les funcions contínuament diferenciables amb suport compacte en $\mathbb{R}^d \times \mathbb{R}^+$.

Les condicions de Rankine-Hugoniot caracteritzen les solucions febles en termes del moviment de les discontinuïtats, i proporciona informació sobre el comportament de les variables conservades al llarg de les discontinuïtats. Aquestes venen donades per

$$[f] \cdot n = [u](n \cdot s), \tag{8}$$

on $f = (f^1, \ldots f^d)$ és una matriu contenint els fluxos, u és la solució, s és la velocitat de propagació de la discontinuïtat i n és el vector normal a la discontinuïtat. La notació $[\cdot]$ indica el salt d'una variable al llarg d'una discontinuïtat. En el cas particular de problemes escalars la condició anterior és

$$f(u_L) - f(u_L) = s(u_L - u_R),$$

on u_L i u_R són els estats a l'esquerra i a la dreta de la discontinuïtat, respectivament.

Les solucions febles no són necessàriament úniques, i per tant es proposen condicions addicionals, anomenades entròpiques, que identifiquen la solució físicament rellevant (entròpica) del problema.

Mètodes numèrics per a lleis de conservació

Tot i que en alguns casos pot comprovar-se l'existència de solucions febles entròpiques per a lleis de conservació hiperbòliques, en molts pocs casos es coneix la solució analítica, on el coneixement d'aquestes queda restringit essencialment a equacions lineals o alguns problemes de Riemann. És per això que en la majoria de casos cal emprar mètodes numèrics per aproximar les solucions.

Considerem un problema de Cauchy escalar en una dimensió espacial

$$\begin{cases} u_t + f(u)_x = 0, \quad x \in \mathbb{R}, \quad , t \in \mathbb{R}^+, \\ u(x,0) = u_0(x), \end{cases} \tag{9}$$

on $u, f : \mathbb{R} \longrightarrow \mathbb{R}$.

Per definir una malla, considerem el subconjunt discret de punts (nodes) $\{x_j\}_{j\in\mathbb{Z}}$, $x_j \in \mathbb{R} \quad \forall j$ i suposem que la malla és uniforme, ço és, $x_j - x_{j-1} = \Delta x > 0$, $\forall j \in \mathbb{Z}$. Aquesta constant s'anomena grandària de la malla i l'abreviem per $h = \Delta x$. A partir dels punts $\{x_j\}$ definim les cel·les c_j com els subintervals el centre de les quals és x_j.

$$c_j = \left[\frac{x_{j-1}+x_j}{2}, \frac{x_j+x_{j+1}}{2}\right] = \left[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}\right].$$

Una malla es defineix, depenent del context, com o bé el conjunt de cel·les $\{c_j\}_{j\in\mathbb{Z}}$ o el conjunt de nodes $\{x_j\}_{j\in\mathbb{Z}}$.

Discretitzem la variable temporal definint punts en temps $\{t^n\}_{n\in\mathbb{N}}$, amb $t^n < t^{n+1}, \quad \forall n \in \mathbb{N}$. Si $t^{n+1} - t^n$ és constant amb respecte de n, ho denotem per Δt i l'anomenem increment temporal. Denotarem per $u^n = \{u_j^n\}_{j\in\mathbb{Z}}$ la informació corresponent a la solució exacta $u(x_j, t^n)$ de (9).

En problemes reals, el domini de definició de les equacions està restringit a un subconjunt fitat de $\mathbb{R}$ i un interval temporal finit, de manera que la malla s'ha de restringir a un nombre finit de nodes o cel·les. Si considerem l'interval $I = [0,1]$ i un temps fixat $T > 0$, aleshores podem prendre nombres positius N i M i definir un conjunt de nodes $\{x_j\}_{0\leq j<M}$ donat per $x_j = (j+\frac{1}{2})\Delta x$, amb $\Delta x = \frac{1}{M}$. Els punts en temps $\{t^n\}_{0\leq n<N}$ poden definir-se per $t^n = n\Delta t$, amb $\Delta t = \frac{1}{N}$.

L'explicació anterior pot estendre's per al cas multidimensional. Considerem per exemple una llei de conservació escalar en 2D de la forma:

$$\begin{cases} u_t(x,y,t) + f(u(x,y,t))_x + g(u(x,y,t))_y = 0, & (x,y) \in \mathbb{R}\times\mathbb{R}, \quad t\times\mathbb{R}^+, \\ u(x,y,0) = u_0(x), \end{cases}$$

i dos conjunts de punts ordenats $\{x_i\}_{i\in\mathbb{Z}}$ i $\{y_j\}_{j\in\mathbb{Z}}$, que satisfan $x_i < x_{i+1}$ per a tot $i\in\mathbb{Z}$ i $y_j < y_{j+1}$ per a tot $j\in\mathbb{Z}$. A més, assumim com abans que $\Delta x = x_{i+1}-x_i$ i $\Delta y = y_{j+1}-y_j$ són constants amb respecte de i i j, respectivament. Podem definir cel·les $c_{i,j}$ per

$$c_{i,j} = \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right] \times \left[y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}\right],$$

de manera que cada node (x_i, y_j) és el centre de la cel·la $c_{i,j}$.

Mètodes conservatius

La forma més senzilla d'aproximar derivades és mitjançant diferències finites lineals. Si la solució presenta alguna singularitat llavors, en general, aquestos mètodes no proporcionen aproximacions satisfactòries de les derivades parcials que apareixen a les equacions. A més, en tractar solucions discontínues, pot haver més d'una solució feble i el mètode pot no convergir a la correcta o fins i tot convergir a una funció que no és una solució feble de l'EDP. Existeix un requeriment simple, el compliment del qual garanteix que el mètode, en cas de convergir, ho faça a una solució feble.

Definició 2. *Un mètode numèric es diu conservatiu si pot escriure's de la forma*

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}\left(\hat{f}(u_{j-p+1}^n,\ldots,u_{j+q}^n) - \hat{f}(u_{j-p}^n,\ldots,u_{j+q-1}^n)\right), \qquad (10)$$

on la funció $\hat{f}:\mathbb{R}^{p+q}\to\mathbb{R}$ *rep el nom de flux numèric i* $p,q\in\mathbb{N},\ p,q\geq 0$.

El propòsit dels mètodes conservatius és el de reproduir a un nivell discret la conservació de les variables físiques en les equacions contínues. De fet, (10) es pot veure com una versió discreta de la forma integral (7) de l'EDP.

Un requeriment essential per al flux numèric és la condició de consistència:

Definició 3. *Direm que el flux numèric d'un mètode numèric conservatiu és consistent amb la llei de conservació si el flux numèric $\hat{f}$ es redueix al flux exacte f per al cas de dades constants, i.e.,*

$$\hat{f}(u, \ldots, u) = f(u).$$

La condició de consistència és necessària per tal de garantir que els mètodes conservatius proporcionen una forma discreta de conservació, anàloga a la llei de conservació.

En general, es requereixen certes condicions de suavitat, en la forma en què $\hat{f}$ s'aproxima a un cert valor $f(u)$, llavors suposem que el flux numèric és localment Lipschitz contínua en cada variable, és a dir, si x és un punt en un espai normat M aleshores existeix una constant K i un entorn $N(x)$ de x tal que $||f(y) - f(x)|| \leq K||y - x||, \quad \forall y \in N(x)$.

El resultat principal sobre mètodes conservatius és el teorema de Lax-Wendroff, que demostra que si produeixen una seqüència d'aproximacions que convergeix a alguna funció $u(x,t)$ segons es refina la malla, aleshores aquesta funció és una solució feble de la llei de conservació:

Teorema 1. *(Lax-Wendroff, [31, 22]) Considerem una seqüència de malles indexades per $k = 1, 2, \ldots$, amb grandàries de malla $(\Delta x_k, \Delta t_k)$, verificant*

$$\lim_{k \to +\infty} \Delta x_k = 0,$$
$$\lim_{k \to +\infty} \Delta t_k = 0.$$

Siga $\{u_k(x,t)\}$ la funció constant a trossos definida a partir de la solució numèrica obtinguda per un mètode conservatiu, consistent amb (1), en la malla k-èssima. Si la variació total de la funció $u_k(\cdot,t)$ està uniformement fitada en k, t, o el que és el mateix, $\sup_{k,t \in [0,T]} TV(u_k(\cdot,t)) < \infty$ i $u_k(x,t)$ convergeix en $\mathcal{L}^1_{loc}$ a una funció $u(x,t)$ quan $k \to \infty$, aleshores u és una solució feble de la llei de conservació.

Cal imposar condicions addicionals per tal de convergir a solucions entròpiques [38, 44]; altrament, els mètodes conservatius poden generar shocks que violen l'entropia, els quals es corresponen a solucions febles no entròpiques.

Esquema conservatiu de diferències finites de Shu-Osher

Per tal d'obtenir esquemes conservatius de diferències finites d'alt ordre per resoldre lleis de conservació hiperbòliques, emprem la tècnica de

Shu-Osher [42]. Els esquemes de tercer ordre o superior multidimensionals que s'obtenen amb aquesta tècnica són més eficients, tant en l'execució com implementació, que el seu esquema homòleg de volums finits. Una restricció per al cas d'esquemes de diferències finites d'alt ordre és que es requereixen malles cartesianes, la qual cosa és un inconvenient seriós a l'hora de tractar amb dominis de frontera corbada. Com veurem, una contribució essencial d'aquest treball és mostrar que és possible superar aquest inconvenient emprant tècniques d'extrapolació adients per a dades a les cel·les fantasma.

La idea bàsica que fa possible el procediment de Shu-Osher es recull al lema següent:

Lema 1. *Si les funcions g, φ satisfan*

$$g(x) = \frac{1}{\Delta x}\int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} \varphi(\xi) d\xi,$$

aleshores

$$g'(x) = \frac{\varphi\left(x+\frac{\Delta x}{2}\right) - \varphi\left(x-\frac{\Delta x}{2}\right)}{\Delta x}.$$

Si apliquem aquest resultat a $g(x) = f(u(x,t))$, per a un valor de t fix, la propietat conservativa de la discretització espacial s'obté definint implícitament la funció φ per:

$$f(u(x,t)) = \frac{1}{\Delta x}\int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} \varphi(\xi, t) d\xi,$$

de manera que la derivada espacial en

$$u_t + f(u)_x = 0$$

s'obté exactament a partir d'una fórmula de diferències finites conservatives a les fronteres de les cel·les,

$$f(u)_x = \frac{\varphi\left(x+\frac{\Delta x}{2}, t\right) - \varphi\left(x-\frac{\Delta x}{2}, t\right)}{\Delta x}.$$

Obviant la dependència de t en la presentació de la semidiscretització espacial, observem que es poden calcular aproximacions d'alt ordre de $\varphi\left(x \pm \frac{\Delta x}{2}\right)$ a partir de valors nodals coneguts de f (que són mitjanes en cel·la de φ) i un procediment de reconstrucció $\mathcal{R}$. Si $\widehat{\varphi}$ és una aproximació de φ obtinguda a partir de valors puntuals de f en un stencil al voltant de $x_{j+\frac{1}{2}}$ tal que $\varphi(x_{j+\frac{1}{2}}) = \widehat{\varphi}(x_{j+\frac{1}{2}}) + d(x_{j+\frac{1}{2}})\Delta x^r + \mathcal{O}(\Delta x^{r+1})$, per a una funció Lipschitz d, llavors podem discretitzar

$$f(u)_x(x_j) = \frac{\widehat{\varphi}(x_{j+\frac{1}{2}}) - \widehat{\varphi}(x_{j-\frac{1}{2}})}{\Delta x} + \mathcal{O}(\Delta x^r),$$

és a dir, l'error local de truncament de l'esquema semidiscret és $\mathcal{O}(\Delta x^r)$.

Denotem per $\mathcal{R}(\bar{f}_{j-s_1}, \ldots, \bar{f}_{j+s_2}, x)$ la reconstrucció local genèrica de $f(x)$ a partir de les seues mitjanes en cel·la $\{\bar{f}_{j-s_1}, \ldots, \bar{f}_{j+s_2}\}$, on s_1 i s_2 són enters no negatius.

A l'hora d'obtenir reconstruccions, un altre aspecte important a tenir en compte és l'upwinding, en el qual la discretització de les equacions en una malla es realitza d'acord amb la direcció de propagació de la informació en dita malla, és a dir, s'ha de tenir en compte el costat en el qual la informació (vent) flueix, donat pels signes dels valors propis de la matriu jacobiana.

Les aproximacions $\hat{f}^n_{j+\frac{1}{2}}$ s'obtenen mitjançant reconstruccions esbiaixades d'alt ordre $\mathcal{R}^{\pm}(\bar{f}_{j-s_1}, \ldots, \bar{f}_{j+s_2}, x)$, ço és, interpoladors de mitjanes en cel·la els stencils dels quals tenen més punts al costat de l'upwind dels punts on aquests s'avaluen. En aquest treball, $\widehat{f}$ s'obté mitjançant la partició de fluxos de Donat-Marquina [11] amb el mètode de reconstrucció WENO, que veurem tot seguit.

El mètode de Donat-Marquina empra descomposicions característiques locals dels jacobians del flux i projeccions de les variables d'estat i fluxos en camps característics. Per al cas de cinquè ordre la fórmula s'escriu com:

$$\begin{aligned} \hat{f}_{i+\frac{1}{2}} = &\sum_{k=1}^{m} r^{+,k} \left(\mathcal{R}^{+} \left(l^{+,k} \cdot f^{+,k}_{i-2}, \ldots, l^{+,k} \cdot f^{+,k}_{i+2}; x_{i+\frac{1}{2}}\right)\right) \\ &+ \sum_{k=1}^{m} r^{-,k} \left(\mathcal{R}^{-} \left(l^{-,k} \cdot f^{-,k}_{i-1}, \ldots, l^{-,k} \cdot f^{-,k}_{i+3}; x_{i+\frac{1}{2}}\right)\right), \end{aligned} \tag{11}$$

on $f^{\pm,k}_l = f^{\pm,k}(u_l)$ es defineix més endavant, $r^{\pm,k} = r^k(u^{\pm}_{i+\frac{1}{2}}), l^{\pm,k} = l^k(u^{\pm}_{i+\frac{1}{2}})$ són els vectors propis normalitzats dreta i esquerra corresponents al valor propi $\lambda_k(f'(u^{\pm}_{i+\frac{1}{2}}))$ del jacobià del flux $f'(u^{\pm}_{i+\frac{1}{2}})$, respectivament, calculat a $u^{\pm}_{i+\frac{1}{2}}$, on

$$u^{+}_{i+\frac{1}{2}} = \mathcal{I}^{+}(u_{i-2}, \ldots, u_{i+2}; x_{i+\frac{1}{2}}), \quad u^{-}_{i+\frac{1}{2}} = \mathcal{I}^{-}(u_{i-1}, \ldots, u_{i+3}; x_{i+\frac{1}{2}}),$$

per a certs interpoladors $I^{\pm}$. $f^{\pm,k}$ satisfan $f^{+,k}+f^{-,k}=f$, $\pm\lambda_k((f^{\pm,k})'(u))>0$ per a u en algun rang rellevant $\mathcal{M}_{i+\frac{1}{2}}$ prop de $u^{\pm}_{i+\frac{1}{2}}$, i venen donats per:

$$(f^{-,k},f^{+,k})(v)=\begin{cases}(0,f(v)), & \lambda_k(f'(u))>0, \quad \forall u\in\mathcal{M}_{i+\frac{1}{2}}\\ (f(v),0), & \lambda_k(f'(u))<0, \quad \forall u\in\mathcal{M}_{i+\frac{1}{2}}\\ (F_{-\alpha^k_{i+\frac{1}{2}}}(v),F_{\alpha^k_{i+\frac{1}{2}}}(v)), & \exists u\in\mathcal{M}_{i+\frac{1}{2}}/\lambda_k(f'(u))=0,\end{cases}$$

on $\alpha^k_{i+\frac{1}{2}}\geq|\lambda_k(f'(u))|$ per a $u\in\mathcal{M}_{i+\frac{1}{2}}$ i $F_\alpha(v)=\frac{1}{2}(f(v)+\alpha v)$.

Esquemes WENO en diferències finites

Les reconstruccions WENO apareixen per primera vegada en [35]. Expliquem el seu funcionament tot seguit. Denotem $h=\Delta x$. Si f és suau en l'stencil $S^{2r-1}_{j+r-1}=\{x_{j-r+1},\ldots,x_{j+r-1}\}$, aleshores pot calcular-se una aproximació d'ordre $(2r-1)$ al punt $x_{j+\frac{1}{2}}$ a partir del polinomi p^{2r-1}_{r-1} que reconstrueix f (f i p^{2r-1}_{r-1} tenen les mateixes mitjanes en cel.la) a eixe stencil:

$$p^{2r-1}_{r-1}(x_{j+\frac{1}{2}})=f(x_{j+\frac{1}{2}})+\mathcal{O}\left(h^{2r-1}\right).$$

Si considerem els r possibles substencils $S^r_{j+k}=\{x_{j-r+1+k},\ldots,x_{j+k}\}$, $k=0,\ldots,r-1$, de grandària r de S^{2r-1}_{j+r-1} i les seues corresponents reconstruccions polinòmiques de grau $r-1$, $p^r_k(x)$, complint $p^r_k(x_{j+\frac{1}{2}})=f(x_{j+\frac{1}{2}})+\mathcal{O}(h^r)$, aleshores una reconstrucció WENO (esbiaixada cap a l'esquerra) de f ve donada per la combinació convexa:

$$q(x_{j+\frac{1}{2}})=\sum_{k=0}^{r-1}w_kp^r_k(x_{j+\frac{1}{2}}), \tag{12}$$

on:

$$w_k\geq 0,\ k=0,\ldots,r-1, \qquad \sum_{k=0}^{r-1}w_k=1.$$

i la corresponent avaluació de l'operador de reconstrucció (esbiaxada cap a l'esquerra) ve donada per:

$$\mathcal{R}(\bar{f}_{j-r+1},\ldots,\bar{f}_{j+r-1})=\sum_{k=0}^{r-1}\omega_{j,k}p^r_{j,k}(x_{j+\frac{1}{2}}).$$

Els pesos haurien de triar-se amb l'objectiu d'assolir el màxim ordre de precisió possible, $2r-1$, allà on f siga suau, i ordre r a la resta de zones.

De la mateixa manera que en la versió WENO original [35], primer notem que per a $r \geq 2$, es poden calcular uns coeficients C_k^r, anomenats pesos òptims, tals que

$$p_{r-1}^{2r-1}(x_{j+\frac{1}{2}}) = \sum_{k=0}^{r-1} C_k^r p_k^r(x_{j+\frac{1}{2}}),$$

on

$$C_k^r \geq 0 \; \forall k, \qquad \sum_{k=0}^{r-1} C_k^r = 1.$$

En [2], Aràndiga et al. proporcionen diferents fórmules explícites per a les reconstruccions polinòmiques i els pesos òptims.

Notem que per tal d'acomplir els requeriments dels pesos no lineals w_k és suficient definir-los de manera que es verifique la condició:

$$w_k = C_k + \mathcal{O}(h^m), \qquad k = 0, \dots, r, \tag{13}$$

amb $m \leq r-1$. Aleshores, es verifica (veure [2], [35]) que

$$f(x_{j+\frac{1}{2}}) - q(x_{j+\frac{1}{2}}) = \mathcal{O}(h^{r+m}), \tag{14}$$

i, si $m = r-1$ en (13), llavors l'aproximació (14) té ordre òptim $2r-1$.

Un altre requeriment que han de verificar els pesos és que aquells que es corresponen amb polinomis construïts emprant stencils on la funció presenta una singularitat haurien de ser molt petits, de manera que la reconstrucció WENO no té en compte eixos polinomis.

A [35] es defineixen pesos satisfent la combinació de condicions anteriors com se segueix:

$$w_k = \frac{\alpha_k}{\sum_{i=0}^{r-1} \alpha_i}, \quad \alpha_k = \frac{C_k^r}{(\varepsilon + I_k)^p}, \quad k = 0, \dots, r-1, \tag{15}$$

on $p \in \mathbb{N}$, C_k^r són els pesos òptims, $I_k = I_k(h)$ és un indicador de suavitat de la funció f a l'stencil S_k i ε és un nombre positiu i petit, possiblement depenent de h, introduït per tal d'evitar denominadors nuls, però, com veurem més endavant en aquesta tesi, té una forta influència en el rendiment global de les aproximacions a punts crítics i discontinuïtats. D'acord amb la definició, els pesos satisfan $\sum_k \omega_k = 1$ independentment de la tria dels indicadors de suavitat.

Emprem els indicadors de suavitat de Jiang i Shu (veure [25]):

$$I_k = \sum_{l=1}^{r-1} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} h^{2l-1} (p_k^{(l)}(x))^2 dx, \tag{16}$$

El terme h^{2l-1} fou introduït per a evitar factors depenents de h a les derivades dels polinomis de reconstrucció $p_k(x)$.

Mètodes de Runge-Kutta TVD

En acabar el procediment de semidiscretització espacial, resolem el sistema d'equacions diferencials ordinàries resultant

$$\frac{du_j(t)}{dt} + \mathcal{D}(u(t))_j = 0, \quad \mathcal{D}(u(t))_j = \frac{\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}}{\Delta x}, \quad \forall j, \tag{17}$$

emprant un resoledor per a equacions diferencials ordinàries (EDO). Entre els més freqüentment emprats en aquest context, trobem els mètodes de Runge-Kutta TVD desenvolupats per Shu i Osher en [41]. La formulació general és la següent:

$$\begin{cases} u^{(0)} = u^n, \\ u^{(i)} = \displaystyle\sum_{k=0}^{i} \left(\alpha_{ik} u^{(k)} - \beta_{ik} \Delta t \mathcal{D}(u^{(k)}) \right), \quad 1 \leq i \leq \bar{r}, \\ u^{n+1} = u^{(\bar{r})}, \end{cases}$$

on $\bar{r}$ depèn de l'ordre de precisió de l'esquema de Runge-Kutta particular i α_{ik}, β_{ik} són els coeficients que també depenen del mètode (per a més detalls, veure [41, 42]). Específicament, en aquest treball emprem la versió de tercer ordre:

$$\begin{cases} u^{(1)} = u^n - \Delta t \mathcal{D}(u^n), \\ u^{(2)} = \dfrac{3}{4} u^n + \dfrac{1}{4} u^{(1)} - \dfrac{1}{4} \Delta t \mathcal{D}(u^{(1)}), \\ u^{n+1} = \dfrac{1}{3} u^n + \dfrac{2}{3} u^{(2)} - \dfrac{2}{3} \Delta t \mathcal{D}(u^{(2)}). \end{cases} \tag{18}$$

Notem que podem desenvolupar l'expressió de (18) com:

$$\begin{aligned} u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x} \bigg[& \left(\frac{1}{6} \hat{f}_{j+\frac{1}{2}}(u^n) + \frac{1}{6} \hat{f}_{j+\frac{1}{2}}(u^{(1)}) + \frac{2}{3} \hat{f}_{j+\frac{1}{2}}(u^{(2)}) \right) \\ & - \left(\frac{1}{6} \hat{f}_{j-\frac{1}{2}}(u^n) + \frac{1}{6} \hat{f}_{j-\frac{1}{2}}(u^{(1)}) + \frac{2}{3} \hat{f}_{j-\frac{1}{2}}(u^{(2)}) \right) \bigg]. \end{aligned} \tag{19}$$

Com que $u^{(1)}$ i $u^{(2)}$ s'obtenen a partir de u^n, podem escriure (19) de manera conservativa:

$$u_j^{n+1} = u_j^n - \Delta t \left(\hat{f}^{RK3}_{j+\frac{1}{2}}(u^n) - \hat{f}^{RK3}_{j-\frac{1}{2}}(u^n) \right),$$

on el flux numèric ve donat per

$$\hat{f}^{RK3}(u^n) = \frac{1}{6}\hat{f}(u^n) + \frac{1}{6}\hat{f}(u^{(1)}) + \frac{2}{3}\hat{f}(u^{(2)}).$$

L'error local de truncament de l'esquema completament discret està relacionat amb els errors locals de truncament de la semidiscretització espacial i la del resoledor d'EDOs per a la discretització temporal.

Mallat de dominis complexos

La primera pregunta que s'esdevé a l'hora d'abordar un problema amb condicions de frontera és com discretitzar aquestes de manera que el mètode numèric puga combinar adequadament les condicions de frontera requerides amb dades procedents dels nodes interiors.

Per a fer-ho, proposem un procediment de mallat basat en el càlcul d'interseccions entre les rectes del mallat, contenint informació nodal i de la frontera $\partial\Omega$. En aquest treball, ens centrarem en la resolució de sistemes en dues dimensions de lleis de conservació amb dominis complexos.

Suposem en primer lloc que $\Omega \subseteq \mathbb{R}^2$ és un domini simplement connex tal que $\exists \alpha : [a, b] \to \mathbb{R}^2$, $\alpha \in C^2$ a trossos corba tancada (és a dir, $\alpha(a) = \alpha(b)$) tal que $\alpha([a, b]) = \partial\Omega$. En un cas més general, podem considerar dominis la frontera dels quals és la unió d'un nombre finit de corbes tancades.

Detecció segura d'interseccions

Hi ha moltes maneres de mallar un conjunt, però com que els mètodes numèrics emprats en aquest treball per a resoldre les equacions físiques són esquemes de diferències finites ens centrarem en el cas de malles cartesianes, és a dir, malles les cel·les de les quals són rectangulars i idènticament distribuïdes. Per tant, hem de desenvolupar una estratègia per a automatitzar el procediment de mallat, ço és, el càlcul de totes les interseccions de la frontera amb les rectes de la malla, cel·les fantasma, rectes normals, etc. independentment de la parametrització de la corba.

Siga $\alpha : [a, b] \to \mathbb{R}^2$ una corba $C^2([a, b], \mathbb{R}^2)$ tal que $\alpha(a) = \alpha(b)$. Suposem que volem establir una malla al seu interior amb rectes horitzontals i verticals de la forma $x_k = x_0 + kh_x$ i $y_k = y_0 + kh_y$, $k \in \mathbb{Z}$, amb $h_x, h_y > 0$ els espaiats vertical i horitzontal, respectivament. Tot seguit il·lustrem el procediment de cerca d'interseccions de les rectes horitzontals del mallat.

Siga L_1 una fita superior de la primera derivada de α_2 en $[a, b]$ (és a dir una constant de Lipschitz) i L_2 una fita superior de la segona derivada de α_2 en $[a, b]$. Llavors es pot comprovar que efectuant el salt de paràmetre

$$\Delta s = \max\{\Delta s_1, \Delta s_2\}, \quad \text{on} \quad \Delta s_1 = \frac{h_y}{L_1}, \quad \Delta s_2 = \frac{2h_y}{\sqrt{|\alpha_2'(s_0)|^2 + 2L_2 h_y} + |\alpha_2'(s_0)|}$$

es garanteix la detecció de totes i cadascuna de les possibles interseccions de la malla amb $\partial\Omega$.

Mètode de Newton amb control de bisecció

El procediment previ redueix el problema del càlcul d'interseccions a la cerca d'arrels d'una funció contínua dins d'un cert interval. Per a ser més precisos, si tenim $y_0 + kh_y \leq \alpha_2(s_0) \leq y_0 + (k+1)h_y$ i $y_0 + (k+1)h_y \leq \alpha_2(s_1) \leq y_0 + (k+2)h_y$ llavors pel teorema de Bolzano $\exists c \in [s_0, s_1]$ tal que $\alpha_2(c) = y_0 + (k+1)h_y$. Si definim $f(s) = \alpha_2(s) - (y_0 + (k+1)h_y)$, aleshores $f(s_0) \leq 0$, $f(s_1) \geq 0$ i c és una arrel de f, és a dir, $f(c) = 0$. Per tant, aquest problema pot traduir-se a la cerca d'una arrel de la funció f. Note's que f és una funció de classe $\mathcal{C}^2([s_0, s_1])$ que canvia de signe a eixe interval, i per tant pel teorema de Bolzano $\exists c \in (s_0, s_1)$ tal que $f(c) = 0$. Denotem per simplicitat $[a, b] = [s_0, s_1]$. El procediment es descriu tot seguit.

Inicialment, prenem $a_0 = a$, $b_0 = b$ tot complint-se $f(a_0)f(b_0) < 0$ i com a punt inicial $x_0 = \frac{a+b}{2}$. Suposem que estem en el pas n, amb $a_{n-1} \leq a_n < b_n \leq b_{n-1}$ complint-se $f(a_n)f(b_n) < 0$ i x_n com a aproximació de c. Aleshores calculem x_{n+1} bé siga pel mètode de Newton o el mètode de la secant i aleshores si $f(x_{n+1})$ està suficientment prop de zero llavors el procediment s'atura ja que s'ha trobat l'arrel. Altrament, considerem els dos casos següents:

- Si $|f(x_{n+1})| > \dfrac{|f(x_n)|}{2}$ o $x_{n+1} \notin [a_n, b_n]$, aleshores el ràtio de convergència és, en termes generals, pitjor que el mètode de la bisecció o bé el nou punt de la iteració està fora dels límits de seguretat, i aleshores redefinim x_{n+1} com $x_{n+1} = \dfrac{a_n + b_n}{2}$.

- Si $|f(x_{n+1})| \leq \dfrac{|f(x_n)|}{2}$ i $x_{n+1} \in [a_n, b_n]$, aleshores el ràtio de convergència local és més o menys el mateix o millor que el mètode de la bisecció i el punt es troba dins de la zona de seguretat, i per tant mantenim el valor de x_{n+1}.

En ambdós casos, si sign$(f(x_{n+1})) =$ sign$(f(a_n))$ aleshores definim $a_{n+1} = x_{n+1}$ i $b_{n+1} = b_n$; altrament, si sign$(f(x_{n+1})) =$ sign$(f(b_n))$ aleshores definim $a_{n+1} = a_n$ i $b_{n+1} = x_{n+1}$.

El procediment s'atura quan $|f(x_{n+1})|$ o $|b_{n+1} - a_{n+1}|$ estan per baix d'una certa tolerància. Aquest algorisme garanteix que el ràtio de convergència és almenys el que proporciona el mètode de la bisecció.

Cel·les fantasma

Els esquemes WENO d'ordre senar, $2\ell - 1$, empren un stencil (conjunt d'índexs consecutius) de 2ℓ punts, per tant es necessiten ℓ cel·les addicionals a ambdós costats de cada cel·la per tal d'efectuar un pas temporal. Per a cel·les properes a la frontera alguna d'aquestes cel·les addicionals pot trobar-se fora del domini computacional i en aquest cas reben el nom de *cel·les fantasma* i, en termes dels seus centres, venen donades per:

$$\mathcal{GC} := \mathcal{GC}_x \cup \mathcal{GC}_y,$$

on

$$\mathcal{GC}_x := \{(x_r, y_s) : \ \ 0 < d\left(x_r, \, \Pi_x\left(\Omega \cap (\mathbb{R} \times \{y_s\})\right)\right) \leq kh_x, \quad r, s \in \mathbb{Z}\},$$

$$\mathcal{GC}_y := \{(x_r, y_s) : \ \ 0 < d\left(y_s, \, \Pi_y\left(\Omega \cap (\{x_r\} \times \mathbb{R})\right)\right) \leq kh_y, \quad r, s \in \mathbb{Z}\},$$

on Ω és el domini computacional, Π_x i Π_y denoten les projeccions a les respectives coordenades i

$$d(a, B) := \inf\{|b - a| : \ b \in B\},$$

per a un $a \in \mathbb{R}$ donat i $B \subseteq \mathbb{R}$. Notem que $d(a, \emptyset) = +\infty$, ja que, per conveni, $\inf \emptyset = +\infty$.

Rectes normals

Ens centrem ara en la configuració 2D i fronteres amb condicions Dirichlet, com ara condicions reflectants per a les equacions d'Euler. En aquesta situació, sembla raonable que l'extrapolació a una determinada cel·la fantasma $P = (x_*, y_*) \in \mathcal{GC}$ estiga basada en el valor especificat al punt de la frontera més proper. Es pot provar que un punt $P_0 \in \partial\Omega$ que compleix

$$\|P - P_0\|_2 = \min\{\|P - B\|_2 : \quad B \in \partial\Omega\}$$

també satisfà que la recta determinada per P i P_0 és normal a la corba $\partial\Omega$ a P_0 si $\partial\Omega$ és diferenciable en P_0. Dit d'una altra manera, suposant que $P_0 = \alpha(s_*)$, es verifica la següent condició:

$$\langle P - P_0, \alpha'(s_*)\rangle = 0.$$

Açò indueix un procediment iteratiu per a aproximar automàticament a través del mètode de Newton o de la secant la recta normal associada a cada node fantasma P mitjançant la cerca de l'arrel de la funció

$$F_P(s) = \langle P - \alpha(s), \alpha'(s)\rangle.$$

La unicitat de P_0 es té sempre que P estiga suficientment prop de la frontera; en tal cas, denotem $N(P) = P_0$.

L'argument anterior suggereix que una bona estratègia és realitzar una rotació (virtual) del domini i obtenir dades en alguns punts $N_i \in \Omega$ de la recta que passa per P i $N(P)$ (recta normal a $\partial\Omega$ que passa per P) i aleshores emprar una extrapolació unidimensional de les dades en eixos punts al segment per a aproximar el valor de P.

Els punts N_q, $1 \leq q \leq R + 1$, s'obtenen a partir d'interpolació o extrapolació unidimensional d'una filera de dades nodals del domini computacional, a partir d'un conjunt de punts $N_{q,i} \in \mathcal{D}$, $1 \leq q, i \leq R + 1$, que es troben a la mateixa coordenada x o y, en funció de l'angle de la recta normal de manera que la distància total siga la mínima possible.

En cas de condicions Dirichlet, anomenant $P_0 = N(P)$, cal efectuar un pas intermig entre l'obtenció dels N_q, $1 \leq q \leq R + 1$, i l'extrapolació a P per tal que l'stencil final d'extrapolació siga equiespaiat. Per a fer-ho, s'extrapola la informació obtinguda al primer pas dels nodes de la recta normal N_q a nous punts P_q, $1 \leq q \leq R$, de manera que juntament amb P_0 formen un stencil equiespaiat.

En el cas que les condicions de frontera proporcionen valors per a la component normal d'una incògnita vectorial $\overrightarrow{v}$ relacionada amb el sistema de coordenades (com és el cas de les condicions de frontera reflectants per a les equacions d'Euler), aleshores hom defineix

$$\overrightarrow{n} = \frac{P - N(P)}{\|P - N(P)\|}, \quad \overrightarrow{t} = \overrightarrow{n}^{\perp},$$

i obté les components normal i tangencial de $\overrightarrow{v}$ a cada punt N_i del segment esmentat per:

$$v^t(N_i) = \overrightarrow{v}(N_i) \cdot \overrightarrow{t}, \quad v^n(N_i) = \overrightarrow{v}(N_i) \cdot \overrightarrow{n}.$$

El procediment d'extrapolació s'aplica a $v^t(N_i)$ per tal d'aproximar $v^t(P)$ i a $v^n(N_i)$ i $v^n(N(P)) = 0$ per a aproximar $v^n(P)$. Una vegada obtingudes les aproximacions dels valors $v^t(P), v^n(P)$, l'aproximació a $\overrightarrow{v}(P)$ s'estableix com

$$\overrightarrow{v}(P) = v^t(P)\overrightarrow{t} + v^n(P)\overrightarrow{n}.$$

Notem que tots els passos del procediment d'extrapolació d'informació a les cel·les fantasma descrit anteriorment requereix únicament la realització d'interpolacions o extrapolacions unidimensionals.

Tècniques d'extrapolació per a condicions de frontera numèriques

Per tal d'aconseguir un esquema espacial completament d'alt ordre, tenint en compte la possible formació o posicionament eventual d'una discontinuïtat prop de la frontera, s'ha de tindre especial cura a l'hora de plenar les cel·les fantasma mitjançant condicions de frontera numèriques, ja que l'interpolació/extrapolació pot produir errors grans si hi ha una discontinuïtat en la regió determinada pels nodes d'interpolació i el punt d'avaluació. A l'hora d'implementar l'extrapolació a les cel·les fantasma, per tal d'evitar aquesta pèrdua de precisió considerable o fins i tot una fallida completa de la simulació, és necessari tractar aquesta qüestió amb molt de compte.

Alguns autors han abordat aquest problema des de perspectives diferents. En [43] els autors desenvolupen una tècnica basada en interpolació de Lagrange amb un limitador el qual està restringit a mètodes de segon ordre amb una única cel·la fantasma. Podem trobar altres articles que també estan relacionats amb el nostre procediment, com ara els treballs de Shu i col·laboradors, [45, 46], on l'equació a resoldre s'empra per a extrapolar valors de les derivades de la solució numèrica a punts de la frontera on hi ha establides condicions inflow i aleshores aproximen valors fantasma mitjançant un desenvolupament de Taylor. Per a fronteres outflow, s'empra una tècnica d'extrapolació basada en el mètode WENO, tot assolint alt ordre quan les dades són suaus en ambdós casos. Els inconvenients d'aquest procediment és que el mètode depèn del problema i requereix tractaments diferents per a diferents tipus de frontera, a més de tenir un cost computacional relativament alt.

En aquest treball s'introdueixen noves tècniques per a l'extrapolació d'informació interior a cel·les fantasma mitjançant condicions de frontera (en cas d'haver-ne) i dades interiors properes a una cel·la fantasma donada. Aquest procediment és capaç de detectar canvis abruptes a les dades.

Totes les tècniques d'extrapolació a la frontera descrites en aquesta secció ens asseguren l'obtenció d'aproximacions de l'ordre adequat sota certes condicions.

Selecció d'stencils mitjançant *thresholding*

Suposem que tenim informació en un stencil amb nodes no necessàriament equiespaiats, $x_0 < \cdots < x_R$ $(R \geq r)$, amb valors nodals corresponents $u_i = u(x_i)$, i que volem interpolar a un cert node x^*.

El node clau a partir del qual establim un criteri de proximitat en el seu corresponent valor nodal és el node interior més proper a x^*, és a dir, triem el node x_{i_0}, $i_0 \in \{0, \ldots, R\}$ tal que:

$$i_0 = \operatorname*{argmin}_{0 \leq i \leq R} |x_i - x^*|.$$

L'objectiu és ara aproximar el valor que el node x^* hauria de tenir, basant-se en la informació del substencil "més suau" i el node x_{i_0}.

Siga r el grau dels polinomis interpoladors que utilitzarem en la reconstrucció, que satisfà $r+1 \leq \lceil \frac{R+1}{2} \rceil$. Aleshores tenim $R-r+1$ substencils possibles:

$$S_m = \{x_m, \ldots, x_{m+r}\}, \quad 0 \leq m \leq R - r.$$

Denotem per $p_m(x)$ l'interpolador associat a l'stencil S_m, $0 \leq m \leq R-r$. Si hi ha suavitat suficient a tot l'stencil, llavors es té

$$u(x_i) - p_m(x_i) \quad = \quad \mathcal{O}(h_x^{r+1}), \quad i = 1, \ldots, R, \tag{20}$$

i per tant

$$u(x_i) = u(x_{i_0}) + (p_m(x_i) - p_m(x_{i_0})) + \mathcal{O}(h_x^{r+1}).$$

Aleshores seleccionem el substencil que satisfà:

$$m_0 := \operatorname*{argmin}_{0 \leq m \leq R-r} \sum_{k=1}^{r} \int_{x_m}^{x_{m+r}} (x_{m+r} - x_m)^{2k-1} p_m^{(k)}(x)^2 dx, \tag{21}$$

i definim

$$v_i := u_{i_0} + (p_{m_0}(x_i) - p_{m_0}(x_{i_0})). \tag{22}$$

Finalment, siga $\delta \in (0, 1]$ un *threshold* i definim el conjunt d'índexs I_δ per

$$I_\delta := \{i \in \{0, \dots, R\} : \quad \delta\left(|u_i - u_{i_0}| + D(x_i)\right) \leq |v_i - u_{i_0}| + D(x_i)\}, \tag{23}$$

on

$$D(x) := \sum_{j=1}^{r} \left|(x - x_{i_0})^j p_{m_0}^{(j)}(x_{i_0})\right|.$$

Notem que $I_\delta \neq \emptyset$, ja que $i_0 \in I_\delta$. L'stencil que utilitzarem per interpolar és el substencil de I_δ de grandària $r+1$ (o inferior si I_δ no conté suficients punts) composat per i_0 i els punts més propers.

Com a darrer filtre (opcional), si u^* és el valor obtingut a partir de l'interpolació Lagrange de l'stencil resultant, $S = \{x_i : \quad i \in I_\delta\}$, aleshores el mateix criteri de threshold pot aplicar-se a eixe valor, tot resultant en un valor d'extrapolació definitiu:

$$u^*_{\text{def}} = \begin{cases} u^* & \text{if} \quad \delta'\left(|u^* - u_{i_0}| + D(x^*)\right) \leq |p_m(x^*) - p_m(x_{i_0})| + D(x^*) \\ u_{i_0} & \text{if} \quad \delta'\left(|u^* - u_{i_0}| + D(x^*)\right) > |p_m(x^*) - p_m(x_{i_0})| + D(x^*) \end{cases} \tag{24}$$

amb $0 \leq \delta' \leq 1$.

Extrapolació amb pesos

Considerem un stencil de nodes equiespaiats $x_0 < \cdots < x_r$ amb valors nodals corresponents $u_j = u(x_j)$. Denotem per $J = \{0, \dots, r\}$ i $X = \{x_j\}_{j \in J}$ i siga x_* el node el qual volem interpolar i j_0 el node interior més proper a x_*, ço es,

$$j_0 = \underset{j \in J}{\text{argmin}} |x_j - x_*|.$$

L'objectiu és novament aproximar el valor que x_* hauria de tenir, basat en la informació de l'stencil "més suau" i el node x_{j_0}. Definim inductivament el següent conjunt d'índexs:

$$J_0 = \{j_0\}, \text{ i } X_0 = \{x_j\}_{j \in J_0} = \{x_{j_0}\}.$$

Suposem que tenim definit $J_k = \{j_k, \dots, j_k + k\}$; aleshores J_{k+1} ve definit seguint un procediment ENO per

$$J_{k+1} = \begin{cases} \{j_k - 1\} \cup J_k & \text{si } j_k > 0 \wedge [u_{j_k - 1}, \dots, u_{j_k + k}] \leq [u_{j_k}, \dots, u_{j_k + k + 1}] \\ J_k \cup \{j_k + k + 1\} & \text{si } j_k < r - k \wedge [u_{j_k}, \dots, u_{j_k + k + 1}] < [u_{j_k - 1}, \dots, u_{j_k + k}] \end{cases}$$

i

$$X_{k+1} = \{x_j\}_{j\in J_{k+1}},$$

on $[v_1, \ldots, v_\ell]$ representa la diferència no dividida de $v_1, \ldots, v_\ell$. Per construcció, és clar que el conjunt X_k pot escriure's com una successió de nodes amb índexs consecutius, és a dir, stencils:

$$X_k = \{x_{i_k+j}\}_{j=0}^{k}$$

per a algun $0 \leq i_k \leq r-k$, $0 \leq k \leq r$.

Ara, per a cada k, $0 \leq k \leq r$, definim p_k com el polinomi interpolador de grau com a molt k tal que $p_k(x_{i_k+j}) = u_{i_k+j}$, $\forall j$, $0 \leq j \leq k$. Donat un conjunt de pesos $\{\omega_k\}_{k=1}^{r}$ tals que $0 \leq \omega_k \leq 1$, definim la recurrència següent:

$$\begin{array}{rcl} u_*^{(0)} & = & p_0(x_*) = u_{i_0}, \\ u_*^{(k)} & = & (1-\omega_k)u_*^{(k-1)} + \omega_k p_k(x_*), \quad 1 \leq k \leq r. \end{array} \tag{25}$$

On els pesos ω_k estan construïts de manera que són essencialment 1 si X_k és un stencil dins d'una zona suau i essencialment 0 en altre cas.

El resultat final de l'extrapolació es defineix aleshores per $u_* := u_*^{(r)}$, que es pren com a aproximació del valor $u(x_*)$.

Existeixen diferents maneres de construir els pesos i variacions del mètode. De manera resumida, tenim les següents:

- **Pesos simples:** Venen donats per

$$\begin{aligned} \omega_k &= 1 - \left(1 - \left(\frac{IS_k}{I_k}\right)^{s_1}\right)^{s_2}, \quad 1 \leq k \leq r_0, \\ \omega_k &= \min\left\{1 - \left(1 - \left(\frac{IS_{r_0}}{I_k}\right)^{s_1}\right)^{s_2}, 1\right\}, \quad r_0 + 1 \leq k \leq r. \end{aligned} \tag{26}$$

- **Pesos millorats:** Es defineixen per

$$\omega_k = \frac{1}{1+\rho_k}, \tag{27}$$

on

$$\rho_k = \tau_k \left(\frac{1-\sigma_k}{\sigma_k}\right)^d, \quad d \geq \frac{r}{2},$$

$$\tau_k := \frac{I_k}{IM_k},$$

$$\sigma_k := \min\left\{\frac{IS_{\min\{k,r_0\}}}{I_k}, 1\right\},$$

i

$$IM_k = \max_{0\leq j\leq r-k} \frac{1}{r}\sum_{\ell=1}^{r_0}\int_{x_0}^{x_r} h^{2\ell-1} q_{k,j}^{(\ell)}(x)^2 dx, \quad 1 \leq k \leq r_0.$$

- **Pes únic:** El resultat d'extrapolació és en aquest cas

  $$u_* = (1-\omega)p_0(x_*) + \omega p_r(x_*) = (1-\omega)u_{i_0} + \omega p_r(x_*),$$

  on ω ve donat per

  $$\omega := (1-(1-\rho)^{s_1})^{s_2}, \tag{28}$$

  amb

  $$\rho := \frac{(r-r_0+1)^2}{\left(\sum_{j=0}^{r-r_0} I_{r_0,j}^m\right)\left(\sum_{j=0}^{r-r_0}\frac{1}{I_{r_0,j}^m}\right)}.$$

Per raons d'estabilitat, aquestes extrapolacions es combinen amb mínims quadrats a zones suaus:

$$u_* := \omega z_* + (1-\omega)v_*, \tag{29}$$

on v_* és el resultat obtingut a partir d'alguna de les dues primeres tècniques o, en el cas del pes únic, $v_* = u_{i_0}$, i ω és el pes donat per l'equació (28).

D'altra banda, z_* és el resultat d'aplicar mínims quadrats a un stencil de grandària $R \geq r$ contenint l'stencil original, a partir d'un polinomi de grau r; és a dir, és el resultat de resoldre $p(x_i) = u_i$, $0 \leq i \leq R$ per mínims quadrats i avaluar en x_*.

Esquemes d'alt ordre temporal

Una vegada completada la formulació d'un esquema d'alt ordre espacial en un context general, presentem un esquema d'alt ordre temporal, que en combinació amb les tècniques anteriors dóna lloc a un esquema d'alt ordre general. Per a la seua derivació prenem com a referència l'esquema presentat per Qiu i Shu en 2003 [39], basat en la conversió de derivades temporals a derivades espacials mitjançant la tècnica de Cauchy-Kowalewski, a través del procediment de Lax-Wendroff.

Per raons de simplicitat, comencem amb el cas unidimensional per a una equació escalar ($d = m = 1$). Per a la solució $u(x,t)$ de $u_t + f(u)_x = 0$ en una malla espacial fixada (x_i), amb espaiat $h = x_{i+1} - x_i$ i un cert temps t_n, a partir d'una malla temporal amb espaiat $\delta = \Delta t = t_{n+1} - t_n > 0$, proporcional a h, $\delta = \tau h$, on τ respon a restriccions d'estabilitat (condició CFL), emprem la següent notació per a les derivades temporals de u i $f(u)$:

$$\begin{aligned} u_{i,n}^{(l)} &= \frac{\partial^l u(x_i, t_n)}{\partial t^l}, \\ f_{i,n}^{(l)} &= \frac{\partial^l f(u)(x_i, t_n)}{\partial t^l}. \end{aligned}$$

El nostre objectiu és el d'obtenir un esquema d'avanç temporal d'ordre R, és a dir, un esquema amb un error local de truncament d'ordre $R+1$, basat en el desenvolupament de Taylor de la solució u des del temps t_n fins al següent temps t_{n+1}:

$$u_i^{n+1} = \sum_{l=0}^{R} \frac{\Delta t^l}{l!} u_{i,n}^{(l)} + \mathcal{O}(\Delta t^{R+1}).$$

Per tal d'aconseguir-ho, definim les corresponents aproximacions

$$\begin{aligned} \widetilde{u}_{i,n}^{(l)} &= u_{i,n}^{(l)} + \mathcal{O}(h^{R+1-l}), \\ \widetilde{f}_{i,n}^{(l)} &= f_{i,n}^{(l)} + \mathcal{O}(h^{R-l}), \end{aligned}$$

per recurrència sobre l, suposant (per una anàlisi de l'error local de truncament) que $\widetilde{u}_{i,n}^{0} = u_{i,n}^{(0)} = u(x_i, t_n)$. L'aproximació $\widetilde{u}_{i,n}^{1}$ es definirà de manera separada.

El fet que u siga solució d'un sistema de lleis de conservació implica que les derivades temporals $u_{i,n}^{(l)}$, $1 \leq l \leq R$, poden escriure's en termes de les derivades espacials d'algunes funcions de $u_{i,n}^{(j)}$, $j < l$,

$$f_{i,n}^{(l-1)} = F_{l-1}(u_i^n, u_{i,n}^{(1)}, \ldots, u_{i,n}^{(l-1)}), \tag{30}$$

seguint el procediment de Cauchy-Kowalewski (o de Lax-Wendroff de segon ordre):

$$\frac{\partial^l u}{\partial t^l} = \frac{\partial^{l-1}}{\partial t^{l-1}}\big(u_t\big) = -\frac{\partial^{l-1}}{\partial t^{l-1}}\big(f(u)_x\big) = -\left[\frac{\partial^{l-1} f(u)}{\partial t^{l-1}}\right]_x, \tag{31}$$

i la fórmula de Faà di Bruno establida al Teorema 2.

Específicament, per a aproximar la primera derivada temporal, $u_t = -f(u)_x$, emprem l'esquema de les diferències finites de Shu-Osher [42] amb reconstruccions espacials WENO upwind d'ordre $2r-1$ del flux.

$$u_{i,n}^{(1)} = u_t(x_i, t_n) = -[f(u)]_x(x_i, t_n) = -\frac{\hat{f}_{i+\frac{1}{2}}^n - \hat{f}_{i-\frac{1}{2}}^n}{h} + \mathcal{O}(h^{2r-1}). \tag{32}$$

Per a les següents derivades s'empren al seu lloc diferències centrades, molt més assequibles en termes de cost computacional.

Així doncs, la segona derivada es calcula per

$$u_{tt} = [u_t]_t = [-f(u)_x]_t = -[f(u)_t]_x = -[f'(u)u_t]_x,$$

on $f'(u)u_t$ és ara una expressió que es coneix de manera aproximada als nodes requerits. Emprem aleshores una diferència centrada de segon ordre per tal d'obtenir l'aproximació:

$$\widetilde{u}^{(2)}_{i,n} = -\frac{\widetilde{f}^{(1)}_{i+1,n} - \widetilde{f}^{(1)}_{i-1,n}}{2h},$$

on

$$\widetilde{f}^{(1)}_{i,n} = F_1(\widetilde{u}^{(0)}_{i,n}, \widetilde{u}^{(1)}_{i,n}) = f'(\widetilde{u}^{(0)}_{i,n})\widetilde{u}^{(1)}_{i,n}.$$

La tercera derivada temporal s'aproxima per

$$u_{ttt} = [u_t]_{tt} = [-f(u)_x]_{tt} = -[f(u)_{tt}]_x = -\Big(f''(u)u_t^2 + f'(u)u_{tt}\Big)_x,$$

on novament la funció $f''(u)u_t^2 + f'(u)u_{tt}$ es coneix de manera aproximada als nodes i per tant u_{ttt} es pot aproximar per

$$\widetilde{u}^{(3)}_{i,n} = -\frac{\widetilde{f}^{(2)}_{j+1,n} - \widetilde{f}^{(2)}_{j-1,n}}{2h},$$

on

$$\widetilde{f}^{(2)}_{i,n} = F_2(\widetilde{u}^{(0)}_{i,n}, \widetilde{u}^{(1)}_{i,n}, \widetilde{u}^{(2)}_{i,n}) = f''(\widetilde{u}^{(0)}_{i,n}) \cdot (\widetilde{u}^{(1)}_{i,n})^2 + f'(\widetilde{u}^{(0)}_{i,n}) \cdot (\widetilde{u}^{(2)}_{i,n})^2.$$

Si, per exemple, es desitja un esquema temporal de tercer ordre, és suficient amb haver realitzat els càlculs anteriors, i podem aproximar la següent iteració temporal per

$$\widetilde{u}^{n+1}_i = \widetilde{u}^n_i + \Delta t \widetilde{u}^{(1)}_{i,n} + \frac{\Delta t^2}{2}\widetilde{u}^{(2)}_{i,n} + \frac{\Delta t^3}{6}\widetilde{u}^{(3)}_{i,n}.$$

El procediment aproximat de Lax-Wendroff

Tal i com indiquen els autors de [39], el càlcul dels valors nodals exactes de $f^{(k)}$ pot arribar a ser molt car en termes computacionals en augmentar k, ja que el nombre d'operacions augmenta exponencialment. A més, implementar-lo és costós i requereix moltes manipulacions amb eines de càlcul simbòlic per a cada equació.

Tot seguit presentem una alternativa, que és menys costosa computacionalment per a k gran i que no necessita disposar de les derivades del flux de l'equació. Aquest procediment també funciona en el cas multidimensional i en el de sistemes (treballant component a component). Aquesta tècnica està basada en l'observació que es poden obtenir fàcilment aproximacions de $\widetilde{f}^{(l-1)} \approx f^{(l-1)}$ mitjançant un ús subtil de diferències finites, en lloc d'emprar l'expressió exacta de F_{l-1} en (30).

Donada una funció $u\colon \mathbb{R} \to \mathbb{R}^m$, denotem la funció en la malla definida per un punt base a i un espaiat de mallat h per

$$G_{a,h}(u)\colon \mathbb{Z} \to \mathbb{R}^m, \quad G_{a,h}(u)_i = u(a+ih).$$

Denotem per $\Delta_h^{p,q}$ a l'operador de diferències finites que aproxima derivades p-èssimes a ordre $2q$ en malles amb espaiat h.

Volem definir aproximacions $\widetilde{u}_{i,n}^{(k)} \approx u_{i,n}^{(k)}$, $k = 0, \dots, R$, recursivament. Comencem la recursió amb

$$\begin{aligned} \widetilde{u}_{i,n}^{(0)} &= u_i^n, \\ \widetilde{u}_{i,n}^{(1)} &= -\frac{\hat{f}^n_{i+\frac{1}{2}} - \hat{f}^n_{i-\frac{1}{2}}}{h}, \end{aligned} \tag{33}$$

on $\hat{f}^n_{i+\frac{1}{2}}$ es calculen mitjançant reconstruccions WENO upwind amb partició de fluxos obtingudes a partir de les dades (u_i^n) al pas temporal n (veure [42, 11, 26] per a més detalls).

Associat a valors h, i, n fixats, una vegada obtinguts $\widetilde{u}_{i,n}^{(l)}$, $l = 0, \dots, k$, en el procés recursiu, definim el polinomi aproximat de Taylor de grau k, $T_k[h, i, n]$, per

$$T_k[h,i,n](\rho) = \sum_{l=0}^{k} \frac{\widetilde{u}_{i,n}^{(l)}}{l!} \rho^l.$$

Per a $k = 1, \dots, R-1$, definim per recurrència

$$\begin{aligned} \widetilde{f}_{i,n}^{(k)} &= \Delta_\delta^{k, \left\lceil \frac{R-k}{2} \right\rceil} \Big(G_{0,\delta}\big(f(T_k[h,i,n])\big) \Big), \\ \widetilde{u}_{i,n}^{(k+1)} &= -\Delta_h^{1, \left\lceil \frac{R-k}{2} \right\rceil} \widetilde{f}_{i+\cdot,n}^{(k)}, \end{aligned} \tag{34}$$

on denotem per $\widetilde{f}_{i+\cdot,n}^{(k)}$ el vector donat pels elements $(\widetilde{f}_{i+\cdot,n}^{(k)})_j = \widetilde{f}_{i+j,n}^{(k)}$.

Amb tots eixos ingredients, l'esquema proposat és:

$$u_i^{n+1} = u_i^n + \sum_{l=1}^{R} \frac{\Delta t^l}{l!} \widetilde{u}_{i,n}^{(l)}. \tag{35}$$

D'una banda, el següent resultat garanteix que el nostre esquema té l'ordre desitjat.

Proposició 1. *L'esquema definit per* (34) *i* (35) *té ordre* R*.*

D'altra banda, també pot provar-se que el nou esquema és conservatiu.

Teorema 2. *L'esquema resultat del procediment d'aproximació de fluxos pot escriure's de manera conservativa, és a dir,*

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{h}\big(\hat{g}^n_{i+\frac{1}{2}} - \hat{g}^n_{i-\frac{1}{2}}\big). \tag{36}$$

Control de fluctuacions

Ara ens centrem en el càlcul dels valors nodals aproximats de la derivada temporal de primer ordre. Típicament, hom empraria directament les aproximacions obtingudes mitjançant el procediment de reconstrucció upwind de les diferències finites de Shu-Osher, ço és,

$$\widetilde{u}^{(1)}_{j,n} = -\frac{\hat{f}_{j+\frac{1}{2},n} - \hat{f}_{j-\frac{1}{2},n}}{h}.$$

De fet, aquest és el procediment que es segueix al treball de Qiu-Shu [39]. No obstant això, prendre directament eixos valors com a aproximacions de la primera derivada per a aproximar les següents derivades amb procediments sense pesos, és a dir, suposant que les dades són suaus, no és segur ja que en realitat les dades no necessàriament són suaus; de fet, inclou termes d'ordre $\mathcal{O}(h^{-1})$ al voltant de les discontinuïtats, que anomenarem d'ara endavant *fluctuacions*. Aquests termes apareixen quan $\hat{f}_{j-\frac{1}{2},n}$ i $\hat{f}_{j+\frac{1}{2},n}$ procedeixen de dos costats diferents d'una discontinuïtat.

Això motiva la necessitat de calcular una aproximació nodal alternativa amb dades completament suaus, que anomenarem $\widetilde{\widetilde{u}}_i^{(1)}$, que s'empraran per a calcular les derivades següents, però no s'empraran com a termes de primer ordre del desenvolupament de Taylor per a avançar en temps, on se seguirà emprant l'aproximació conservativa i upwind de Shu-Osher de la primera derivada $\widetilde{u}^{(1)}_j = -\dfrac{\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}}{h}$.

Per a fer-ho, s'empren reconstruccions WENO centrals, que es descriuen tot seguit.

Suposem que el nostre esquema espacial empra reconstruccions WENO d'ordre $2r-1$. Considerem l'stencil de $2r-1$ punts

$$S^{2r-1}_{i+r-1} := \{i-r+1, \dots, i, \dots, i+r-1\}, \tag{37}$$

Per un i fix, siga q^r_k el polinomi interpolador de grau $\leq r-1$ tal que $q^r_k(x_j) = f_j$, $j \in S^r_{i+k} := \{i-r+1+k, \dots, i+k,\}$, $0 \leq k \leq r-1$. Tenint en compte la discussió anterior, el nostre objectiu és obtenir una aproximació de la derivada del flux $f(u)_x(x_i)$ a partir de l'stencil S^{2r-1}_{i+r-1}, que té ordre $2r-1$ si els nodes de l'stencil es troben en una regió suau. Per a fer-ho, emprem tècniques WENO, amb l'ajuda del següent resultat.

Lema 2. *Existeix un conjunt de constants* $\{c^r_k\}^r_{k=1}$ *que satisfà* $0 < c^r_k < 1$, *per a* $0 \leq k \leq r-1$, $\sum_{k=0}^{r-1} c^r_k = 1$, *tals que*

$$\sum_{k=0}^{r-1} c^r_k (q^r_k)'(x_i) = (q^{2r-1}_{r-1})'(x_i).$$

Tenint en compte aquest resultat, definim l'aproximació suavitzada com

$$\widetilde{\widetilde{u}}^{(1)}_{i,n} = -\sum_{k=1}^{r} \omega_k q'_k(x_i),$$

on

$$\omega_k = \frac{\alpha_k}{\sum_{l=1}^r \alpha_l}, \quad \alpha_k = \frac{c_k}{(I_k+\varepsilon)^m}, \tag{38}$$

amb I_k els corresponents indicadors de suavitat de Jiang-Shu:

$$I_k = \sum_{\ell=1}^{r-1} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} h^{2\ell-1} p^{(\ell)}_k(x)^2 dx, \quad 0 \leq k \leq r-1 \tag{39}$$

on p_k és el polinomi de grau $r-1$ que satisfà

$$\frac{1}{h}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} p_k(x)dx = f_j, \quad i-r+1+k \leq j \leq i+k, \quad 0 \leq k \leq r-1.$$

Abstract

High-Resolution Shock-Capturing (HRSC) schemes constitute the state of the art for computing accurate numerical approximations to the solution of many hyperbolic systems of conservation laws, especially in computational fluid dynamics.

In this context, the application of suitable numerical boundary conditions on domains with complex geometry has become a problem with certain difficulty that has been tackled in different ways according to the nature of the numerical methods and mesh type. In this work we present a new technique for the extrapolation of information from the interior of the computational domain to ghost cells designed for structured Cartesian meshes (which, as opposed to non-structured meshes, cannot be adapted to the morphology of the domain boundary).

The aformentioned technique is based on the application of Lagrange interpolation equipped with detection of discontinuities that permits a data dependent extrapolation, with higher order at smooth regions and essentially non oscillatory properties near discontinuities.

We also propose an alternative approach to develop a high order accurate scheme both in space and time, with the one that was proposed by Qiu and Shu for numerically solving hyperbolic conservation laws as starting point. Both methods are based on the conversion of time derivatives to spatial derivatives through the Cauchy-Kowalewski technique, following the Lax-Wendroff procedure. Such spatial derivatives are then discretized through the Shu-Osher finite difference procedure with an adequate upwind scheme. The alternative approach replaces the exact derivatives of the flux by approximations of the suitable order in order to reduce both the implementation and the computational cost, as well as a fluctuation control which avoids the expansion of large terms at the discretization of the high order derivatives.

1

Introduction

1.1 Motivation

Systems of hyperbolic conservation laws and related equations have been the focus of many research lines in the past four decades, for example in modeling the flow of air around a vehicle, meteorology and weather prediction, or modeling the flow of the water over a channel or the sedimentation of small solid particles dispersed in a viscous fluid.

Since no analytic solution is known for many of these equations, different techniques have been developed in order to tackle these problems from a numerical point of view, with methodologies that have evolved and improved along these years. Our interest concerns obtaining the results from the simulations as fast as possible and with the highest possible accuracy, but the numerical simulation of physical problems modeled by systems of conservation laws is a delicate issue, due to the presence of discontinuities in the solution. These discontinuities are developed even when the initial flow is smooth. If we compute discontinuous solutions to conservation laws using standard methods developed under the assumption of smooth solutions, we typically obtain numerical results that are not accurate enough or even a failure in the simulation.

So, we require the use of shock-capturing schemes, developed to produce sharp approximations to discontinuous solutions automatically, without explicit tracking or using jump conditions, in order to ensure a proper handling of discontinuities in numerical simulations.

Low-order methods are faster and easier to implement, but provide less accurate results than high-resolution methods. High-Resolution Shock-Capturing (HRSC) schemes are the state of the art for numerical simulations of physical problems. The aim of those methods is to obtain high-order resolution wherever the solution is smooth, while maintaining sharp profiles of the discontinuities and avoiding the formation of spurious oscillations near them.

Since the drawback of a high-order reconstruction is the oscillations it might create, several methods were suggested to combine the upwinding framework, in which the discretization of the equations on a mesh is performed according to the direction of propagation of information on that mesh, with a mechanism to prevent the creation and evolution of such spurious numerical oscillations. Therefore, most of these schemes emerge from a combination of upwinding and high-order interpolation.

Robust and accurate HRSC schemes often have a high computational cost, which is related to their incorporating upwinding through characteristic information required at each cell boundary in the computational domain and high-order reconstruction procedures.

To solve partial differential equations (PDEs) we replace the continuous problem represented by the PDEs by a finite set of discrete values. These are obtained by first discretizing the domain of the PDEs into a finite set of points or volumes via a mesh or grid. Typically the computational domain is divided into cells, and the continuous equations are replaced by a discrete approximation at each cell. Boundary conditions are also discretized and the above concerns about the scheme used in the interior (accuracy, presence of discontinuities, etc.) also apply to them.

1.2 Previous work

Weighted essentially non-oscillatory (WENO) finite-difference spatial discretization schemes have become one of the most popular methods to approximate the solutions of hyperbolic equations, so, a lot of development has been done on them. These schemes have as a basic ingredient: the WENO reconstructions, i.e, “cell-average interpolators”, with a high order of accuracy and a control of the oscillations.

These schemes were developed by Liu, Osher and Chan in [35] as an improvement of ENO (essentially non-oscillatory) schemes, originally introduced and developed in [16, 18]. The only difference between these schemes and the standard cell-average version of ENO is the definition of the reconstruction procedure which produces a high-order accurate global approximation to the solution from its given cell-averages.

In [25], Jiang and Shu improved the high-order WENO finite-difference schemes by defining a new way of measuring the smoothness of the numerical solution, which results in a fifth-order WENO scheme for five-points stencils, instead of the fourth-order scheme obtained with the original smoothness measurement by Liu et al. [35].

Regarding high order boundary conditions, some authors have approached this problem from different perspectives. In [43] the authors develop a technique based on Lagrange interpolation with a limiter which is restricted to second order methods and a single ghost cell. Also related to our approach are the works of Shu and collaborators [45, 46] where the equation to be solved is used to extrapolate derivative values of the numerical solution to the boundary points where inflow conditions are prescribed and then approximate ghost values by a Taylor expansion; such technique is known as the inverse Lax-Wendroff procedure. For outflow boundaries an extrapolation technique based on the WENO method is used, achieving high order when the data is smooth in both cases. The drawbacks of this approach are that it is problem-dependent (see [23, 52] for a similar methodology applied to other equations), that it requires a different treatment for different types of boundary and its relatively high computational cost.

As for the time discretization, the most tipically used time discretization scheme with excellent stability, efficiency and low storage properties, which has been vastly used in the literature, is the third order Runge-Kutta 3 TVD (total variation diminishing) scheme [15]. Since stability issues arise for fourth and higher order Runge-Kutta methods, in an attemp to develop a family of schemes with arbitrarily high order in time, Qiu and Shu [39] developed in 2003 an scheme based on the Lax-Wendroff procedure, also known as Cauchy-Kowalewski. The drawback in this case is that again the implementation relies strongly on the equation and the corresponding derivatives of the flux, as well as a high implementation and computational cost.

1.3 Scope of the work

In this work we develop some techniques addressed to obtain a fully high order accurate scheme, mainly pursuing two main goals:

- To develop a high order accurate method to perform numerical boundary conditions and store the information at the ghost cells at each time step. This procedure must take into account the possible complex geometry of the boundary so that the process is properly accurate and the eventual presence of discontinuities near the boundary, with the design of weights containing the information related with this issue.

- To design a high order accurate time scheme, competitive with Runge-Kutta TVD schemes, overcoming some implementation and computational time issues inherent to the scheme originally proposed by Qiu and Shu. The implementation of these schemes should not be much harder than the implementation of Runge-Kutta schemes and the computational cost involving the high order derivatives should not be too high. This, together with the fact that only one spectral decomposition per time step is needed, should yield a more efficient scheme than the family of Runge-Kutta methods. In this sense, we also develop a mechanism to avoid the propagation of large terms at the approximation of the high order derivatives near discontinuities.

1.4 Organization of the text

The text is organized as follows:

In Chapter 2 we recall the basic concepts and ideas concerning hyperbolic conservation laws and numerical methods for their solution, focusing on the description of Shu-Osher's finite-difference approach and the weighted essentially non-oscillatory (WENO) reconstruction procedure.

In Chapter 3 we describe a procedure to automatically mesh through a Cartesian grid the boundary of a two-dimensional set described by a closed curve. The procedure ensures the computation of all the intersections of the mesh lines with the boundary, as well as the computation of all the normal lines to the boundary passing through each ghost cell, both of them computed at the desired precision.

In Chapter 4 we introduce some techniques to perform the extrapolations associated to the numerical boundary conditions with arbitrarily high order accuracy, with a procedure that takes into account the possible formation or approaching of discontinuities to the boundary. To do so, we develop two kind of methods, respectively based on thresholds and weights, in both cases scale and dimension independent, which allow to perform the extrapolations under the aforementioned considerations. This chapter is based on "A. Baeza, P. Mulet, D. Zorío", *High order boundary extrapolation technique for finite difference methods on complex domains with Cartesian meshes*, *Journal of Scientific Computing*, 66: 761-791, 2016 and "A. Baeza, P. Mulet, D. Zorío", *High order weighted extrapolation for boundary conditions for finite difference methods on complex domains with Cartesian meshes*, to appear in *Journal of Scientific Computing.*

Chapter 5 stands for the development of a high order time scheme, based on the Lax-Wendroff procedure proposed by Qiu and Shu, with some implementation, performance and resolution improvements. More precisely, we develop an scheme where no flux derivative is required to be computed, faster than the original scheme under common circumstances due to a considerable simplification of the computation of high order terms and capable of capturing better the discontinuities. This chapter is partially based on "D. Zorío, A. Baeza, P. Mulet", *An approximate Lax-Wendroff procedure for high order accurate in space and time scheme for hyperbolic conservation laws*, submitted to *Journal of Scientific Computing.*

Finally, some conclusions and future research lines to be followed from this work are pointed out in Chapter 6.

2

Preliminaries

In this chapter we collect some basic facts about hyperbolic conservation laws and numerical methods for them, focusing on finite difference Weighted Essentially Non-Oscillatory (WENO) methods applied to fluid dynamics equations, mainly to the Euler equations of gas dynamics. More information can be obtained from classic books such as Landau and Lifshitz [28], Chorin and Marsden [8], Whitham [50], Dafermos [10] and Lax [30]. Other interesting references are the books by LeVeque [32, 33], Evans [12] and Toro [47].

We also include the statement and proof of a generalization of the chain rule for higher order derivatives of composition of functions, known as Faà di Bruno's formula.

2.1 Hyperbolic conservation laws

Conservation laws are systems of first order partial differential equations that can be written as:

$$\frac{\partial u}{\partial t} + \sum_{i=1}^{d} \frac{\partial f^i(u)}{\partial x_i} = 0, \quad x \in \mathbb{R}^d, \quad t \in \mathbb{R}^+, \tag{2.1}$$

where $u = (u_1, \ldots, u_m)^T : \mathbb{R}^d \times \mathbb{R}^+ \longrightarrow \mathbb{R}^m$ is the vector of conserved variables and $f^i : \mathbb{R}^m \longrightarrow \mathbb{R}^m$ are the flux functions, $i = 1, \ldots, d$.

Equation (2.1) is provided with initial conditions

$$u(x, 0) = u_0(x), \quad x \in \mathbb{R}^d,$$

in order to solve a Cauchy problem, i.e., to find the state of the system after a certain time $t = T$, given the state at time $t = 0$. System (2.1) is hyperbolic if any linear combination of the Jacobian matrices of f^i, $\sum_{i=1}^{d} \alpha_i (f^i)'(v)$, is diagonalizable with real eigenvalues $\forall v \in \mathbb{R}^m$. This conditions ensures the stability of Cauchy problems for the linearized systems about constant states.

Boundary conditions have to be also specified when considering a bounded domain $\Omega \subseteq \mathbb{R}^d$. Part of this thesis will be focused on handling numerical boundary conditions on complex domains in multiple dimensions.

System (2.1) can be written in quasi-linear form as:

$$\frac{\partial u}{\partial t} + \sum_{i=1}^{d} (f^i)'(u) \frac{\partial u}{\partial x_i} = \frac{\partial u}{\partial t} + \sum_{i=1}^{d} \sum_{j=1}^{m} \frac{\partial f^i(u)}{\partial u_j} \frac{\partial u_j}{\partial x_i} = 0.$$

The particular case $m = 1$, often referred as scalar conservation law, is one of the systems most used in this work due to their simplicity. In 1D, when $d = 1$, this conservation law can be written as

$$u_t + f(u)_x = 0, \quad x \in \mathbb{R}, \quad t \in \mathbb{R}^+,$$

with the conserved variable $u : \mathbb{R} \times \mathbb{R}^+ \longrightarrow \mathbb{R}$ and flux function $f : \mathbb{R} \longrightarrow \mathbb{R}$. Many examples will use scalar conservations laws, such as linear advection

$$u_t + a u_x = 0, \quad a \in \mathbb{R},$$

and Burger's equation

$$u_t + (\frac{u^2}{2})_x = 0.$$

Conservation laws regularly come from an integral relationship representing the conservation of a certain quantity u. Conservation means that the amount of quantity contained in a given volume can only change due to the flux of this quantity crossing the interfaces of the given volume. In one space dimension it is written as:

$$\int_{x_1}^{x_2} (u(x,t_2) - u(x,t_1))dx = \int_{t_1}^{t_2} f(u(x_1,t))dt - \int_{t_1}^{t_2} f(u(x_2,t))dt, \qquad (2.2)$$

where the control volume in the $x-t$ plane is $V = [x_1,x_2] \times [t_1,t_2] \subseteq \mathbb{R} \times \mathbb{R}$.

The characteristic structure of the hyperbolic conservation laws refers to the eigenstructure of the Jacobian matrix of the fluxes. The characteristic structure is important both for exact and approximate solutions of the equations. The characteristic speeds are the eigenvalues of the Jacobian matrices. For one-dimensional systems of conservation laws, we will assume that there are smooth functions $\lambda_k \colon \mathbb{R}^m \to \mathbb{R}$, $k = 1, \ldots, m$, such that $\lambda_k(u)$ is the k-th eigenvalue of $f'(u)$. For scalar conservation laws, these characteristic speeds are just the flux derivatives $f'(u)$. For one-dimensional systems of conservation laws, characteristics for a solution u are curves $(t, x(t))$ satisfying $x'(t) = \lambda_k(u(x(t),t))$. For scalar equations, this reduces to $x'(t) = f'(u(x(t),t) = f'(u(x(0),0))$, so in this case characteristics are straight lines of slope $f'(u_0(x(0)))$.

2.1.1 Weak solutions and Rankine-Hugoniot conditions

A classical solution of (2.1) is a smooth function $u : \mathbb{R}^d \times \mathbb{R}^+ \longrightarrow \mathbb{R}^m$ that satisfies the equations point-wise. A key feature of nonlinear conservation laws is the general lack of classical solutions of (2.1) beyond some finite time interval, even when the initial condition u_0 is a smooth function. This is due to the fact that the classical method of characteristics for obtaining solutions of first order PDE may fail to give global existence of classical solutions at all time when characteristics cross, which may happen since nonlinearity of fluxes is equivalent to characteristic speeds, i.e., the eigenvalues of the Jacobian matrices, being non constant.

In order to be able to consider non-smooth solutions, the classical concept of solution is relaxed by using a weak, distributional formulation that involves no derivatives of u.

Definition 1. *A function $u(x,t)$ is a weak solution of (2.1) with given initial data $u_0(x)$ if*

$$\int_{\mathbb{R}^+}\int_{\mathbb{R}^d}\left[u(x,t)\frac{\partial\phi}{\partial t}(x,t)+\sum_{j=1}^{d}f^j(u)\frac{\partial\phi}{\partial x_j}\right]dxdt=-\int_{\mathbb{R}^d}\phi(x,0)u_0(x)dx \qquad (2.3)$$

is satisfied for all $\phi\in C_0^1(\mathbb{R}^d\times\mathbb{R}^+)$, where $C_0^1(\mathbb{R}^d\times\mathbb{R}^+)$ is the space of continuously differentiable functions with compact support in $\mathbb{R}^d\times\mathbb{R}^+$.

Weak solutions provide an adequate generalization of the concept of classical solution for hyperbolic conservation laws. It is easy to see that strong solutions are also weak solutions, and continuously differentiable weak solutions are strong solutions. Furthermore, the weak formulation (2.3) is equivalent to the integral formulation (2.2) and the satisfaction of the initial conditions in $\mathcal{L}^1_{\text{loc}}$.

The Rankine-Hugoniot condition [24, 40], whose derivation can be found for example in [8, 19, 20], follows from the definition of weak solution. This condition characterizes weak solutions in terms of the discontinuity movement, and gives information about the behavior of the conserved variables across discontinuities.

For a general conservation law the Rankine-Hugoniot condition reads:

$$[f]\cdot n=[u](n\cdot s), \qquad (2.4)$$

where $f=(f^1,\dots f^d)$ is a matrix containing the fluxes, u is the solution, s is the speed of propagation of the discontinuity and n is the vector normal to the discontinuity. The notation $[\cdot]$ indicates the jump on a variable across the discontinuity. For scalar problems this simply gives:

$$f(u_L)-f(u_R)=s(u_L-u_R),$$

where u_L and u_R are the states at the left and the right side of the discontinuity, respectively.

At discontinuities, weak solutions have to satisfy the Rankine-Hugoniot condition. It can be shown that a function $u(x,t)$ is a weak solution of (2.1) if and only if equation (2.1) holds wherever u is smooth at (x,t) and the Rankine-Hugoniot condition is satisfied if u is not smooth in (x,t), see e.g. [8].

However, weak solutions are often not unique (see e.g. [32]), and there are *entropy conditions* proposed to single out a unique weak solution, known as entropy solution. The most well-known entropy condition characterizes a p-shock, $1\leq p\leq m$, as a discontinuity of u, defined by

$x = s(t)$, separating two states $u_L(t)$ and $u_R(t)$, such that the Rankine-Hugoniot conditions holds and

$$\lambda_p(u_L(t)) \geq s'(t) \geq \lambda_p(u_R(t)), \tag{2.5}$$

where λ_p is the p-th eigenvalue of the flux Jacobian.

Condition (2.5) is called Lax's E-condition [29]. There is also an entropy inequality for entropy-entropy flux pairs, due also to Lax [29], which is closely linked to vanishing viscosity solutions. There are other entropy conditions such as Oleinik's condition [37], Kružkov's condition [27], Wendroff's condition [49] or Liu's condition [34].

Establishing the existence of weak solutions satisfying entropy conditions is a great challenge. Positive answers to this existence question can be found for wide classes of multi-dimensional scalar conservation laws or one-dimensional systems. Knowledge of the characteristic structure, Riemann invariants and solution of Rankine-Hugoniot conditions yields answers to this question for Riemann problems for some hyperbolic systems of conservation laws, i.e., problems for which the initial value data is piecewise constant with only one discontinuity.

The solution to Riemann problems may be used as a building block for theoretical or practical purposes when the initial data belongs to a more general class of functions. For scalar conservation laws and some hyperbolic systems of conservations laws, existence can be established by the front tracking method [9, 21, 22]. Other simpler means of establishing existence may apply in some cases, such as Lax-Oleinink's formula [12] for scalar conservation laws with convex flux.

2.1.2 Euler equations

One of the most well-studied hyperbolic systems of conservation laws is that formed by the Euler equations, which model the dynamics of a Newtonian, ideal, inviscid fluid. They are derived from the conservation of mass, linear momentum and energy of the fluid in motion, and represent a simplified model for the Navier-Stokes equations, which are the most complete model used up to now for the simulation of Newtonian fluid dynamics.

The two-dimensional Euler equations can be written as:

$$u_t + f(u)_x + g(u)_y = 0 \tag{2.6}$$

with

$$u = \begin{bmatrix} \rho \\ \rho v^x \\ \rho v^y \\ E \end{bmatrix}, \quad f(u) = \begin{bmatrix} \rho v^x \\ \rho (v^x)^2 + p \\ \rho v^x v^y \\ v^x (E + p) \end{bmatrix}, \quad g(u) = \begin{bmatrix} \rho v^y \\ \rho v^y v^x \\ \rho (v^y)^2 + p \\ v^y (E + p) \end{bmatrix}, \tag{2.7}$$

where ρ denotes density, v^x and v^y are the Cartesian components of the velocity vector v, E is energy and p is pressure, where the energy (density) E is defined as the sum of the kinetic energy and the internal energy ρe

$$E = \frac{1}{2}\rho((v^x)^2 + (v^y)^2) + \rho e, \tag{2.8}$$

with e denoting the specific internal energy, linked with pressure and density through a thermodynamical equation of state $e = e(p, \rho)$. We will use a perfect gas equation of state

$$e = \frac{p}{(\gamma - 1)\rho},$$

where

$$\gamma = \frac{c_p}{c_v} \tag{2.9}$$

is called the specific heat ratio, and depends on the gas. For air it takes the value $\gamma \approx 1.4$.

The one dimensional version of the equations are obtained by postulating that all quantities depend only on x and v^y is constant, so that we get

$$\begin{bmatrix} \rho \\ \rho v^x \\ E \end{bmatrix}_t + \begin{bmatrix} \rho v^x \\ \rho (v^x)^2 + p \\ v^x (E + p) \end{bmatrix}_x = 0. \tag{2.10}$$

The hyperbolicity of system (2.7) is better established by formally rewriting it in terms of the primitive variables

$$w = \begin{bmatrix} \rho \\ v^x \\ v^y \\ p \end{bmatrix},$$

as follows

$$w_t + A^x(w)w_x + A^y(w)w_y = 0$$

$$A^x(w) = W'(U(w))f'(U(w))(W'(U(w)))^{-1} = \begin{bmatrix} v^x & \rho & 0 & 0 \\ 0 & v^x & 0 & 1/\rho \\ 0 & 0 & v^x & 0 \\ 0 & \gamma p & 0 & v^x \end{bmatrix}$$

$$A^y(w) = W'(U(w))g'(U(w))(W'(U(w)))^{-1} = \begin{bmatrix} v^y & 0 & \rho & 0 \\ 0 & v^y & 0 & 0 \\ 0 & 0 & v^y & 1/\rho \\ 0 & 0 & \gamma p & v^y \end{bmatrix}$$

where W is the function that transforms conserved variables into primitive variables

$$W(\rho, \rho v^x, \rho v^y, E) = (\rho, (\rho v^x)/\rho, (\rho v^y)/\rho, (\gamma - 1)\Big(E - \frac{1}{2}\rho^{-1}\big((\rho v^x)^2 + (\rho v^y)^2\big)\Big).$$

Assume, without loss of generality that the coefficients in the linear combination of Jacobian matrices satisfy $\alpha^2 + \beta^2 = 1$. Since

$$\alpha f'(u) + \beta g'(u) = W'(u)^{-1}\big(\alpha A^x(W(u)) + \beta A^y(W(u))\big)W'(u),$$

we deduce that the eigenvalues of $\alpha f'(u) + \beta g'(u)$ coincide with those of $\alpha A^x(W(u)) + \beta A^y(W(u))$ and the eigenvectors of both matrices can be related by product by the matrix $W'(u)$ or $W'(u)^{-1}$. The advantage of this algebraic manipulation is that $A = \alpha A^x(w) + \beta A^y(w)$ is much simpler than the other matrix and its eigenstructure can be readily obtained. Simple calculations yield that A has eigenvalues $z, z, z - c, z + c$, where $z = \alpha v^x + \beta v^y$ and $c = \sqrt{\frac{\gamma p}{\rho}}$ is the sound velocity. The corresponding right eigenvectors form the matrix $\overline{R}$ columnwise and the (normalized) left eigenvectors form the matrix $\overline{L} = (\overline{R})^{-1}$ rowwise, in such a way that $\overline{L}A\overline{R} = \Lambda$, for the following matrices:

$$\overline{R} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & -\beta & -\frac{\alpha c}{\rho} & \frac{\alpha c}{\rho} \\ 0 & \alpha & -\frac{\beta c}{\rho} & \frac{\beta c}{\rho} \\ 0 & 0 & c^2 & c^2 \end{bmatrix}, \overline{L} = \begin{bmatrix} 1 & 0 & 0 & -\frac{1}{c^2} \\ 0 & -\beta & \alpha & 0 \\ 0 & -\frac{\alpha\rho}{2c} & -\frac{\beta\rho}{2c} & \frac{1}{2c^2} \\ 0 & \frac{\alpha\rho}{2c} & \frac{\beta\rho}{2c} & \frac{1}{2c^2} \end{bmatrix}, \Lambda = \begin{bmatrix} z & 0 & 0 & 0 \\ 0 & z & 0 & 0 \\ 0 & 0 & z - c & 0 \\ 0 & 0 & 0 & z + c \end{bmatrix}.$$

If we denote $\|v\|^2 = (v^x)^2 + (v^y)^2$, the total enthalpy by $H = (E + p)/\rho = \|v\|^2 + c^2/(\gamma - 1)$, $z_\perp = -\beta v^x + \alpha v^y$, then the right and left eigenvectors form the following matrices

$$R = \big(W'(u)\big)^{-1}\,\overline{R} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ v^x & -\rho\,\beta & v^x - \alpha\,c & v^x + \alpha\,c \\ v^y & \alpha\,\rho & v^y - \beta\,c & v^y + \beta\,c \\ \frac{\|v\|^2}{2} & \rho\,z_\perp & H - z\,c & H + z\,c \end{bmatrix}$$

$$L = \overline{L}\,W'(u) = \begin{bmatrix} 1 - \frac{(\gamma-1)\,\|v\|^2}{2\,c^2} & \frac{(\gamma-1)\,v^x}{c^2} & \frac{(\gamma-1)\,v^y}{c^2} & -\frac{\gamma-1}{c^2} \\ -\frac{z_\perp}{\rho} & -\frac{\beta}{\rho} & \frac{\alpha}{\rho} & 0 \\ \frac{z}{2\,c} + \frac{(\gamma-1)\,\|v\|^2}{4\,c^2} & -\frac{\alpha\,c+(\gamma-1)\,v^x}{2\,c^2} & -\frac{(\gamma-1)\,v^y+c\,\beta}{2\,c^2} & \frac{\gamma-1}{2\,c^2} \\ -\frac{z}{2\,c} + \frac{(\gamma-1)\,\|v\|^2}{4\,c^2} & \frac{\alpha\,c-(\gamma-1)\,v^x}{2\,c^2} & -\frac{(\gamma-1)\,v^y-c\,\beta}{2\,c^2} & \frac{\gamma-1}{2\,c^2} \end{bmatrix},$$

that satisfy $L(\alpha f'(u) + \beta g'(u))R = \Lambda$, hence the Euler equations are hyperbolic. We can further obtain the eigenvectors of $f'(u)$ and $g'(u)$, by setting $\alpha = 1, \beta = 0$ and $\alpha = 0, \beta = 1$, respectively. For the first case, $z = v^x, z_\perp = v^y$ and the eigenvector matrices are:

$$R_x = \begin{bmatrix} 1 & 0 & 1 & 1 \\ v^x & 0 & v^x - c & v^x +, c \\ v^y & \rho & v^y & v^y \\ \frac{\|v\|^2}{2} & \rho\,v^y & H - v^x\,c & H + v^x\,c \end{bmatrix}$$

$$L_x = \begin{bmatrix} 1 - \frac{(\gamma-1)\,\|v\|^2}{2\,c^2} & \frac{(\gamma-1)\,v^x}{c^2} & \frac{(\gamma-1)\,v^y}{c^2} & -\frac{\gamma-1}{c^2} \\ -\frac{v^y}{\rho} & 0 & \frac{1}{\rho} & 0 \\ \frac{v^x}{2\,c} + \frac{(\gamma-1)\,\|v\|^2}{4\,c^2} & -\frac{c+(\gamma-1)\,v^x}{2\,c^2} & -\frac{(\gamma-1)\,v^y}{2\,c^2} & \frac{\gamma-1}{2\,c^2} \\ -\frac{v^x}{2\,c} + \frac{(\gamma-1)\,\|v\|^2}{4\,c^2} & \frac{c-(\gamma-1)\,v^x}{2\,c^2} & -\frac{(\gamma-1)\,v^y}{2\,c^2} & \frac{\gamma-1}{2\,c^2} \end{bmatrix},$$

For the second case, $z = v^y, z_\perp = -v^x$ and the eigenvector matrices are:

$$R_y = \begin{bmatrix} 1 & 0 & 1 & 1 \\ v^x & \rho & v^x & v^x \\ v^y & 0 & v^y - c & v^y + c \\ \frac{\|v\|^2}{2} & \rho\,v^x & H - v^y\,c & H + v^y\,c \end{bmatrix}$$

$$L_y = \begin{bmatrix} 1 - \frac{(\gamma-1)\,\|v\|^2}{2\,c^2} & \frac{(\gamma-1)\,v^x}{c^2} & \frac{(\gamma-1)\,v^y}{c^2} & -\frac{\gamma-1}{c^2} \\ -\frac{v^x}{\rho} & \frac{1}{\rho} & 0 & 0 \\ \frac{v^y}{2\,c} + \frac{(\gamma-1)\,\|v\|^2}{4\,c^2} & -\frac{(\gamma-1)\,v^x}{2\,c^2} & -\frac{(\gamma-1)\,v^y+c}{2\,c^2} & \frac{\gamma-1}{2\,c^2} \\ -\frac{v^y}{2\,c} + \frac{(\gamma-1)\,\|v\|^2}{4\,c^2} & -\frac{(\gamma-1)\,v^x}{2\,c^2} & -\frac{(\gamma-1)\,v^y-c}{2\,c^2} & \frac{\gamma-1}{2\,c^2} \end{bmatrix},$$

where we have further changed the sign to the second left and right eigenvectors for symmetry.

2.2 Numerical methods

Although existence of entropy weak solutions to hyperbolic systems of conservation laws can be established in some cases, practical closed formulas only exist in very limited cases, such as linear equations or some Riemann problems. Therefore, numerical methods should be used to obtain approximations to the solutions. In this section we review some basic concepts and results related to numerical methods for hyperbolic systems of conservation laws, paying special attention to finite difference conservative methods [42] and Weighted Essentially Non-Oscillatory (WENO) reconstructions [35, 26].

2.2.1 Computational grids

The first step to numerically solve partial differential equations is to replace the continuous problem, represented by the PDE's, by a discrete representation of it. First of all we discretize the $x-t$ plane by choosing a mesh (or grid) composed by a finite set of points or volumes defined below. Then the PDE is discretized on this grid, and the resulting discrete, finite-dimensional problem, is solved. We use a point-value discretization if we regard these discrete values as point values defined at grid points. On the other hand, we use a cell-average discretization if those discrete values represent the average value over cells.

Consider a scalar Cauchy problem in one space dimension,

$$\begin{cases} u_t + f(u)_x = 0, \quad x \in \mathbb{R}, \quad , t \in \mathbb{R}^+, \\ u(x,0) = u_0(x), \end{cases} \tag{2.11}$$

where $u, f : \mathbb{R} \longrightarrow \mathbb{R}$.

To define a mesh, we consider a discrete subset of points (nodes) $\{x_j\}_{j\in\mathbb{Z}}$, $x_j \in \mathbb{R} \quad \forall j$ and assume that the grid is uniform, i.e., $x_j - x_{j-1} = \Delta x > 0$, $\forall j \in \mathbb{Z}$. This constant is called mesh size and we abbreviate it as $h = \Delta x$. From the points $\{x_j\}$ we define the cells c_j as the subintervals whose respective centers are x_j:

$$c_j = \left[\frac{x_{j-1}+x_j}{2}, \frac{x_j+x_{j+1}}{2}\right] = \left[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}\right].$$

A grid is defined, depending on the context, to be either the set of cells $\{c_j\}_{j\in\mathbb{Z}}$ or the set of nodes $\{x_j\}_{j\in\mathbb{Z}}$.

We discretize the time variable by defining points in time $\{t^n\}_{n\in\mathbb{N}}$, with $t^n < t^{n+1}, \quad \forall n \in \mathbb{N}$. If $t^{n+1} - t^n$ is constant with respect to n, we denote it by Δt and call it the time increment. We will denote by $u^n = \{u_j^n\}_{j\in\mathbb{Z}}$ the computed approximation to the exact solution $u(x_j, t^n)$ of (2.11).

In real problems, the domain of definition of the equations is restricted to a bounded subset of $\mathbb{R}$ and a finite time interval, so the grid has to be restricted to a finite number of nodes or cells. If we consider the interval $I = [0, 1]$ and a fixed time $T > 0$, then we can take positive numbers M and N and define a set of nodes $\{x_j\}_{0\le j<M}$ given by $x_j = (j+\frac{1}{2})\Delta x$, with $\Delta x = \frac{1}{M}$. The points in time $\{t^n\}_{0\le n<N}$ can be defined by $t^n = n\Delta t$, with $\Delta t = \frac{1}{N}$.

We can extend all this explanation to the two-dimensional case. Let us consider a scalar conservation law in 2D with the form:

$$\begin{cases} u_t(x,y,t) + f(u(x,y,t))_x + g(u(x,y,t))_y = 0, & (x,y) \in \mathbb{R}\times\mathbb{R}, \quad t \times \mathbb{R}^+, \\ u(x,y,0) = u_0(x), \end{cases}$$

and two sets of ordered points, $\{x_i\}_{i\in\mathbb{Z}}$ and $\{y_j\}_{j\in\mathbb{Z}}$, satisfying $x_i < x_{i+1}$ for all $i \in \mathbb{Z}$ and $y_j < y_{j+1}$ for all $j \in \mathbb{Z}$. Moreover, we assume as before that $\Delta x = x_{i+1} - x_i$ and $\Delta y = y_{j+1} - y_j$ are constant with respect to i and j respectively. We can define cells $c_{i,j}$ by

$$c_{i,j} = \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right] \times \left[y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}\right],$$

so that each node (x_i, y_j) is the center of the cell $c_{i,j}$.

2.2.2 Conservative methods

The simplest way to approximate derivatives is by means of linear finite differences. If a solution presents some singularity then, in principle, finite-differences do not yield a satisfactory approximation of the partial derivatives appearing in the equations. Finite volume methods (Godunov's method, Roe's method, etc.), based on approximating (2.2), and Discontinuous Galerkin methods, based on (2.3), overcome this difficulty by resorting to weak formulations that do not require derivatives of the unknowns.

When we deal with discontinuous solutions, as mentioned in section 2.1.1, there may be more than one weak solution and the method may not converge to the right one or it may converge to a function that is not a weak solution of the PDE. Some examples of these facts can be found e.g. in [32]. There exists a simple requirement that we can impose on the numerical methods to guarantee that they do not converge to non-solutions. Conservative methods ensure that convergence can only be achieved to weak solutions (Lax-Wendroff's theorem).

Definition 2. *A numerical method is said to be conservative if it can be written in the form*

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}\left(\hat{f}(u_{j-p+1}^n, \dots, u_{j+q}^n) - \hat{f}(u_{j-p}^n, \dots, u_{j+q-1}^n)\right), \tag{2.12}$$

where the function $\hat{f} : \mathbb{R}^{p+q} \to \mathbb{R}$ is called the numerical flux function and $p, q \in \mathbb{N},\ p, q \geq 0$.

The purpose of conservative methods is to reproduce at a discrete level the conservation of the physical variables in the continuous equations. In fact (2.12) can be seen as a discrete version of the integral form (2.2) of the PDE.

An essential requirement on the numerical flux is the consistency condition:

Definition 3. *We say that the numerical flux function of a conservative numerical method is consistent with the conservation law if the numerical flux function $\hat{f}$ reduces to the exact flux f for the case of constant flow, i.e,*

$$\hat{f}(u, \dots, u) = f(u).$$

The consistency condition is necessary to ensure that a discrete form of conservation, analogous to the conservation law, is provided by conservative methods.

In general, some smoothness is required in the way in which $\hat{f}$ approaches a certain value $f(u)$, then we suppose that the flux function is locally Lipschitz continuous in each variable, i.e., if x is a point in a normed space M then there exists a constant K and a neighborhood $N(x)$ of x such that $||f(y) - f(x)|| \leq K||y - x||, \quad \forall y \in N(x)$.

The main result about conservative methods is the Lax-Wendroff theorem, that proves that if they produce a sequence of approximations that converges to some function $u(x, t)$ as the grid is refined, then this function will be a weak solution of the conservation law:

Theorem 1. *(Lax-Wendroff, [31, 22]) Consider a sequence of grids indexed by $k = 1, 2, \dots$ with grid sizes $(\Delta x_k, \Delta t_k)$, satisfying*

$$\lim_{k\to+\infty} \Delta x_k = 0,$$
$$\lim_{k\to+\infty} \Delta t_k = 0.$$

Let $\{u_k(x,t)\}$ denote the piecewise constant function defined from the numerical solution obtained by a conservative numerical method, consistent with (2.1), on the k-th grid. If the total variation of the function $u_k(\cdot,t)$ is uniformly bounded in k, t, i.e., $\sup_{k,t\in[0,T]} TV(u_k(\cdot,t)) < \infty$ and $u_k(x,t)$ converges in $\mathcal{L}^1_{\text{loc}}$ to a function $u(x,t)$ as $k \to \infty$, then u is a weak solution of the conservation law.

Some extra conditions for convergence to entropy solutions need to be imposed [38, 44], otherwise entropy violating shocks, which are non-entropic weak solutions, may result from conservative methods.

2.2.3 High-resolution conservative methods

The term “high-resolution” is applied to methods whose local truncation error has order higher than two, thus giving second or even higher order global errors in smooth parts of the solution, while giving well-resolved non-oscillatory approximations near discontinuities.

Godunov’s method [14] is a first order accurate method based on the computation of local Riemann problems located at each cell interface and subsequent computation of cell-averages of the numerical solution after a time step short enough so that the waves emanating from the cell-interfaces do not interact.

The idea of solving Riemann problems forward in time is at the basis of many modern high-resolution shock-capturing finite volume methods. A common practice to design numerical methods with order of accuracy higher than one and suitable for non-linear systems is using piecewise constant initial data obtained by a high-order reconstruction at the cell interfaces (see [48]).

To achieve higher order some techniques have been developed as the essentially non-oscillatory (ENO) methods, introduced by Harten, Engquist, Osher and Chakravarthy in [16] and the weighted essentially non-oscillatory (WENO) methods [25, 35], explained in more detail in section 2.2.6.

2.2.4 Semi-discrete formulation

The previously mentioned schemes have all been fully discrete methods, discretized in both space and time. Let us now consider the discretization process in two stages, following the so called “method of lines”: we first discretize only in space, leaving the problem continuous in time. This leads to a system of ordinary differential equations in time, called “semi-discrete equations”, that can be written as

$$\frac{d\,u_j(t)}{dt} + \mathcal{D}(u(t))_j = 0, \quad \forall j, \tag{2.13}$$

where $\mathcal{D}(u(t))_j \approx f(u)_x(x_j, t)$. The discrepancy in this approximation is defined as the local truncation error of the semi-discrete scheme (2.13), whose spatial order is precisely the order of the local truncation error.

If we compute the spatial approximation using a conservative formulation, we can rewrite the ODE system (2.13) as:

$$\frac{du_j(t)}{dt} + \frac{\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}}{\Delta x} = 0, \quad \forall j, \tag{2.14}$$

where $\hat{f}_{j+\frac{1}{2}} = \hat{f}(u_{j-p+1}(t), \ldots, u_{j+q}(t))$, so that $\mathcal{D}(u(t))_j = \frac{\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}}{\Delta x}$.

After that, we solve the system of ordinary differential equations (2.14) using an ODE solver. Among the most widely used ODE solvers in this context are the TVD Runge-Kutta methods developed by Shu and Osher in [41]. The general formulation of these methods is as follows:

$$\begin{cases} u^{(0)} = u^n, \\ u^{(i)} = \displaystyle\sum_{k=0}^{i} \left(\alpha_{ik} u^{(k)} - \beta_{ik} \Delta t \mathcal{D}(u^{(k)}) \right), \quad 1 \le i \le \bar{r}, \\ u^{n+1} = u^{(\bar{r})}, \end{cases}$$

where $\bar{r}$ depends on the order of accuracy of the particular Runge-Kutta scheme and α_{ik}, β_{ik} are coefficients that also depend on the method (for more details see [41, 42]). Specifically, in this work we use the third-order version:

$$\begin{cases} u^{(1)} = u^n - \Delta t \mathcal{D}(u^n), \\ u^{(2)} = \frac{3}{4} u^n + \frac{1}{4} u^{(1)} - \frac{1}{4} \Delta t \mathcal{D}(u^{(1)}), \\ u^{n+1} = \frac{1}{3} u^n + \frac{2}{3} u^{(2)} - \frac{2}{3} \Delta t \mathcal{D}(u^{(2)}). \end{cases} \tag{2.15}$$

The overall local truncation error of the fully-discrete scheme is related to the local truncation errors of the spatial semi-discretization and of the ODE solver.

If we use this TVD Runge-Kutta method as a ODE solver together with spatial operators that lead to ODE's of the form (2.14), then we obtain conservative schemes that can be expressed in the conservative form (2.12). For example, if we expand (2.15) for each node x_j, supposing that $\mathcal{D}(u^n)_j = \frac{\hat{f}_{j+\frac{1}{2}}(u^n)-\hat{f}_{j-\frac{1}{2}}(u^n)}{\Delta x}$, then we can write

$$\begin{aligned} u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}\Bigg[&\left(\frac{1}{6}\hat{f}_{j+\frac{1}{2}}(u^n) + \frac{1}{6}\hat{f}_{j+\frac{1}{2}}(u^{(1)}) + \frac{2}{3}\hat{f}_{j+\frac{1}{2}}(u^{(2)})\right) \\ &- \left(\frac{1}{6}\hat{f}_{j-\frac{1}{2}}(u^n) + \frac{1}{6}\hat{f}_{j-\frac{1}{2}}(u^{(1)}) + \frac{2}{3}\hat{f}_{j-\frac{1}{2}}(u^{(2)})\right)\Bigg]. \end{aligned} \tag{2.16}$$

Since $u^{(1)}$ and $u^{(2)}$ are obtained from u^n we can write (2.16) in terms of a numerical flux function

$$\hat{f}^{RK3}(u^n) = \frac{1}{6}\hat{f}(u^n) + \frac{1}{6}\hat{f}(u^{(1)}) + \frac{2}{3}\hat{f}(u^{(2)}),$$

which is consistent, as

$$u_j^{n+1} = u_j^n - \Delta t\left(\hat{f}^{RK3}_{j+\frac{1}{2}}(u^n) - \hat{f}^{RK3}_{j-\frac{1}{2}}(u^n)\right).$$

2.2.5 Shu-Osher's finite-difference conservative schemes

In order to obtain high-order finite-difference conservative schemes to solve hyperbolic systems of conservation laws, we use Shu and Osher's technique [42]. Third or higher order multidimensional schemes obtained with this technique are more efficient, both in execution and implementation, than their finite volume counterparts. A restriction of high order finite-difference conservative schemes is that uniformly-spaced Cartesian grids are required, which is a serious drawback when dealing with domains with curved boundaries. As we shall see, an essential contribution of this work is to show that we can overcome this drawback by using suitable extrapolations techniques for ghost-cell data.

The basic idea that makes possible Shu-Osher's approach is stated in the following lemma:

Lemma 1. *If the functions g, φ satisfy*

$$g(x) = \frac{1}{\Delta x}\int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} \varphi(\xi)d\xi,$$

then

$$g'(x) = \frac{\varphi\left(x+\frac{\Delta x}{2}\right)-\varphi\left(x-\frac{\Delta x}{2}\right)}{\Delta x}.$$

Applying this result to $g(x) = f(u(x,t))$, for a fixed t, the conservative property of the spatial discretization is obtained by implicitly defining the function φ as:

$$f(u(x,t)) = \frac{1}{\Delta x}\int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} \varphi(\xi,t)d\xi,$$

so that the spatial derivative in

$$u_t + f(u)_x = 0$$

is exactly obtained by a conservative finite-difference formula at the cell boundaries,

$$f(u)_x = \frac{\varphi\left(x+\frac{\Delta x}{2},t\right)-\varphi\left(x-\frac{\Delta x}{2},t\right)}{\Delta x}.$$

Dropping the dependence on t for the presentation of the spatial semi-discretization, we notice that highly accurate approximations to $\varphi\left(x \pm \frac{\Delta x}{2}\right)$ are computed using known grid values of f (which are cell-averages of φ) and a reconstruction procedure $\mathcal{R}$. If $\widehat{\varphi}$ is an approximation to φ obtained from point values of f in an stencil around $x_{j+\frac{1}{2}}$ such that $\varphi(x_{j+\frac{1}{2}}) = \widehat{\varphi}(x_{j+\frac{1}{2}}) + d(x_{j+\frac{1}{2}})\Delta x^r + \mathcal{O}(\Delta x^{r+1})$, for a Lipschitz function d, then we can discretize

$$f(u)_x(x_j) = \frac{\widehat{\varphi}(x_{j+\frac{1}{2}})-\widehat{\varphi}(x_{j-\frac{1}{2}})}{\Delta x} + \mathcal{O}(\Delta x^r),$$

i.e., the local truncation error of the semi-discrete scheme is $\mathcal{O}(\Delta x^r)$.

We denote as $\mathcal{R}(\bar{f}_{j-s_1},\ldots,\bar{f}_{j+s_2},x)$ the generic local reconstruction procedure for $f(x)$ from its cell-averages $\{\bar{f}_{j-s_1},\ldots,\bar{f}_{j+s_2}\}$, where s_1 and s_2 are non-negative integers. The most important properties that has to satisfy this local reconstruction procedure are:

- Preservation of the cell-averages:

$$\frac{1}{\Delta x}\int_{x_{k-\frac{1}{2}}}^{x_{k+\frac{1}{2}}} \mathcal{R}(\bar{f}_{j-s_1},\ldots,\bar{f}_{j+s_2},x)dx = \bar{f}_k, \quad k = j-s_1,\ldots,j+s_2.$$

- Accuracy:

$$\mathcal{R}(\bar{f}_{j-s_1},\dots,\bar{f}_{j+s_2},x)=f(x)+\mathcal{O}(\Delta x^r),\quad x\in[x_{j-s_1-\frac{1}{2}},x_{j+s_2+\frac{1}{2}}].$$

  wherever f is smooth, for some $r>0$.

- The total variation of $\mathcal{R}(\bar{f}_{j-s_1},\dots,\bar{f}_{j+s_2},x)$ is essentially bounded by the total variation of $f(x)$, i.e., for some $p>0$:

$$TV(\mathcal{R}(\bar{f}_{j-s_1},\dots,\bar{f}_{j+s_2},x))\leq C\cdot TV(f(x))+\mathcal{O}(\Delta x^p).$$

When computing reconstructions, another essential point is the use of an upwinding framework, in which the discretization of the equations on a mesh is performed according to the direction of propagation of information on that mesh, i.e we have into account the side from which information (wind) flows, given by the signs of the eigenvalues of the Jacobian matrix. For instance, for scalar equations, the direction of propagation of the solution is locally given by the sign of $f'(u)$ and we use the value of $f'(u)$ to perform reconstructions biased towards the correct direction: if $f'(u)>0$, the upwind side is the left side whereas if $f'(u)<0$, the upwind side is the right side.

The approximations $\hat{f}^n_{j+\frac{1}{2}}$ are obtained by high-order upwind-biased reconstructions $\mathcal{R}^{\pm}(\bar{f}_{j-s_1},\dots,\bar{f}_{j+s_2},x)$, i.e., cell-average interpolators whose stencils have more points at the upwind side of the points where they are evaluated. In this work, we obtain $\widehat{f}$ by the WENO reconstruction method which will be explained in the next section.

Summarizing, the computation of the numerical fluxes with Shu-Osher's procedure is performed as follows:

Algorithm 1. *(Shu-Osher's algorithm for scalar equations)*

```
Define $\beta_{j+\frac{1}{2}} = \max_{u\in[u_j,u_{j+1}]} |f'(u)|$

if $f'(u) \neq 0 \quad \forall u \in [u_j, u_{j+1}]$
  if $f'(u) > 0$
    $\hat{f}_{j+\frac{1}{2}} = \mathcal{R}^+(f_{j-s_1}, \dots, f_{j+s_2}, x_{j+\frac{1}{2}})$
  else
    $\hat{f}_{j+\frac{1}{2}} = \mathcal{R}^-(f_{j-s_1+1}, \dots, f_{j+s_2+1}, x_{j+\frac{1}{2}})$
  end
else
  $\hat{f}^+_{j+\frac{1}{2}} = \mathcal{R}^+(f^+_{j-s_1}, \dots, f^+_{j+s_2}, x_{j+\frac{1}{2}})$
  $\hat{f}^-_{j+\frac{1}{2}} = \mathcal{R}^-(f^-_{j-s_1+1}, \dots, f^-_{j+s_2+1}, x_{j+\frac{1}{2}})$
  $\hat{f}_{j+\frac{1}{2}} = \hat{f}^+_{j+\frac{1}{2}} + \hat{f}^-_{j+\frac{1}{2}}.$
end
```

where the functions $f^{\pm}$ define a flux-splitting that satisfies $f^+ + f^- = f$ and the eigenvalues λ^k satisfy $\pm\lambda^k\left((f^{\pm}(u))'\right) \geq 0$ ($f^{\pm}$ are upwind fluxes) for $u \in [u_j, u_{j+1}]$. In their work, Shu and Osher [41] use a local Lax-Friedrichs (LLF) flux-splitting version of the ENO algorithms.

To extend these schemes to systems of conservation laws we can compute the numerical flux $\hat{f}_{i+\frac{1}{2}}$ (we drop the j subscript for simplicity) by using a fifth order Donat-Marquina's flux-splitting [11], which uses local characteristic decompositions of the flux Jacobians and projections of the state variables and fluxes onto characteristic fields. for the case of a fifth order method the formula is:

$$\begin{aligned}\hat{f}_{i+\frac{1}{2}} = &\sum_{k=1}^{m} r^{+,k}\left(\mathcal{R}^+\left(l^{+,k}\cdot f^{+,k}_{i-2}, \dots, l^{+,k}\cdot f^{+,k}_{i+2}; x_{i+\frac{1}{2}}\right)\right)\\ &+\sum_{k=1}^{m} r^{-,k}\left(\mathcal{R}^-\left(l^{-,k}\cdot f^{-,k}_{i-1}, \dots, l^{-,k}\cdot f^{-,k}_{i+3}; x_{i+\frac{1}{2}}\right)\right),\end{aligned} \tag{2.17}$$

where $f^{\pm,k}_l = f^{\pm,k}(u_l)$ as defined below, $r^{\pm,k} = r^k(u^{\pm}_{i+\frac{1}{2}}), l^{\pm,k} = l^k(u^{\pm}_{i+\frac{1}{2}})$ are the right and left normalized eigenvectors corresponding to the eigenvalue $\lambda_k(f'(u^{\pm}_{i+\frac{1}{2}}))$ of the flux Jacobian $f'(u^{\pm}_{i+\frac{1}{2}})$, respectively, computed at $u^{\pm}_{i+\frac{1}{2}}$, where

$$u^+_{i+\frac{1}{2}} = \mathcal{I}^+(u_{i-2}, \dots, u_{i+2}; x_{i+\frac{1}{2}}), \quad u^-_{i+\frac{1}{2}} = \mathcal{I}^-(u_{i-1}, \dots, u_{i+3}; x_{i+\frac{1}{2}}),$$

for some interpolators $I^{\pm}$. The functions $f^{\pm,k}$ satisfy $f^{+,k}+f^{-,k}=f$, $\pm\lambda_k((f^{\pm,k})'(u))>0$ for u in some relevant range $\mathcal{M}_{i+\frac{1}{2}}$ near $u^{\pm}_{i+\frac{1}{2}}$, and are given by:

$$(f^{-,k},f^{+,k})(v)=\begin{cases}(0,f(v)), & \lambda_k(f'(u))>0, \quad \forall u\in\mathcal{M}_{i+\frac{1}{2}}\\ (f(v),0), & \lambda_k(f'(u))<0, \quad \forall u\in\mathcal{M}_{i+\frac{1}{2}}\\ (F_{-\alpha^k_{i+\frac{1}{2}}}(v),F_{\alpha^k_{i+\frac{1}{2}}}(v)), & \exists u\in\mathcal{M}_{i+\frac{1}{2}}/\lambda_k(f'(u))=0,\end{cases}$$

where $\alpha^k_{i+\frac{1}{2}}\geq|\lambda_k(f'(u))|$ for $u\in\mathcal{M}_{i+\frac{1}{2}}$ and $F_\alpha(v)=\frac{1}{2}(f(v)+\alpha v)$. For the Euler equations we can simply take $\mathcal{M}_{i+\frac{1}{2}}=\{u_i,u_{i+1}\}$.

2.2.6 WENO reconstruction method

For the Essentially Non Oscillatory (ENO) schemes, introduced by Harten et al. in [16], a given cell interface reconstruction is obtained by choosing one of the different polynomial reconstructions of a given degree that can be constructed using stencils that contain one of the cells that define the given interface. The stencil choice is based on the smoothness of the numerical solution on it and the obtained reconstructions are r-th order accurate when considering r stencils (consecutive indexes) of length r containing the target cell, with the condition that at least one of the stencils does not contain a singularity. During the stencil selection procedure the ENO method considers r possible stencils, which altogether contain $2r-1$ cells.

Weighted Essentially Non Oscillatory (WENO) reconstructions, introduced by Liu, Osher and Chan in [35], are based on the idea of increasing the order of accuracy of the method in smooth regions by considering a reconstruction given by a convex combination of the different polynomial reconstruction candidates of the ENO method, with spatially varying weights designed to increase the accuracy of the individual reconstructions corresponding to the different stencils. In [35], the r-th order of accuracy of the ENO method obtained with stencils of r points was raised to $r+1$ in smooth regions, whilst retaining the r-th order near discontinuities. The weight assigned to the polynomial reconstruction associated to a given stencil depends on a smoothness indicator, for which they used a suitably weighted sum of squares of (undivided) differences of the data corresponding to that stencil. A new smoothness indicator was proposed by Jiang and Shu in [25] to achieve fifth-order reconstructions from third-order ENO reconstructions, i.e. an order of $2r-1$ when $r=3$.

We describe next the ENO and WENO reconstruction schemes used in this work.

Let $h=\Delta x$ be the grid spacing. In the ENO algorithm [16] a left-biased approximation to the value $f(x_{j+\frac{1}{2}})$ is computed using the values $\bar{f}_l$ at stencils of r nodes ($r \geq 2$) that contain the node x_j. There are r stencils of r nodes that contain x_j, given by

$$S^r_{j+k} = \{x_{j+k-r+1}, \dots, x_{j+k}\}, \quad k=0,\dots,r-1.$$

From them, r different polynomial reconstructions of degree at most $r-1$, denoted by $p^r_k(x)$, can be constructed, each of them satisfying

$$p^r_k(x_{j+\frac{1}{2}}) = f(x_{j+\frac{1}{2}}) + \mathcal{O}(h^r)$$

if f is smooth in the corresponding stencil.

Among all the candidate substencils the ENO algorithm selects the substencil producing the smallest divided differences, in an attempt to produce less oscillatory interpolants, see [1, 16] for further details. The polynomial reconstruction $p^r_k(x_{j+\frac{1}{2}})$ would be the r-th order accurate approximation of the numerical flux computed by the ENO algorithm if the stencil S^r_{j+k} had been chosen in the stencil selection procedure.

Weighted ENO reconstructions appeared in [35] as an improvement upon ENO reconstructions. In [35], Liu et al. state that there is no need of selecting just one of the possible stencils, and that a combination of them can give better results in smooth regions. If f is smooth in all stencils, a $(2r-1)$-th order reconstruction

$$p^{2r-1}_{r-1}(x_{j+\frac{1}{2}}) = f(x_{j+\frac{1}{2}}) + \mathcal{O}\left(h^{2r-1}\right)$$

can be computed using the stencil $S^{2r-1}_{j+r-1} = \{x_{j-r+1}, \dots, x_{j+r-1}\}$, instead of the r-th order reconstruction provided by the ENO algorithm.

If we consider the r candidate stencils of the ENO algorithm, $S^r_{j+k} = \{x_{j-r+1+k}, \dots, x_{j+k}\}$, $k=0,\dots,r-1$, and the $(r-1)$-th degree polynomial reconstructions $p^r_k(x)$, defined on each stencil S^r_{j+k}, satisfying $p^r_k(x_{j+\frac{1}{2}}) = f(x_{j+\frac{1}{2}}) + \mathcal{O}(h^r)$, then a (left-biased) WENO reconstruction of f is given by the convex combination:

$$q(x_{j+\frac{1}{2}}) = \sum_{k=0}^{r-1} w_k p^r_k(x_{j+\frac{1}{2}}), \tag{2.18}$$

where:

$$w_k \geq 0,\ k = 0, \dots, r-1, \qquad \sum_{k=0}^{r-1} w_k = 1$$

and the corresponding (left-biased) reconstruction evaluation operator is given by:

$$\mathcal{R}(\bar{f}_{j-r+1}, \dots, \bar{f}_{j+r-1}) = \sum_{k=0}^{r-1} \omega_{j,k} p^r_{j,k}(x_{j+\frac{1}{2}}).$$

The weights should be selected with the goal of achieving the maximal order of accuracy $2r-1$ wherever f is smooth, and $r-$th order, as the ENO algorithm, elsewhere.

As in the original WENO approach [35], we first note that for $r \geq 2$, coefficients C^r_k, called optimal weights, can be computed such that:

$$p^{2r-1}_{r-1}(x_{j+\frac{1}{2}}) = \sum_{k=0}^{r-1} C^r_k p^r_k(x_{j+\frac{1}{2}}),$$

where,

$$C^r_k \geq 0\ \forall k, \qquad \sum_{k=0}^{r-1} C^r_k = 1.$$

In [2], Aràndiga et al. give different explicit formulae for the polynomial reconstructions and the optimal weights.

Notice that to accomplish the requirements on the non-linear weights w_k one can define them satisfying the condition:

$$w_k = C_k + \mathcal{O}(h^m), \qquad k = 0, \dots, r, \tag{2.19}$$

with $m \leq r-1$. Then, there holds (see [2], [35]) that

$$f(x_{j+\frac{1}{2}}) - q(x_{j+\frac{1}{2}}) = \mathcal{O}(h^{r+m}), \tag{2.20}$$

and, if $m = r-1$ in (2.19), then the approximation (2.20) has maximal order $2r-1$.

Another requirement for the weights is that the ones corresponding to polynomials constructed using stencils where the function has a singularity should be very small, so that the WENO reconstruction does not take those polynomials into account and, as required, yields an approximation of an order not worse than that of the ENO interpolators. Besides, the weights should be smooth functions of the cell-averages of the reconstructed function and efficiently computable.

Weights satisfying these conditions are defined in [35] as follows:

$$w_k = \frac{\alpha_k}{\sum_{i=0}^{r-1} \alpha_i}, \quad \alpha_k = \frac{C_k^r}{(\varepsilon + I_k)^p}, \quad k = 0, \ldots, r-1, \tag{2.21}$$

where $p \in \mathbb{N}$, C_k^r are the optimal weights, $I_k = I_k(h)$ is an smoothness indicator of the function f on the stencil S_k and ε is an small positive number, possibly dependent on h, introduced to avoid null denominators, but, as we will see later on in this thesis, it has a strong influence in the overall performance of the approximations at critical points and at discontinuities. The weights thus defined satisfy $\sum_k \omega_k = 1$ independently of the smoothness indicator choice.

We use Jiang and Shu's smoothness indicator (see [25]):

$$I_k = \sum_{l=1}^{r-1} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} h^{2l-1} (p_k^{(l)}(x))^2 dx, \tag{2.22}$$

with which they obtained WENO schemes with optimal order $2r-1$ for $r = 2, 3$. The term h^{2l-1} was introduced to remove h-dependent factors in the derivatives of the polynomial reconstructions $p_k(x)$.

In [2], the authors give explicit formulae for the optimal weights C_k^r and polynomial values $p_k^r(x_{j+\frac{1}{2}})$ for $k = 0, \ldots, r-1$.

The optimal weights for $r = 2, 3, 4, 5$ obtained using these explicit formulae are displayed in Table 2.1.

r	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$
2	1/3	2/3			
3	1/10	6/10	3/10		
4	1/35	12/35	18/35	4/35	
5	1/126	20/126	60/126	40/126	5/126

Table 2.1: Optimal weights for $r = 2, 3, 4, 5$.

In [2], Aràndiga et al. prove that the order of accuracy of the scheme is $2r-1$ when using stencils of length $2r-1$ contained in smooth regions, regardless of neighboring extrema, whereas this order is at least r when at least one of the substencils involved in the weighted average does not cross a discontinuity. They also show that for achieving the maximal order $2r-1$ at any smooth region with the original weights proposed by Liu, Osher and Chan in [35] (given by (2.21)), the choice of ε being proportional to h^2 is optimal.

2.3 Faà di Bruno's formula

High order derivatives of composition of functions are ubiquitous in numerical analysis and the following generalization of the chain rule, known as Faà di Bruno's formula [13], comes in very handy in many situations. For the sake of completeness, we include a proof of Theorem 2, for which we have not found satisfactory references.

Theorem 2 (Faà di Bruno formula). *Let $f : \mathbb{R}^m \to \mathbb{R}^p$, $u : \mathbb{R} \to \mathbb{R}^m$ n times continuously differentiable. Then*

$$\frac{d^n f(u(t))}{dt^n} = \sum_{s\in\mathcal{P}_n} \left[\begin{array}{c} n \\ s \end{array} \right] f^{(|s|)}(u(t))D^s u(t), \tag{2.23}$$

where $\mathcal{P}_n = \{s \in \mathbb{N}^n / \sum_{j=1}^n js_j = n\}$, $|s| = \sum_{j=1}^n s_j$, $\left[\begin{array}{c} n \\ s \end{array} \right] = \dfrac{n!}{s_1! \cdots s_n!}$, $D^s u(t)$ is an $m \times |s|$ matrix whose $(\sum\limits_{l<j} s_l + k)$-th column is given by

$$(D^s u(t))_{\sum\limits_{l<j} s_l + k} = \frac{1}{j!}\frac{\partial^j u(x)}{\partial t^j}, \quad k = 1, \dots, s_j, \quad j = 1, \dots, n, \tag{2.24}$$

and the action of the k-th derivative tensor of f on a $m \times k$ matrix A is given by

$$f^{(k)}(u)A = \sum_{i_1,\dots,i_k=1}^{m} = \frac{\partial^k f}{\partial u_{i_1} \dots \partial u_{i_k}}(u) A_{i_1,1} \dots A_{i_k,k} \in \mathbb{R}^p. \tag{2.25}$$

Denote by $\mathcal{M}(s,n)$ the vector space of multilinear functions (tensors),

$$T\colon \overbrace{\mathbb{R}^n \times \cdots \times \mathbb{R}^n}^{s} \to \mathbb{R}.$$

Since $\overbrace{\mathbb{R}^n \times \cdots \times \mathbb{R}^n}^{s}$ is isomorphic to the vector space of $n \times s$ matrices, we can regard s-tensors as acting on the columns of $n \times s$ matrices. Tensors can be characterized as $\overbrace{n \times \cdots \times n}^{s}$ matrices $(T_{i_1,\dots,i_s})$, i.e.,

$$T(A) = TA = \sum_{i_1=\cdots=i_s=1}^{n} T_{i_1,\dots,i_s} A_{i_1,1} \dots A_{i_s,s}.$$

The following result is easily established.

Lemma 2. *Assume* $T\colon \mathbb{R}^n \to \mathcal{M}(s,n)$ *is differentiable (equivalently,* $T_{i_1,\dots,i_{i_s}}$ *are differentiable) and that* $A\colon \mathbb{R} \to \mathbb{R}^{n\times s}$*,* $u : \mathbb{R} \to \mathbb{R}^n$ *are also differentiable. Then,* $\forall x \in \mathbb{R}$

$$\frac{d}{dx}T(u(x))A(x) = T'(u(x))[u'(x)\ A(x)] + T(u(x))\sum_{j=1}^{s} d_jA(x),$$

where we have used the notation $d_jA(x)$ *for the* $n \times s$ *matrix given by the columns:*

$$(d_jA(x))_k = \begin{cases} A_k(x) & k \neq j \\ A'_j(x) & k = j \end{cases}$$

We introduce some further notation for the proof of Theorem 2. For $s \in \mathbb{N}$, we denote

$$\mathcal{P}_{s,j} = \{m \in \mathcal{P}_s / m_j \neq 0\}.$$

We denote also

$$S_0\colon \mathcal{P}_s \to \mathcal{P}_{s+1,1}, \quad S_0(m)_k = \begin{cases} 0 & k = s+1 \\ m_k & s \geq k \neq 1 \\ m_1 + 1 & k = 1, \end{cases}$$

$$S_j\colon \mathcal{P}_{s,j} \to \mathcal{P}_{s+1,j+1}, \quad S_j(m)_k = \begin{cases} 0 & k = s+1 \\ m_k & s \geq k \neq j, j+1 \\ m_j - 1 & s \geq k = j \\ m_{j+1} + 1 & s \geq k = j+1. \end{cases}$$

for $1 \leq j < s$, and S_s that maps $(0,\dots,0,1) \in \mathbb{N}^s$ to $(0,\dots,0,1) \in \mathbb{N}^{s+1}$.

Proof. (of Theorem 2) We use induction on s, the case $s = 1$ being the chain rule. By the induction hypothesis for s and Lemma 2 we deduce:

$$\frac{d^{s+1}f(u(x))}{dx^{s+1}} = \sum_{m\in\mathcal{P}_s} \left[\begin{array}{c} s \\ m \end{array}\right] \frac{d}{dx}\left(f^{(|m|)}(u(x))D^m u(x)\right)$$

$$= \sum_{m\in\mathcal{P}_s} \left[\begin{array}{c} s \\ m \end{array}\right] \left((f^{(|m|)})'(u(x))[u'(x)\ D^m u(x)] + f^{(|m|)}(u(x))\sum_{j=1}^{n} d_jD^m u(x)\right)$$

$$= \sum_{m\in\mathcal{P}_s} \left[\begin{array}{c} s \\ m \end{array}\right] \left(f^{(|m|+1)}(u(x))[u'(x)\ D^m u(x)] + f^{(|m|)}(u(x))\sum_{j=1}^{n} d_jD^m u(x)\right).$$

Now,

$$d_j D^m u(x) = D^{S_j(m)} u(x) P E,$$

where P is a permutation matrix correspondig to the transposition of j and $\sum_{l\leq k} m_l$, with $\sum_{l<k} m_l < j \leq \sum_{l\leq k} m_l$ and E is a diagonal matrix with $k+1$ in the $\sum_{l\leq k} m_l$ entry and 1 in the rest.

By the symmetry of $f^{(|m|)}$, if $\sum_{l<k} m_l < j \leq \sum_{l\leq k} m_l$

$$f^{(|m|)}(u(x)) d_j D^m u(x) = (k+1) f^{(|S_k(m)|}(u(x)) D^{S_k(m)} u(x),$$

therefore, collecting identical terms,

$$\frac{d^{s+1} f(u(x))}{dx^{s+1}} = \sum_{m\in\mathcal{P}_s} \begin{bmatrix} s \\ m \end{bmatrix} \big(f^{(|S_0(m)|)}(u(x)) D^{S_0(m)} u(x) \\ + \sum_{j=1}^{n} f^{(|m|)}(u(x)) d_j D^{S_j(m)} u(x) \big)$$

can be written as

$$\begin{aligned} \frac{d^{s+1} f(u(x))}{dx^{s+1}} = \sum_{m\in\mathcal{P}_s} \begin{bmatrix} s \\ m \end{bmatrix} \big(& f^{(|S_0(m)|)}(u(x)) D^{S_0(m)} u(x) \\ & + \sum_{k=1}^{n} m_k (k+1) f^{(|S_k(m)|)}(u(x)) D^{S_k(m)} u(x) \big), \end{aligned} \tag{2.26}$$

where we point out that in the last expression the only terms that actually appear are those for which $m_k > 0$. Since $m_k - 1 = (S_k(m))_k$, by collecting the terms for m, k such that $S_k(m) = \widehat{m}$, (2.26) can be written as

$$\frac{d^{s+1} f(u(x))}{dx^{s+1}} = \sum_{\widehat{m}\in\mathcal{P}_{s+1}} a_{\widehat{m}} f^{(|\widehat{m}|)}(u(x)) D^{\widehat{m}} u(x), \tag{2.27}$$

where

$$a_{\widehat{m}} = \begin{cases} \widetilde{a_{\widehat{m}}} & \text{if } \widehat{m_1} = 0 \\ \widetilde{a_{\widehat{m}}} + \begin{bmatrix} s \\ S_0^{-1}(\widehat{m}) \end{bmatrix} & \text{if } \widehat{m_1} \neq 0, \end{cases} \quad \widetilde{a_{\widehat{m}}} = \sum_{\substack{\widehat{m} = S_k(m), \\ k \in \{1,\dots,s\}, \\ m \in \mathcal{P}_{s,k}}} \begin{bmatrix} s \\ m \end{bmatrix} m_k (k+1). \tag{2.28}$$

For $k \in \{1, \dots, s\}$, and $m \in \mathcal{P}_{s,k}$, such that $\widehat{m} = S_k(m)$, i.e., $\widehat{m}_i = m_i$, $i \neq k, k+1$, $\widehat{m}_k = m_k - 1$, $\widehat{m}_{k+1} = m_{k+1} + 1$, we deduce:

$$\begin{aligned}\begin{bmatrix} s \\ m \end{bmatrix} m_k(k+1) &= \frac{s!}{m_1! \dots (m_k - 1)! m_{k+1}! \dots m_s!}(k+1) \\ &= \frac{s!}{\widehat{m}_1! \dots \widehat{m}_k! (\widehat{m}_{k+1} - 1)! \dots \widehat{m}_s!}(k+1) \\ &= \frac{s!}{\widehat{m}_1! \dots \widehat{m}_k! \widehat{m}_{k+1}! \dots \widehat{m}_s!} \widehat{m}_{k+1}(k+1).\end{aligned}$$

Let $\widehat{m} = S_k(m)$ with $k < s$, then one has $\widehat{m}_{s+1} = 0$. The only element $m \in \mathcal{P}_{s,s}$ is $(0, \dots, 0, 1) \in \mathbb{N}^s$ and $S_s(m) = (0, \dots, 0, 1) \in \mathbb{N}^{s+1}$. Therefore

$$\widetilde{a_{\widehat{m}}} = \frac{s!}{\widehat{m}_1! \dots \widehat{m}_{s+1}!} \sum_{\substack{\widehat{m} = S_k(m), \\ k \in \{1, \dots, s\}, \\ m \in \mathcal{P}_{s,k}}} \widehat{m}_{k+1}(k+1)$$

$$\widetilde{a_{\widehat{m}}} = \frac{s!}{\widehat{m}_1! \dots \widehat{m}_{s+1}!} \sum_{k=1}^{s} \widehat{m}_{k+1}(k+1) = \frac{s!}{\widehat{m}_1! \dots \widehat{m}_{s+1}!} \sum_{k=2}^{s+1} \widehat{m}_k k. \tag{2.29}$$

On the other hand, if $\widehat{m}_1 \neq 0$, then:

$$\begin{bmatrix} s \\ S_0^{-1}(\widehat{m}) \end{bmatrix} = \frac{s!}{(\widehat{m}_1 - 1)! \widehat{m}_2! \cdots \widehat{m}_s!} = \frac{s!}{\widehat{m}_1! \widehat{m}_2! \cdots \widehat{m}_s! \widehat{m}_{s+1}!} \widehat{m}_1, \tag{2.30}$$

where the last equality holds since, as before, we have $\widehat{m}_{s+1} = 0$. Then, regardless of $\widehat{m}_1$, (2.29) and (2.30) yield for $\widehat{m} \in \mathcal{P}_{s+1}$

$$a_{\widehat{m}} = \frac{s!}{\widehat{m}_1! \dots \widehat{m}_{s+1}!} \sum_{k=1}^{s+1} \widehat{m}_k k = \frac{s!}{\widehat{m}_1! \dots \widehat{m}_{s+1}!}(s+1) = \begin{bmatrix} s+1 \\ \widehat{m} \end{bmatrix}, \tag{2.31}$$

since $\widehat{m} \in \mathcal{P}_{s+1}$ means $\sum_{k=1}^{s+1} \widehat{m}_k k = s+1$. We deduce from (2.27), (2.28) and (2.31) that

$$\frac{d^{s+1} f(u(x))}{dx^{s+1}} = \sum_{\widehat{m} \in \mathcal{P}_{s+1}} \begin{bmatrix} s+1 \\ \widehat{m} \end{bmatrix} f^{(|\widehat{m}|)}(u(x)) D^{\widehat{m}} u(x),$$

which concludes the proof by induction. □

3

Meshing procedure for complex domains

The first question that may arise when tackling a problem with boundary conditions is how to discretize them in a way such that the numerical method can properly mix the required boundary information with the data from the interior nodes.

In order to do so we propose a meshing procedure that relies on the computation of the intersections between the mesh lines, where the nodal information is contained, and the domain boundary. In this work, we will focus on solving two-dimensional systems of conservation laws with complex domains, and therefore we will assume $d = 2$ along this chapter.

As a first simplification, for now we will assume that $\Omega \subseteq \mathbb{R}^2$ is a simply connected open domain such that $\exists \alpha : [a, b] \to \mathbb{R}^2$, $\alpha \in C^2([a, b], \mathbb{R}^2)$ closed curve (namely, $\alpha(a) = \alpha(b)$) such that $\alpha([a, b]) = \partial\Omega$. In a more general case scenario we can consider domains whose boundary is the union of a finite number of closed curves.

There are mainly two ways in which such a boundary can be described:

1. By means of a differentiable curve parametrized in a compact interval.

2. Through an implicit equation of the form $F(x, y) = 0$. In this case, it can be determined if a point $(x, y) \in \mathbb{R}^2$ belongs to Ω, $\partial\Omega$ or $\mathbb{R}^2 \setminus \overline{\Omega}$ in terms of the sign of $F(x, y)$.

Our analysis will be focused on the first case, since, although working with parametrizations is more complicated in some senses, such as computing intersections, it is convenient to tackle this case because it is easier to describe a curve through parametrizations rather than by an implicit equation. Our goal is thus obtaining an automated process to mesh the interior of a closed curve, in terms of an algorithm which can be implemented in a computer.

3.1 Safe detection of intersections

There are many ways to mesh a set, but as the numerical methods used in this work to solve the physical equations are finite-difference schemes we will focus on the case of Cartesian meshes, i.e., meshes whose cells are rectangular and identically distributed. We must therefore design a strategy to automate the meshing procedure, that is, the computation of all the intersections of the boundary with the mesh lines, ghost cells, normal lines, etc. regardless the parametrization of the curve.

Let $\alpha : [a, b] \to \mathbb{R}^2$ be a piecewise C^2 curve such that $\alpha(a) = \alpha(b)$. Let us assume we want to establish a grid into its interior with horizontal and vertical lines of the form $x_k = x_0 + kh_x$ and $y_k = y_0 + kh_y$, $k \in \mathbb{Z}$, with $h_x, h_y > 0$ the vertical and horizontal spacings, respectively.

We will show the procedure for the computation of the intersection between the horizontal mesh lines and the boundary, being the vertical case completely analogous. Starting at $\alpha(a)$ the procedure follows the boundary curve trying to find the values of the parameter defining the boundary where horizontal intersections occur. Hence, we will focus only on the second component of α, α_2. We take our starting parameter $s_0 = a$ and consider the value $\alpha_2(s_0)$. The first step is to find k such that $\alpha_2(s_0)$ is between $y_0 + kh_y$ and $y_0 + (k+1)h_y$. By continuity considerations it is clear that the closest intersection involves one of these two lines. It holds that $\alpha_2(s_0) = y_0 + \tilde{k}h_y$, where $\tilde{k} := \frac{\alpha_2(s_0)-y_0}{h_y}$. Therefore, the values we are looking for are $y_0 + kh$ and $y_0 + (k+1)h$, with $k = \lfloor \tilde{k} \rfloor$.

Now we must perform a suitable parameter displacement in order to approach the first intersection, with enough care in order not to surpass two or more lines in a single step as in that case some intersections could be unintentionally discarded. This is where we take advantage of the assumption that curve is twice differentiable with second derivative continuous.

We are thus interested in finding $\Delta s > 0$ such that the inequality $|\alpha_2(s_0 + \Delta s) - \alpha_2(s_0)| \leq h_y$ holds and on the other hand $\alpha_2(s_0 + \Delta s)$ is as big as possible in order to reduce the number of iterations (curve evaluations) required to find the intersection as much as possible.

For the enforcement of the above inequality a natural way to proceed is to use the mean value theorem for real functions:

$$|\alpha_2(s_0 + \Delta s) - \alpha_2(a)| = \Delta s|\alpha_2'(s_0 + \xi)|, \quad \xi \in (0, \Delta s).$$

The expression involving α_2' can be bounded through different considerations. The simplest one is by using its boundedness as α_2' is continuous at the compact set $[a, b]$: $\exists L_1 > 0$ such that $|\alpha_2'(s)| \leq L_1$ $\forall s \in [a, b]$; in particular, taking $s = a + \xi$ we have $|\alpha_2'(a + \xi)| \leq L_1$ as well. Therefore:

$$|\alpha_2(s_0 + \Delta s) - \alpha_2(s_0)| = \Delta s|\alpha_2'(s_0 + \xi)| \leq L_1 \Delta s.$$

Hence, if we want $|\alpha_2(s_0+\Delta s)-\alpha_2(s_0)| \leq h_y$ it suffices to impose $L_1\Delta s \leq h_y$, and therefore the choice

$$\Delta s_1 := \frac{h_y}{L_1}$$

is valid.

As indicated previously, we are interested in finding Δs as large as possible in order to avoid unnecessary evaluations of the curve, thus the optimal value for L_1 is

$$L_1 = \max_{s\in[a,b]} |\alpha_2'(s)|.$$

Therefore, if the maximum can be computed, we take its value as L_1 and an upper bound as fine as possible otherwise.

The bound can be optionally refined by increasing the accuracy order of the approximation from first to second order. To do so, we perform a second order Taylor expansion:

$$\alpha_2(s_0 + \Delta s) = \alpha_2(s_0) + \Delta s\alpha_2'(s_0) + \frac{\Delta s^2}{2}\alpha_2''(s_0 + \xi), \quad \xi \in (0, \Delta s).$$

Therefore

$$|\alpha_2(s_0+\Delta s)-\alpha_2(s_0)| \le \Delta s|\alpha_2'(s_0)| + \frac{\Delta s^2}{2}|\alpha_2''(s_0+\xi)| \le \Delta s|\alpha_2'(s_0)| + \frac{\Delta s^2}{2}L_2,$$

where

$$L_2 = \max_{s\in[a,b]} |\alpha_2''(s)|,$$

which exists since by assumption α_2'' is continuous in the compact set $[a,b]$.

Thus, to satisfy

$$|\alpha_2(s_0+\Delta s)-\alpha_2(s_0)| \le h_y$$

it suffices to impose

$$\Delta s|\alpha_2'(s_0)| + \frac{\Delta s^2}{2}L_2 \le h_y,$$

which solving for $\Delta s > 0$ yields

$$\Delta s \le \frac{\sqrt{|\alpha_2'(s_0)|^2 + 2L_2h_y} - |\alpha_2'(s_0)|}{L_2} = \frac{2h_y}{\sqrt{|\alpha_2'(s_0)|^2 + 2L_2h_y} + |\alpha_2'(s_0)|}.$$

Therefore, the optimal parameter step in this case is

$$\Delta s_2 = \frac{2h_y}{\sqrt{|\alpha_2'(s_0)|^2 + 2L_2h_y} + |\alpha_2'(s_0)|}.$$

In practice, we will take the maximum value between the first and second order approach as optimal step:

$$\Delta s = \max\{\Delta s_1, \Delta s_2\}.$$

We denote $s_1 = \alpha_2(s_0+\Delta s)$.

The procedure that has been described up to now is a method to capture every single possible intersection of the grid lines with the boundary of the domain, which is an indispensable information to mesh the domain. After each step, we must check if a mesh line has been crossed (i.e., $\alpha_2(s_0) \in [y_k, y_{k+1}]$ and $\alpha_2(s_1) \in [y_{k+1}, y_{k+2}]$ or $\alpha_2(s_1) \in [y_{k-1}, y_k]$). If this is the case, we know by continuity arguments that there is an intersection between the two parameters. In this situation, we can compute through a safe procedure an accurate approximation of the parameter corresponding to such intersection, which is described next. Otherwise we replace s_0 by s_1 and repeat the procedure until an horizontal mesh line is crossed.

3.2 Newton's method with bisection control

The previous approach reduces the problem to the one of finding a root of a continuous function within a certain interval. More precisely, if we have $y_0{+}kh_y \leq \alpha_2(s_0) \leq y_0{+}(k{+}1)h_y$ and $y_0{+}(k{+}1)h_y \leq \alpha_2(s_1) \leq y_0{+}(k{+}2)h_y$ then by Bolzano's theorem $\exists c \in [s_0, s_1]$ such that $\alpha_2(c) = y_0{+}(k{+}1)h_y$. If we define $f(s) = \alpha_2(s) - (y_0 + (k{+}1)h_y)$, then $f(s_0) \leq 0$, $f(s_1) \geq 0$ and c is a root of f, namely, $f(c) = 0$. Therefore this problem can be translated to find the root of the function f. Recall that f is a function of class $\mathcal{C}^2([s_0, s_1])$ that changes sign at that interval, and hence by Bolzano's theorem $\exists c \in (s_0, s_1)$ such that $f(c) = 0$. Let us denote by simplicity $[a, b] = [s_0, s_1]$.

Let us recall that the bisection method consists on taking the middle point x_1 of the interval (a, b), $x_1 := \frac{a+b}{2}$ and then evaluate the sign of $f(x_1)$. If sign$(f(x_1))$ = sign$(f(a))$ then a root is located at (x_1, b), an interval whose length is half the original one. Analogously, if sign$(f(x_1))$ = sign$(f(b))$ then a root is located at (a, x_1), which is again an interval of half the size of the original. Then we repeat the process by computing x_2 as the midpoint of the current interval and so on. The decreasing size of the intervals and the continuity of f yields the convergence of this algorithm.

On the other hand, Newton's method takes a starting point x_0 reasonably close to the root c in order to generate a recurrence in an attempt to approximate it:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n \in \mathbb{N}^*.$$

This method is much faster than the bisection method with an excellent convergence rate provided f' is far enough from zero in a neighborhood of c.

Alternatively, the secant method allows to approximate the root as well without using the expression of the first derivative, by approximating it through

$$x_{n+1} = x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})},$$

with $x_0 = a$ and $x_1 = b$.

The last two methods, however, can be extremely slow or even divergent if f' is close to zero around c. We next describe how to obtain a hybrid procedure combining the robustness of the bisection method with the efficiency of methods as Newton's or secant method. The algorithm produces a sequence of intervals $[a_n, b_n]$ of decreasing size that contain the root of f, with a convergence rate equal or higher to the one of the bisection method.

Initially we take $a_0 = a$, $b_0 = b$ such that $f(a_0)f(b_0) < 0$ and a starting point $x_0 = \frac{a+b}{2}$. Assume we are in step n, with $a_{n-1} \leq a_n < b_n \leq b_{n-1}$, $f(a_n)f(b_n) < 0$ and x_n as approximation of c. Then we compute x_{n+1} either by Newton's or secant method and then: if $f(x_{n+1})$ is close enough to zero then the procedure stops since the root has been found. Otherwise, we consider the following two cases:

- If $|f(x_{n+1})| > \dfrac{|f(x_n)|}{2}$ or $x_{n+1} \notin [a_n, b_n]$, then the convergence rate is roughly worse than the bisection method or the new iteration point is out of the safe bounds, and then we redefine x_{n+1} as $x_{n+1} = \dfrac{a_n + b_n}{2}$.

- If $|f(x_{n+1})| \leq \dfrac{|f(x_n)|}{2}$ and $x_{n+1} \in [a_n, b_n]$, then the local convergence rate is roughly the same or better than the bisection method and inside the safe zone, and then we keep the value of x_{n+1}.

In both cases, if $\text{sign}(f(x_{n+1})) = \text{sign}(f(a_n))$ then we define $a_{n+1} = x_{n+1}$ and $b_{n+1} = b_n$; on the other hand, if $\text{sign}(f(x_{n+1})) = \text{sign}(f(b_n))$ then we define $a_{n+1} = a_n$ and $b_{n+1} = x_{n+1}$.

The procedure stops once $|f(x_{n+1})|$ or $|b_{n+1} - a_{n+1}|$ are below a certain tolerance. This algorithm ensures that the convergence rate is at least the one offered by the bisection method.

3.3 Ghost cells

WENO schemes of odd order, say $2\ell - 1$, use a stencil (set of consecutive indexes) of 2ℓ points to perform a reconstruction at each cell interface, therefore ℓ additional cells are needed at both sides of each cell in order to perform a time step. For cells close to the boundary some of these additional cells may fall outside the computational domain and in that case they are usually named *ghost cells* and, in terms of their centers, are given by:

$$\mathcal{GC} := \mathcal{GC}_x \cup \mathcal{GC}_y,$$

where

$$\mathcal{GC}_x := \{(x_r, y_s): \ \ 0 < d\,(x_r,\ \Pi_x\,(\Omega \cap (\mathbb{R} \times \{y_s\}))) \leq kh_x, \quad r, s \in \mathbb{Z}\}\,,$$

$$\mathcal{GC}_y := \{(x_r, y_s): \ \ 0 < d\,(y_s,\ \Pi_y\,(\Omega \cap (\{x_r\} \times \mathbb{R}))) \leq kh_y, \quad r, s \in \mathbb{Z}\}\,,$$

where Ω is the computational domain, Π_x and Π_y denote the projections on the respective coordinates and

$$d(a, B) := \inf\{|b - a| : \ b \in B\},$$

for given $a \in \mathbb{R}$ and $B \subseteq \mathbb{R}$. Notice that $d(a, \emptyset) = +\infty$, since, by convention, $\inf \emptyset = +\infty$.

3.4 Normal lines

We focus now on the two-dimensional setting and boundaries with prescribed Dirichlet conditions, e.g., reflective boundary conditions for the Euler equations. In this situation, it seems reasonable that the extrapolation at a certain ghost cell $P = (x_*, y_*) \in \mathcal{GC}$ is based on the prescribed value at the nearest boundary point. It can be proven that a point $P_0 \in \partial\Omega$ satisfying

$$\|P - P_0\|_2 = \min\{\|P - B\|_2 : \quad B \in \partial\Omega\}$$

also satisfies that the line determined by P and P_0 is normal to the curve $\partial\Omega$ at P_0, if $\partial\Omega$ is differentiable at P_0. Namely, assuming $P_0 = \alpha(s_*)$, the following condition is verified:

$$\langle P - P_0, \alpha'(s_*)\rangle = 0.$$

This yields an iterative procedure to automatically approximate through Newton's or secant method the normal line associated to each ghost node P by finding the root of the function

$$F_P(s) = \langle P - \alpha(s), \alpha'(s)\rangle.$$

Uniqueness of P_0 holds whenever P is close enough to the boundary, so we will henceforth denote $N(P) = P_0$.

This argument suggests that a good strategy is to perform a (virtual) rotation of the domain and obtain data on some points $N_i \in \Omega$ on the line that passes through P and $N(P)$ (normal line to $\partial\Omega$ passing by P)

and then use a one-dimensional extrapolation from the data on these points on the segment to approximate the value at P. The details of this procedure are described in Section 3.4.1.

In case that the boundary conditions prescribe values for the normal component of a vectorial unknown $\overrightarrow{v}$ related to the coordinate frame (as is the case for reflective boundary conditions for the Euler equations), then one defines

$$\overrightarrow{n} = \frac{P - N(P)}{\|P - N(P)\|}, \quad \overrightarrow{t} = \overrightarrow{n}^{\perp},$$

and obtains normal and tangential components of $\overrightarrow{v}$ at each point N_i of the mentioned segment by:

$$v^t(N_i) = \overrightarrow{v}(N_i) \cdot \overrightarrow{t}, \quad v^n(N_i) = \overrightarrow{v}(N_i) \cdot \overrightarrow{n}.$$

The extrapolation procedure is applied to $v^t(N_i)$ to approximate $v^t(P)$ and to $v^n(N_i)$ and $v^n(N(P)) = 0$ to approximate $v^n(P)$. Once $v^t(P), v^n(P)$ are approximated, the approximation to $\overrightarrow{v}(P)$ is set to

$$\overrightarrow{v}(P) = v^t(P)\overrightarrow{t} + v^n(P)\overrightarrow{n}.$$

3.4.1 Choice of nodes on normal lines

If we wish to compute boundary data in a way such that a certain precision in the resulting scheme is formally preserved it is necessary to extrapolate information from the domain interior in an adequate manner. Therefore, if the basic numerical scheme has order r it is reasonable to use extrapolation of this order at least. For the sake of clarity, we will not distinguish between interpolation or extrapolation when these take place at the interior of the domain.

At this point there are many possibilities. However, as expected, not all of them yield the same quality in the results nor the same computational efficiency. The following configuration aims to represent a reasonable balance between both factors.

We proceed in a fashion similar to [43]. Let $(x_*, y_*) \in \mathcal{GC}$ and consider the corresponding point in $\partial\Omega$ at minimal distance, $N(x_*, y_*)$. As already mentioned, the vector determined by both points is orthogonal to $\partial\Omega$ at $N(x_*, y_*)$. Let us suppose that we wish to use an extrapolation of order r at the ghost cell center (x_*, y_*).

At first place, one needs to obtain data from the information in Ω at a set of points $\mathcal{N}(x_*, y_*) = \{N_1, \ldots, N_{R+1}\}$, with $R \geq r$, on the line determined by the points (x_*, y_*) and $N(x_*, y_*)$. By a CFL stability motivation, we will do the selection with a spacing between them of at least the distance between (x_*, y_*) and $N(x_*, y_*)$ We will choose the nodes depending on the slope of the normal line, so that the use of interior information is maximized. We denote by $v = (v_1, v_2)$ the vector determined by (x_*, y_*) and $N(x_*, y_*)$, so that the normal line passing through (x_*, y_*) is given by the parametric equations:

$$\begin{aligned} x &= x_* + sv_1, \\ y &= y_* + sv_2. \end{aligned}$$

Depending on the angle θ of the vector $v = (v_1, v_2)$, we consider two possibilities:

1. $|v_1| \geq |v_2|$.
2. $|v_1| < |v_2|$.

In the first case we take points $N'_q = (x_* + qC_x h_x, y_* + qC_x h_x \frac{v_2}{v_1})$, with $C_x \in \mathbb{Z}$ chosen with the same sign as v_1 and so that:

$$\|N'_q - N'_{q+1}\|_2 \geq \|v\|_2.$$

As

$$\|N'_q - N'_{q+1}\|_2 = \frac{h_x|C_x|}{|v_1|}\|v\|_2 \geq \|v\|_2 \Leftrightarrow |C_x| \geq \frac{|v_1|}{h_x},$$

our choice is $C_x = \lceil \frac{v_1}{h_x} \rceil$. Now, if Dirichlet boundary conditions at $P_0 := N(x_*, y_*)$ are prescribed, we take the nodes

$$\mathcal{N}(x_*, y_*) = \begin{cases} \{P_0, N'_1, \ldots, N'_R\} & \text{if } \|P_0 - N'_1\|_2 \geq \|v\|_2 \\ \{P_0, N'_2, \ldots, N'_{R+1}\} & \text{if } \|P_0 - N'_1\|_2 < \|v\|_2. \end{cases} \tag{3.1}$$

If no boundary condition is specified at $N(x_*, y_*)$ then

$$\mathcal{N}(x_*, y_*) = \{N'_1, N'_2, \ldots, N'_{R+1}\}. \tag{3.2}$$

In this fashion, the chosen nodes $\mathcal{N}(x_*, y_*) = \{N_1, \ldots, N_{R+1}\}$ satisfy $\|N_q - N_{q+1}\|_2 \geq \|v\|_2$, $q = 1, \ldots, R$.

Let us denote $N_q = (\widetilde{x}_q, \widetilde{y}_q)$. For each q for which $u(\widetilde{x}_q, \widetilde{y}_q)$ is not known, we need to obtain a sufficiently accurate approximation of this value from the information on the interior nodes. Since the second coordinate, $\widetilde{y}_q$, of N_q does not need to coincide with the center of a vertical cell, we will use interpolation from the cells in the line $x = \widetilde{x}_q$ by using the following set of points:

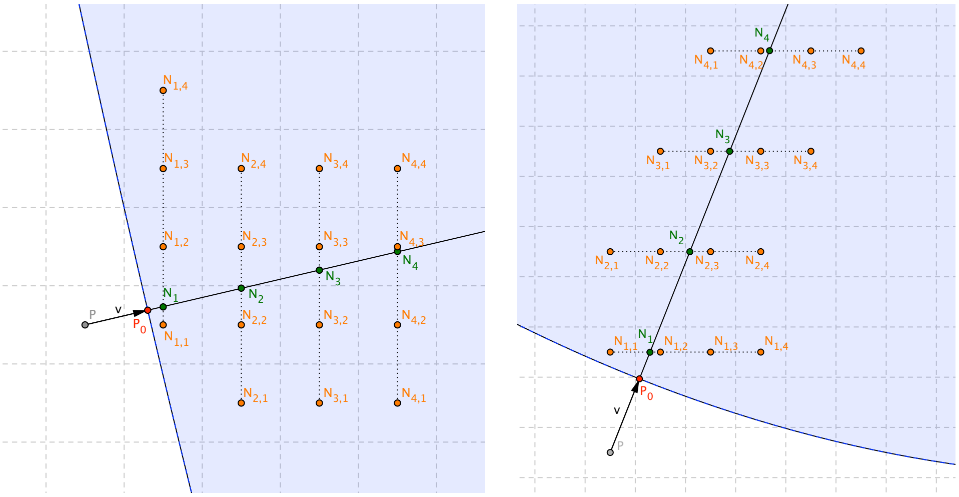
$$\mathcal{S}_q = \{N_{q,1}, \dots, N_{q,R+1}\} := \underset{A \in \mathcal{A}}{\text{argmin}} \sum_{(\widetilde{x}_q, y_s) \in A} |y_s - \widetilde{y}_q|,$$
$$\mathcal{A} := \{A = \{(\widetilde{x}_q, y_j), \dots, (\widetilde{x}_q, y_{j+R})\} / A \subseteq \Omega\}.$$

That is, we select the vertical stencil of length $R+1$ with a first coordinate fixed to $\widetilde{x}_q$ such that it be as centered as possible with respect to the point N_q, see Figure 3.1 (a) for a graphical example.

In a dual fashion, in the second case ($|v_1| < |v_2|$) we take points $N'_q = (x_* + qC_y h_y \frac{v_1}{v_2}, y_* + qC_y h_y)$, with $C_y = \lceil \frac{v_2}{h_y} \rceil$ and

$$\mathcal{S}_q = \{N_{q,1}, \dots, N_{q,R+1}\} := \underset{A \in \mathcal{A}}{\text{argmin}} \sum_{(x_s, \widetilde{y}_q) \in A} |x_s - \widetilde{x}_q|,$$
$$\mathcal{A} := \{A = \{(x_j, \widetilde{y}_q), \dots, (x_{j+R}, \widetilde{y}_q)\} / A \subseteq \Omega\}.$$

See Figure 3.1 (b) for a graphical example.



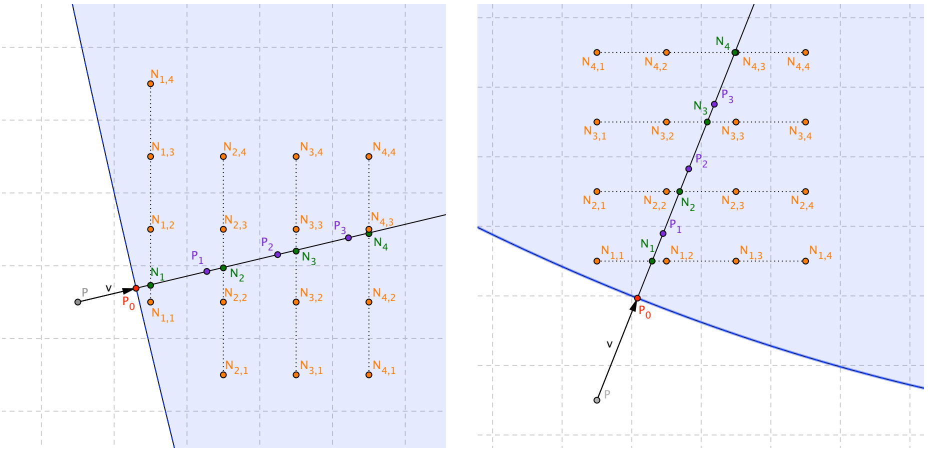
(a) (b)

Figure 3.1: Examples of choice of stencil: (a) $C_x = 1$, $N_{q,i} \in \mathcal{S}_q$; (b) $C_y = 2$, $N_{q,i} \in \mathcal{S}_q$

As will be expounded in Chapter 4, there are extrapolation methods which, due to efficiency reasons, should be used using stencils containing equally spaced nodes, mainly because of the computation of smoothness indicators, which can be very computationally expensive if the data is not equally spaced, as it will generally happen on Dirichlet boundaries. To overcome this issue, we perform an additional step before extrapolating the value at the ghost cell in order to generate a new stencil that includes the boundary node and is composed by equally spaced points.

Therefore, if Dirichlet conditions are prescribed, we use the data obtained in N_q, $1 \leq q \leq R+1$ to perform 1D interpolations at the points P_q, $1 \leq q \leq R$, where $P_q = (P_0^x + q h_x, P_0^y + q\frac{v_2}{v_1}h_x)$, $0 \leq q \leq R$ if $|v_1| \geq |v_2|$ or $P_q = (P_0^x + q\frac{v_1}{v_2}h_y, P_0^y + q h_y)$, $0 \leq q \leq R$, otherwise, and use the data from the stencil $\mathcal{S}(P) = \{P_0, P_1, \ldots, P_R\}$ to extrapolate it at the ghost cell P.

In case of outflow conditions, we extrapolate directly the data from the stencil $\mathcal{S}(P) = \{N_1, N_2, \ldots, N_{R+1}\}$. See Figure 3.2 for graphical examples.



(a) (b)

Figure 3.2: Examples of choice of stencil to perform extrapolations with stencils including equally spaced nodes. We use the stencil $\mathcal{S}(P) = \{N_1, N_2, N_3, N_4\}$ in case of outflow boundary and conditions and the stencil $\mathcal{S}(P) = \{P_0, P_1, P_2, P_3\}$ in case of Dirichlet boundary conditions.

If the boundary does not change with time the elements involved in the computation of the extrapolated value at P are determined only once at the beginning of the simulation.

The above procedure for the selection of the interpolation nodes at the normal lines and their corresponding sets $\mathcal{S}_q$ is performed only once at the beginning of the simulation as long as the boundary does not change. With an adequate use of this data structure, one can reconstruct data at order r (in case of smoothness) at the points $N_1, \ldots, N_R + 1$ on the normal line. Once these values are obtained, they are used to finally extrapolate to the given ghost cell (x_*, y_*).

The full extrapolation procedure is thus done in two stages in the case of outflow boundaries and in three in the case of Dirichlet boundaries (if one wants to use stencils with equally spaced nodes in all the extrapolation phases): in a first step, data located at the normal lines is computed from the numerical solution by (horizontal or vertical) 1D interpolation; in the second one, only performed on Dirichlet boundaries, the nodes obtained in the first step are used to interpolate at new points at the normal line so that they compose an equally spaced stencil together with the point $N(P)$; finally, values for the ghost cell are obtained by 1D extrapolation along the normal line obtained in the first stage (in case of outflow boundary) or the second stage (in case of Dirichlet boundary). Note that stencils with equally spaced notes are used in all the above approximations.

4

Extrapolation techniques for numerical boundary conditions

In order to have a fully high order accurate spatial scheme while taking into account the possible formation or eventual positioning of a discontinuity at the vicinity of the boundary, special care should be taken when filling the ghost cells through numerical boundary conditions, because interpolation/extrapolation can produce large errors if there is a discontinuity in the region determined by the interpolation nodes and the evaluation point. When implementing extrapolation at ghost cells, in order to avoid this considerable loss of precision or even a complete failure of the simulation, it is necessary to handle this situation carefully.

Some authors have approached this problem from different perspectives. In [**?**] a second order extrapolation procedure is developed for elliptic interface problems with Cartesian meshes. A second order procedure for the Poisson equation is developed in [**?**], and extended to fourth order for the Laplace and heat equation in [**?**]. Ghost-point-based methods on elliptic problems are developed in [**?**] for arbitrary interfaces on one dimensional problems and through multigrid methods in [**?**]. In [43] the authors develop a technique based on Lagrange interpolation with a limiter which is restricted to second order methods and a single ghost cell. Also related to our approach are the works of Shu and collaborators [45, 46] where the equation to be solved is used to extrapolate derivative values of the numerical solution to the boundary points where inflow conditions are prescribed and then approximate ghost values by a Taylor expansion. For outflow boundaries an extrapolation technique based on the WENO method is used, achieving high order when the data is smooth in both cases. The drawbacks of this approach are that it is problem-dependent (see [23, 52] for a similar methodology applied to other equations), that it requires a different treatment of different types of boundary and its relatively high computational cost.

In this work we introduce new techniques for the extrapolation of interior information to ghost cells (cells outside the domain, but within the stencils of interior points) making use of boundary conditions (if available) and interior data near a given ghost cell. This procedure is able to detect abrupt data changes. Our approach can be understood as an extension of [43] in the sense that it is based on Lagrange extrapolation with filters, both through a Boolean approach and in a WENO sense, but without imposing limitations on the order of the method or the number of ghost cells. Further, albeit the description is made for hyperbolic conservation laws, the procedure is agnostic about the equation and can be applied to other hyperbolic problems. Finally, the methodology is the same for inflow and outflow boundaries, just by considering the boundary node as an interpolation node in the case of inflow data.

Since in our procedure the interpolator is evaluated at a point which is not necessarily centered with respect to the interpolation nodes, we cannot directly use techniques based on the partition of the stencil in substencils and/or the weighting of these, such as it is done in ENO [17] or WENO schemes, because in this case not all the substencils are useful, this depending on the localization of the discontinuity and the evaluation point.

Consider for instance the function $u := \chi_{[1/2,+\infty)}$ and take the nodes $x_i := i$, $0 \leq i \leq 4$, with nodal values $u_i := u(x_i)$ and suppose we want to extrapolate this information at $x^* = -1$. Our nodal values are thus

$u_0 = 0$ and $u_i = 1$ for $1 \leq i \leq 4$. It is well-known that the ENO3 technique divides the global stencil of five points into three substencils, $S_m := \{x_m, x_{m+1}, x_{m+2}\}$, $0 \leq m \leq 2$, and chooses the one with maximal smoothness in terms of its divided differences, in this case, S_1 or S_2, both with all nodal values equal to 1 and thus all derivatives are zero. However, the result of this extrapolation at $x^* = -1$ is 1, which corresponds to the other state of the discontinuity from where x^* is located. The same applies for WENO.

Therefore, the interpolation strategy should be made more flexible, in order to choose certain nodes as valid according to some criterion and reject the rest. The strategy expounded in Chapter 3 lets us focus on a one-dimensional setting.

In Section 4.1 we introduce an extrapolation method that, starting from a wide stencil, selects the substencil that, in some sense, is the most adequate and then computes the extrapolation by means of ordinary Lagrange polynomials computed on that stencil [4].

In section 4.2 another technique for extrapolation, based on weights, is introduced. In some sense it represents an evolution of the previous one, as it also ponders the contributions of different stencils but now based on a weighted combination of their reconstructions that improves the method in Section 4.1 and depends on less parameters [5]. Several options for the design of the weights used in the method are given and analyzed.

4.1 Stencil selection by thresholding

Let r be the sought degree of the interpolating polynomial used for the extrapolation and assume to have information on a stencil of not necessarily equispaced nodes, $x_0 < \cdots < x_R$ $(R \geq r)$, with corresponding nodal values $u_i = u(x_i)$, and that we wish to interpolate at a certain node x^*. The procedure described in this section selects a substencil of size not bigger than $r+1$, contained in $\{x_0, \ldots, x_R\}$ where the data is smooth. The criterion for the selection of the nodes that will compose that substencil merges two complementary considerations: on the one hand, the comparison with a reference value which is the value of the function at the node closest to x^*, and on the other hand, smoothness information, obtained from smoothness indicators.

The key node on which we establish a proximity criterion on its corresponding nodal value is the interior node which is the closest to x^*, i.e., we choose the node x_{i_0}, $i_0 \in \{0,\dots,R\}$ such that:

$$i_0 = \operatorname*{argmin}_{0\leq i\leq R} |x_i - x^*|.$$

Now, the goal is to approximate the value that that node should have, based on the information of the “smoothest” substencil and the node x_{i_0}.

We consider all possible substencils of size $r+1$. There exist therefore $R-r+1$ possible substencils:

$$S_m = \{x_m,\dots,x_{m+r}\}, \quad 0 \leq m \leq R-r.$$

We denote by $p_m(x)$ the interpolator associated to the stencil S_m, $0 \leq m \leq R-r$. If sufficient smoothness at the whole stencil holds, then one has:

$$u(x_i) - p_m(x_i) \quad = \quad \mathcal{O}(h^{r+1}), \quad i = 0,\dots,R, \ \ , \tag{4.1}$$

therefore

$$u(x_i) = u(x_{i_0}) + (p_m(x_i) - p_m(x_{i_0})) + \mathcal{O}(h^{r+1}).$$

We select the substencil that solves:

$$m_0 := \operatorname*{argmin}_{0\leq m\leq R-r} \sum_{k=1}^{r} \int_{x_m}^{x_{m+r}} (x_{m+r} - x_m)^{2k-1} p_m^{(k)}(x)^2 dx, \tag{4.2}$$

and define

$$v_i := u_{i_0} + (p_{m_0}(x_i) - p_{m_0}(x_{i_0})). \tag{4.3}$$

From (4.1), we have that $v_i = u_i + \mathcal{O}(h^{r+1})$ if there is smoothness up to the r-th derivative of u. On the other hand, assuming that u is smooth on an open set that contains S_{m_0}, if there is a discontinuity within the whole stencil $\bigcup_{m=0}^{R-r} S_m$ and u_i is quite far from u_{i_0}, since by construction $v_i = u_{i_0} + \mathcal{O}(h)$, then it can be expected that v_i also be quite different from u_i.

In order for the smoothness assumption on S_{m_0} to make sense in a general setting, one needs $r+1 \leq \lceil \frac{R+1}{2} \rceil$, because all substencils would overlap in some common central nodes otherwise, leading to a situation where all substencils contain a discontinuity if it is contained in the overlapping region.

Therefore, one can conclude that the proximity of v_i with respect to u_i indicates the stencil smoothness that would entail including or not a node x_i in the stencil used for the final extrapolation.

Finally, let $\delta \in (0,1]$ be a *threshold* and define the set of indexes

$$I_\delta := \{i \in \{0,\dots,R\} : \quad \delta\left(|u_i - u_{i_0}| + D(x_i)\right) \leq |v_i - u_{i_0}| + D(x_i)\}, \tag{4.4}$$

where

$$D(x) := \sum_{j=1}^{r} \left|(x - x_{i_0})^j p_{m_0}^{(j)}(x_{i_0})\right|.$$

Notice that $I_\delta \neq \emptyset$, since $i_0 \in I_\delta$.

The term $D(x_i)$ is used in (4.4) to avoid an order loss at smoothness regions whenever $\exists j_0,\ 1 \leq j_0 \leq r : |u^{(j_0)}| \geq \mathcal{O}(h^{r+1})$ near x_{i_0}. When the first derivative is close to zero, despite both $|v_i - u_{i_0}|$ and $|u_i - u_{i_0}|$ are still $r+1$-th order close, its quotient may be far from 1, specially when one of the previous expressions is close to zero or even exactly zero (for instance, in zeros of even degree functions). The terms $D(x_i)$, alleviate this discrepancy by adding a $\mathcal{O}(h^{k_0}) \neq \mathcal{O}(h^{k_0+1})$ term, with $k_0 \leq r$ being the minimum index such that the k_0-th derivative does not have a zero around x_{i_0}. The definitive stencil is the largest stencil in I_δ containing i_0.

As last (optional) filter, if u^* is the value obtained from Lagrange interpolation from the resulting stencil, then the same threshold criterion can be applied to that value, resulting in the definitive extrapolation value:

$$u^*_{\text{def}} = \begin{cases} u^* & \text{if} \quad \delta'\left(|u^* - u_{i_0}| + D(x^*)\right) \leq |p_m(x^*) - p_m(x_{i_0})| + D(x^*) \\ u_{i_0} & \text{if} \quad \delta'\left(|u^* - u_{i_0}| + D(x^*)\right) > |p_m(x^*) - p_m(x_{i_0})| + D(x^*) \end{cases} \tag{4.5}$$

with $0 \leq \delta' \leq 1$.

This last criterion can be useful to detect wrong extrapolations (even when data are apparently smooth and previous criteria are met). Since it is an a posteriori criterion, we may generally use it with threshold values that are more permissive (i.e., much smaller than one) than those used for the node acceptance check. By construction, the closer the parameter δ is to one the lesser the tolerance to high gradients will be (with the consequent risk of eliminating some nodes from smooth regions). On the other hand, if δ is set to too low values, there may appear some oscillations or artifacts near discontinuities.

The quality of the smoothness criterion is enhanced with a larger substencil size (there is less risk of rejecting “correct” nodes). Furthermore a larger substencil can be also used to avoid a loss of precision order when consecutive derivatives are null at some point (precisely, until the $(r-1)$-th order derivative). Nevertheless, this would force increasing R to work with a wider initial stencil, i.e., obtain more data from the general problem in order to avoid the previously mentioned problem.

In summary, the extrapolation of the nodal data $\{(x_i, u_i)\}_{i=0}^{R}$ to the point x^* consists of the following steps:

1. Find i_0 such that x_{i_0} is the closest node to x^*.
2. Find the $(r+1)$-point stencil $\mathcal{S}_{m_0} = \{x_{m_0+j}\}_{j=0}^{r}$ with maximal smoothness. We use the smoothness indicators in (4.2) for this purpose.
3. For $i \in \{0, \dots, R\}$ compute candidate approximations v_i of u_i using (4.3).
4. Fix a value $0 < \delta \leq 1$ and compute the set of nodes I_δ according to (4.4).
5. Extract the substencil in I_δ composed by x_{i_0} and its r closest points. If I_δ contains less that $r+1$ points then extract the largest stencil in I_δ containing i_0.
6. Compute the extrapolated value u^* at x^* using the stencil in the previous step.
7. Optionally, fix $0 < \delta' \leq 1$ and replace u^* by u^*_{def} computed from (4.5).

Let us apply the previous steps to the toy example in page 46. We have $R = 4, r = 2$ in that example and we assume the values $\delta = \delta' = 0.5$, although any other choice of δ and δ' in the range $(0, 1)$ would give the same result. The stencil selection procedure is as follows:

1. The closest node to $x^* = -1$ is $x_0 = 0$, whose nodal value is $u_0 = 0$.
2. There are two stencils where the information is constant, S_1 and S_2 and therefore any of them would be selected in this step leading to the same result. Assume S_1 is chosen.
3. $v_i = u_0 = 0$, $0 \leq i \leq 4$, because $p_1 = 1$ for all $i \in \{0, \dots, 4\}$ and thus $D(x) = 0$.
4. The differences $|u_i - v_i|$ are all equal to 1 except for x_0 for which it is equal to 0. Therefore $I_\delta = \{x_0\}$ and the result of the extrapolation is $u^* = u_0 = 0$.
5. If the a posteriori filter is applied the result is kept as $\delta'|u^* - u_0| = 0 \leq 0 = |p_1(x^*) - p_1(x_1)|$.

More numerical experiments on this technique are presented in Section 4.3.

4.2 Weighted extrapolation

The method described in Section 4.1 tries to produce extrapolations at the ghost nodes that maintain the target order of accuracy even increasing the size of the global stencil whenever required.

We now present a new technique, which can be considered as an evolution of the thresholding method, based on the computation of dimensionless and scale independent weights that are used to combine different polynomial reconstructions in several ways according to the weight design. This method outperforms the method based on thresholds and depends on less parameters. The extrapolation keeps maximal order if the data is smooth and produces a lower order approximation otherwise.

Consider a stencil of equally spaced nodes $x_0 < \cdots < x_r$ and their corresponding nodal values $u_j = u(x_j)$. Denote $J = \{0, \dots, r\}$ and $X = \{x_j\}_{j\in J}$ and let x_* be the node where we wish to interpolate and j_0 the interior node which is closest to x_*, i.e.,

$$j_0 = \operatorname*{argmin}_{j\in J} |x_j - x_*|.$$

The goal is again to approximate the value that x_* should have, based on the information of the "smoothest" substencil and the node x_{j_0}. We define inductively the following set of indexes:

$$J_0 = \{j_0\}, \text{ and } X_0 = \{x_j\}_{j\in J_0} = \{x_{j_0}\}.$$

Assume we have defined $J_k = \{j_k, \dots, j_k + k\}$; then J_{k+1} is defined in an ENO fashion by

$$J_{k+1} = \begin{cases} \{j_k - 1\} \cup J_k & \text{if } j_k > 0 \wedge [u_{j_k-1}, \dots, u_{j_k+k}] \leq [u_{j_k}, \dots, u_{j_k+k+1}] \\ J_k \cup \{j_k + k + 1\} & \text{if } j_k < r - k \wedge [u_{j_k}, \dots, u_{j_k+k+1}] < [u_{j_k-1}, \dots, u_{j_k+k}] \end{cases}$$

and

$$X_{k+1} = \{x_j\}_{j\in J_{k+1}},$$

where $[v_1, \dots, v_\ell]$ represents the undivided difference of $v_1, \dots, v_\ell$. By construction, it is clear that the set X_k can be written as a sequence of nodes with successive indexes, i.e., stencils:

$$X_k = \{x_{i_k+j}\}_{j=0}^{k}$$

for some $0 \leq i_k \leq r - k$, $0 \leq k \leq r$.

Now, for each k, $0 \leq k \leq r$, we define p_k as the interpolating polynomial of degree at most k such that $p_k(x_{i_k+j}) = u_{i_k+j}$, $\forall j$, $0 \leq j \leq k$. Given a set of weights $\{\omega_k\}_{k=1}^r$ such that $0 \leq \omega_k \leq 1$, we define the following recurrence:

$$\begin{array}{rcl} u_*^{(0)} & = & p_0(x_*) = u_{i_0}, \\ u_*^{(k)} & = & (1-\omega_k)u_*^{(k-1)} + \omega_k p_k(x_*), \quad 1 \leq k \leq r. \end{array} \tag{4.6}$$

We define the final result of the weighted extrapolation as

$$u_* := u_*^{(r)},$$

which will be taken as an approximation for the value $u(x_*)$.

The idea is to increase the degree of the interpolating polynomial only if the solution in the corresponding stencil is smooth, and therefore the chosen weights should verify that $\omega_k \approx 0$ if the stencil J_k crosses a discontinuity and $\omega_k \approx 1$ if the data from the stencil is smooth. We will show below a weight construction that verifies that property as well as the capability of preserving the accuracy order of the extrapolation in case of smoothness.

From now on, we will assume that the nodes X are equally spaced and define $h = x_{j+1} - x_j$.

For each $1 \leq k \leq r$, we define a slight modification of the Jiang-Shu smoothness indicator [26] associated to the stencil J_k as the following value:

$$I_k = \frac{1}{r}\sum_{\ell=1}^{k}\int_{x_0}^{x_r} h^{2\ell-1} p_k^{(\ell)}(x)^2 dx.$$

Now, given $1 \leq r_0 \leq \lfloor \frac{r}{2} \rfloor$, we will seek for a smoothness zone along the stencils of $r_0 + 1$ points as a reference.

This procedure will work correctly if there is only one discontinuity in the stencil, and the restriction $r_0 \leq \lfloor \frac{r}{2} \rfloor$ is set in order to avoid a stencil overlapping, since a discontinuity might eventually be in the overlapping zone and thus none of the stencils would include smooth data. Define

$$IS_k = \min_{0 \leq j \leq r-k} \frac{1}{r}\sum_{\ell=1}^{r_0}\int_{x_0}^{x_r} h^{2\ell-1} q_{k,j}^{(\ell)}(x)^2 dx, \quad 1 \leq k \leq r_0,$$

where $q_{k,j}$ is the polynomial of degree at most k such that $q_{k,j}(x_{j+i}) = u_{j+i}$ for $0 \leq i \leq k$, $0 \leq j \leq r-k$.

There are many possibilities for defining the weights in (4.6). We next introduce several possibilities that might be suitable for different scenarios. We start with two designs that we name Simple Weights (SW) and

Improved Weights (IW) that show good performance for problems that have smooth solutions but might misbehave otherwise. Albeit the interest of such approaches is reduced to academic problems we describe them in Sections 4.2.1 and 4.2.2 with some detail as they are illustrative of the ideas behind other methods described in the rest of the section, which are much more oriented to more challenging problems that may include the typical non-smooth features of hyperbolic problems.

Numerical experimentation shows that the use of the SW and IW methods in complex problems can lead to poor results, probably because of the low numerical viscosity introduced by the methods at the boundary. For this reason, we introduce new weight designs which are derived from the two aforementioned extrapolation techniques. These are the unique weight (UW), where the extrapolation is performed by computing one weight (Section 4.2.4), a tuned version (λ-UW), where the tuning parameter λ can magnetize the weight to 0 or 1 in order to improve the quality of the extrapolation depending on the context we are working in (Section 4.2.4) and the global average weight (GAW), also based on a single weight, but more robust than the UW version, described in Section 4.2.5. Finally, since some stability issues may appear if the extrapolation is solely based on Lagrange extrapolation, we introduce a least-squares extrapolation procedure, which can be ultimately combined with any of the the weighted extrapolation techniques. We refer to this combination as Weighted Least Squares (WLS) so that WLS-X denotes the WLS technique combined with weights computed by the method X. We describe this approach in Section 4.2.6. Numerical experiments analyzing all the above possibilities are shown in Section 4.3.

4.2.1 Simple weights (SW)

The weights are defined as follows

$$\begin{aligned}
\omega_k &= 1 - \left(1 - \left(\frac{IS_k}{I_k}\right)^{s_1}\right)^{s_2}, \quad 1 \le k \le r_0, \\
\omega_k &= \min\left\{1 - \left(1 - \left(\frac{IS_{r_0}}{I_k}\right)^{s_1}\right)^{s_2}, 1\right\}, \quad r_0 + 1 \le k \le r.
\end{aligned} \tag{4.7}$$

A small positive number $\varepsilon > 0$ is added to each smoothness indicator in order to avoid the denominator to become zero (in all our experiments, we take $\varepsilon = 10^{-100}$). The parameter s_1 enforces the convergence to 0 when the stencil is not smooth, while the parameter s_2 enforces the convergence to 1 when it is smooth.

It can be shown that for a smooth stencil, if there exists some $1 \leq k_0 \leq r_0$ such that $|u^{(k_0)}| >> 0$ around the stencil, then

$$\omega_k = 1 - \mathcal{O}(h^{s_2})$$

and if the stencil crosses a discontinuity, then

$$\omega_k = \mathcal{O}(h^{2s_1}).$$

A drawback of this weight design, apart from the loss of accuracy when all the k-th derivatives $1 \leq k \leq r_0$ vanish near the stencil, is the fact that sometimes the optimal order cannot be attained regardless of the values of s_1 and s_2. Consider for instance $r = 5$ and $r_0 = 2$ and the function

$$u(x) = \left\{ \begin{array}{cc} x^2, & x \leq 0 \\ 1, & x > 0 \end{array} \right.$$

Now we take the nodes $x_i = -0.5 + 0.2i$, $0 \leq i \leq 5$, and the corresponding values u_i are $U = \{0.25, 0.09, 0.01, 1, 1, 1\}$. Since one of the substencils of three points contains $\{1, 1, 1\}$, it is clear then that $IS = 0$ and therefore $\omega_k = 0$, $\forall k$, $1 \leq k \leq 5$. If we performed a weighted extrapolation to $x_* = -0.7$ then it would be obtained the nodal value from the corresponding closest node, $x_0 = -0.5$, that is, $u_* = u_0 = 0.25$, while the most reasonable thing to do would be to perform a second order extrapolation taking the first three nodes (take all the nodes from the correct side of the discontinuity), which gives 0.49. The aforementioned scenario is depicted in Figure 4.1.

This has occurred because in this case the smoothest substencil belongs to the other side of the discontinuity, and thus the information about the derivatives is wrong. We next present an alternative weight design that overcomes the above issue.

4.2.2 Improved weights (IW)

The previously mentioned issue can be solved through the following modification of the weights design. In this new weight design we will define and combine additional parameters in order to ensure that, for stencils with discontinuities, the information is taken from the correct side of the

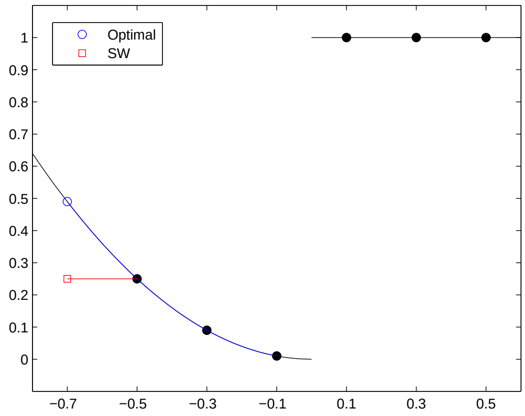


Figure 4.1: Comparison of the SW extrapolation results against the expected optimal result.

discontinuity. Let us now consider the maximum value of the up to r_0-th order smoothness indicators:

$$IM_k = \max_{0\leq j\leq r-k} \frac{1}{r}\sum_{\ell=1}^{r_0}\int_{x_0}^{x_r} h^{2\ell-1} q_{k,j}^{(\ell)}(x)^2 dx, \quad 1\leq k\leq r_0.$$

We define

$$\sigma_k := \min\left\{\frac{IS_{\min\{k,r_0\}}}{I_k}, 1\right\},$$

$$\tau_k := \frac{I_k}{IM_k},$$

and

$$\rho_k = \tau_k\left(\frac{1-\sigma_k}{\sigma_k}\right)^d, \quad d\geq \frac{r}{2},$$

to finally define our new weights as

$$\omega_k = \frac{1}{1+\rho_k}. \tag{4.8}$$

We now see in detail the reason for this choice. These are the possible values in terms of powers of $\mathcal{O}(h)$ that both quotients between smoothness indicators can take:

$$\sigma_k = \begin{cases} 1-\mathcal{O}(h) & \text{if there is smoothness,} \\ \mathcal{O}(h^2) & \text{if } X_k \text{ crosses a discontinuity.}\end{cases}$$

However, if there is not smoothness but still X_k does not cross a discontinuity there may be two possibilities: $IS_{\min\{k,r_0\}}$ is obtained in the "correct" side or in the "wrong" one. In the first case, we would have $\frac{IS_{\min\{k,r_0\}}}{I_k} = 1 + \mathcal{O}(h)$, as desired, but otherwise, if the derivatives are different from one side and another in a random fashion, that would lead to quotients with values in a random fashion as well. Here is thus the importance of the second quotient in order to fix such issue:

$$\tau_k = \begin{cases} 1 + \mathcal{O}(h) & \text{if smoothness,} \\ \mathcal{O}(h^2) & \text{if not smoothness and } X_k \text{ belongs to a smooth zone,} \\ \mathcal{O}(1) & \text{if not smoothness and } X_k \text{ crosses a discontinuity,} \end{cases}$$

where $\mathcal{O}(1)$ means that it is a value comprised between 0 and 1 in a random fashion, but not $\mathcal{O}(h)$ since in that case it is a quotient between two smoothness indicators, both in non-smooth zones. Note from that the term IM_k from the definition of τ_k can be replaced by I_r, since in this case τ will still verify the above properties and less smoothness indicators will be required to be computed. In practice, we will work through this modification.

One can show that this way

$$\rho_k = \begin{cases} \mathcal{O}(h^d) & \text{if } X_k \text{ belongs to a smooth zone,} \\ \mathcal{O}(h^{-2d}) & \text{if } X_k \text{ crosses a discontinuity,} \end{cases}$$

as desired, and thus

$$\omega_k = \begin{cases} 1 - \mathcal{O}(h^d) & \text{if } X_k \text{ belongs to a smooth zone,} \\ \mathcal{O}(h^{2d}) & \text{if } X_k \text{ crosses a discontinuity,} \end{cases}$$

except when all derivatives up to the r_0-th one are close to be zero in some of the two discontinuity sides, where the value of the weight cannot be predicted, although it will take values more likely close to 0 rather than 1.

We can fix this issue by redefining σ_k as

$$\sigma_k = \frac{IS_k + \beta}{I_k + \beta}, \tag{4.9}$$

where β is assumed to be a small quantity, $\beta = \mathcal{O}(h^b)$, $b \leq 2r_0$. We propose two different choices of β satisfying that condition:

- $\beta = \lambda^2 h^{2r_0}$, where λ is a parameter proportional to the scaling of the solution. That is, if we re-scale the data by a factor of μ, then the value λ should be replaced by $\mu\lambda$.

- In our context, we have knowledge about a wide range of point values of a function (the numerical data from a computational domain in a certain scheme). Hence, we can naturally define a quantity $\mathcal{O}(h^{2r_0})$ depending directly on the data of the problem.

  For instance, if we assume we have a 1D simulation with data U_j, $0 \leq j \leq n$, then one can define

  $$\beta = \left(\frac{1}{n} \sum_{j=1}^{n} |u_j - u_{j-1}| \right)^{2r_0} = TV(u)^{2r_0} h^{2r_0}.$$

  The above value can be generalized to any dimension as a global average of all the absolute values of all the directional undivided differences (in all directions).

  When discontinuities and zeros at the first derivative are supposed to be in a region of measure 0, the above value verifies $\beta = \mathcal{O}(h^{2r_0})$, while keeping σ_k scaling independent. This argument is also valid even when the first derivative is zero almost everywhere, but there is a discontinuity on the data, which is also a common case as initial condition for shock problems. Another valid case is when, despite having zeros in the derivative in a non-null region, there is a non-null region having non-zero derivatives as well provided that the discontinuities, if any, are located in a null region.

  If one wants to keep some stricter control of the above parameter due to the presence of very strong discontinuities which might make β too big, one can always consider a tuning parameter κ (in this case independent of the scaling as well) and redefine σ_k as

  $$\sigma_k = \frac{IS_k + \kappa\beta}{I_k + \kappa\beta}.$$

If one uses the above technique to avoid a loss of accuracy near zeros on the corresponding derivatives and makes sure that the parameters, if any, are well tuned, it makes no sense to use a smoothness control stencil longer than a two-points one, and thus for that case we will always use $r_0 = 1$. Moreover, in this particular case it is no longer needed to define σ_k such that does not surpass the unity, since we have the following result.

Proposition 1. *If* $r_0 = 1$ *then*

$$\alpha_k := \frac{I_1}{I_k}$$

verifies $0 \leq \alpha_k \leq 1$, $1 \leq k \leq r$.

Proof. Given $f, g \in L^2([x_{i-1}, x_i])$ we define the following scalar products

$$\langle f, g\rangle_i = \int_{x_{i-1}}^{x_i} f(x)g(x)dx$$

and their induced norms as

$$\|f\|_i^2 = \langle f, f\rangle_i, \quad 1 \leq i \leq r.$$

Now, taking into account that $x_i - x_{i-1} = h$, we have

$$\begin{aligned}
I_k &= \frac{1}{r}\sum_{\ell=1}^{k}\int_{x_0}^{x_r} h^{2\ell-1}p_k^{(\ell)}(x)^2dx \geq \frac{1}{r}\int_{x_0}^{x_r} hp_k'(x)^2dx = \frac{1}{r}\sum_{i=1}^{r} h\int_{x_{i-1}}^{x_i} p_k'(x)^2dx \\
&= \frac{1}{r}\sum_{i=1}^{r}\int_{x_{i-1}}^{x_i} dx \int_{x_{i-1}}^{x_i} p_k'(x)^2dx = \frac{1}{r}\sum_{i=1}^{r}\|f_i\|_i^2\|g_i\|_i^2,
\end{aligned}$$

where $f_i(x) = 1$ and $g_i(x) = p_k'(x)$ for $x \in [x_{i-1}, x_i]$. By the Cauchy-Schwarz Inequality

$$\langle f_i, g_i\rangle_i^2 \leq \|f_i\|_i^2\|g_i\|_i^2$$

we have

$$\begin{aligned}
I_k &\geq \frac{1}{r}\sum_{i=1}^{r}\|f_i\|_i^2\|g_i\|_i^2 \geq \frac{1}{r}\sum_{i=1}^{r}\langle f_i, g_i\rangle_i^2 = \frac{1}{r}\sum_{i=1}^{r}\left(\int_{x_{i-1}}^{x_i} f_i(x)g_i(x)dx\right)^2 \\
&= \frac{1}{r}\sum_{i=1}^{r}\left(\int_{x_{i-1}}^{x_i} p_k'(x)dx\right)^2 = \frac{1}{r}\sum_{i=1}^{r}(p_k(x_i) - p_k(x_{i-1}))^2 = \frac{1}{r}\sum_{i=1}^{r}(u_i - u_{i-1})^2 \\
&\geq \frac{1}{r}\sum_{i=1}^{r}\min_{1\leq j\leq r}(u_j - u_{j-1})^2 = \min_{1\leq j\leq r}(u_j - u_{j-1})^2 = I_1.
\end{aligned}$$

□

With these modifications, weighted extrapolation will not suffer from a loss of accuracy in any case and it will have the optimal order in presence of sharp discontinuities.

If we now apply this technique, it will still capture well sharp discontinuities while keeping the highest order as possible when there is a discontinuity in the stencil.

4.2.3 Examples

Example 1. Let $u : \mathbb{R} \to \mathbb{R}$ a function defined by

$$u(x) = \begin{cases} x^2 & \text{if} \quad x \leq 1, \\ 1 + x^3 & \text{if} \quad x > 1. \end{cases}$$

We study numerically the accuracy behavior of our scheme in a six point stencil around the discontinuity point $x = 1$ when $h \to 0$. Given $h > 0$ we select as stencil the set of nodes $x_i = 1 + (-2.5 + i)h$, $0 \leq i \leq 5$. Figure 4.2 shows a graphical example of the grid points for $h = 0.2$ and $h = 0.1$.

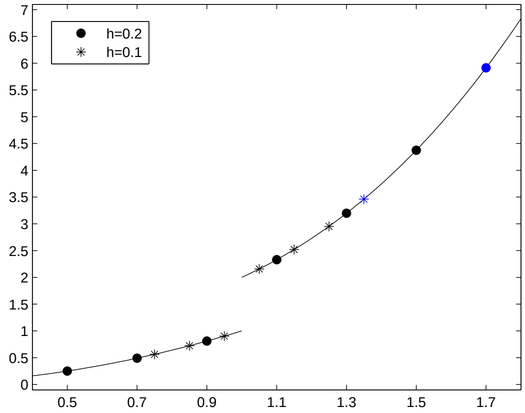


Figure 4.2: Illustration of the grid configuration in Example 1.

We start with $h_0 = 0.04$ and define $h_i = h_{i-1}/2$ for $1 \leq i \leq 6$ and compute the exact errors extrapolating at $x_* = 1 + 3.5h$ using the above techniques. A successful one should give third order accuracy since there are available three points from the right side of the discontinuity.

The results obtained with the simple weights (4.7) for $s_1 = s_2 = 3$ and $r_0 = 2$ are shown in Table 4.1.

It can be clearly seen that the optimal accuracy is not attained by this weight design.

The technical reason for this happening is that the derivative of u for $x \neq 1$ is

$$u'(x) = \begin{cases} 2x & \text{if} \quad x < 1, \\ 3x^2 & \text{if} \quad x > 1, \end{cases}$$

and thus $\lim_{x \to 1^-} u'(x) = 2$ and $\lim_{x \to 1^+} u'(x) = 3$, hence since the "smoothest" information is taken from the left side of the discontinuity but the

Resolution	Error	Order
h_0	5.72E−2	–
h_1	1.18E−2	1.48
h_2	7.40E−3	1.32
h_3	3.24E−3	1.19
h_4	1.51E−3	1.10
h_5	7.26E−4	1.06
h_6	3.56E−4	1.03

Table 4.1: Simple weights (4.7), example 1.

actual information should be taken from the right side of the discontinuity, where there is smoothness as well. Therefore, the weights ω_k, $1 \leq k \leq 2$, converge to $(1-(\frac{2}{3})^3)^3$ as $h \to 0$ rather than 1 for the above explained reason.

Now, we repeat the same test using the new weights defined in (4.8) for $d=3$ and $r_0=1$ and we present in Table 4.2 the errors for the same setup as above, where it can be clearly seen that the optimal third order accuracy is obtained.

Resolution	Error	Order
h_0	3.89E−4	–
h_1	4.82E−5	3.01
h_2	6.01E−6	3.00
h_3	7.50E−7	3.00
h_4	9.38E−8	3.00
h_5	1.17E−8	3.00
h_6	1.46E−9	3.00

Table 4.2: Improved weights (4.8), example 1.

Example 2. We now consider an example where one derivative vanishes. This example is very similar to the one presented to motivate the definition of the improved weights. In this case, we define $u:\mathbb{R}\to\mathbb{R}$ as

$$u(x)=\begin{cases} \sin(x) & \text{if} \quad x \leq 0, \\ 1 & \text{if} \quad x > 0. \end{cases}$$

We define now the grid points as $x_i=(-2.5+i)h$, $0 \leq i \leq 5$, whose configuration is depicted in Figure 4.3 for $h=0.2$ and $h=0.1$.

Taking the same values of h as in the above experiment, we now extrapolate at $x_*=-3.5h$ and we obtain the results in Table 4.3 using the

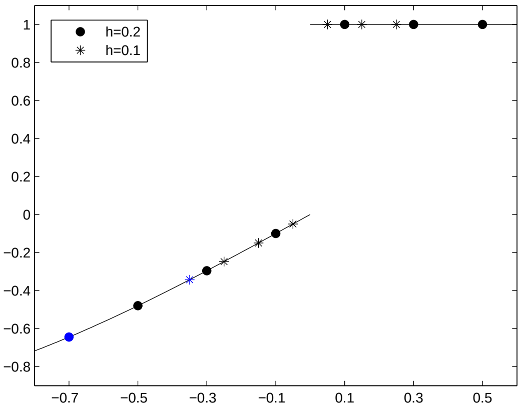


Figure 4.3: Illustration of the grid configuration in Example 2.

improved weights with the same parameters as above. This order decay is now due to the fact that one of the derivatives vanishes.

Resolution	Error	Order
h_0	3.97E−2	–
h_1	2.00E−2	0.99
h_2	1.00E−2	1.00
h_3	5.00E−3	1.00
h_4	2.50E−3	1.00
h_5	1.25E−3	1.00
h_6	6.25E−4	1.00

Table 4.3: Improved weights (4.8), example 2.

To fix this, we use the weight modification suggested in (4.9) by taking $\beta = h^2$, obtaining third order accuracy as can be observed in Table 4.4.

4.2.4 Unique weight extrapolation (UW)

In this section we propose a method that uses only one weight to decide the extrapolation method attending to the global smoothness in the stencil X that contains all $r+1$ nodes. The idea is that switching to a low order reconstruction as soon as a lack of smoothness is detected increases the robustness of the procedure in the non-smooth case when the extrapolation is performed in the context of PDEs, where disconti-

Resolution	Error	Order
h_0	6.38E−4	–
h_1	7.99E−5	3.00
h_2	1.00E−5	3.00
h_3	1.25E−6	3.00
h_4	1.56E−7	3.00
h_5	1.95E−8	3.00
h_6	2.44E−9	3.00

Table 4.4: Improved weights (4.8), $\beta = h^2$, example 2.

nuities get smeared, while maintaining high order in the smooth case, besides being more computationally efficient.

As our purpose, apart from achieving robustness, is also designing an efficient method, we will propose for the UW method a simplified version of the smoothness indicators introduced previously. We will also introduce in Section 4.2.4 an additional (and optional) tuning parameter, with which we can map the original weight ω to another one through a transformation that magnetizes values far from 1 (but still far from 0 as well) to 0. This variant is particularly useful in problems with strong shocks –as in this case the weights should be very close to 0– or in problems with complex smooth structure –where the results are better if the weights are close to 1 near them–.

The computation of the weights in the previous cases implies the use of logical structures since a minimum has to be computed. We overcome this issue as well by designing a weight capable of capturing well the discontinuities while keeping high order accuracy on smooth zones. This will be discussed in Section 4.2.5.

The procedure for the extrapolation with only one weight is performed in the following sense: Instead of gradually increasing the degree of the interpolating polynomials, we will just average the constant extrapolation ($k = 0$) and maximum degree extrapolation ($k = r$), that is, we will consider

$$u_* = (1-\omega)p_0(x_*) + \omega p_r(x_*) = (1-\omega)u_{i_0} + \omega p_r(x_*),$$

where

$$\omega = \min\left\{1 - \left(1 - \left(\frac{IS_{r_0}}{I_r}\right)^{s_1}\right)^{s_2}, 1\right\}.$$

It can be shown that for a smooth stencil, if there exists some $1 \leq k_0 \leq$

r_0 such that $|u^{(k_0)}| >> 0$ around the stencil, then

$$\omega = 1 - \mathcal{O}(h^{s_2})$$

and if the stencil crosses a discontinuity, then

$$\omega = \mathcal{O}(h^{2s_1}).$$

In order to lower the computational cost and ensure that $0 \leq \omega \leq 1$ without having to artificially bound it by 1 when $r_0 > 1$, we can replace the definition of I_r, which is a smoothness indicator of the whole $r+1$ points stencil, by the average of all smoothness indicators of the substencils of r_0+1 points, i.e.:

$$I_r^* := \frac{1}{r - r_0 + 1} \sum_{j=0}^{r-r_0} I_{r_0,j},$$

where

$$I_{r_0,j} = \frac{1}{r_0} \sum_{\ell=1}^{r_0} \int_{x_j}^{x_{r_0+j}} h^{2\ell-1} q_{r_0,j}^{(\ell)}(x)^2 dx. \tag{4.10}$$

Then one can define

$$\omega = 1 - \left(1 - \left(\frac{IS_{r_0}}{I_r^*}\right)^{s_1}\right)^{s_2},$$

which in this case it clearly verifies $0 \leq \omega \leq 1$.

Under the hypothesis $\exists k_0 \in \mathbb{N}$, $1 \leq k_0 \leq r_0$ such that $|u^{(k_0)}| >> 0$ around the stencil, then

$$u_* = u(x^*) + \mathcal{O}(h^{r'+1}),$$

where $r' = \min\{s_2(r_0 - k_0 + 1), r\}$.

Tuned weight (λ-UW)

When the stencil includes globally some smeared discontinuity, one can map a ω value such that $\omega >> 0$ and $\omega << 1$ to a new one $\widetilde{\omega} \approx 0$, as desired in this case, but that at same time $\widetilde{\omega} \approx 1$ when $\omega \approx 1$ as well.

Let $\widetilde{\omega} = F_\lambda(\omega)$ be

$$\widetilde{\omega} := \begin{cases} \frac{e^{\lambda\omega}-1}{e^{\lambda}-1} & \text{if} \quad \lambda \neq 0 \\ \omega & \text{if} \quad \lambda = 0 \end{cases},$$

for some $\lambda \in \mathbb{R}$. It can be proven that $\forall \omega \in [0,1]$, $G_\omega \in \mathcal{C}^\infty(\mathbb{R})$, where $G_\omega(\lambda) := F_\lambda(\omega)$. The larger λ is, the stricter the discontinuity detection filter will be. The lower (negative) λ is, the more permissive it will be.

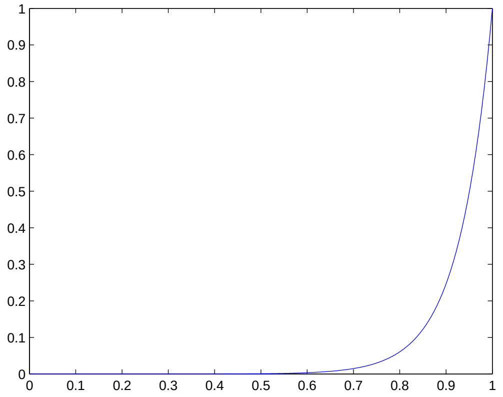


Figure 4.4: Plot of the mapping $\omega \to \widetilde{\omega}$ for $\lambda = 14$.

Assume smoothness conditions, therefore $\omega = 1 - \delta$, where $\delta = \mathcal{O}(h^{s_1})$. Assume $s_1 \geq r$ in order to achieve the maximum order accuracy. We have:

$$\begin{aligned} e^{\lambda\omega} - 1 &= e^{\lambda(1-\delta)} - 1 = e^{\lambda}e^{-\lambda\delta} - 1 = e^{\lambda}e^{-\lambda\delta} - 1 = e^{\lambda}e^{-\lambda\delta} - e^{\lambda} + e^{\lambda} - 1 \\ &= (e^{\lambda} - 1) + e^{\lambda}(e^{-\lambda\delta} - 1). \end{aligned}$$

Therefore

$$\widetilde{\omega} = \frac{(e^{\lambda} - 1) + e^{\lambda}(e^{-\lambda\delta} - 1)}{e^{\lambda} - 1} = 1 + \frac{e^{\lambda}}{e^{\lambda} - 1}(e^{-\lambda\delta} - 1).$$

On the other hand, using the Taylor expansion:

$$e^{-\lambda\delta} - 1 = \sum_{k=1}^{\infty} \frac{(-\lambda\delta)^k}{k!} = -\lambda\mathcal{O}(h^{s_1}).$$

Therefore

$$\widetilde{\omega} = 1 - \frac{e^{\lambda}}{e^{\lambda} - 1}\lambda\mathcal{O}(h^{s_1})$$

and the desired accuracy is attained provided that $0 << \lambda = \mathcal{O}(1)$.

Figure 4.4 shows a plot of the mapping $\omega \to \widetilde{\omega}$ for $\lambda = 14$.

An alternative expression to tune the weights without using exponential functions is, given $0 < \mu < 1$,

$$W_{\mu}(\omega) = \frac{\mu\omega}{\mu\omega + (1-\mu)(1-\omega)}, \quad \omega \in [0,1].$$

In this case, it holds $W_{\frac{1}{2}} = \mathrm{Id}_{[0,1]}$, $\lim_{\mu\to 0^+} W_{\mu}(\omega) = 0 \ \forall\omega \in [0,1)$ and $\lim_{\mu\to 1^-} W_{\mu}(\omega) = 1 \ \forall\omega \in (0,1]$.

4.2.5 Global average weight (GAW)

Using the weight defined through the smoothness indicator replacement in (4.2.4), a natural substitute of IS_{r_0} using every available r_0-th order smoothness indicator is their harmonic mean:

$$IS^*_{r_0} := \frac{1}{\frac{1}{r-r_0+1}\sum_{j=0}^{r-r_0}\frac{1}{I_{r_0,j}}}.$$

We can thus define the new weight ω as

$$\omega := (1-(1-\rho)^{s_1})^{s_2},$$

where

$$\rho := \frac{IS^*_{r_0}}{I^*_r} = \frac{\frac{1}{\frac{1}{r-r_0+1}\sum_{j=0}^{r-r_0}\frac{1}{I^m_{r_0,j}}}}{\frac{1}{r-r_0+1}\sum_{j=0}^{r-r_0}I^m_{r_0,j}} = \frac{(r-r_0+1)^2}{\left(\sum_{j=0}^{r-r_0}I^m_{r_0,j}\right)\left(\sum_{j=0}^{r-r_0}\frac{1}{I^m_{r_0,j}}\right)},$$

and m is a parameter that enforces the convergence to 0 of the weight in a discontinuity. We next show that $0\leq\rho\leq 1$ (and therefore ω verifies this property as well) as well as the desired properties both in smooth and non-smooth cases.

Proposition 2. $0\leq\rho\leq 1$ *and verifies*

$$\rho = \begin{cases} 1-\mathcal{O}(h^{2c_s}) & \text{if the stencil is } C^{r_0} \text{ with a } s\text{-th order zero derivative,} \\ \mathcal{O}(h^{2ms}) & \text{if the stencil contains a discontinuity,} \end{cases}$$

with $c_s := \max\{r_0-s,0\}$. *Therefore*

$$\omega = \begin{cases} 1-\mathcal{O}(h^{2s_1c_s}) & \text{if the stencil is } C^{r_0} \text{ with a } s\text{-th order zero derivative,} \\ \mathcal{O}(h^{2s_2ms}) & \text{if the stencil contains a discontinuity.} \end{cases}$$

Proof. Let $a_j>0$, $1\leq j\leq k$. We show that the quotient between their harmonic mean and their mean is bounded by 1, that is

$$0\leq\rho=\frac{\frac{1}{\frac{1}{k}\sum_{j=1}^{k}\frac{1}{a_j}}}{\frac{1}{k}\sum_{j=1}^{k}a_j}\leq 1.$$

Since

$$\rho=\frac{\frac{1}{\frac{1}{k}\sum_{j=1}^{k}\frac{1}{a_j}}}{\frac{1}{k}\sum_{j=1}^{k}a_j}=\frac{k^2}{(\sum_{j=1}^{k}\frac{1}{a_j})(\sum_{j=1}^{k}a_j)}$$

it suffices to show that

$$A := \left(\sum_{j=1}^{k} \frac{1}{a_j}\right) \left(\sum_{j=1}^{k} a_j\right) \geq k^2.$$

Indeed,

$$\begin{aligned} A &= \sum_{i=1}^{k} \sum_{j=1}^{k} \frac{a_i}{a_j} = k + \sum_{i=1}^{k} \sum_{j=1}^{i-1} \left(\frac{a_i}{a_j} + \frac{a_j}{a_i}\right) = k + \sum_{i=1}^{k} \sum_{j=1}^{i-1} \frac{(a_i - a_j)^2 + 2a_i a_j}{a_i a_j} \\ &= k + 2\sum_{i=1}^{k}(i-1) + \sum_{i=1}^{k} \sum_{j=1}^{i-1} \frac{(a_i - a_j)^2}{a_i a_j} = k^2 + \sum_{i=1}^{k} \sum_{j=1}^{i-1} \frac{(a_i - a_j)^2}{a_i a_j} \geq k^2. \end{aligned}$$

Let us now assume that the stencil is C^{r_0} smooth with a s-th zero in the derivative, then

$$I_{r_0,j} = h^{2s}(1 + \mathcal{O}(h^{\max\{r-s,0\}})) = h^{2s}C(1 + \mathcal{O}(h^{c_s})).$$

Since we have actually proven that

$$A = k^2 + \sum_{i=1}^{k} \sum_{j=1}^{i-1} \frac{(a_i - a_j)^2}{a_i a_j},$$

replacing the a terms with the corresponding smoothness indicators to the power of m in case of smoothness

$$I_{r_0,j}^m = h^{2ms}C^m(1 + \mathcal{O}(h^{c_s}))^m = h^{2ms}C^m(1 + \mathcal{O}(h^{c_s})),$$

we have, denoting $k := r - r_0 + 1$,

$$\begin{aligned} A &= k^2 + \sum_{i=1}^{k} \sum_{j=1}^{i-1} \frac{(h^{2ms}C^m(1 + \mathcal{O}(h^{c_s})) - h^{2ms}C^m(1 + \mathcal{O}(h^{c_s})))^2}{h^{4ms}C^{2m}(1 + \mathcal{O}(h^{c_s}))^2} \\ &= k^2 + \sum_{i=1}^{k} \sum_{j=1}^{i-1} \frac{h^{4ms}C^{2m}(\mathcal{O}(h^{c_s}))^2}{h^{4ms}C^{2m}(1 + \mathcal{O}(h^{c_s}))} = k^2 + \sum_{i=1}^{k} \sum_{j=1}^{i-1} \frac{\mathcal{O}(h^{2c_s})}{1 + \mathcal{O}(h^{c_s})} \\ &= k^2 + \mathcal{O}(h^{2c_s}). \end{aligned}$$

Therefore,

$$\rho = \frac{k^2}{A} = \frac{k^2}{k^2 + \mathcal{O}(h^{2c_s})} = 1 + \mathcal{O}(h^{2c_s}).$$

Finally, if there is smoothness at least in one substencil with corresponding smoothness indicator $I_{r_0,i_0} = h^{2s}C(1+\mathcal{O}(h^{c_s})) = \mathcal{O}(h^{2s})$ and a discontinuity crosses the stencil then there will be another substencil such that its smoothness indicator verifies $I_{r_0,j_0} = \mathcal{O}(1)$, then the corresponding term in the above sum (swapping the indexes if necessary, that is, if $i_0 < j_0$) satisfies:

$$\frac{(I^m_{r_0,i_0} - I^m_{r_0,j_0})^2}{I^m_{r_0,i_0} I^m_{r_0,j_0}} = \frac{(\mathcal{O}(h^{2ms}) - \mathcal{O}(1))^2}{\mathcal{O}(h^{2ms})\mathcal{O}(1)} = \frac{\mathcal{O}(1)}{\mathcal{O}(h^{2ms})} = \mathcal{O}(h^{-2ms}),$$

and thus

$$A = \mathcal{O}(h^{-2ms}).$$

Therefore,

$$\rho = \frac{k^2}{A} = \frac{k^2}{\mathcal{O}(h^{-2ms})} = \mathcal{O}(h^{2ms}).$$

□

It follows from the result that to obtain an order of accuracy as large as possible, and if $r = 2r_0$, as it will be our usual choice of r_0 from now on, we must take $s_1 \geq r_0$ (to prevent from a possible extreme case $s = r_0 - 1$). Note that in the worst case scenario, $s \geq r_0$, implies an unavoidable downgrade of the accuracy as it happens with the original WENO-JS weights for reconstructions. The only way to overcome this is issue is to add to each smoothness indicator the β parameter defined above, but it will not be done in the forthcoming tests as we do not do it either in the definition of the WENO-JS weights in our numerical solver.

4.2.6 Weighted least squares extrapolation (WLS)

After some 2D order accuracy tests for smooth solutions with different spacing setups, it has been noticed that depending on the complexity of the domain, not only high order may not be achieved using straight Lagrange extrapolation at the ghost cells, but also the scheme might turn mildly unstable in some extreme cases, this independently of the choice for the spacing of the normal line points to be extrapolated at the ghost cell.

To avoid this phenomena, it is necessary to find an alternative to Lagrange (and weighted) extrapolation that stabilizes the scheme while keeping high order accuracy in terms of the global error. A possibility is to use least squares fitting as described in [45], [46].

Let $R \geq r$ and let $\{(x_i, u_i)\}_{i=0}^{R}$ be the stencil with nodal data, where $x_{i+1} - x_i = h$. Let us assume that we want to extrapolate or interpolate at the point x_* with a certain r-th degree polynomial

$$p(x) = \sum_{j=0}^{r} a_j x^j$$

using the data from the whole stencil. Since in general there is no polynomial of r passing through the R points, we find the polynomial $p(x)$ of degree r which minimizes the error with respect to $\{(x_i, u_i)\}_{i=0}^{R}$ in the L^2-norm, i.e., we solve $p(x_i) = u_i$, $0 \leq i \leq R$ by least squares, which will be still a $(r+1)$-th order accurate approximation if the data is smooth.

We can now combine the least squares extrapolation, conceived for smooth regions with the already described techniques techniques based on weights. Let z_* be the result of the least squares extrapolation and v_* the result of some chosen modality of the weighted extrapolation techniques. To this aim we define a weight ω by using the information of the whole stencil of $R+1$ points through the λ-UW or GAW technique and then take the final result of the extrapolation as

$$u_* := \omega z_* + (1-\omega) v_*. \tag{4.11}$$

The simplest and most robust case for the election of v_* is to take $v_* = u_{i_0}$. This is ultimately the technique that will be used both for smooth and non-smooth problems in our numerical experiments, so that we will indicate by WLS-X the weighted least squares method with ω in (4.11) computed through the method X.

We next show a step-by-step algorithm of the chosen WLS-GAW technique, for a stencil $\{x_i\}_{i=0}^{R}$ with nodal values $\{u_i\}_{i=0}^{R}$, to be extrapolated at the point x_*, is thus:

1. Find the index $0 \leq i_0 \leq R$ such that x_{i_0} is the closest point to x_*. Then u_{i_0} is the reference value.

2. Compute the least-square polynomial of degree r at x_*, $p(x_*)$, using the whole stencil data.

3. Obtain the corresponding smoothness indicators from the substencils of size r_0.

4. Compute the global average weight, w from the previously computed smoothness indicators.

5. The final extrapolated value, u_*, is then obtained as a weighted average of the least-square polynomial and the reference value, namely:

$$u_* = \omega p(x_*) + (1-\omega)u_{i_0}.$$

We next summarize all parameters related to the size of the various stencils involved in the extrapolation:

- $r+1$: Stencil size for Lagrange extrapolation. The accuracy order is thus $r+1$ in case of smoothness.
- r_0+1: Substencil size used in the computation of the smoothness indicators.
- $R+1$: Stencil size for least-squares extrapolation. In this context, we refer to r as the degree of the computed polynomial (hence the accuracy order is $r+1$ as well).

Table 4.5 shows all the parameters involved in the different weight designs and indications on whether a parameter is involved or not on a particular method.

Method (columns) / Parameter (rows)	SW	IW	UW	GAW
s_1: $\omega = \mathcal{O}(h^{\xi s_1})$ if discontinuity	Y	N	Y	Y
s_2: $\omega = 1 - \mathcal{O}(h^{\xi s_2})$ if smoothness	Y	N	Y	Y
d: $\omega = \mathcal{O}(h^{\xi d})$ if discontinuity $\omega = 1 - \mathcal{O}(h^{\xi d})$ if smoothness	N	Y	N	N
λ: $\omega \to 0$ if $\lambda \to +\infty$ $\omega \to 1$ if $\lambda \to -\infty$	N	N	O	N
m: $\omega = \mathcal{O}(h^{\xi m})$ if discontinuity	N	N	N	Y

Table 4.5: List of parameters for the weights (Y: yes, N: no, O: optional). In each case, ξ is a parameter that depends on the method, the substencils size r_0, the number of consecutive zero-derivatives and other parameters.

In the experiments shown in Section 4.3, the parameters are set to $R = 8$ (a stencil of 9 nodes to perform a least squares extrapolation), $r = 4$ (degree of the least squares interpolating polynomial), $r_0 = 2$ (substencils of size 3 where smoothness indicators are computed), $s_1 = r_0 = 2$ (requirement to match the whole scheme accuracy at the boundary provided that the first and second derivatives do not vanish simultaneously), $s_2 = 1$ and $m = 2$ (in order to mimic the exponent choice at the WENO-JS weights computation for spatial biased reconstructions, although in the boundary case it does not affect the order of the extrapolation when a discontinuity crosses the stencil since the alternative choice is just the constant extrapolation).

4.3 Numerical experiments

4.3.1 One-dimensional experiments

In this section we present some one-dimensional numerical experiments where both the accuracy of the extrapolation method for smooth solutions and its behavior in presence of discontinuities will be tested and analyzed.

Let us remark that for one-dimensional tests it is not necessary to develop a procedure as in the two-dimensional case described in Chapter 3, because one can set up initially a proper spacing between the nodes and perform a straight extrapolation at the ghost cells without having stability issues due to the presence of small-cut cells. However, to present accuracy and stability analysis in an easier setup, we perform the one-dimensional extrapolation, which directly corresponds to the two-dimensional extrapolation procedure that is proposed in this paper. This approach will illustrate that the accuracy order will still be the expected one in the smooth case and that the extrapolation method shows good performance in the non-smooth case.

Linear advection, $\mathcal{C}^\infty$ solution.

We start with a simple one-dimensional test case that will be used to illustrate the performance of the proposed method and also to analyze the importance and relative influence of some elements of the algorithm along the four examples detailed below. The problem statement for this test is the same as in [45]. We consider the linear advection equation

$$u_t + u_x = 0, \quad \Omega := (-1, 1),$$

with initial condition given by $u(x, 0) = 0.25 + 0.5\sin(\pi x)$ and boundary condition $u(-1, t) = 0.25 - 0.5\sin(\pi(1 + t))$, $t \geq 0$. We apply a numerical outflow condition at $x = 1$, where Dirichlet boundary conditions cannot be imposed due to the direction of propagation of the information.

It is immediately checked that the unique (smooth) solution to this problem is

$$u(x,t) = 0.25 + 0.5\sin(\pi(x-t)).$$

Example 1. In order to numerically test the order of accuracy we perform tests at resolutions given by $n = 20 \cdot 2^j$ points, $j = 1, \ldots, 5$. The cell centers are $x_j := -1 + (j + \frac{1}{2})h$, with $h := \frac{2}{n}$. We recall that the set of all cell centers which are interior to Ω is

$$\mathcal{D} := \{x_j : \ j \in \{0, \ldots, n-1\}\}.$$

Since we use WENO5 reconstruction, we require 3 extra cells at each side of the boundary, where extrapolation from the interior will take place.

- $x = -1$: x_j, $-3 \leq j \leq -1$.
- $x = 1$: x_j, $n \leq j \leq n+2$.

Given that the ODE solver is third order accurate, in order to attain fifth order accuracy in the overall scheme, we need to select a time step given by $\Delta t = \left(\frac{2}{n}\right)^{\frac{5}{3}}$, with corresponding Courant numbers $\Delta t/h = (2/n)^{2/3} \leq 1/20^{2/3}$.

Since the left boundary conditions are time dependent, we also have to take into account that a specific approximation is needed in each of the 3 stages in each RK3-TVD time step. In general, if the inflow condition is given by some function $g(t)$ which is at least twice continuously differentiable, we have to use the following values at the boundary to preserve third order accuracy [7]:

- First stage: $g(t_k)$.
- Second stage: $g(t_k) + \Delta t g'(t_k)$.
- Third stage: $g(t_k) + \frac{1}{2}\Delta t g'(t_k) + \frac{1}{4}\Delta t^2 g''(t_k)$.

Taking into account all the previous considerations, we execute the simulation until $t = 1$ for all the previously specified resolutions and we study the errors in the 1 and ∞ norms, together with the order deduced from them. We consider different modalities of boundary extrapolation: Constant extrapolation using only the closest node value (Table 4.6), five points stencil Lagrange extrapolation without discontinuity filters (Table 4.7) and with filters by thresholding described in Section 4.1 for different choices of the thresholds (Tables 4.8–4.10).

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
40	2.07E-3	–	3.87E-2	–
80	5.32E-4	1.96	1.96E-2	0.98
160	1.34E-4	1.99	9.81E-3	1.00
320	3.38E-5	1.99	4.91E-3	1.00
640	8.48E-6	1.99	2.45E-3	1.00

Table 4.6: Example 1: constant extrapolation (first order).

The Table 4.6 illustrates that a low order extrapolation affects the order of the global scheme. We can see that in this case is downgraded to second order in $\|\cdot\|_1$, while it is first order in $\|\cdot\|_\infty$.

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
40	8.73E-6	–	2.44E-5	–
80	2.70E-7	5.01	7.35E-7	5.05
160	8.45E-9	5.00	2.31E-8	4.99
320	2.64E-10	5.00	6.95E-10	5.06
640	8.26E-12	5.00	2.13E-11	5.03

Table 4.7: Example 1: Lagrange extrapolation (without filter).

From Table 4.8 on, we add the last column with the percentage of extrapolations for which no rejection, either in the 5 nodes or in the final result in the a posteriori criterion, has taken place along the complete simulation.

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$	% Success
40	5.45E-5	–	3.81E-4	–	86.18 %
80	3.06E-6	4.15	3.65E-5	3.38	95.77 %
160	1.34E-8	7.83	2.10E-7	7.44	99.55 %
320	2.64E-10	5.67	6.95E-10	8.93	100.00 %
640	8.26E-12	5.00	2.13E-11	5.03	100.00 %

Table 4.8: Example 1: thresholding, $\delta = \delta' = 0.99$.

From the results in those tables one can conclude that the thresholding detection behavior improves with increasing resolution. The technical reason for this is that the quotient between the quantities appearing in (4.4) satisfies

$$\lim_{h\to 0} \frac{|u_i - u_{i_0}| + D(x_i)}{|v_i - u_{i_0}| + D(x_i)} = 1.$$

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$	% Success
40	1.95E-5	–	1.38E-4	–	98.75 %
80	2.70E-7	6.17	7.35E-7	7.55	100.00 %
160	8.45E-9	5.00	2.31E-8	4.99	100.00 %
320	2.64E-10	5.00	6.95E-10	5.06	100.00 %
640	8.26E-12	5.00	2.13E-11	5.03	100.00 %

Table 4.9: Example 1: thresholding, $\delta = 0.9$, $\delta' = 0.75$.

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$	% Success
40	8.73E-6	–	2.44E-5	–	100.00 %
80	2.70E-7	5.01	7.35E-7	5.05	100.00 %
160	8.45E-9	5.00	2.31E-8	4.99	100.00 %
320	2.64E-10	5.00	6.95E-10	5.06	100.00 %
640	8.26E-12	5.00	2.13E-11	5.03	100.00 %

Table 4.10: Example 1: thresholding, $\delta = 0.75$, $\delta' = 0.5$.

Even at low resolutions, we observe that it is sufficient to use a relatively restrictive threshold for not rejecting any point in the extrapolations procedure at each time step.

Example 2. We now perform a test omitting the $D(x_i)$ terms, which, as stated in the previous section, help avoiding erratic node eliminations when the differences are very close to be 0. The results can be seen at Table 4.11, illustrating the importance of such terms.

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$	% Success
40	2.60E-5	–	1.72E-4	–	99.73 %
80	3.25E-7	6.32	1.60E-6	6.75	99.92 %
160	8.45E-9	5.27	2.31E-8	6.11	99.97 %
320	2.64E-10	5.00	6.95E-10	5.06	99.99 %
640	8.26E-12	5.00	2.13E-11	5.03	99.99 %

Table 4.11: Example 2: filter without $D(x_i)$ terms, $\delta = 0.2$, $\delta' = 0.1$.

We see that, indeed, without the $D(x_i)$ terms there are always some nodes removed even using very low threshold values.

Example 3. In order to illustrate the behavior of our method in presence of small-cut cells, we now perform a test changing the location of the nodes by $x_j = -1 + \left(j + \frac{1}{8}\right) h$. For instance, to extrapolate data to $x_* := x_{-1} = -1 - \frac{7}{8}h$, one first computes $v = N(x_*) - x_* = -1 - x_{-1} = \frac{7}{8}h$,

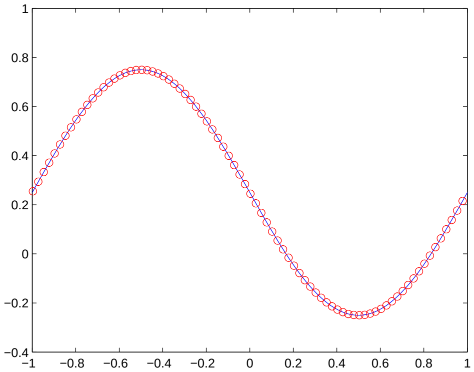


Figure 4.5: Example 3: stability.

$C_x = \lceil v/h \rceil = 1$ and considers points $N'_q = x_* + qC_xh = x_{-1} + qh = x_{q-1}$. Since there is a boundary condition at $N'_0 = N(x_*) = -1$ and $|N'_0 - N'_1| = \frac{h}{8} < |v| = \frac{7h}{8}$, then $N'_1 = x_0$ is not considered for extrapolation and the selected five nodes are $\{-1, N'_2, \ldots, N'_5\} = \{-1, x_1, \ldots, x_4\}$

The results obtained for $\delta = 0.75$, $\delta' = 0.35$ are shown in Table 4.12. No node rejection occurred in this experiment.

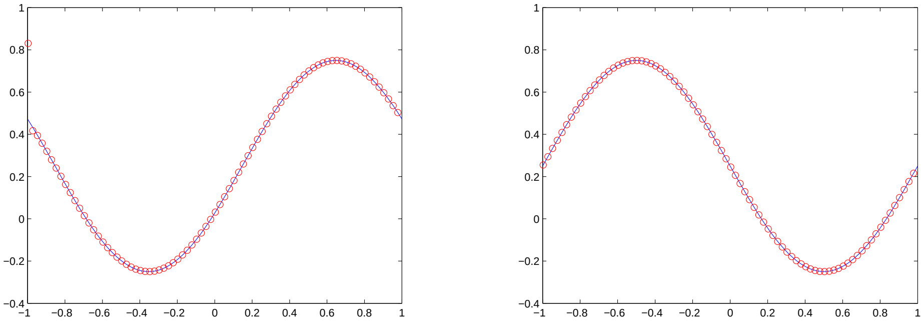
n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
40	9.81E−6	–	2.39E−5	–
80	3.06E−7	5.00	7.56E−7	4.98
160	9.52E−9	5.00	2.28E−8	5.05
320	2.97E−10	5.00	7.03E−10	5.02
640	9.23E−12	5.01	2.12E−11	5.06

Table 4.12: Example 3: Lagrange extrapolation (node removal).

Note that in our numerical scheme and for accuracy reasons we have used $\Delta t = h^{\frac{5}{3}}$ and, therefore, no stability issue should appear anyway for big enough n. Forgetting about matching the spatial accuracy order with the time accuracy order, we set $n = 80$, $\Delta t = 0.9h$, that is, a CFL value of 0.9, and see that our scheme is indeed stable and obtains good results as can be seen in Figure 4.5.

Example 4. To complete the previous examples we now analyze what happens if we attempt to extrapolate directly information at ghost cells without the removal of nodes too close to the boundary, i.e., for $x_* = x_{-1}$ the stencil would be $\{-1, x_0, \ldots, x_3\}$. For this experiment, we use the grid from Example 3, a Courant number of 0.9, i.e., $\Delta t = 0.9h$, and a five nodes extrapolation at both sides of the boundary as done in the previous experiments. The crucial difference with respect to Example 3 is that now we do not remove N_1', thus resulting in a stability problem clearly visible already at the early stages of the simulation shown in Figure 4.6 (a), which ultimately lead to failure by numeric overflow.

In order to illustrate that it is actually a CFL issue, we now repeat the simulation with a Courant number set again to 0.9 but based on the distance of the closest node of the inflow boundary to this last one (based on a spacing of $\frac{h}{8}$), i.e., $\Delta t = 0.9\frac{h}{8}$. In Figure 4.6 (b) it can be seen that now the scheme is stable. We conclude that the intermediate step consisting in extrapolating the information on nodes with adequate spacing is necessary in order to avoid unnecessarily severe time step restrictions.



(a)

(b)

Figure 4.6: Example 4: (a) $\Delta t = 0.9h$, $t = 0.147$. Oscillations appear; (b) $\Delta t = 0.9\frac{h}{8}$, $t = 1$. No oscillations

We repeat the test comparing some weighted modalities described in Section 4.2: IW, WLS-UW and WLS-GAW. We show the errors corresponding to $1-$ and $\infty-$norm, and the numerical order computed from them in Tables 4.13–4.15.

From the results, and comparing with those obtained with the thresholding technique, it can be seen that IW behaves essentially as Lagrange extrapolation even for low resolutions, as it happens with thresholding extrapolation with not excessively restrictive parameter for the detection of discontinuities.

On the other hand, the two remaining techniques involving least squares extrapolation produce slightly less accurate results, but still fifth

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
40	8.73E-6	$-$	2.44E-5	$-$
80	2.70E-7	5.01	7.35E-7	5.05
160	8.45E-9	5.00	2.31E-8	4.99
320	2.64E-10	5.00	6.95E-10	5.06
640	8.26E-12	5.00	2.13E-11	5.03

Table 4.13: Error table for linear advection problem, $t = 1$, IW.

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
40	4.28E-5	$-$	1.99E-4	$-$
80	5.32E-7	6.33	1.86E-6	6.74
160	1.38E-8	5.26	4.65E-8	5.32
320	4.16E-10	5.06	1.43E-9	5.02
640	1.27E-11	5.03	4.49E-11	5.00

Table 4.14: Error table for linear advection problem, $t = 1$, WLS-UW.

order accurate. This is what should be expected since a wider stencil is used, involving a polynomial of the same degree than IW. Albeit this fact, we will see in further experiments that the WLS-X techniques are more robust than the ones based on Lagrange extrapolation for more demanding problems.

Linear advection, discontinuous solution.

We illustrate with this experiment the behavior of the scheme based on thresholds when discontinuities are present and the entailed improvement with respect to using Lagrange extrapolation with no filters. We consider the same meshing and data as in Example 1 for the previous

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
40	1.50E-5	$-$	4.29E-5	$-$
80	4.56E-7	5.04	1.44E-6	4.90
160	1.37E-8	5.06	4.57E-8	4.98
320	4.16E-10	5.04	1.43E-9	5.00
640	1.27E-11	5.03	4.49E-11	4.99

Table 4.15: Error table for linear advection problem, $t = 1$, WLS-GAW.

problem, but now the boundary condition is defined by:

$$u(-1,t) = g(t) = \begin{cases} 0.25 & \text{if} \quad t \leq 1 \\ -1 & \text{if} \quad t > 1 \end{cases}$$

With this definition, the unique (weak) solution to this problem has a moving discontinuity and is given by:

$$u(x,t) = \begin{cases} -1 & \text{if} \quad x < t-2 \\ 0.25 & \text{if} \quad t-2 \leq x \leq t-1 \\ 0.25 + 0.5\sin(\pi(x-t)) & \text{if} \quad x \geq t-1 \end{cases}$$

In Figure 4.7 we check the graphical results that correspond to the simulation until $t = 1.5$, first using Lagrange extrapolation with no filters and afterwards with a filter with $\delta = 0.75$ and $\delta' = 0.5$, the same values that have achieved no node rejections in the first test. As it can be seen in Figure 4.7, Lagrange extrapolation without filters leads to spurious oscillations around the left side of the discontinuity, while thresholding removes them.

Burgers equation.

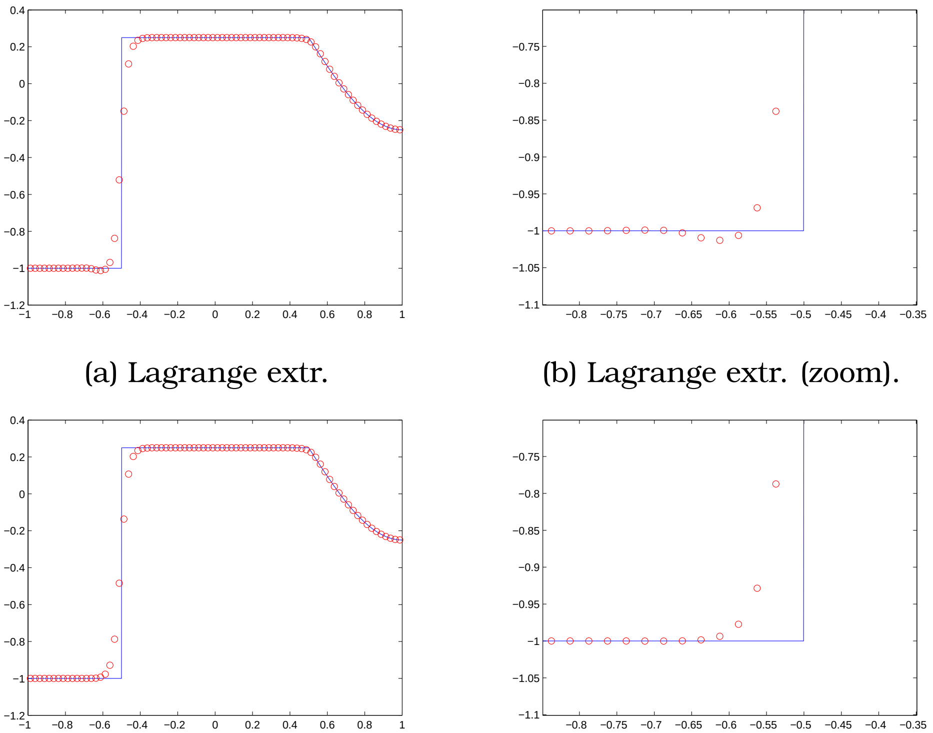
Let us now perform some tests using Burgers equation

$$u_t + \left(\frac{u^2}{2}\right)_x = 0, \quad \Omega = (-1,1),$$

with initial condition $u(x,0) = 0.25 + 0.5\sin(\pi x)$, outflow condition at the right boundary and a left inflow boundary conditions given by $u(-1,t) = g(t)$, where $g(t) = w(-1,t)$, with w the exact solution of the problem using periodic boundary conditions.

For $t = 0.3$ the solution is smooth and we get the following error table for $n = 40 \cdot 2^k$, $0 \leq k \leq 5$, and the same spacing as the first test, using, on the one hand, threshold values of $\delta = 0.75$ and $\delta' = 0.5$, where no node rejection occurs at any resolution, whose result is shown in Table 4.16, and, on the other hand, some weighted approaches shown in Tables 4.17-4.19.

At $t = 1.1$, a shock is fully developed in the interior of the computational domain and enters the inflow boundary at $t = 8$. At $t = 12$ it is located at $x = 0$. We can see in Figure 4.8 that in this case the discontinuities are well captured by our scheme as well. The thresholding version is run with the parameters $\delta = 0.75$, $\delta' = 0.5$ and the WLS-GAW version is used for the weighted extrapolation.



(a) Lagrange extr.

(b) Lagrange extr. (zoom).

(c) Extr. with thresholds.

(d) Extr. with thresholds. (zoom).

Figure 4.7: Comparison of different extrapolations for the linear advection test with discontinuous solution.

Euler equations.

We end the one-dimensional experiments with an experiment using the Euler equations

$$
\begin{aligned}
&u_t + f(u)_x = 0, \quad u = u(x,t), \quad \Omega = (0,1),\\
&u = \begin{bmatrix} \rho \\ \rho v \\ E \end{bmatrix}, \quad f(u) = \begin{bmatrix} \rho v \\ p + \rho v^2 \\ v(E+p) \end{bmatrix},
\end{aligned}
\tag{4.12}
$$

where ρ is the density, v is the velocity and E is the specific energy of the system. The variable p stands for the pressure and is given by the equation of state:

$$
p = (\gamma - 1)\left(E - \frac{1}{2}\rho v^2\right),
$$

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
40	3.66E−5	–	7.45E−4	–
80	6.96E−7	5.72	1.73E−5	5.43
160	1.33E−8	5.70	3.58E−7	5.59
320	3.34E−10	5.32	1.15E−8	4.96
640	1.02E−11	5.04	3.43E−10	5.06
1280	3.19E−13	4.99	1.03E−11	5.06

Table 4.16: Error table for Burgers equation, thresholding, $t = 0.3$.

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
40	2.03E−5	–	3.57E−4	–
80	6.56E−7	4.95	1.47E−5	4.60
160	1.36E−8	5.59	2.81E−7	5.71
320	2.82E−10	5.58	9.16E−9	4.94
640	7.58E−12	5.22	2.76E−10	5.05
1280	2.23E−13	5.09	8.30E−12	5.06

Table 4.17: Error table for Burgers equation, $t = 0.3$. IW.

where γ is the adiabatic constant, that will be taken as 1.4.

We simulate the interaction of two blast waves [51] by using the following initial data

$$u(x,0) = \begin{cases} u_L & 0 < x < 0.1, \\ u_M & 0.1 < x < 0.9, \\ u_R & 0.9 < x < 1, \end{cases}$$

where $\rho_L = \rho_M = \rho_R = 1$, $v_L = v_M = v_R = 0$, $p_L = 10^3, p_M = 10^{-2}, p_R = 10^2$. We set reflecting boundary conditions at $x = 0$ and $x = 1$, simulating a solid wall at both sides. This problem involves multiple reflections of

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
40	1.88E−3	–	3.65E−2	–
80	6.62E−5	4.82	3.94E−3	3.21
160	1.72E−7	8.52	9.45E−6	8.70
320	2.50E−9	6.10	9.45E−6	5.37
640	4.49E−11	5.80	6.73E−9	5.08
1280	9.61E−13	5.55	1.92E−10	5.13

Table 4.18: Error table for Burgers equation, $t = 0.3$. WLS-UW.

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
40	7.80E-4	–	2.61E-2	–
80	2.82E-6	8.11	8.14E-5	8.32
160	1.11E-7	4.66	6.31E-6	3.69
320	2.40E-9	5.53	2.28E-7	4.79
640	4.48E-11	5.74	6.73E-9	5.08
1280	9.61E-13	5.54	1.92E-10	5.13

Table 4.19: Error table for Burgers equation, $t = 0.3$. WLS-GAW.

shocks and rarefactions off the walls and many interactions of waves inside the domain. We will use the same extrapolation nodes setup as in the previous tests as well as the same threshold values.

Figure 4.9 shows the density profile at $t = 0.038$ at two different resolutions, using thresholding extrapolation with $\delta = 0.75$, $\delta' = 0.5$ and the WLS-GAW weighted extrapolation, being the reference solution computed at a resolution of $h = 1/16000$. The figure clearly shows that the results are satisfactory.

4.3.2 Two-dimensional experiments

2D linear advection, $\mathcal{C}^\infty$ solution.

We consider the 2D linear advection equation

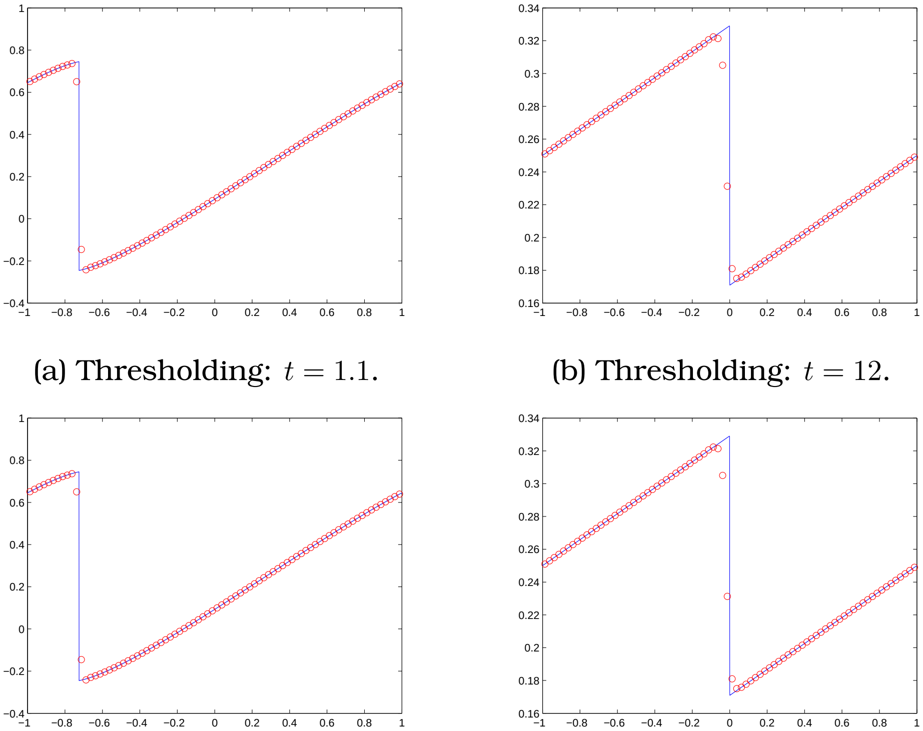
$$u_t + u_x + u_y = 0, \tag{4.13}$$

with

$$u_0 = u(x, y, 0) = 0.25 + 0.5\sin(\pi(x + y)) \tag{4.14}$$

for different domains Ω. We start with a square, $\Omega = (-1, 1)$ where inflow conditions $g(t) = u(x, y, t) = 0.25 + 0.5\sin(\pi(x + y - 2t))$ are imposed at the left and bottom boundary and outflow conditions at the rest. We compute the solution with the same setup and techniques to achieve fifth order accuracy as the above 1D tests for resolutions $n \times n$, for $n = 10 \cdot 2^j$, $1 \leq j \leq 6$, and we obtain the results in Tables 4.20-4.22 using different extrapolation techniques, running a simulation until $t = 1$.

We now change the domain and set Ω with respective boundary conditions as indicated in Figure 4.10, where Ω is the bounded connected component of $\mathbb{R}^2 \setminus K$, where K is the closed curve given by



(a) Thresholding: $t = 1.1$.

(b) Thresholding: $t = 12$.

(c) Weighted: $t = 1.1$.

(d) Weighted: $t = 12$.

Figure 4.8: Shock in Burgers equation, $n = 80$.

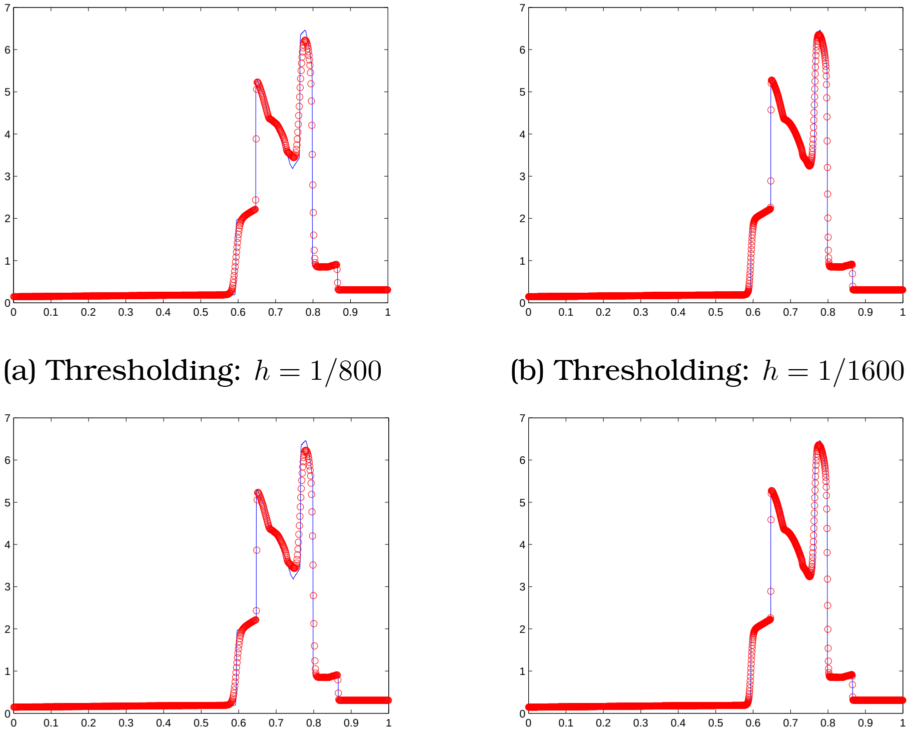
$$K = A_1 \cup A_2 \cup A_3 \cup A_4 \cup A_5 \cup A_6 \cup A_7 \cup A_8,$$

with

$$\begin{array}{lcllcl}
A_1 & = & \{(\cos(s), \sin(s)),\ s \in [0, \frac{\pi}{2}]\}, & A_5 & = & \{(\cos(s), \sin(s)),\ s \in [\pi, \frac{3\pi}{2}]\}, \\
A_2 & = & \{(-s, 1),\ s \in [0, \frac{3}{4}]\}, & A_6 & = & \{(s, -1),\ s \in [0, \frac{3}{4}]\}, \\
A_3 & = & \{(-\frac{3}{4} - s, 1 - 2s),\ s \in [0, \frac{1}{4}]\}, & A_7 & = & \{(\frac{3}{4} + s, -1 + 2s),\ s \in [0, \frac{1}{4}]\}, \\
A_4 & = & \{(-1, \frac{1}{2} - s),\ s \in [0, \frac{1}{2}]\}, & A_8 & = & \{(1, -\frac{1}{2} + s),\ s \in [0, \frac{1}{2}]\}.
\end{array}$$

Note that $\partial\Omega = K$.

This time, the complexity of the domain makes Lagrange extrapolation mildly unstable, whereas WLS is still stable and fifth order accurate as we can see numerically in Tables 4.23-4.24 that it is indeed achieved, running a simulation until $t = 0.85$.



(a) Thresholding: $h = 1/800$

(b) Thresholding: $h = 1/1600$

(c) Weights: $h = 1/800$

(d) Weights: $h = 1/1600$

Figure 4.9: Density profile for the blast wave problem, $t = 0.038$.

Euler equations.

The equations that will be considered in this section are the two-dimensional Euler equations for inviscid gas dynamics

$$
\begin{aligned}
&u_t + f(u)_x + g(u)_y = 0, \quad u = u(x,y,t),\\
&u = \begin{bmatrix} \rho \\ \rho v^x \\ \rho v^y \\ E \end{bmatrix}, \quad f(u) = \begin{bmatrix} \rho v^x \\ p + \rho (v^x)^2 \\ \rho v^x v^y \\ v^x(E+p) \end{bmatrix}, \quad g(u) = \begin{bmatrix} \rho v^y \\ \rho v^x v^y \\ p + \rho (v^y)^2 \\ v^y(E+p) \end{bmatrix}.
\end{aligned}
\tag{4.15}
$$

In these equations, ρ is the density, (v^x, v^y) is the velocity and E is the specific energy of the system. The variable p stands for the pressure and is given by the equation of state:

$$
p = (\gamma - 1)\left(E - \frac{1}{2}\rho((v^x)^2 + (v^y)^2)\right),
$$

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
20	5.36E−4	–	1.56E−3	–
40	1.66E−5	5.01	5.40E−5	4.85
80	5.24E−7	4.99	1.71E−6	4.98
160	1.65E−8	4.99	5.33E−8	5.00
320	5.15E−10	5.00	1.62E−9	5.04
640	1.63E−11	4.98	5.04E−11	5.01

Table 4.20: Error table for problem (4.13) - (4.14), Square domain, IW.

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
20	6.14E−3	–	9.07E−2	–
40	4.22E−5	7.19	2.98E−4	8.25
80	6.90E−7	5.93	3.47E−6	6.42
160	1.95E−8	5.15	9.13E−8	5.25
320	5.94E−10	5.04	2.81E−9	5.02
640	1.84E−11	5.01	8.90E−11	4.98

Table 4.21: Error table for problem (4.13) - (4.14), Square domain, WLS-UW.

where γ is the adiabatic constant, that will be taken as 1.4 in all the experiments.

Double Mach Reflection

This experiment uses the Euler equations to model a vertical right-going Mach 10 shock colliding with an equilateral triangle. By symmetry, this is equivalent to a collision with a ramp with a slope of 30 degrees with respect to the horizontal line, which is how we will model the simulation

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
20	8.22E−4	–	2.07E−3	–
40	2.12E−5	5.28	8.10E−5	4.68
80	6.39E−7	5.05	2.89E−6	4.81
160	1.94E−8	5.04	9.03E−8	5.00
320	5.94E−10	5.03	2.80E−9	5.01
640	1.84E−11	5.01	8.91E−11	4.98

Table 4.22: Error table for problem (4.13) - (4.14), Square domain, WLS-GAW.

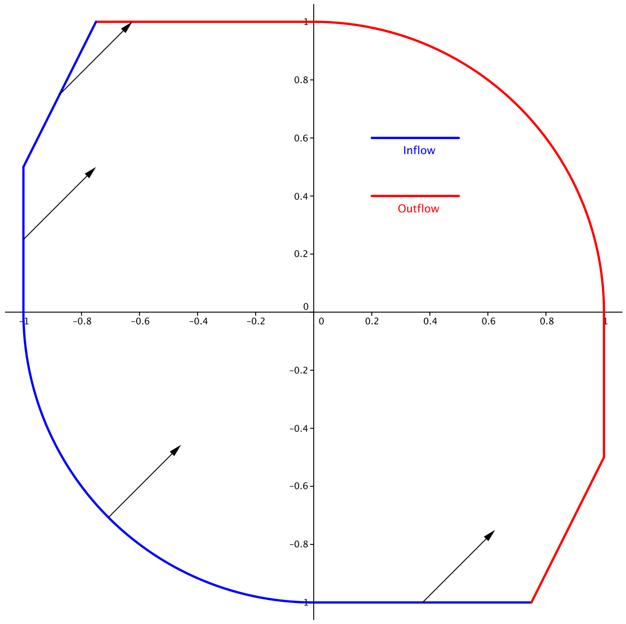


Figure 4.10: Complex domain for 2D experiment.

to half the computational cost.

The data for this problem are the following:

$$\Omega = \left\{ (x,y) \in (0,4) \times (0,4) : \; y > \frac{\sqrt{3}}{3}\left(x - \frac{1}{4}\right) \right\}.$$

The domain with the corresponding boundary conditions is sketched in Figure 4.11.

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
20	1.14E−2	–	1.12E−1	–
40	4.82E−3	1.56	4.54E−2	1.40
80	8.33E−6	9.18	2.25E−4	7.66
160	7.53E−8	6.79	3.00E−6	6.23
320	2.14E−9	5.14	1.01E−7	4.89
640	6.59E−11	5.02	3.57E−9	4.82

Table 4.23: Error table for problem (4.13) - (4.14), Complex domain, WLS-UW.

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
20	3.78E-3	$-$	3.06E-2	$-$
40	8.04E-5	5.56	1.41E-3	4.44
80	1.81E-6	5.48	2.39E-5	5.88
160	6.81E-8	4.73	2.42E-6	3.31
320	2.13E-9	5.00	9.96E-8	4.60
640	6.58E-11	5.01	3.57E-9	4.80

Table 4.24: Error table for problem (4.13) - (4.14), Complex domain, WLS-GAW.

The initial conditions are the following:

$$\begin{array}{rcll} u=(\rho,v^x,v^y,E) & = & (8.0,8.25,0,563.5) & \text{if} \quad x\leq\frac{1}{4} \\ u=(\rho,v^x,v^y,E) & = & (1.4,0,0,2.5) & \text{if} \quad x>\frac{1}{4} \end{array}$$

The most commonly used strategy for this simulation (see [51]) is to rotate the reference frame by -30 degrees, so that the simulation is cast into a rectangular domain, with a shock that is inclined 60 degrees with respect to the horizontal. In our case, we perform the simulation with the original problem to see that the improvement achieved by increasing the order of the extrapolations at the boundary leads to results that are comparable to those obtained with the rotated version.

Following the notation of the previous section, we have selected the values $R=10$, $M=3$ for the boundary in the thresholding case (substencils are of the same size as those in WENO5). The reason for selecting $R=10$ is not achieving an order higher than the one of the method, but having a wider stencil with more room for a safe selection of a substencil in smoothness regions.

We perform the simulation until $t=0.2$. The experiment consists in different simulations with different threshold values, considering also a version with a unique point in the stencil (order 1). In Figure 4.12 we present the result for the density ρ at a resolution of $h_x=h_y=\frac{\sqrt{3}}{2}\frac{1}{640}$, which is equivalent to a resolution $\hat{h}_x=\hat{h}_y=\frac{1}{640}$ in the rotated experiment. A comparison of the results for the original and the rotated experiment for different extrapolation options is shown in Figure 4.13.

The Schlieren plots shown in Figures 4.13, 4.15, 4.16 display the gradients of the density field in an exponential scale in a gray scale, where darker tonalities correspond to higher values (see [36] and references therein for details).

As it can be seen, lower threshold values lead to better defined vortices. Also, note that according to the results on the figures, the rotated

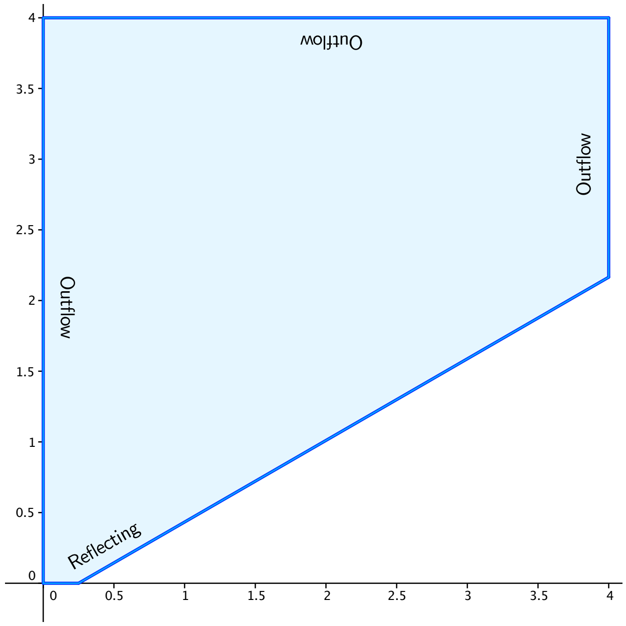


Figure 4.11: Domain for the double Mach reflection test.

problem (with first order extrapolations) looks better than the original problem using also first order extrapolations. One of the reasons might be the fact that the cell centers are exactly located on each normal line, leading to exact values on the first step of the extrapolation process (we recall the reader that this step consists in extrapolating information to points on normal lines from the values of the computational domain).

Interaction of a shock with a circular obstacle

We now change our data to a right-going vertical Mach 3 shock initially located at $x = 0.1$ with a circular obstacle with center $(0.5, 1)$ and radius 0.2 into a square domain $(0, 2) \times (0, 2)$. This experiment has already been performed in [6] using penalization techniques. The technique used here is the thresholding extrapolation for $\delta = \delta' = 0.99$. To halve the computational time by exploiting the symmetry of the solution, we run a simulation until $t = 0.4$ and a mesh size of $h_x = h_y = \frac{1}{512}$ on the upper half of the domain, by adding reflecting boundary conditions at the bottom. A Schlieren plot of the result can be seen at Figure 4.15. As it can be seen, the results are very similar to those obtained in [6].

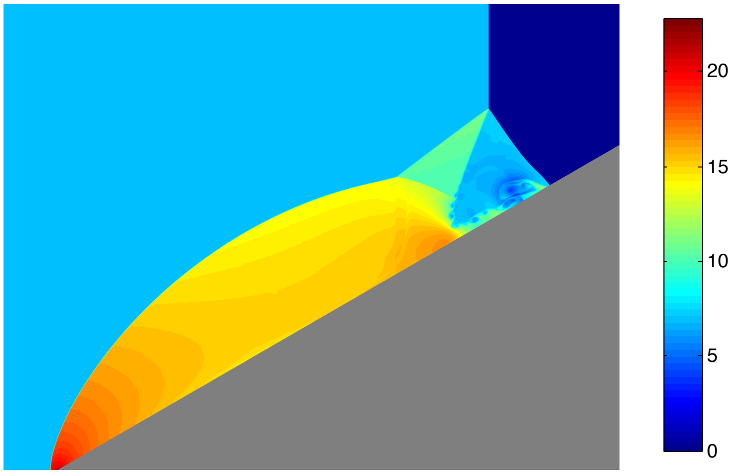


Figure 4.12: Double Mach Reflection: original problem.

Interaction of a shock with multiple circular obstacles

We repeat the previous experiment by adding multiple circles in the domain as shown in Figure 4.16. This test can also be found in [6]. In this case, we run the simulation until $t = 0.5$ and a mesh size of $h_x = h_y = \frac{1}{512}$ on the whole domain. As in the previous experiment, we present a Schlieren plot for the last time step in Figure 4.16. These results are again consistent with those obtained in [6].

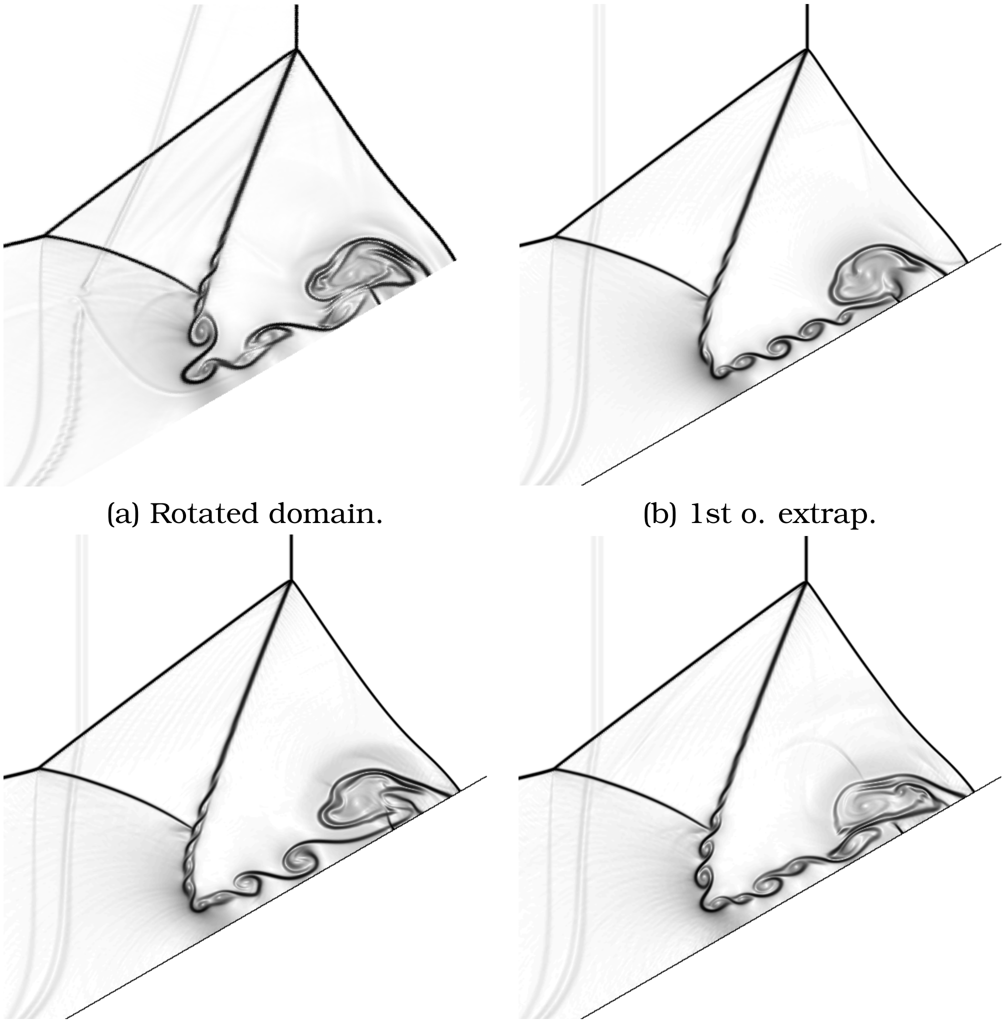
Steady-state supersonic flow around a triangle

We next simulate the flow field over a solid triangle with height $h = 0.5$ and half angle $\theta = 20$ deg moving at supersonic speeds. The initial conditions are

$$u = (\rho, v^x, v^y, p) = (1, \sqrt{\gamma}M_1, 0, 1)$$

and the computation is stopped when a steady state is obtained from the position of the shock waves. In order to halve the computational cost, we perform the simulation in the upper half of the domain, imposing appropriate reflecting boundary conditions at the symmetry axis. We solve this problem using the WLS-GAW technique at the boundary.

Figure 4.17 shows results that are consistent with those obtained in [6] and have sharper resolution that the ones reported in that paper for the same resolution.



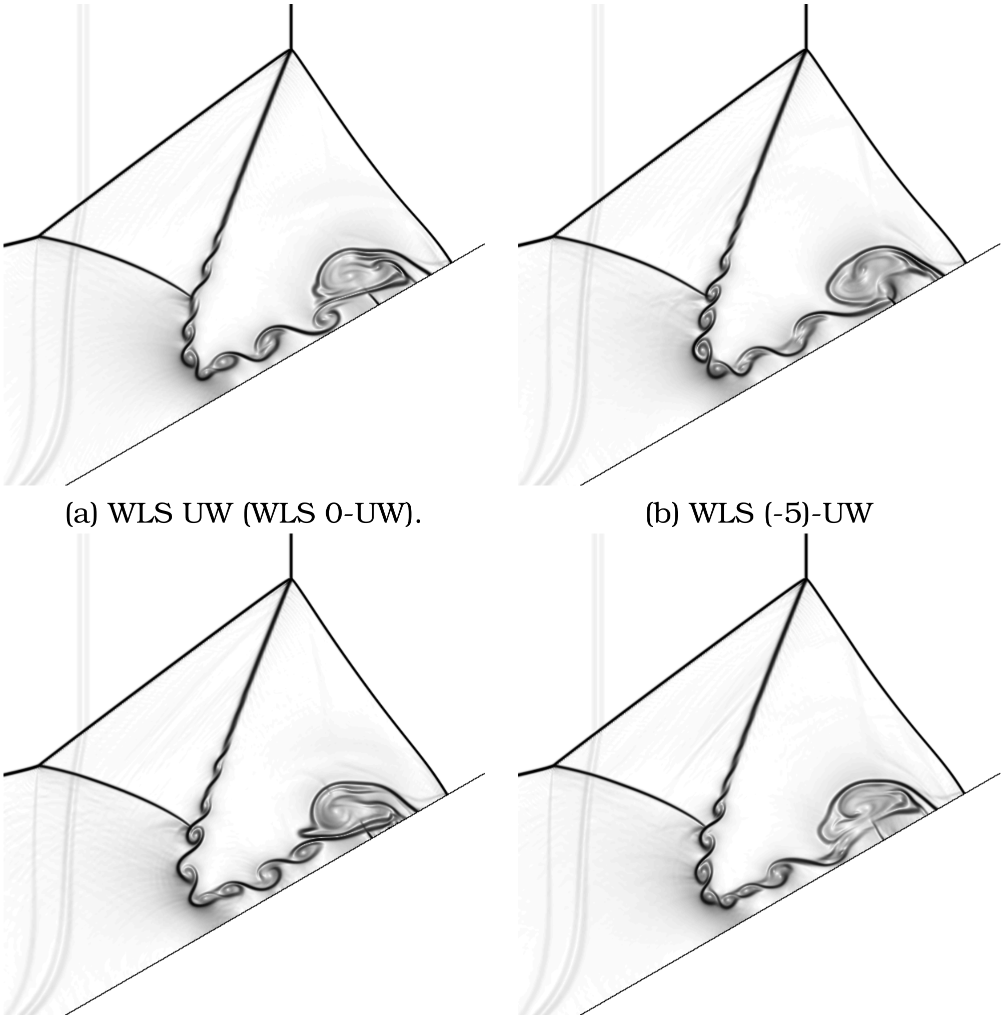
(a) Rotated domain.

(b) 1st o. extrap.

(c) 5th o. extrap. $\delta = \delta' = 0.9$.

(d) 5th o. extrap. $\delta = \delta' = 0.35$.

Figure 4.13: Double Mach Reflection: rotated version, first order and high order thresholding extrapolation. Enlarged view of the turbulence zone (Schlieren).



(a) WLS UW (WLS 0-UW).

(b) WLS (-5)-UW

(c) WLS (-50)-UW

(d) WLS GAW.

Figure 4.14: Double Mach Reflection: weighted extrapolation techniques. Enlarged view of the turbulence zone (Schlieren).

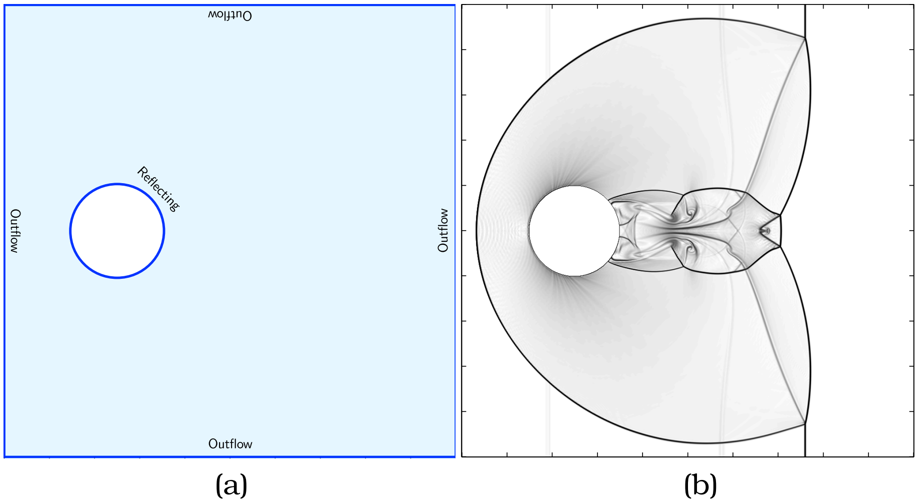


Figure 4.15: Circle reflection test: (a) domain; (b) simulation for $t = 0.4$.

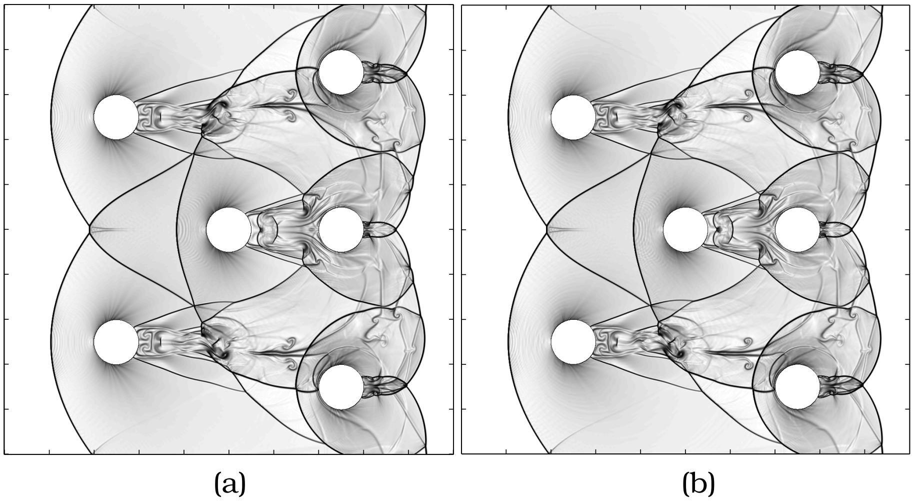


Figure 4.16: Circles reflection test, $t = 0.5$: (a) thresholding extrapolation, $\delta = \delta' = 0.99$; (b) WLS-GAW.

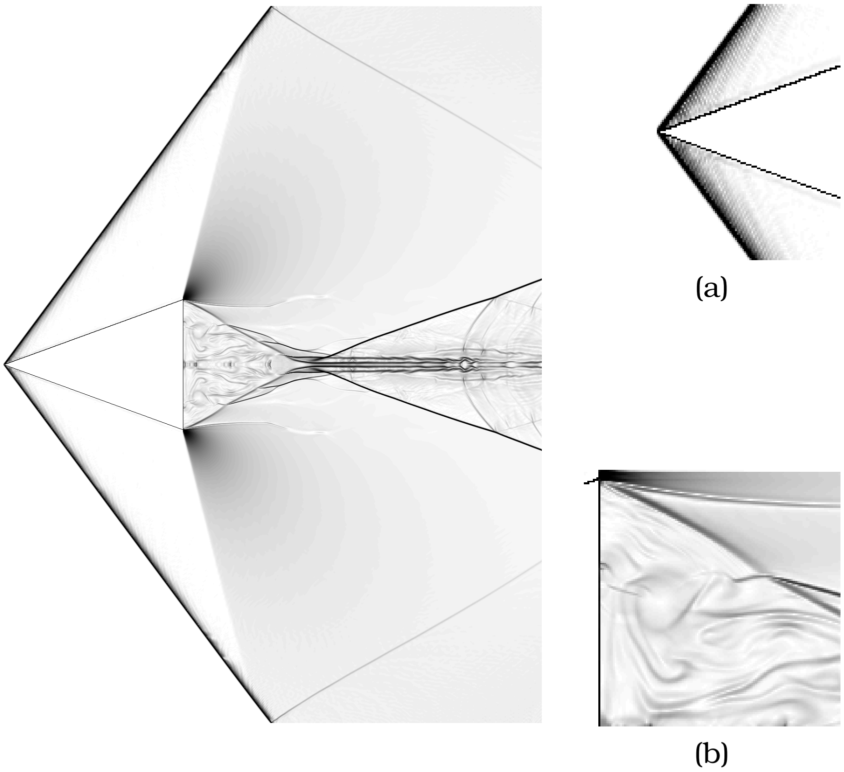


Figure 4.17: Triangle: (a) Left wedge (enlarged view). (b) Turbulence aside upper right corner of the triangle (enlarged view).

5

High order accurate time discretizations

In this chapter we present a high order accurate temporal scheme, which combined with the spatial HRSC scheme introduced in Chapter 2 and the high order boundary extrapolation techniques expounded in Chapter 4 yields a fully high order accurate scheme. For the derivation of the high order time scheme we take as starting point the one that was proposed in 2003 by Qiu and Shu [39], for numerically solving hyperbolic conservation laws, based on the conversion of time derivatives to spatial derivatives through the Cauchy-Kowalewski technique, following the Lax-Wendroff procedure.

For the sake of simplicity, we start with the one-dimensional scalar case ($d = m = 1$). For the solution $u(x, t)$ of $u_t + f(u)_x = 0$ on a fixed spatial grid (x_i) with spacing $h = x_{i+1} - x_i$ and some time t_n from a temporal grid with spacing $\delta = \Delta t = t_{n+1} - t_n > 0$, proportional to h, $\delta = \tau h$, where τ is dictated by stability restrictions (CFL condition) we use the following notation for time derivatives of u and $f(u)$:

$$u_{i,n}^{(l)} = \frac{\partial^l u(x_i, t_n)}{\partial t^l},$$
$$f_{i,n}^{(l)} = \frac{\partial^l f(u)(x_i, t_n)}{\partial t^l}.$$

Our goal is to obtain an R-th order accurate numerical scheme, i.e., a scheme with a local truncation error of order $R+1$, based on the Taylor expansion of the solution u from time t_n to the next time t_{n+1}:

$$u_i^{n+1} = \sum_{l=0}^{R} \frac{\Delta t^l}{l!} u_{i,n}^{(l)} + \mathcal{O}(\Delta t^{R+1}).$$

To achieve this we aim to define corresponding approximations

$$\widetilde{u}_{i,n}^{(l)} = u_{i,n}^{(l)} + \mathcal{O}(h^{R+1-l}),$$
$$\widetilde{f}_{i,n}^{(l)} = f_{i,n}^{(l)} + \mathcal{O}(h^{R-l}),$$

by recursion on l, assuming (for a local truncation error analysis) that $\widetilde{u}_{i,n}^{0} = u_{i,n}^{(0)} = u(x_i, t_n)$.

The fact that u solves the system of conservation laws implies that the time derivatives $u_{i,n}^{(l)}$, $1 \leq l \leq R$, can be written in terms of spatial derivatives of some functions of $u_{i,n}^{(j)}$, $j < l$,

$$f_{i,n}^{(l-1)} = F_{l-1}(u_i^n, u_{i,n}^{(1)}, \dots, u_{i,n}^{(l-1)}), \tag{5.1}$$

following the Cauchy-Kowalewski (or Lax-Wendroff for second order) procedure:

$$\frac{\partial^l u}{\partial t^l} = \frac{\partial^{l-1}}{\partial t^{l-1}}(u_t) = -\frac{\partial^{l-1}}{\partial t^{l-1}}(f(u)_x) = -\left[\frac{\partial^{l-1} f(u)}{\partial t^{l-1}}\right]_x, \tag{5.2}$$

and Faà di Bruno's formula stated in Theorem 2.

Specifically, to approximate the first time derivative, $u_t = -f(u)_x$, we use the Shu-Osher finite difference scheme [42] with upwinded WENO spatial reconstructions [26] of order $2r-1$ in the flux function:

$$u_{i,n}^{(1)} = u_t(x_i, t_n) = -[f(u)]_x(x_i, t_n) = -\frac{\hat{f}_{i+\frac{1}{2}}^n - \hat{f}_{i-\frac{1}{2}}^n}{h} + \mathcal{O}(h^{2r-1}). \tag{5.3}$$

Much cheaper centered differences are used instead for the next derivatives. We expound the general procedure for a third order accurate scheme ($R = 3$) for a scalar one-dimensional conservation law.

Assume we have numerical data, $\{\widetilde{u}_i^n\}_{i=0}^{M-1}$, which approximates $u(\cdot, t_n)$ and want to compute an approximation for $u(\cdot, t_{n+1})$ at the same nodes, namely, $\{\widetilde{u}_i^{n+1}\}_{i=0}^{M-1}$.

First, we compute an approximation of u_t by the procedure stated above:

$$\widetilde{u}_{i,n}^{(1)} = -\frac{\hat{f}_{i+\frac{1}{2}}^n - \hat{f}_{i-\frac{1}{2}}^n}{h},$$

with

$$\hat{f}_{i+\frac{1}{2}}^n = \hat{f}(\widetilde{u}_{i-r+1}^n, \dots, \widetilde{u}_{i+r}^n)$$

the numerical fluxes, which are obtained through upwind WENO spatial reconstructions of order $2r-1$, with $r = \lceil \frac{R+1}{2} \rceil = 2$.

Once the corresponding nodal data is obtained for the approximated values of u_t, we compute

$$u_{tt} = [u_t]_t = [-f(u)_x]_t = -[f(u)_t]_x = -[f'(u)u_t]_x,$$

where $f'(u)u_t$ is now an approximately known expression for the required nodes. We use then a second order centered difference in order to obtain the approximation:

$$\widetilde{u}_{i,n}^{(2)} = -\frac{\widetilde{f}_{i+1,n}^{(1)} - \widetilde{f}_{i-1,n}^{(1)}}{2h},$$

where

$$\widetilde{f}_{i,n}^{(1)} = F_1(\widetilde{u}_{i,n}^{(0)}, \widetilde{u}_{i,n}^{(1)}) = f'(\widetilde{u}_{i,n}^{(0)})\widetilde{u}_{i,n}^{(1)},$$

Finally, we approximate the third derivative:

$$u_{ttt} = [u_t]_{tt} = [-f(u)_x]_{tt} = -[f(u)_{tt}]_x = -\Big(f''(u)u_t^2 + f'(u)u_{tt}\Big)_x,$$

where again the function $f''(u)u_t^2 + f'(u)u_{tt}$ is approximately known at the nodes and therefore u_{ttt} can be computed by second order accurate centered differences (note that in this case it would be required only a first order accurate approximation; however, the order of centered approximations is always even):

$$\widetilde{u}_{i,n}^{(3)} = -\frac{\widetilde{f}_{i+1,n}^{(2)} - \widetilde{f}_{i-1,n}^{(2)}}{2h},$$

where

$$\widetilde{f}_{i,n}^{(2)} = F_2(\widetilde{u}_{i,n}^{(0)}, \widetilde{u}_{i,n}^{(1)}, \widetilde{u}_{i,n}^{(2)}) = f''(\widetilde{u}_{i,n}^{(0)}) \cdot (\widetilde{u}_{i,n}^{(1)})^2 + f'(\widetilde{u}_{i,n}^{(0)}) \cdot (\widetilde{u}_{i,n}^{(2)})^2.$$

Once all the needed data is obtained, we advance in time by replacing the terms of the third order Taylor expansion in time of $u(\cdot, t_{n+1})$ by their corresponding nodal approximations:

$$\widetilde{u}_i^{n+1} = \widetilde{u}_i^n + \Delta t \widetilde{u}_{i,n}^{(1)} + \frac{\Delta t^2}{2} \widetilde{u}_{i,n}^{(2)} + \frac{\Delta t^3}{6} \widetilde{u}_{i,n}^{(3)}.$$

As we shall see, the above example can be extended to arbitrarily high order time schemes through the computation of the suitable high order central differences of the nodal values

$$\widetilde{f}_{i,n}^{(l)} = F_l(\widetilde{u}_{i,n}^{(0)}, \widetilde{u}_{i,n}^{(1)}, \dots, \widetilde{u}_{i,n}^{(l)}) = f_{i,n}^{(l)} + \mathcal{O}(h^{R-l+1}).$$

The generalization to multiple dimensions is straightforward, since now the Cauchy-Kowalewski procedure, being based on the fact that $u_t = -\nabla \cdot f(u)$, yields

$$\frac{\partial^l u}{\partial t^l} = -\nabla \cdot \left(\frac{\partial^{l-1} f(u)}{\partial t^{l-1}} \right) = -\sum_{i=1}^{d} \frac{\partial}{\partial x_i} \left(\frac{\partial^{l-1} f^i(u)}{\partial t^{l-1}} \right)$$

and that the spatial reconstruction procedures are done separately for each dimension. For the case of the systems of equations, the time derivatives are now computed through tensorial products of the corresponding derivatives of the Jacobian of the fluxes and Faà di Bruno's formula in Theorem 2 describes a procedure to compute them. The general procedure for systems and multiple dimensions is thus easily generalizable and further details about the procedure can be found in [39].

5.1 The approximate Lax-Wendroff procedure

As reported by the authors of [39], the computation of the exact nodal values of $f^{(k)}$ can be very expensive as k increases, since the number of required operations increases exponentially. Moreover, implementing it is costly and requires large symbolic computations for each equation.

We now present an alternative, which is much less expensive for large k and agnostic about the equation, in the sense that its only requirement is the knowledge of the flux function. This procedure also works in the multidimensional case and in the case of systems as well (by working componentwise). This technique is based on the observation that approximations $\widetilde{f}^{(l-1)} \approx f^{(l-1)}$ can be easily obtained by a clever use of suitable finite differences, rather than using the exact expression F_{l-1} in (5.1).

5.1.1 Scheme formulation and theoretical results

We next introduce some notation which will help in the description of the approximate fluxes technique along this section. We assume a one-dimensional system for the sake of simplicity.

For a function $u\colon \mathbb{R} \to \mathbb{R}^m$, we denote the function on the grid defined by a base point a and grid space h by

$$G_{a,h}(u)\colon \mathbb{Z} \to \mathbb{R}^m, \quad G_{a,h}(u)_i = u(a+ih).$$

We denote by $\Delta_h^{p,q}$ the centered finite differences operator that approximates p-th order derivatives to order $2q$ on grids with spacing h. For any u sufficiently differentiable, it satisfies:

$$\Delta_h^{p,q} G_{a,h}(u) = u^{(p)}(a) + \alpha^{p,q} u^{(p+2q)}(a) h^{2q} + \mathcal{O}(h^{2q+2}), \tag{5.4}$$

see Proposition 4 for more details.

We aim to define approximations $\widetilde{u}^{(k)}_{i,n} \approx u^{(k)}_{i,n}$, $k = 0, \dots, R$, recursively. We start the recursion with

$$\begin{aligned} \widetilde{u}^{(0)}_{i,n} &= u_i^n, \\ \widetilde{u}^{(1)}_{i,n} &= -\frac{\hat{f}^n_{i+\frac{1}{2}} - \hat{f}^n_{i-\frac{1}{2}}}{h}, \end{aligned} \tag{5.5}$$

where $\hat{f}^n_{i+\frac{1}{2}}$ are computed by applying upwind WENO reconstructions to split fluxes obtained from the data (u_i^n) at time step n (see [42, 11, 26] for further details)

Associated to fixed h, i, n, once obtained $\widetilde{u}^{(l)}_{i,n}$, $l = 0, \dots, k$, in the recursive process we define the k-th degree approximated Taylor polynomial $T_k[h, i, n]$ by

$$T_k[h,i,n](\rho) = \sum_{l=0}^{k} \frac{\widetilde{u}^{(l)}_{i,n}}{l!} \rho^l.$$

By recursion, for $k = 1, \dots, R-1$, we define

$$\begin{aligned} \widetilde{f}^{(k)}_{i,n} &= \Delta_\delta^{k,\left\lceil \frac{R-k}{2} \right\rceil} \Big(G_{0,\delta}\big(f(T_k[h,i,n])\big) \Big), \\ \widetilde{u}^{(k+1)}_{i,n} &= -\Delta_h^{1,\left\lceil \frac{R-k}{2} \right\rceil} \widetilde{f}^{(k)}_{i+\cdot,n}, \end{aligned} \tag{5.6}$$

where we denote by $\widetilde{f}^{(k)}_{i+\cdot,n}$ the vector given by the elements $(\widetilde{f}^{(k)}_{i+\cdot,n})_j = \widetilde{f}^{(k)}_{i+j,n}$, recall that $\delta = \Delta t$ and, as previously mentioned, $\widetilde{u}^{(0)}_{i,n} = u_i^n$ is the data at the n-th time step. With all these ingredients, the proposed scheme is:

$$u_i^{n+1} = u_i^n + \sum_{l=1}^{R} \frac{\Delta t^l}{l!} \widetilde{u}^{(l)}_{i,n}. \tag{5.7}$$

Proposition 3. *The scheme defined by* (5.6) *and* (5.7) *is R-th order accurate.*

Proof. For the accuracy analysis of the local truncation error, we take

$$\widetilde{u}^{(0)}_{i,n} = u(x_i, t_n). \tag{5.8}$$

We now use induction on $k = 0, \dots, R$ to prove that

$$\widetilde{u}^{(k)}_{i,n} = u^{(k)}_{i,n} + c^k(x_i, t_n)h^{R-k+1} + \mathcal{O}(h^{R-k+2}), \tag{5.9}$$

for continuously differentiable functions c^k. The result in (5.9) for $k = 1$ immediately follows from the fact that WENO finite differences applied to the exact data in (5.8) yield approximations

$$\frac{\hat{f}_{i+\frac{1}{2},n} - \hat{f}_{i-\frac{1}{2},n}}{h} = f(u)_x(x_i, t_n) + \widetilde{c}^1(x_i, t_n)h^{2r-1} + \mathcal{O}(h^{2r}), \quad r = \lceil \frac{R+1}{2} \rceil.$$

From the definition in (5.5) we deduce

$$\widetilde{u}^{(1)}_{i,n} = -\frac{\hat{f}_{i+\frac{1}{2},n} - \hat{f}_{i-\frac{1}{2},n}}{h} = u^{(1)}_{i,n} - \widetilde{c}^1(x_i, t_n)h^{2r-1} + \mathcal{O}(h^{2r}), \quad 2r \geq R+1,$$

thus proving the case $k = 1$, by taking $c^1 = -\widetilde{c}^1$ if $2r = R+1$ or $c^1 = 0$ if $2r > R+1$.

Assume now the result to hold for k and aim to prove it for $k+1 \leq R$. For this purpose we first prove the following estimate:

$$\widetilde{f}^{(k)}_{i,n} = f^{(k)}_{i,n} + a^k(x_i, t_n)h^{R-k} + b^k(x_i, t_n)h^{R-k+1} + \mathcal{O}(h^{R-k+2}), \tag{5.10}$$

for continuously differentiable functions a^k, b^k.

From (5.6) and (5.4), with $q = \lceil \frac{R-k}{2} \rceil$ and the notation $v = T_k[h, i, n]$:

$$\widetilde{f}^{(k)}_{i,n} = (f(v))^{(k)}(0) + \alpha^{k,q}(f(v))^{(k+2q)}(0)h^{2q} + \mathcal{O}(h^{2q+2}). \tag{5.11}$$

Now, Faà di Bruno's formula (2.23) yields:

$$\begin{aligned} (f(v))^{(k)}(0) &= \sum_{s\in\mathcal{P}_k} \begin{bmatrix} k \\ s \end{bmatrix} f^{(|s|)}(v(0))\big(D^s v(0)\big), \\ D^s v(0) &= \left[\overbrace{\frac{v^{(1)}(0)}{1!} \quad \dots \quad \frac{v^{(1)}(0)}{1!}}^{s_1} \quad \dots \quad \overbrace{\frac{v^{(k)}(0)}{k!} \quad \dots \quad \frac{v^{(k)}(0)}{k!}}^{s_k} \right], \\ D^s v(0) &= \left[\overbrace{\frac{\widetilde{u}_{i,n}^{(1)}}{1!} \quad \dots \quad \frac{\widetilde{u}_{i,n}^{(1)}}{1!}}^{s_1} \quad \dots \quad \overbrace{\frac{\widetilde{u}_{i,n}^{(k)}}{k!} \quad \dots \quad \frac{\widetilde{u}_{i,n}^{(k)}}{k!}}^{s_k} \right]. \end{aligned} \tag{5.12}$$

Since $v = T_k[h, i, n]$ is a k-th degree polynomial, $v^{(j)} = 0$ for $j > k$. Therefore, in the same fashion as before,

$$(f(v))^{(k+2q)}(0) = \sum_{s\in\mathcal{P}_{k+2q}^{k}} \begin{bmatrix} k \\ s \end{bmatrix} f^{(|s|)}(v(0))\big(D^s v(0)\big), \tag{5.13}$$

where $\mathcal{P}_{k+2q}^{k} = \{s \in \mathcal{P}_{k+2q} / s_j = 0, j > k\}$.

On the other hand, another application of Faà di Bruno's formula to $f(u)$, yields:

$$\begin{aligned} f_{i,n}^{(k)} = f(u)^{(k)}(x_i, t_n) &= \sum_{s\in\mathcal{P}_k} \begin{bmatrix} k \\ s \end{bmatrix} f^{(|s|)}(u(x_i, t_n))\big(D^s u(x_i, t_n)\big), \\ D^s u(x_i, t_n) &= \left[\overbrace{\frac{u^{(1)}(x_i,t_n)}{1!} \quad \dots \quad \frac{u^{(1)}(x_i,t_n)}{1!}}^{s_1} \quad \dots \quad \overbrace{\frac{u^{(k)}(x_i,t_n)}{k!} \quad \dots \quad \frac{u^{(k)}(x_i,t_n)}{k!}}^{s_k} \right], \\ D^s u(x_i, t_n) &= \left[\overbrace{\frac{u_{i,n}^{(1)}}{1!} \quad \dots \quad \frac{u_{i,n}^{(1)}}{1!}}^{s_1} \quad \dots \quad \overbrace{\frac{u_{i,n}^{(k)}}{k!} \quad \dots \quad \frac{u_{i,n}^{(k)}}{k!}}^{s_k} \right]. \end{aligned} \tag{5.14}$$

We have $v(0) = \widetilde{u}_{i,n}^{(0)} = u(x_i, t_n)$ and, by induction,

$$\widetilde{u}_{i,n}^{(l)} = u_{i,n}^{(l)} + c^l(x_i, t_n)h^{R-l+1} + \mathcal{O}(h^{R-l+2}), \quad l = 1, \dots, k. \tag{5.15}$$

For any $s \in \mathcal{P}_k$, $D^s v(0)$ is a $m \times |s|$ matrix, and for any $\mu \in \{1, \dots, m\}$ and $\nu \in \{1, \dots, |s|\}$, we have from (2.24), (5.12), (5.14) and (5.15) that

$$(D^s v(0) - D^s u(x_i, t_n))_{\mu,\nu} = \frac{(\widetilde{u}_{i,n}^{(l)} - u_{i,n}^{(l)})_\mu}{l!} = \frac{c_\mu^l(x_i, t_n)}{l!} h^{R-l+1} + \mathcal{O}(h^{R-l+2}), \tag{5.16}$$

for some $l = l(s, \nu) \leq k$. From the definition of the set $\mathcal{P}_k$, the only k-tuple $s \in \mathcal{P}_k$ such that $s_k \neq 0$ is $s^* = (0, \dots, 1)$. Therefore, from the definition of the operator D^s in (2.24) (or (5.14) (5.12)), the only $s \in \mathcal{P}_k$, $\nu \leq |s|$, such that $l(s,\nu) = k$ is s^*, $\nu = |s^*| = 1$. We deduce from (5.16) that

$$(D^s v(0) - D^s u(x_i,t_n))_{\mu,\nu} = \mathcal{O}(h^{R-k+2}), \quad \forall s \in \mathcal{P}_s, s \neq s^*, \forall \mu \leq m, \forall \nu \leq |s|, \tag{5.17}$$

$$(D^{s^*} v(0) - D^{s^*} u(x_i,t_n))_{\mu,1} = \frac{c^k_\mu(x_i,t_n)}{k!} h^{R-k+1} + \mathcal{O}(h^{R-k+2}). \tag{5.18}$$

We deduce from (5.16), (2.25), (5.12), (5.14), (5.17) that

$$\begin{aligned} f(v)^{(k)}(0) - f(u)^{(k)}(x_i,t_n) &= \begin{bmatrix} k \\ s^* \end{bmatrix} f^{(|s^*|)}(u(x_i,t_n))\big(D^{s^*} v(0) - D^{s^*} u(x_i,t_n)\big) \\ &\quad + \sum_{\substack{s \in \mathcal{P}_k \\ s \neq s^*}} \begin{bmatrix} k \\ s \end{bmatrix} f^{(|s|)}(u(x_i,t_n))\big(D^s v(0) - D^s u(x_i,t_n)\big), \\ f(v)^{(k)}(0) - f(u)^{(k)}(x_i,t_n) &= \sum_{\mu=1}^{m} \frac{\partial f}{\partial u_\mu}(u(x_i,t_n)) c^k_\mu(x_i,t_n) h^{R-k+1} + \mathcal{O}(h^{R-k+2}). \end{aligned} \tag{5.19}$$

With a similar argument, taking into account that $k+1 \leq R$, we deduce from (5.13) and (5.16) that

$$\begin{aligned} (f(v))^{(k+2q)}(0) &= e^{k,q}(x_i,t_n) + \mathcal{O}(h^{R-k+1}) = e^{k,q}(x_i,t_n) + \mathcal{O}(h^2), \\ e^{k,q}(x,t) &= \sum_{s \in \mathcal{P}^k_{k+2q}} \begin{bmatrix} k \\ s \end{bmatrix} f^{(|s|)}(v(0))\big(D^s u(x,t)\big). \end{aligned} \tag{5.20}$$

Now, (5.11), (5.14), (5.20) and (5.19) yield:

$$\begin{aligned} \widetilde{f}^{(k)}_{i,n} - f^{(k)}_{i,n} &= \sum_{\mu=1}^{m} \frac{\partial f}{\partial u_\mu}(u(x_i,t_n)) c^k_\mu(x_i,t_n) h^{R-k+1} + \mathcal{O}(h^{R-k+2}) \\ &\quad + e^{k,q}(x_i,t_n) h^{2q} + \mathcal{O}(h^{2q+2}). \end{aligned}$$

Since $2q = R-k$ or $2q = R-k+1$, we deduce (5.10) with

$$a^k(x,t) = \begin{cases} e^k(x,t) & 2q = R-k \\ 0 & 2q = R-k+1 \end{cases}$$

$$b^k(x,t) = \begin{cases} \sum_{\mu=1}^{m} \frac{\partial f}{\partial u_\mu}(u(x,t)) c^k_\mu(x,t) & 2q = R-k \\ \sum_{\mu=1}^{m} \frac{\partial f}{\partial u_\mu}(u(x,t)) c^k_\mu(x,t) + e^{k,q}(x,t) & 2q = R-k+1. \end{cases}$$

To prove (5.9) for $k+1$, we apply the linear operator $-\Delta_h^{1,q}$, for $q = \lceil \frac{R-k}{2} \rceil$, to both sides of the already established equality (5.10), taking into account (5.4) and that $2q \geq R-k$:

$$\begin{aligned}
\widetilde{u}^{(k+1)}_{i,n} &= -\Delta_h^{1,q} \widetilde{f}^{(k)}_{i+\cdot,n} \\
&= -\Delta_h^{1,q} f^{(k)}_{i+\cdot,n} - h^{R-k}\Delta_h^{1,q} G_{x_i,h}(a^k(\cdot,t_n)) - h^{R-k+1}\Delta_h^{1,q} G_{x_i,h}(b^k(\cdot,t_n)) \\
&+ \mathcal{O}(h^{R-k+1}) \\
&= -\Delta_h^{1,q} G_{x_i,h}\big(f(u)^{(k)}(\cdot,t_n)\big) - h^{R-k}\big(\frac{\partial a^k}{\partial x}(x_i,t_n) + \mathcal{O}(h^{2q})\big) \\
&- h^{R-k+1}\big(\frac{\partial b^k}{\partial x}(x_i,t_n) + \mathcal{O}(h^{2q})\big) + \mathcal{O}(h^{R-k+1}) \\
&= -[f(u)^{(k)}]_x(x_i,t_n) - \alpha^{1,q}\frac{\partial^{k+2q+1} f(u)}{\partial x^{2q+1}\partial t^k}(x_i,t_n)h^{2q} + \mathcal{O}(h^{2q+2}) \\
&- h^{R-k}\frac{\partial a^k}{\partial x}(x_i,t_n) + \mathcal{O}(h^{R-k+1}) \\
&= u^{(k+1)}(x_i,t_n) + c^{k+1}(x_i,t_n)h^{R-k} + \mathcal{O}(h^{R-k+1}),
\end{aligned}$$

where

$$c^{k+1}(x,t) = -\frac{\partial a^k}{\partial x}(x,t) - \begin{cases} 0 & 2q > R-k \\ \alpha^{1,q}\frac{\partial^{k+2q+1} f(u)}{\partial x^{2q+1}\partial t^k}(x,t) & 2q = R-k. \end{cases}$$

The local truncation error is given by

$$u^{(0)}_{i,n+1} - \sum_{l=0}^{R} \frac{(\Delta t)^l}{l!}\widetilde{u}^{(l)}_{i,n},$$

where $\widetilde{u}^{(l)}_{i,n}$ are computed from $\widetilde{u}^{(0)}_{i,n} = u(x_i,t_n)$. Taylor expansion of the first term and the estimates in (5.9) yield that the local truncation error is:

$$\begin{aligned}
&\sum_{l=1}^{R} \frac{(\Delta t)^l}{l!}(u^{(l)}_{i,n} - \widetilde{u}^{(l)}_{i,n}) + \mathcal{O}(h^{R+1}) \\
&= \sum_{l=1}^{R} \frac{(\Delta t)^l}{l!}\mathcal{O}(h^{R-l+1}) + \mathcal{O}(h^{R+1}) = \mathcal{O}(h^{R+1}),
\end{aligned}$$

since Δt is proportional to h. □

The following result yields optimal central finite difference approximations to derivatives of any order.

Proposition 4. *For any* $p, q \in \mathbb{N}$, *there exist* $\beta_l^{p,q}$, $l = 0, \ldots, s := \lfloor \frac{p-1}{2} \rfloor + q$ *such that*

$$\Delta_h^{p,q} v = \frac{1}{h^p} \sum_{l=0}^{s} \beta_l^{p,q} (v_l + (-1)^p v_{-l}) \tag{5.21}$$

verifies (5.4).

Proof. We set

$$\Delta_h^{p,q} v = \frac{1}{h^p} \sum_{l=-s}^{s} \beta_l^{p,q} v_l, \quad s = \left\lfloor \frac{p-1}{2} \right\rfloor + q \tag{5.22}$$

for $\beta_l^{p,q}$ to determine such that

$$\psi(h) = \psi^{p,q}(h) = \sum_{l=-s}^{s} \beta_l^{p,q} u(a + lh),$$

satisfies

$$\psi^{(r)}(0) = 0, \quad r = 0, \ldots, 2s, r \neq p, \quad \psi^{(p)}(0) = p! u^{(p)}(a). \tag{5.23}$$

Since

$$\psi^{(r)}(0) = \sum_{l=-s}^{s} \beta_l^{p,q} l^r u^{(r)}(a),$$

(5.23) is equivalent to the system of $2s+1$ equations and $2s+1$ unknowns

$$\begin{aligned} &\sum_{l=-s}^{s} \beta_l^{p,q} l^r = 0, \quad r = 0, \ldots, 2s, r \neq p, \\ &\sum_{l=-s}^{s} \beta_l^{p,q} l^p = p!, \end{aligned} \tag{5.24}$$

whose coefficient matrix is a Vandermonde invertible matrix, and it thus have a unique solution. We see now that if p is even then $\beta_{-l}^{p,q} = \beta_l^{p,q}$, $l = 1, \ldots, s$ and if it is odd then $\beta_{-l}^{p,q} = -\beta_l^{p,q}$, $l = 0, \ldots, s$. For the first case (5.24) yields

$$\sum_{l=-s}^{s} \beta_l^{p,q} l^r = 0, \quad r = 1, 3, \ldots, 2s-1,$$

$$\sum_{l=1}^{s} (\beta_l^{p,q} - \beta_{-l}^{p,q}) l^r = 0, \quad r = 1, 3, \ldots, 2s-1,$$

which is a homogeneous system of s equations with s unknowns, with an invertible (Vandermonde) matrix, therefore $\beta_l^{p,q} - \beta_{-l}^{p,q} = 0$, $l = 1, \ldots, s$. The case for odd p is handled similarly.

With this, we have from a Taylor expansion of ψ that:

$$\begin{aligned}\Delta_h^{p,q} G_{a,h} u = \frac{1}{h^p}\psi(h) &= \frac{1}{h^p}(\frac{\psi^{(p)}(0)}{p!}h^p + \sum_{r=2s+1}^{\infty}\frac{\psi^{(r)}(0)}{r!}h^r)\\ &= u^{(p)}(0) + \sum_{r=2s+1}^{\infty}\sum_{l=-s}^{s}\beta_l^{p,q} l^r \frac{u^{(r)}(a)}{r!}h^{r-p}\\ &= u^{(p)}(0) + \sum_{r=2s+1}^{\infty}\sum_{l=1}^{s}(\beta_l^{p,q} + (-1)^r\beta_{-l}^{p,q})l^r\frac{u^{(r)}(a)}{r!}h^{r-p}.\end{aligned}$$

Now, if p is even, then $\beta_l^{p,q} = \beta_{-l}^{p,q}$ and, therefore, the only remaining terms are those with even r:

$$\begin{aligned}&\sum_{r=2s+1}^{\infty}\sum_{l=1}^{s}(\beta_l^{p,q} + (-1)^r\beta_{-l}^{p,q})l^r\frac{u^{(r)}(a)}{r!}h^{r-p}\\ &= \sum_{m=s+1}^{\infty}\alpha_m^{p,q}u^{(2m)}(a)h^{2m-p}, \quad \alpha_m^{p,q} = \frac{2}{(2m)!}\sum_{l=1}^{s}\beta_l^{p,q}l^{2m}.\end{aligned}$$

On the other hand, if p is odd, then $\beta_{-l}^{p,q} = -\beta_l^{p,q}$ and, therefore, the only remaining terms are those with odd r:

$$\begin{aligned}&\sum_{r=2s+1}^{\infty}\sum_{l=1}^{s}(\beta_l^{p,q} + (-1)^r\beta_{-l}^{p,q})l^r\frac{u^{(r)}(a)}{r!}h^{r-p}\\ &= \sum_{m=s}^{\infty}\alpha_m^{p,q}u^{(2m+1)}(a)h^{2m+1-p}, \quad \alpha_m^{p,q} = \frac{2}{(2m)!}\sum_{l=1}^{s}\beta_l^{p,q}l^{2m+1}.\end{aligned}$$

One can check that the definition of s gives that the smallest exponent in the remainder terms is $2q$. The result follows easily if one redefines $\beta_0^{p,q} = \beta_0^{p,q}/2$ for even p (for odd p it is 0). □

Finally, we next present a result which ensures that our scheme, being based on approximations of flux derivatives, is conservative.

Theorem 3. *The scheme resulting of the flux approximation procedure can be written in conservation form, namely,*

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{h}(\hat{g}_{i+\frac{1}{2}}^n - \hat{g}_{i-\frac{1}{2}}^n). \tag{5.25}$$

Proof. The key to (5.25) is to express $\Delta_h^{1,q}v$ in (5.22) in a conservative way:

$$\begin{aligned}\Delta_h^{1,q}v := \frac{1}{h}\sum_{l=-q}^{q}\beta_l^{1,q}v_l &= \frac{1}{h}\left(\sum_{l=-q}^{q-1}\gamma_l^q v_{l+1} - \sum_{l=-q}^{q-1}\gamma_l^q v_l\right)\\ &= \frac{1}{h}\left(\gamma_{q-1}^q v_q + \sum_{l=-q+1}^{q-1}(\gamma_{l-1}^q - \gamma_l^q)v_l - \gamma_{-q}v_{-q}^q\right),\end{aligned} \qquad (5.26)$$

with γ_l^q to be determined. Since the latter ought to be satisfied by any v, we deduce that

$$\begin{aligned}\gamma_{q-1}^q &= \beta_q^{1,q},\\ \gamma_{l-1}^q - \gamma_l^q &= \beta_l^{1,q}, \quad l = -q+1, \dots q-1,\\ -\gamma_{-q}^q &= \beta_{-q}^{1,q}.\end{aligned}$$

This is a system of $2q+1$ equations with $2q$ unknowns:

$$\begin{bmatrix} -1 & 0 & 0 & \dots & \dots & 0\\ 1 & -1 & 0 & \dots & \dots & 0\\ 0 & 1 & -1 & \dots & \dots & 0\\ \vdots & & \ddots & \ddots & & \vdots\\ \vdots & & & \ddots & \ddots & \vdots\\ 0 & \dots & \dots & 0 & 1 & -1\\ 0 & \dots & & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} \gamma_{-q}^q\\ \gamma_{-q+1}^q\\ \vdots\\ \vdots\\ \gamma_{q-2}^q\\ \gamma_{q-1}^q \end{bmatrix} = \begin{bmatrix} \beta_{-q}^{1,q}\\ \beta_{-q+1}^{1,q}\\ \vdots\\ \vdots\\ \vdots\\ \beta_{q-1}^{1,q}\\ \beta_q^{1,q} \end{bmatrix}. \qquad (5.27)$$

The subsystem formed by the first $2q$ equations has a lower triangular invertible matrix, hence the first $2q$ equations can be uniquely solved. Elimination of the elements in the subdiagonal from those in the diagonal yields the determinant of the matrix:

$$\det\begin{bmatrix} -1 & 0 & 0 & \dots & \dots & 0 & \beta_{-q}^{1,q}\\ 1 & -1 & 0 & \dots & \dots & 0 & \beta_{-q+1}^{1,q}\\ 0 & 1 & -1 & \dots & \dots & 0 & \vdots\\ \vdots & & \ddots & \ddots & & \vdots & \vdots\\ \vdots & & & \ddots & \ddots & \vdots & \vdots\\ 0 & \dots & \dots & 0 & 1 & -1 & \beta_{q-1}^{1,q}\\ 0 & \dots & & \dots & 0 & 1 & \beta_q^{1,q} \end{bmatrix} = (-1)^{2q}\sum_{l=-q}^{q}\beta_l^{1,q}.$$

By (5.24) with $r = 0$, $\sum_{l=-q}^{q} \beta_l^{1,q} = 0$, therefore system (5.27) has a unique solution.

From (5.6), (5.7) and (5.26) we deduce:

$$
\begin{aligned}
u_i^{n+1} &= u_i^n - \frac{\Delta t}{h}(\hat{f}^n_{i+\frac{1}{2}} - \hat{f}^n_{i-\frac{1}{2}}) - \sum_{l=1}^{R-1} \frac{(\Delta t)^{l+1}}{(l+1)!} \Delta_h^{1,\lceil \frac{R-l}{2} \rceil} \widetilde{f}^{(l)}_{i+\cdot,n} \\
&= u_i^n - \frac{\Delta t}{h}(\hat{f}^n_{i+\frac{1}{2}} - \hat{f}^n_{i-\frac{1}{2}}) \\
&- \sum_{l=1}^{R-1} \frac{(\Delta t)^{l+1}}{(l+1)!} \frac{1}{h} \Big(\sum_{s=-\lceil \frac{R-l}{2} \rceil}^{\lceil \frac{R-l}{2} \rceil - 1} \gamma_s^{\lceil \frac{R-l}{2} \rceil} \widetilde{f}^{(l)}_{i+s+1,n} - \sum_{s=-\lceil \frac{R-l}{2} \rceil}^{\lceil \frac{R-l}{2} \rceil - 1} \gamma_s^{\lceil \frac{R-l}{2} \rceil} \widetilde{f}^{(l)}_{i+s,n} \Big) \\
&= u_i^n - \frac{\Delta t}{h}(\hat{f}^n_{i+\frac{1}{2}} - \hat{f}^n_{i-\frac{1}{2}}) \\
&- \frac{\Delta t}{h} \sum_{l=1}^{R-1} \frac{(\Delta t)^{l}}{(l+1)!} \Big(\sum_{s=-\lceil \frac{R-l}{2} \rceil}^{\lceil \frac{R-l}{2} \rceil - 1} \gamma_s^{\lceil \frac{R-l}{2} \rceil} \widetilde{f}^{(l)}_{i+s+1,n} - \sum_{s=-\lceil \frac{R-l}{2} \rceil}^{\lceil \frac{R-l}{2} \rceil - 1} \gamma_s^{\lceil \frac{R-l}{2} \rceil} \widetilde{f}^{(l)}_{i+s,n} \Big)
\end{aligned}
$$

and we deduce equation (5.25) with

$$
\hat{g}^n_{i+\frac{1}{2}} = \hat{f}^n_{i+\frac{1}{2}} + \sum_{l=1}^{R-1} \frac{(\Delta t)^{l}}{(l+1)!} \sum_{s=-\lceil \frac{R-l}{2} \rceil}^{\lceil \frac{R-l}{2} \rceil - 1} \gamma_s^{\lceil \frac{R-l}{2} \rceil} \widetilde{f}^{(l)}_{i+s+1,n}. \tag{5.28}
$$

□

Remark 1. *From* (5.6) *and* (5.21) *we may deduce that for each* $i = 0, \ldots, M-1$, *the computation of the coefficients* $\widetilde{u}^{(l)}_{i,n}$, *for* $l = 2, \ldots, k$, *requires* $\frac{R^2}{2} + \mathcal{O}(R)$ *flux evaluations, and* $\frac{3R^2}{2} + \mathcal{O}(R)$ *floating point operations. Therefore, the time step can be performed with one extra WENO reconstruction for* $\widetilde{u}^{(1)}_{i,n}$ *and about* $2R$ *more floating point operations to evaluate the polynomial in* (5.7).

5.2 Fluctuation control

Now we focus on the computation of the approximate nodal values of the first order time derivative. Tipically, one would simply take the approximations obtained through the upwinded reconstruction procedure in the

Shu-Osher's finite difference approach, that is,

$$\widetilde{u}^{(1)}_{j,n} = -\frac{\hat{f}_{j+\frac{1}{2},n} - \hat{f}_{j-\frac{1}{2},n}}{h}.$$

In fact, this is what it was done in the Qiu-Shu work [39]. However, taking directly these values as the first derivative used to compute the next derivatives through weightless procedures, namely, assuming that the data is smooth, is not safe because the data is not actually smooth; in fact, it will include $\mathcal{O}(h^{-1})$ terms wherever there is a discontinuity, which we will call from now on *fluctuations*. These terms will appear provided $\hat{f}_{j-\frac{1}{2},n}$ and $\hat{f}_{j+\frac{1}{2},n}$ come from different sides of a discontinuity (or some of them has mixed information of both sides due to a previous flux splitting procedure to reconstruct the interface values), since in that case $\hat{f}_{j+\frac{1}{2},n} - \hat{f}_{j-\frac{1}{2},n} = \mathcal{O}(1)$.

Such fluctuations are necessary to make a discontinuity move at the right speed, corresponding to an upwind procedure, and thus must be used as a first order term of the Taylor expansion to advance in time; but if we want to compute the next time derivatives using smooth data, we have to find an alternative approximation of the first derivative with fully smooth data, and this is what is going to be discussed in this section.

As stated previously, if careful enough control is not performed and we neglect the fact that big values are generated at the discontinuities, the scheme may turn too dissipative, or even unstable, in terms of severe CFL restrictions or even unconditional failure of the numerical scheme under some circumstances. From now on we will omit the time dependence, since in this section we only focus on spatial affairs.

Let us first clarify why this phenomena happens and what are exactly the interfaces and cells involved, assuming we have an entirely sharp discontinuity. Assume the nodes x_i and x_{i+1} contain a discontinuity between them, so that $u_{i+1} - u_i = \mathcal{O}(1)$. Then, denoting $f_j = f(u_j)$ and assuming $f(u_L) \neq f(u_R)$, with u_L and u_R the left and right states of the discontinuity, respectively, we have $f_{i+1} - f_i = \mathcal{O}(1)$ as well.

Now, the resulting numerical fluxes at the interfaces depend on the reconstruction technique used. In any case, the value $\hat{f}_{i-\frac{1}{2}}$ is reconstructed using essentially only the data at the left side of the discontinuity, while the value $\hat{f}_{i+\frac{3}{2}}$ is reconstructed through essentially only the data at its right side. As for the central interface value $\hat{f}_{i+\frac{1}{2}}$, if the upwind technique is based on an spectral decomposition, then its value will be based on the left data if the eigenvalues both at x_i and x_{i+1} are positive, the right side if both are negative or mixed information from both sides otherwise or, more generally, if the upwind is based on a flux-splitting technique. The last case can be considered the worst case scenario and will generate two fluctuations, since

$$\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}} = \mathcal{O}(1),$$

$$\hat{f}_{i+\frac{3}{2}} - \hat{f}_{i+\frac{1}{2}} = \mathcal{O}(1),$$

where the values both at left at right neighborhood, assuming there is no other discontinuity close, are $\mathcal{O}(h)$.

Hence we have

$$u_i^{(1)} = -\frac{\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}}{h} = \mathcal{O}(h^{-1}),$$

$$u_{i+1}^{(1)} = -\frac{\hat{f}_{i+\frac{3}{2}} - \hat{f}_{i+\frac{1}{2}}}{h} = \mathcal{O}(h^{-1}),$$

where the values both at left and right cell neighborhoods are $\mathcal{O}(1)$.

Therefore, if one works with this approximation of the first time derivative in order to compute approximations of the next time derivatives, these $\mathcal{O}(h^{-1})$ terms will be dragged to the next derivatives and, what is worse, even more cells will be contaminated with such incorrect information; the longer the stencils for the central differences are, the more cell values will be corrupted.

In practice, this implies that the k-th derivative, $1 \leq k \leq R$, will have terms of magnitude $\mathcal{O}(h^{-k})$, therefore, the term which appears on the Taylor expansion term, which is multiplied by $\frac{\Delta t^k}{k!}$, a term of magnitude $\mathcal{O}(h^k)$, will be ultimatelly $\mathcal{O}(1)$. That is, each derivative will include $\mathcal{O}(1)$ terms at each time integration, the more terms of that kind as the longer the degree of the derivative is, which results in undesired diffusion, oscillations or even a complete failure of the scheme in some cases.

As Qiu and Shu stated in [39], there is no apparent need to control these spurious oscillations, since in the experiments shown therein they end up stabilized at the ending time. However, we have noticed several failures in experiments such as the Shu-Osher problem in 1D or the double mach reflection test in 2D using for instance the Donat-Marquina [11] upwind reconstruction scheme, yielding severe CFL stability restrictions when, in particular, high order linearizations are used to reconstruct data at the cell interfaces in order to compute the two sided spectral decompositions for the flux Jacobians.

All the reasons stated above motivate the need of computing an alternative nodal approximation of the first derivative with fully smooth data, that we will call $\widetilde{\widetilde{u}}_j^{(1)}$, which will be used only to compute the next derivatives, but not as a first order term of the Taylor expansion to advance in time, where the Shu-Osher conservative upwind approximation of the first derivative, $\widetilde{u}_j^{(1)} = -\dfrac{\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}}{h}$, will be used instead in order to preserve the upwind features of our scheme (and thus the stability). We detail below the whole procedure to compute the alternative approximation of the first derivative, to be used only to obtain the approximations of the next derivatives.

5.2.1 Central WENO reconstructions

Let us assume that our spatial scheme is $(2r-1)$-th order accurate WENO. After all the operations performed for the reconstruction of the interfaces, the stencil of points that is used in order to approximate the derivative at the node x_i is the following set of $2r+1$ points:

$$\{x_{i-r}, \dots, x_i, \dots, x_{i+r}\}, \tag{5.29}$$

whose corresponding flux values, $f_j = f(u_j)$, are

$$\{f_{i-r}, \dots, f_i, \dots, f_{i+r}\}.$$

The procedure that we next expound only uses information from the stencil

$$\mathcal{S}_{i+r-1}^{2r-1} := \{i-r+1, \dots, i, \dots, i+r-1\}, \tag{5.30}$$

thus ignoring the flux values f_{i-r}, f_{i+r} at the edges of the stencil in (5.29).

For fixed i, let q_k^r be the interpolating polynomial of degree $\leq r-1$ such that $q_k^r(x_j) = f_j$, $j \in \mathcal{S}_{i+k}^r := \{i+k-r+1, \dots, i+k,\}$, $0 \leq k \leq r-1$. After the previous discussion, our goal is to obtain an approximation of the flux derivative $f(u)_x(x_i)$ from the stencil $\mathcal{S}_{i+r-1}^{2r-1}$ which is $(2r-1)$-th order accurate if the nodes in the stencil lie within a smoothness region for u or is $\mathcal{O}(1)$ otherwise. We use Weighted Essentially Non Oscillatory techniques to achieve this purpose.

Lemma 3. *There exists a set of constants* $\{c_k^r\}_{k=1}^r$ *satisfying* $0 < c_k^r < 1$, *for* $0 \leq k \leq r-1$, $\sum_{k=0}^{r-1} c_k^r = 1$, *such that*

$$\sum_{k=0}^{r-1} c_k^r (q_k^r)'(x_i) = (q_{r-1}^{2r-1})'(x_i).$$

Proof. We show by induction on $s = 1, \dots, r-1$ that there exist $a_{p,l}^{r,s} \in (0,1)$, $p = s, \dots, r-1$, $l = 0, \dots, s$, such that $\sum_{l=0}^{s} a_{p,l}^{r,s} = 1$ and

$$\sum_{l=0}^{s} a_{p,l}^{r,s} (q_{p-l}^{r})'(x_i) = (q_p^{r+s})'(x_i), \tag{5.31}$$

the stated result being the final case $s = p = r-1$, for $c_k^r = a_{r-1,r-1-k}^{r,r-1}$.

The case $s = 1$ is obtained by Neville's algorithm and the fact that $q_p^r(x_i) = q_{p-1}^r(x_i) = f_i$, as long as $1 \le p \le r-1$:

$$\begin{aligned} q_p^{r+1}(x) &= \frac{q_p^r(x)(x - x_{i+p-r}) - q_{p-1}^r(x)(x - x_{i+p})}{x_{i+p} - x_{i+p-r}}, \\ (q_p^{r+1})'(x_i) &= \frac{(q_p^r)'(x_i)(x_i - x_{i+p-r}) - (q_{p-1}^r)'(x_i)(x_i - x_{i+p}) + q_p^r(x_i) - q_{p-1}^r(x_i)}{x_{i+p} - x_{i+p-r}}, \\ (q_p^{r+1})'(x_i) &= \frac{r-p}{r}(q_p^r)'(x_i) + \frac{p}{r}(q_{p-1}^r)'(x_i), \end{aligned}$$

so the result for $s = 1$ is obtained.

Assume now that (5.31) holds and aim to prove it for $s+1$, assuming $s+1 \le p \le r-1$. To achieve this, we use again Neville's algorithm and the previous argument to get

$$(q_p^{r+s+1})'(x_i) = \frac{r+s-p}{r+s}(q_p^{r+s})'(x_i) + \frac{p}{r+s}(q_{p-1}^{r+s})'(x_i).$$

The induction hypothesis now yields:

$$\begin{aligned} (q_p^{r+s+1})'(x_i) &= \frac{r+s-p}{r+s}\sum_{l=0}^{s} a_{p,l}^{r,s}(q_{p-l}^r)'(x_i) + \frac{p}{r+s}\sum_{l=0}^{s} a_{p-1,l}^{r,s}(q_{p-1-l}^r)'(x_i) \\ &= \sum_{l=0}^{s+1} a_{p,l}^{r,s+1}(q_{p-l}^r)'(x_i), \end{aligned}$$

for the coefficients given by:

$$a_{p,l}^{r,s+1} = \begin{cases} \frac{r+s-p}{r+s} a_{p,0}^{r,s} & l = 0 \\ \frac{r+s-p}{r+s} a_{p,l}^{r,s} + \frac{p}{r+s} a_{p-1,l-1}^{r,s} & l = 1, \dots, s \\ \frac{p}{r+s} a_{p-1,s}^{r,s} & l = s+1, \end{cases}$$

which clearly satisfy $a_{p,l}^{r,s+1} \in (0,1)$ and, by induction, also satisfy

$$\sum_{l=0}^{s+1} a_{p,l}^{r,s+1} = \frac{r+s-p}{r+s}\sum_{l=0}^{s} a_{p,l}^{r,s} + \frac{p}{r+s}\sum_{l=0}^{s} a_{p-1,l}^{r,s} = \frac{r+s-p}{r+s} + \frac{p}{r+s} = 1.$$

□

If $f_j = f(u(x_j, t_n))$, for smooth enough u and fix t_n, then

$$(q_k^r)'(x_i) = f(u)_x(x_i, t_n) + d_k^r(x_i)h^{r-1} + \mathcal{O}(h^r), k = 0, \dots, r-1, \tag{5.32}$$

$$(q_{r-1}^{2r-1})'(x_i) = f(u)_x(x_i, t_n) + d_{r-1}^{2r-1}(x_i)h^{2r-2} + \mathcal{O}(h^{2r-1}). \tag{5.33}$$

for continuously differentiable d_k^r, d_{r-1}^{2r-1}. The goal is to obtain the accuracy in (5.33) by a suitable nonlinear convex combination of (5.32)

$$\sum_{k=0}^{r-1} w_k^r (q_k^r)'(x_i) = f(u)_x(x_i, t) + \widetilde{d}_{r-1}^{2r-1}(x_i)h^{2r-2} + \mathcal{O}(h^{2r-1}), \tag{5.34}$$

where $w_k^r = c_k^r(1 + \mathcal{O}(h^{r-1})$ if the whole stencil $x_{i-r+1}, \dots, x_{i+r-1}$ lies within a smoothness region for u and $w_k^r = \mathcal{O}(h^{r-1})$ if the k-th stencil crosses a discontinuity and there are at least another stencil which does not. We follow Weighted Essentially Non Oscillatory classical techniques [35, 26]. From now on we drop the superscript r in q_k^r.

Furthermore, we need the approximation in (5.34) to be in conservation form. To achieve this we use the polynomial p_k of degree $r-1$ satisfying

$$\frac{1}{h}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} p_k(x)dx = f_j, \quad i-r+1+k \leq j \leq i+k, \quad 0 \leq k \leq r-1,$$

and $\widetilde{p}_k(x)$ a primitive of it. Then the polynomial

$$\widetilde{q}_k(x) = \frac{\widetilde{p}_k(x + \frac{h}{2}) - \widetilde{p}_k(x - \frac{h}{2})}{h},$$

can be seen to have degree $\leq r-1$ and $\widetilde{q}_k(x_j) = f_j$, $j = i-r+1+k, \dots, i+k$, and must therefore be q_k. It therefore follows that

$$q_k'(x_j) = \frac{(\widetilde{p}_k)'(x_{j+\frac{1}{2}}) - (\widetilde{p}_k)'(x_{j-\frac{1}{2}})}{h} = \frac{p_k(x_{j+\frac{1}{2}}) - p_k(x_{j-\frac{1}{2}})}{h}.$$

Now, let us define the following Jiang-Shu smoothness indicators using the definition of p_k:

$$I_k = \sum_{\ell=1}^{r-1} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} h^{2\ell-1} p_k^{(\ell)}(x)^2 dx, \quad 0 \leq k \leq r-1 \tag{5.35}$$

so that we can define the weights as follows:

$$\omega_k = \frac{\alpha_k}{\sum_{l=1}^{r} \alpha_l}, \quad \alpha_k = \frac{c_k}{(I_k + \varepsilon)^m}, \tag{5.36}$$

with $\varepsilon > 0$ a small positive quantity, possibly depending on h. Following the techniques in [2], since $p_k^{(\ell)} - p_j^{(\ell)} = \mathcal{O}(h^{r-\ell})$ at regions of smoothness, whereas $\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} (p_k')^2 dx = \mathcal{O}(h^{-2})$ if the corresponding stencil $\mathcal{S}_k^r$ crosses a discontinuity, the smoothness indicators satisfy $I_k - I_j = \mathcal{O}(h^{r+1})$ at regions of smoothness and $I_k \not\to 0$ if the k-th stencil crosses a discontinuity. Therefore, the definition (5.36) satisfies the requirements mentioned above in order to achieve maximal order even at smooth extrema, if the parameter $\varepsilon > 0$, besides avoiding divisions by zero, is chosen as $\varepsilon = \lambda h^2$, with $\lambda \sim f(u)_x$, and the exponent m in (5.36) makes the weight $\omega_k = \mathcal{O}(h^{r-1})$ if there is a discontinuity at that stencil. Since one wants to attain the maximal possible order in such case, which corresponds to the value interpolated from a smooth substencil, which is $\mathcal{O}(h^r)$, then it suffices to set $m = \lceil \frac{r}{2} \rceil$.

Finally, we define $\widetilde{\widetilde{u}}_{i,n}^{(1)}$, the smoothened approximation of $u_t(x_i, t_n)$, as the result of the following convex combination:

$$\widetilde{\widetilde{u}}_{i,n}^{(1)} = -\sum_{k=1}^{r} \omega_k q_k'(x_i).$$

5.3 Boundary conditions

In order to keep a stencil of $M+1$ points after each time integration, we need to impose boundary conditions and fill r ghost cells at both sides of each line.

In case of LW/LWA schemes, we need to fill these amount of ghost cells after each computation of the successive time derivatives of u. Depending on the nature of the boundary condition, we will have to proceed in one way or another. So, in case of outflow or periodic boundary conditions, there are no changes in the way we impose the boundary condition in the nodes approximating each derivative.

However, in case of time dependent inflow boundary conditions, given by $g(t)$, if we work with the k-th order derivative nodes we will have to use the inflow condition $g^{(k)}(t)$.

The techniques used for high order numerical boundary conditions have been studied in the literature and can be found in works such as [45], [46], [4] and [5].

Note that the domain of dependence for a fifth order WENO spatial scheme with a third order Lax-Wendroff type time discretization is composed by 9 points at most, meanwhile the domain of dependence has a total of at most 17 points if a fifth order Lax-Wendroff type time discretization is used instead. On the other hand, the domain of dependence for a RK3 time scheme has a length of at most 19 points. The upper bound is always achieved when a flux splitting technique, such as LLF, is used. If the fluctuation control with central WENO reconstructions is used on the Lax-Wendroff technique, the number of points in the domain of dependence is reduced by 2 units.

5.4 Numerical scheme examples

For the sake of illustration, we next detail the specific numerical scheme we use for the numerical experiments that will be shown in the next section, detailing how the aforementioned recursive procedure is performed. In order to simplify as most as possible the notation, we only show it for the scalar 1D case, as in the case of systems it consists on working through components and the multidimensional case working through each line and respective flux, yielding a rather simple generalization. We use a fifth order accurate in space scheme ($r = 3$), with fifth order accurate time discretizations ($R = 5$), yielding a fifth order accurate scheme.

The scheme to obtain the upwinded approximation of the first derivative, $u^{(1)}$, is based on the Shu-Osher finite differences of cell interfaces,

$$\widetilde{u}_{i,n}^{(1)} = -\frac{\hat{f}_{i+\frac{1}{2},n} - \hat{f}_{i-\frac{1}{2},n}}{h}, \quad 0 \leq i < M,$$

where each interface value $\hat{f}_{i-\frac{1}{2},n}$, $0 \leq i < M$, is computed through upwinded fifth order WENO reconstructions. In order to obtain the last three approximations from both corners we need three additional ghost cell values at each side of the stencil, $u_{-3,n}$, $u_{-2,n}$, $u_{-1,n}$ and $u_{M,n}$, $u_{M+1,n}$, $u_{M+2,n}$, which are obtained through the suitable numerical boundary conditions, involving the computational domain and the boundary conditions themselves, if any.

Below it is expounded how to obtain the next derivatives through the flux approximation procedure.

First, we compute $v_i^{(1)}$, $0 \leq i < M$, whose value depends on the usage or not of the fluctuation control:

$$v_i^{(1)} = \begin{cases} \widetilde{\widetilde{u}}_i^{(1)} & \text{if fluctuation control,} \\ \widetilde{u}_i^{(1)} & \text{if no fluctuation control,} \end{cases}$$

where $\widetilde{\widetilde{u}}_i^{(1)}$ are the smoothened first derivative approximation cell values through the technique expounded in Section 5.2.1, where, following the notation therein, we have the following sided approximations of the first time derivative at x_i,

$$\begin{aligned} (q_0^3)'(x_i) &= \frac{f_{i-2} - 4f_{i-1} + 3f_i}{2h}, \\ (q_1^3)'(x_i) &= \frac{f_{i+1} - f_{i-1}}{2h}, \\ (q_2^3)'(x_i) &= \frac{-3f_i + 4f_{i+1} - f_{i+2}}{2h}, \end{aligned}$$

with the corresponding ideal weights

$$c_0 = \frac{1}{6}, \quad c_1 = \frac{2}{3}, \quad c_2 = \frac{1}{6},$$

where the associate smoothness indicators are

$$\begin{aligned} I_0 &= \frac{f_{i-2}(4f_{i-2} - 19f_{i-1} + 11f_i) + f_{i-1}(25f_{i-1} - 31f_i) + 10f_i^2}{3}, \\ I_1 &= \frac{f_{i-1}(4f_{i-1} - 13f_i + 5f_{i+1}) + f_i(13f_i - 13f_{i+1}) + 4f_{i+1}^2}{3}, \\ I_2 &= \frac{f_i(10f_i - 31f_{i+1} + 11f_{i+2}) + f_{i+1}(25f_{i+1} - 19f_{i+2}) + 4f_{i+2}^2}{3}. \end{aligned}$$

Then we define

$$\widetilde{\widetilde{u}}_i^{(1)} = \omega_0 q_0'(x_i) + \omega_1 q_1'(x_i) + \omega_2 q_2'(x_i),$$

where

$$\omega_k = \frac{\alpha_k}{\alpha_0 + \alpha_1 + \alpha_2}, \quad \alpha_k = \frac{c_k}{(I_k + \varepsilon)^2}, \quad 0 \leq k \leq 2,$$

with $\varepsilon = 10^{-100}$.

Once the above step is performed, we compute nodal approximations of the second order time derivative of u, $\widetilde{u}^{(2)}$, by performing the following operation:

$$\begin{aligned} \widetilde{f}_{i,n}^{(1)} &= \Delta_\delta^{1,\lceil\frac{5-1}{2}\rceil}\big(G_{0,\delta}\big(f(T_1[h,i,n])\big)\big) = \Delta_\delta^{1,2}\big(G_{0,\delta}\big(f(T_1[h,i,n])\big)\big) \\ &= \frac{\varphi_{i,n}^1(2\delta) - 8\varphi_{i,n}^1(\delta) + 8\varphi_{i,n}^1(-\delta) - \varphi_{i,n}^1(-2\delta)}{12\delta}, \quad -2 \leq i < M+2, \end{aligned}$$

where

$$\varphi^1_{i,n}(\rho) = f(\widetilde{u}^n_i + \rho v^{(1)}_{i,n}).$$

We need thus to apply previously boundary conditions in order to obtain two ghost cell values at both sides, $v^{(1)}_{-2}, v^{(1)}_{-1}$ and $v^{(1)}_{M}, v^{(1)}_{M+1}$.

We can now define the approximation of the second time derivative of u as the following fourth order accurate central divided difference

$$\widetilde{u}^{(2)}_i = -\frac{\widetilde{f}^{(1)}_{i-2,n} - 8\widetilde{f}^{(1)}_{i-1,n} + 8\widetilde{f}^{(1)}_{i+1,n} - \widetilde{f}^{(1)}_{i+2,n}}{12h}.$$

The nodal approximations of $u^{(3)}$ are obtained in a similar fashion:

$$\begin{aligned}\widetilde{f}^{(2)}_{i,n} &= \Delta^{2,\lceil\frac{5-2}{2}\rceil}_{\delta}\big(G_{0,\delta}\big(f(T_2[h,i,n])\big)\big) = \Delta^{2,2}_{\delta}\big(G_{0,\delta}\big(f(T_2[h,i,n])\big)\big)\\ &= \frac{-\varphi^2_{i,n}(2\delta) + 16\varphi^2_{i,n}(\delta) - 30\varphi^2_{i,n}(0) + 16\varphi^2_{i,n}(-\delta) - \varphi^2_{i,n}(-2\delta)}{12\delta^2},\\ &-2 \le i < M+2,\end{aligned}$$

where

$$\varphi^2_{i,n}(\rho) = f(\widetilde{u}^n_i + \rho v^{(1)}_{i,n} + \frac{\rho^2}{2}\widetilde{u}^{(2)}_{i,n}),$$

by previously having computed through boundary conditions $\widetilde{u}^{(2)}_{-2}, \widetilde{u}^{(2)}_{-1}$ and $\widetilde{u}^{(2)}_{M}, \widetilde{u}^{(2)}_{M+1}$ and, again, use a fourth order central divided difference to approximate the third time derivative of u:

$$\widetilde{u}^{(3)}_i = -\frac{\widetilde{f}^{(2)}_{i-2} - 8\widetilde{f}^{(2)}_{i-1} + 8\widetilde{f}^{(2)}_{i+1} - \widetilde{f}^{(2)}_{i+2}}{12h}.$$

Since for the fourth and fifth time derivative of u it is only required approximations of second and first order, respectively, we approximate the corresponding time derivatives of the flux through second order central differences and perform as well second order central differences between them.

On the one hand, we have

$$\begin{aligned}\widetilde{f}^{(3)}_{i,n} &= \Delta^{3,\lceil\frac{5-3}{2}\rceil}_{\delta}\big(G_{0,\delta}\big(f(T_3[h,i,n])\big)\big) = \Delta^{3,1}_{\delta}\big(G_{0,\delta}\big(f(T_3[h,i,n])\big)\big)\\ &= \frac{\varphi^3_{i,n}(2\delta) - 2\varphi^3_{i,n}(\delta) + 2\varphi^3_{i,n}(-\delta) - \varphi^3_{i,n}(-2\delta)}{2\delta^3}, \quad -1 \le i < M+1,\end{aligned}$$

with

$$\varphi^3_{i,n}(\rho) = f(\widetilde{u}^n_i + \rho v^{(1)}_{i,n} + \frac{\rho^2}{2}\widetilde{u}^{(2)}_{i,n} + \frac{\rho^3}{6}\widetilde{u}^{(3)}_{i,n}),$$

where $\widetilde{u}_{-1}^{(3)}$ and $\widetilde{u}_{M}^{(3)}$ have been computed using the adequate numerical boundary conditions.

Then we define

$$\widetilde{u}_{i,n}^{(4)} = -\frac{\widetilde{f}_{i+1,n}^{(3)} - \widetilde{f}_{i-1,n}^{(3)}}{2h}.$$

On the other hand,

$$\begin{aligned}\widetilde{f}_{i,n}^{(4)} &= \Delta_{\delta}^{4,\lceil\frac{5-4}{2}\rceil}\big(G_{0,\delta}\big(f(T_4[h,i,n])\big)\big) = \Delta_{\delta}^{4,1}\big(G_{0,\delta}\big(f(T_4[h,i,n])\big)\big)\\ &= \frac{\varphi_{i,n}^{4}(2\delta) - 4\varphi_{i,n}^{4}(\delta) + 6\varphi_{i,n}^{4}(0) - 4\varphi_{i,n}^{4}(-\delta) + \varphi_{i,n}^{4}(-2\delta)}{\delta^4}, \quad -1 \le i < M+1,\end{aligned}$$

with

$$\varphi_{i,n}^{4}(\rho) = f(\widetilde{u}_i^n + \rho v_{i,n}^{(1)} + \frac{\rho^2}{2}\widetilde{u}_{i,n}^{(2)} + \frac{\rho^3}{6}\widetilde{u}_{i,n}^{(3)} + \frac{\rho^4}{24}\widetilde{u}_{i,n}^{(4)}),$$

where $\widetilde{u}_{-1}^{(4)}$ and $\widetilde{u}_{M}^{(4)}$ are obtained through numerical boundary conditions.

We then define

$$\widetilde{u}_{i,n}^{(5)} = -\frac{\widetilde{f}_{i+1,n}^{(4)} - \widetilde{f}_{i-1,n}^{(4)}}{2h}.$$

The next time step is then computed through the fifth order Taylor expansion replacing the derivatives with their corresponding approximations:

$$\widetilde{u}_i^{n+1} = \widetilde{u}_i^n + \Delta t\widetilde{u}_{i,n}^{(1)} + \frac{\Delta t^2}{2}\widetilde{u}_{i,n}^{(2)} + \frac{\Delta t^3}{6}\widetilde{u}_{i,n}^{(3)} + \frac{\Delta t^4}{24}\widetilde{u}_{i,n}^{(4)} + \frac{\Delta t^5}{120}\widetilde{u}_{i,n}^{(5)}.$$

5.5 Alternative approach

In this section we present an alternative approach for time discretizations pursuing the same idea consisting on avoiding the computation of large terms for high order time derivatives, as expounded in the original work of Qiu and Shu. The following result provides a relationship involving unmixed spatial and time derivatives for 1D scalar conservation laws.

Theorem 4. *For any 1D scalar conservation law*

$$u_t + f(u)_x = 0$$

and $\forall n \in \mathbb{N}$ such that the function $(f')^n$ admits antiderivative, it holds

$$\frac{\partial^n u}{\partial t^n} = \frac{\partial^n F_n(u)}{\partial x^n},$$

where $F_n(u)$ is a primitive of the function $(-1)^n f'(u)^n$, namely, F_n is a real function such that $F_n'(u) = (-1)^n f'(u)^n$.

Proof. We proceed by induction on n.

For $n = 1$ we have $F_1'(u) = -f'(u)$, being $F_1(u) = -f(u)$ a primitive. Therefore, the result is true since by the conservation law itself it holds that

$$\frac{\partial u}{\partial t} = -\frac{\partial f(u)}{\partial x} = \frac{\partial F_1(u)}{\partial x}.$$

Let us now assume that the result is true for n and it will be proven for $n+1$. Indeed,

$$\begin{aligned}\frac{\partial^{n+1} u}{\partial t^{n+1}} &= \frac{\partial}{\partial t}\left[\frac{\partial^n u}{\partial t^n}\right] = \frac{\partial}{\partial t}\left[\frac{\partial^n F_n(u)}{\partial x^n}\right] = \frac{\partial^n}{\partial x^n}\left[F_n(u)_t\right] = \frac{\partial^n}{\partial x^n}\left[F_n'(u)u_t\right]\\ &= \frac{\partial^n}{\partial x^n}\left[(-1)^n f'(u)^n u_t\right] = \frac{\partial^n}{\partial x^n}\left[(-1)^n f'(u)^n(-f(u)_x)\right]\\ &= \frac{\partial^n}{\partial x^n}\left[(-1)^{n+1} f'(u)^n f(u)_x\right] = \frac{\partial^n}{\partial x^n}\left[(-1)^{n+1} f'(u)^n f'(u)u_x\right]\\ &= \frac{\partial^n}{\partial x^n}\left[(-1)^{n+1} f'(u)^{n+1} u_x\right].\end{aligned}$$

Let $F_{n+1}(u)$ be a function such that $F_{n+1}'(u) = (-1)^{n+1} f'(u)^{n+1}$. Then $F_{n+1}(u)_x = F_{n+1}'(u)u_x = (-1)^{n+1} f'(u)^{n+1} u_x$. Hence,

$$\frac{\partial^{n+1} u}{\partial t^{n+1}} = \frac{\partial^n}{\partial x^n}\left[(-1)^{n+1} f'(u)^{n+1} u_x\right] = \frac{\partial^n}{\partial x^n}\left[F_{n+1}(u)_x\right] = \frac{\partial^{n+1} F_{n+1}(u)}{\partial x^{n+1}}.$$

□

Remark 2. *When computing the primitive, the integration constant does not affect the procedure since at least one spatial derivative is computed for $F_n(u)$ in the equation (to be exact, n, $n \geq 1$), where it vanishes.*

Example 1. *For the linear advection equation*

$$u_t + u_x = 0$$

we have $f(u) = u$ and thus $f'(u) = 1$. Therefore

$$F_n(u) = \int (-1)^n f'(u)^n du = \int (-1)^n 1^n du = (-1)^n u.$$

Example 2. *For the Burgers equation*

$$u_t + f(u)_x = 0,$$

where $f(u) = \frac{1}{2}u^2$*, we have* $f'(u) = u$*. Hence*

$$F_n(u) = \int (-1)^n f'(u)^n du = (-1)^n \int u^n du = \frac{(-1)^n}{n+1} u^{n+1}.$$

This formulation can be useful to build easily high order numerical schemes with a small domain of dependence for 1D scalar conservation laws and cheap in terms of computational cost. Unfortunately, this previous result cannot be generalized to systems unless very restrictive compatibility conditions are hold. Moreover, the extension to 2D would involve crossed derivatives, which would suppose an additional and undesirable computational load.

5.6 Numerical experiments

In this section we present some 1D and 2D experiments both for scalar and system of equations involving comparisons of the fifth order both in space ($r = 3$) and time ($R = 2r - 1 = 5$) exact and approximate Lax-Wendroff schemes, together with the results obtained using the third order TVD Runge-Kutta time discretization.

From now on we will refer as WENO[]-LW[] to the exact Lax-Wendroff procedure, WENO[]-LWA[] to the approximate Lax-Wendroff procedure, WENO[]-LWF[] if a fluctuation control is used in the exact procedure, WENO[]-LWAF[] if the fluctuation control comes together with the approximate procedure and WENO[]-RK[] when a Runge-Kutta method is used. In each case, the first bracket stands for the value of the spatial accuracy order and the second one for the time accuracy order.

5.6.1 One-dimensional experiments

We start with some tests involving 1D conservation laws.

1D Linear advection equation

Smooth initial conditions: We set as initial condition $u(x,0) = 0.25 + 0.5\sin(\pi x)$, periodic boundary conditions at both sides, which leads to a problem wose exact solution is $u(x,t) = 0.25 + 0.5\sin(\pi(x-t))$, using both the exact and approximate Lax-Wendroff procedure (without and with fluctuation control) with fifth order accuracy both in space and time (WENO5-LW5, WENO5-LWA5 and WENO5-LWAF5, respectively) and run the simulation up to $t = 1$, with CFL $= 0.5$ (except for RK3, where we set $k = h^{\frac{5}{3}}$ in order to achieve fifth order accuracy in time), and for resolutions $n = 20 \cdot 2^n$ points, $1 \leq n \leq 5$, obtaining the results shown in Tables 5.1-5.4.

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
40	1.13E−5	–	2.39E−5	–
80	3.49E−7	5.02	7.17E−7	5.06
160	1.09E−8	5.00	2.25E−8	4.99
320	3.41E−10	5.00	6.77E−10	5.06
640	1.15E−11	4.89	2.23E−11	4.93
1280	3.51E−12	1.71	8.32E−12	1.42

Table 5.1: Error table for linear advection equation, $t = 1$. WENO5-RK3.

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
40	1.09E−5	–	2.37E−5	–
80	3.29E−7	5.05	7.00E−7	5.08
160	1.02E−8	5.01	2.21E−8	4.98
320	3.19E−10	5.00	6.65E−10	5.06
640	9.96E−12	5.00	2.02E−11	5.04
1280	3.12E−13	4.99	6.12E−13	5.04

Table 5.2: Error table for linear advection equation, $t = 1$. WENO5-LW5.

From the results, we can conclude that all the proposed schemes achieve the fifth order accuracy. We must remark that the loss of accuracy appreciable in the last row of the RK3 version with $\Delta t = h^{\frac{5}{3}}$ is due to accumulation of machine errors because of a major number of required iterations (produced by the time-space re-scaling performed to achieve the fifth order accuracy). On the other hand, the results obtained through the approximated scheme, WENO5-LWA5, are almost identical than those obtained through the original version, WENO5-LW5, as

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
40	1.09E-5	$-$	2.37E-5	$-$
80	3.29E-7	5.05	7.00E-7	5.08
160	1.02E-8	5.01	2.21E-8	4.98
320	3.19E-10	5.00	6.65E-10	5.06
640	9.96E-12	5.00	2.02E-11	5.04
1280	3.12E-13	5.00	6.12E-13	5.04

Table 5.3: Error table for linear advection equation, $t = 1$. WENO5-LWA5.

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
40	5.17E-6	$-$	1.27E-5	$-$
80	1.46E-7	5.15	3.87E-7	5.04
160	4.34E-9	5.07	1.09E-8	5.14
320	1.33E-10	5.03	3.44E-10	4.99
640	4.13E-12	5.01	1.04E-11	5.05
1280	1.31E-13	4.98	3.02E-13	5.10

Table 5.4: Error table for linear advection equation, $t = 1$. WENO5-LWAF5.

should be expected, since in this case (linear flux) both the exact and the approximate formulation yield theoretically the same results. On the other hand, we can see that the fifth order accuracy is also achieved by the approximate Lax-Wendroff scheme with fluctuation control, WENO5-LWAF5, providing even more accurate results. One of the reasons may be the fact that in this case an essentially central approximation of the first derivative is used to compute the (also central) approximations of the next degree derivatives.

1D Burgers equation

Smooth initial conditions: We perform an accuracy test in this equation with the same setup (initial and boundary conditions as well as the spatial resolutions) as in the previous example, except that now we set the end time to $t = 0.3$ with CFL $= 0.5$. The results for WENO5-LW5, WENO5-LWA5 and WENO5-LWAF5 are presented in Tables 5.5-5.7.

In this case, we can see again that the fifth order accuracy is achieved and the errors both in $\|\cdot\|_1$ and $\|\cdot\|_\infty$ of the exact and approximate version are very close as well.

Discontinuous solution: If we now change the final time to $t = 12$,

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
40	2.38E-5	–	2.09E-4	–
80	7.93E-7	4.91	9.44E-6	4.47
160	2.45E-8	5.01	3.01E-7	4.97
320	7.48E-10	5.04	9.13E-9	5.05
640	2.32E-11	5.01	2.81E-10	5.02
1280	7.22E-13	5.00	8.69E-12	5.01

Table 5.5: Error table for Burgers equation, $t = 0.3$. WENO5-LW5.

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
40	2.38E-5	–	2.09E-4	–
80	7.94E-7	4.91	9.46E-6	4.47
160	2.46E-8	5.01	3.02E-7	4.97
320	7.50E-10	5.04	9.15E-9	5.05
640	2.32E-11	5.01	2.81E-10	5.02
1280	7.23E-13	5.00	8.71E-12	5.01

Table 5.6: Error table for Burgers equation, $t = 0.3$. WENO5-LWA5.

the wave breaks at $t = 1.1$ and a shock is then formed. We compare the WENO5-LW5 and WENO5-LWA5 techniques with WENO5-RK3, whose results are shown in Figure 5.1. We run this simulation using a resolution of $n = 80$ points.

One can conclude from the results shown in Figure 5.1 that even in the discontinuous case the approximate formulation results are quite close to those obtained through the exact version. It can be seen as well that the version with fluctuation control captures better the discontinuity and is less oscillatory around the discontinuity.

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
40	2.98E-5	–	3.01E-4	–
80	1.06E-6	4.82	1.24E-5	4.60
160	3.22E-8	5.04	4.19E-7	4.88
320	9.75E-10	5.05	1.29E-8	5.02
640	2.99E-11	5.03	3.96E-10	5.02
1280	9.26E-13	5.01	1.23E-11	5.01

Table 5.7: Error table for Burgers equation, $t = 0.3$. WENO5-LWAF5.

Euler equations

We now work with 1D Euler equations of gas dynamics (2.10).

Smooth solution: We set as initial conditions

$$\left\{\begin{array}{rcl} \rho(x,t) & = & 0.75+0.5\sin(\pi x) \\ \rho v(x,t) & = & 0.25+0.5\sin(\pi x) \\ E(x,t) & = & 0.75+0.5\sin(\pi x) \end{array}\right. , \quad x\in(-1,1),$$

and periodic boundary conditions for all the quantities. For $t=0.1$ the characteristic lines do not cross so that the solution remains smooth. We compute a reference solution at that time with a very fine mesh and perform an order test with WENO5-LW5, WENO5-LWA5 and WENO5-LWAF5 using CFL $=0.5$ for the same spatial resolutions as the previous examples.

Both the errors and order quantities presented in the tables are an average of the three unknowns of u. The obtained results are presented in Tables 5.8-5.10. In this case it can be seen that the results shown in the tables are identical at the accuracy in which the errors have been displayed. This again indicates that the approximate version provides essentially the same results than the exact version. For instance, the

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
40	2.98E-4	–	4.70E-3	–
80	3.36E-5	3.15	5.49E-4	3.10
160	1.60E-6	4.39	4.59E-5	3.58
320	5.53E-8	4.85	1.78E-6	4.69
640	1.76E-9	4.98	6.01E-8	4.89
1280	5.65E-11	4.96	1.84E-9	5.03

Table 5.8: Error table for 1D Euler equation, $t=0.1$. WENO5-LW5.

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
40	2.98E-4	–	4.70E-3	–
80	3.36E-5	3.15	5.49E-4	3.10
160	1.60E-6	4.39	4.59E-5	3.58
320	5.53E-8	4.85	1.78E-6	4.69
640	1.76E-9	4.98	6.01E-8	4.89
1280	5.65E-11	4.96	1.84E-9	5.03

Table 5.9: Error table for 1D Euler equation, $t=0.1$. WENO5-LWA5.

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
40	2.93E-4	$-$	4.89E-3	$-$
80	3.25E-5	3.17	5.95E-4	3.04
160	1.71E-6	4.25	4.69E-5	3.66
320	6.15E-8	4.80	1.92E-6	4.61
640	2.00E-9	4.94	6.31E-8	4.93
1280	6.47E-11	4.95	1.95E-9	5.01

Table 5.10: Error table for 1D Euler equation, $t = 0.1$. WENO5-LWAF5.

global average error ($\|\cdot\|_1$) including all three components and cells of the numerical solution at $t = 0.1$ at the resolution $n = 1280$ of the approximate flux approach with respect to the exact flux approach is 2.82E-15. It can be clearly seen that the fifth order is achieved as well by the scheme with fluctuation control.

Shu-Osher problem: We now consider the interaction with a Mach 3 shock and a sine wave. The spatial domain is now given by $\Omega := (-5, 5)$, with initial conditions

$$\begin{cases} \left.\begin{array}{rcl} \rho(x,t) & = & 3.857143 \\ v(x,t) & = & 2.629369 \\ p(x,t) & = & 10.33333 \end{array}\right\} & \text{if } x \leq -4 \\ \left.\begin{array}{rcl} \rho(x,t) & = & 1.0 + 0.2\sin(5x) \\ v(x,t) & = & 0 \\ p(x,t) & = & 1 \end{array}\right\} & \text{if } x > -4 \end{cases}$$

with left inflow and right outflow boundary conditions.

We run one simulation until $t = 1.8$ and compare the results obtained with WENO5-RK3, WENO5-LW5, WENO5-LWA5 and WENO5-LWAF5, $n = 400$ cells, CFL $= 0.5$ with a reference solution computed with 16000 grid points. The results are shown in Figure 5.2. We can see from the results that again the version with approximate fluxes yields essentially the same results than the version with exact fluxes and that the version equipped with a fluctuation control captures slightly better the shock.

Blast wave: Now the initial data is the following one, corresponding to the interaction of two blast waves:

$$u(x,0) = \begin{cases} u_L & 0 < x < 0.1, \\ u_M & 0.1 < x < 0.9, \\ u_R & 0.9 < x < 1, \end{cases}$$

where $\rho_L=\rho_M=\rho_R=1$, $v_L=v_M=v_R=0$, $p_L=10^3, p_M=10^{-2}, p_R=10^2$. Reflecting boundary conditions are set at $x=0$ and $x=1$, simulating a solid wall at both sides. This problem involves multiple reflections of shocks and rarefactions off the walls and many interactions of waves inside the domain. We use here the same node setup as in the previous tests.

We compute a reference solution, this time using a resolution of $n=16000$ points and compare the performance of the results setting $n=800$ with the WENO5-RK3, WENO5-LW5, WENO5-LWA5 and WENO5-LWAF5 schemes with CFL $=0.5$. The results of the density field are shown in Figure 5.3, where the conclusions are the same than those obtained in the previous experiments.

5.6.2 Two-dimensional experiments

To illustrate that these techniques work as well in the multidimensional case, we next show some results involving two-dimensional experiments.

2D Euler equations

We now show some experiments involving 2D Euler equations.

Smooth solution: In order to test the accuracy of our scheme in the general scenario of a multidimensional system of conservation laws, we perform a test using the 2D Euler equations with smooth initial conditions, given by

$$\begin{aligned} u_0(x,y) &= (\rho(x,y), v^x(x,y), v^y(x,y), E(x,y)) \\ &= \left(\frac{3}{4}+\frac{1}{2}\cos(\pi(x+y)), \frac{1}{4}+\frac{1}{2}\cos(\pi(x+y)),\right. \\ &\left.\frac{1}{4}+\frac{1}{2}\sin(\pi(x+y)), \frac{3}{4}+\frac{1}{2}\sin(\pi(x+y))\right), \end{aligned}$$

where $x\in\Omega=[-1,1]\times[-1,1]$, with periodic boundary conditions.

In order to perform the smoothness analysis, we compute a reference solution in a fine mesh and then compute numerical solutions for the resolutions $n\times n$, for $n=10\cdot 2^k$, $1\leq k\leq 5$, obtaining the results shown in Tables 5.11-5.13 at the time $t=0.025$ for CFL $=0.5$. We can see thus that our scheme achieves the desired accuracy even in the general scenario of a multidimensional system of conservation laws, which is

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
40	1.80E-5	$-$	2.74E-4	$-$
80	1.09E-6	4.05	1.80E-5	3.93
160	3.89E-8	4.80	7.36E-7	4.61
320	1.29E-9	4.92	2.49E-8	4.88
640	4.11E-11	4.97	8.07E-10	4.95
1280	1.23E-12	5.06	2.43E-11	5.06

Table 5.11: Error table for 2D Euler equation, $t = 0.025$. WENO5-LW5.

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
40	1.80E-5	$-$	2.74E-4	$-$
80	1.09E-6	4.05	1.80E-5	3.93
160	3.89E-8	4.80	7.36E-7	4.61
320	1.29E-9	4.92	2.49E-8	4.88
640	4.11E-11	4.97	8.07E-10	4.95
1280	1.23E-12	5.06	2.43E-11	5.06

Table 5.12: Error table for 2D Euler equation, $t = 0.025$. WENO5-LWA5.

consistent with our theoretical results. Also, we can see that again the results obtained through the approximate Lax-Wendroff procedure are almost the same than those obtained using the exact version.

Double Mach Reflection: This experiment uses the Euler equations to model a vertical right-going Mach 10 shock colliding with an equilateral triangle. By symmetry, this is equivalent to a collision with a ramp with a slope of 30 degrees with respect to the horizontal line.

For the sake of simplicity, we consider the equivalent problem in a rectangle, consisting on a rotated shock, whose vertical angle is $\frac{\pi}{6}$ rad. The domain we consider in this problem is the rectangle $\Omega = [0, 4] \times [0, 1]$,

n	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_\infty$	Order $\|\cdot\|_\infty$
40	2.63E-5	$-$	2.97E-4	$-$
80	1.58E-6	4.06	2.01E-5	3.89
160	6.66E-8	4.57	1.06E-6	4.24
320	2.33E-9	4.84	4.08E-8	4.70
640	7.60E-11	4.94	1.34E-9	4.93
1280	2.35E-12	5.02	4.06E-11	5.04

Table 5.13: Error table for 2D Euler equation, $t = 0.025$. WENO5-LWAF5.

whose initial conditions are

$$u_0(x,y) = \begin{cases} C_1 & y \leq \frac{1}{4} + \tan(\frac{\pi}{6})x, \\ C_2 & y > \frac{1}{4} + \tan(\frac{\pi}{6})x, \end{cases}$$

where

$$\begin{aligned} C_1 &= (\rho_1, v_1^x, v_1^y, E_1)^T = (8, 8.25\cos(\frac{\pi}{6}), -8.25\sin(\frac{\pi}{6}), 563.5)^T, \\ C_2 &= (\rho_2, v_2^x, v_2^y, E_2)^T = (1.4, 0, 0, 2.5)^T. \end{aligned}$$

We impose inflow boundary conditions, with value C_1, at the left side, $\{0\} \times [0,1]$, outflow boundary conditions both at $[0, \frac{1}{4}] \times \{0\}$ and $\{4\} \times [0,1]$, reflecting boundary conditions at $]\frac{1}{4}, 4] \times \{0\}$ and inflow boundary conditions at the upper side, $[0,4] \times \{1\}$, which mimics the shock at its actual traveling speed:

$$u(x,1,t) = \begin{cases} C_1 & x \leq \frac{1}{4} + \frac{1+20t}{\sqrt{3}}, \\ C_2 & x > \frac{1}{4} + \frac{1+20t}{\sqrt{3}}. \end{cases}$$

We run different simulations until $t = 0.2$ at a resolution of 2048×512 points for CFL $= 0.4$ and a different combination of techniques, involving WENO5-RK3, WENO5-LW5 and WENO5-LWA5.

The results are presented as a Schlieren plot of the turbulence zone and they are shown in Figure 5.4. From Figure 5.4 it can be concluded that the results obtained through the exact and approximate Lax-Wendroff techniques are again quite similar, and that the results obtained through the technique with fluctuation control provides a slightly sharper profile.

Finally, in order to illustrate that the LW techniques are more efficient than the RK time discretization, we show a performance test involving the computational time required by each technique by running the Double Mach Reflection problem for the resolution 200×50. The results are shown in Table 5.14, where the field "Efficiency" stands for $\frac{t_{\text{RK3}}}{t_{\text{LW*}}}$.

We can see from Table 5.14 that even the fifth order Lax-Wendroff technique is more efficient than the third order accurate Runge-Kutta scheme. One of the main reasons is that fact that only a spectral decomposition per time step is needed to be performed at the LW technique, whereas three are needed by the RK3 scheme, one per stage.

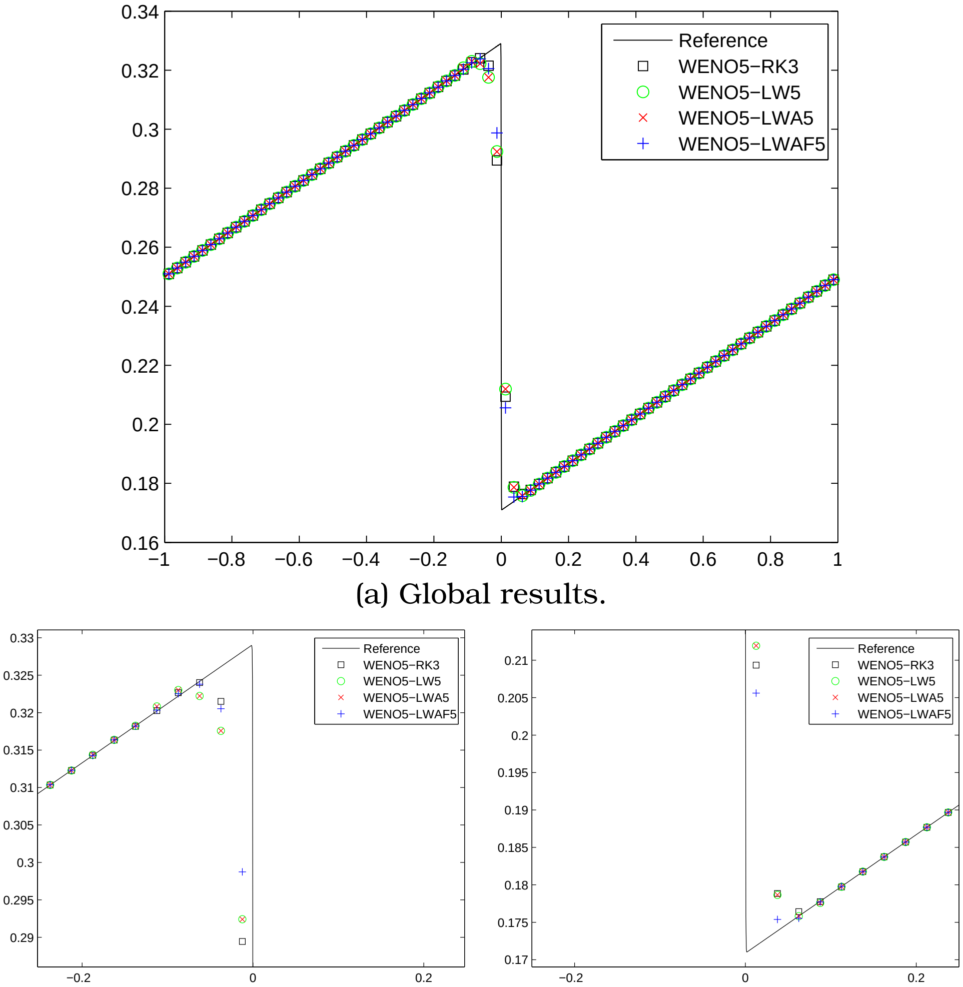
On the other hand, we see that the version with approximate fluxes has a better performance than the main formulation, since less computations are required for high order derivatives. On the other hand, if

Method	Efficiency
WENO5-LW5	1.44
WENO5-LWA5	1.54
WENO5-LWF5	1.33
WENO5-LWAF5	1.44

Table 5.14: Performance table.

the fluctuation control is used then the performance is lower, since an additional step where smoothness indicators are computed is required. However, the combination of the approximate fluxes with the fluctuation control yields a fifth order accurate with approximately the same efficiency than the original formulation, but providing better results.

As a final remark, it must be pointed out that the schemes without fluctuation control fail whenever a high order WENO linearization for the computation of the two sided Jacobian matrices fail unless the first time steps are shortened in the Shu-Osher, blast wave and Double Mach Reflection problems. This is due to the fact that initial conditions in these cases are sharp discontinuities, which aggravates the phenomena introduced by the propagation of the fluctuations through the adjacent cells of these discontinuities involving the nodal approximations of the derivatives of degree two and higher. This issue disappears in each case when the fluctuation control is used.

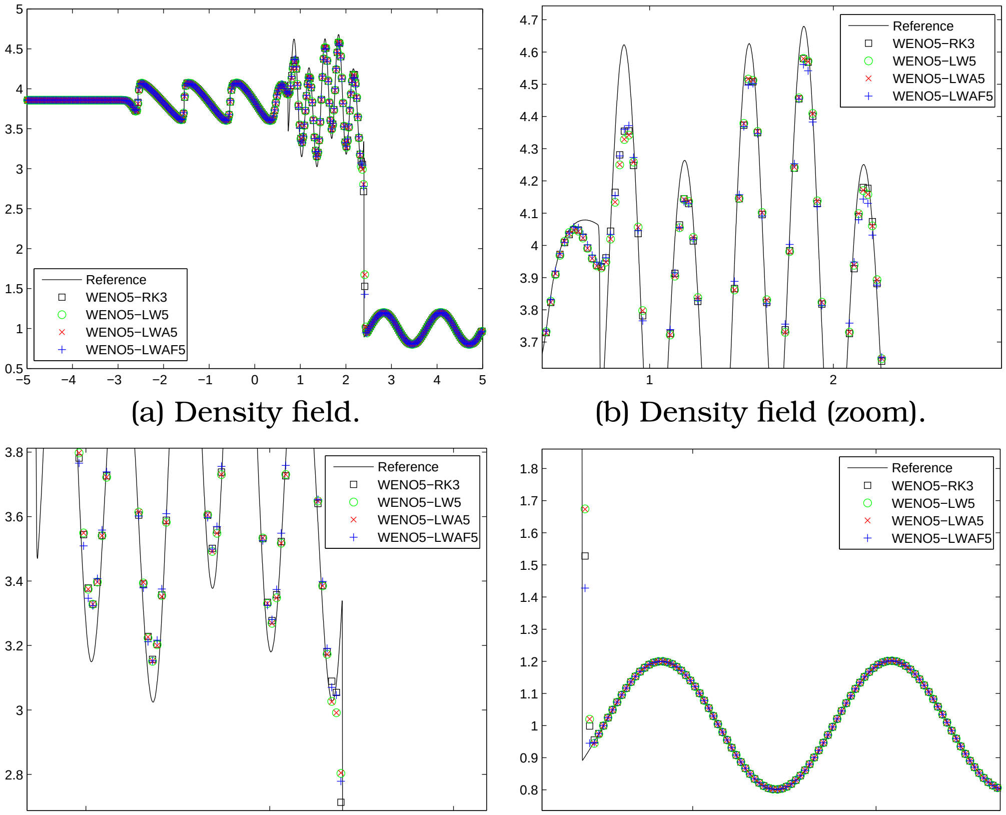


(a) Global results.

(b) Enlarged view.

(c) Enlarged view.

Figure 5.1: Discontinuous solution for Burgers equation, $t = 12$.



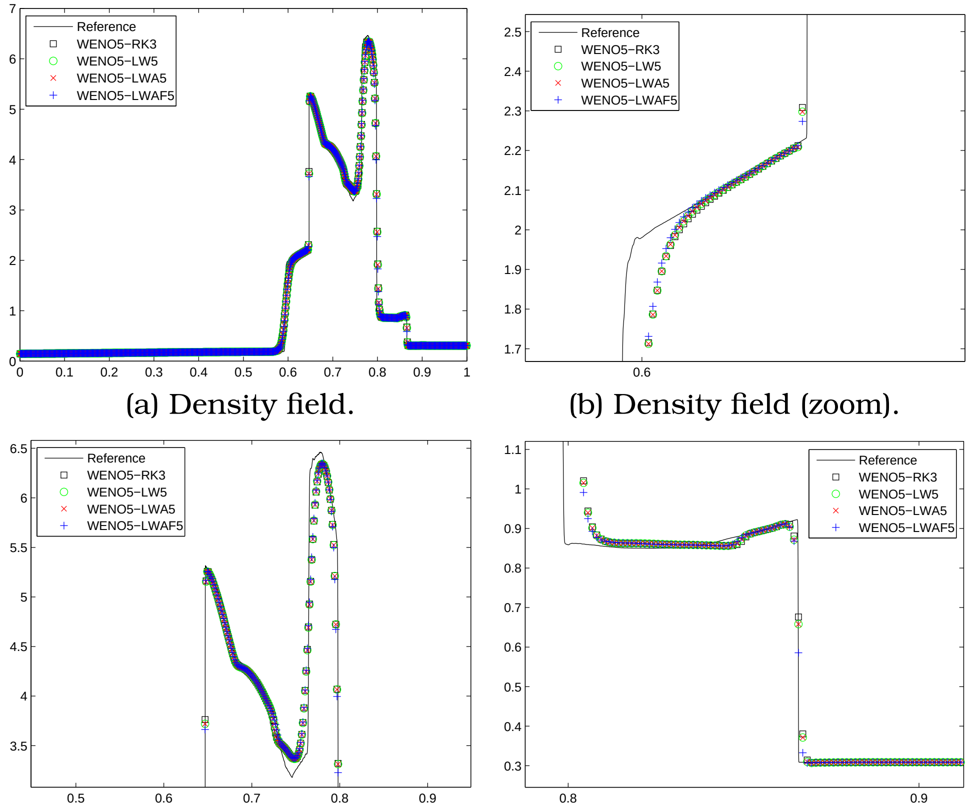
(a) Density field.

(b) Density field (zoom).

(c) Density field (zoom).

(d) Density field (zoom).

Figure 5.2: Shu-Osher problem.



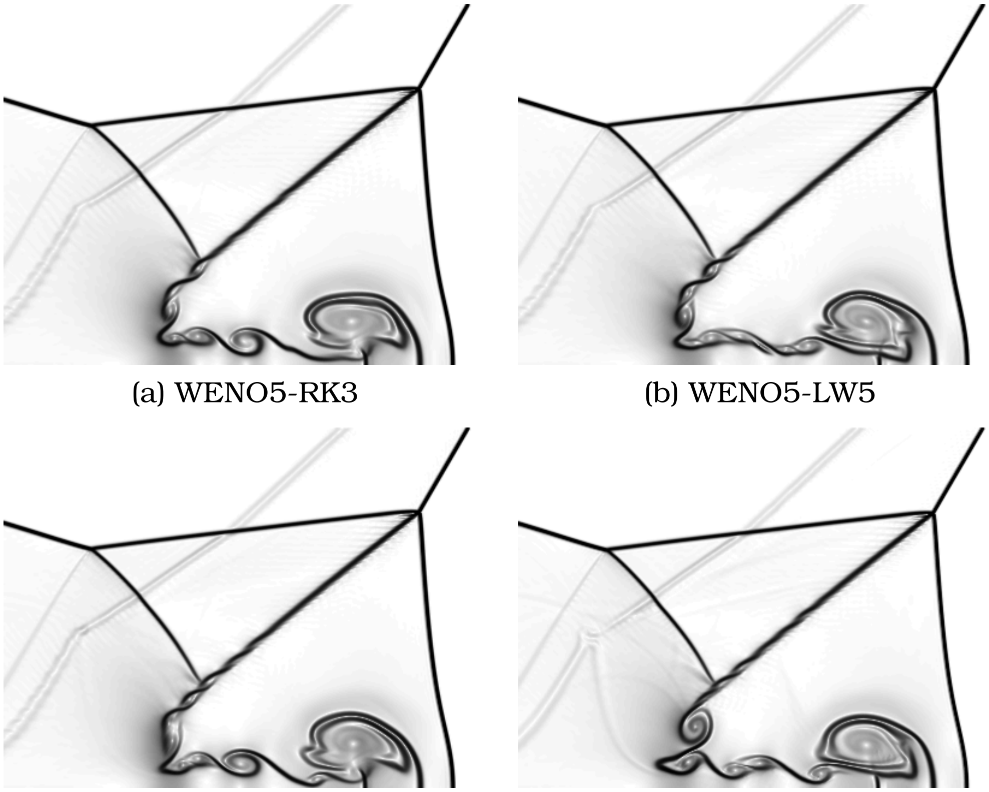
(a) Density field.

(b) Density field (zoom).

(c) Density field (zoom).

(d) Density field (zoom).

Figure 5.3: Blast wave results.


(a) WENO5-RK3

(b) WENO5-LW5

(c) WENO5-LWA5

(d) WENO5-LWAF5

Figure 5.4: Double Mach Reflection results. Density field.

6

Conclusions and further work

6.1 Conclusions

In this thesis we have presented some techniques for data extrapolation to handle boundary conditions for finite difference numerical methods for hyperbolic conservation laws. We have obtained some successful simulations in non rectangular domains. This illustrates that Lagrange extrapolation is a viable technique for filling-in auxiliary data at the ghost cells, as long as sufficient care is taken for accounting for possible discontinuities.

Furthermore, these techniques are designed to avoid an order loss at the boundary of complex domains in methods that require Cartesian meshes, a loss that can propagate to the rest of the data, thus notably decreasing the simulation quality. The results that have been obtained with these techniques entail an improvement that solves the previous problem without a significant increase in computational time at not excessively low resolutions.

The extrapolation techniques proposed in this work have the advantage of letting a regulation of the tolerance to some variations at the boundary by using a threshold parameter. This, besides being data scale independent, is useful in simulations with strong turbulence or, in general, with wide regions where the data is not smooth. However, the need of tuning the thresholding parameters to the particular problem represents a drawback of the method.

To overcome this issue, we have introduced some possible weighted extrapolations akin to WENO reconstructions capable of keeping high order accuracy for the global scheme, which entails an improvement with respect to the techniques based on thresholding parameters. On the other hand, as stated in [45], straight Lagrange extrapolation may lead to a mildly unstable scheme for multidimensional problems with some complex domains.

We have seen that an appropriate and efficient option to achieve good results both on smooth and non-smooth problems is to combine least squares with an unique weight design to reduce to the constant extrapolation (copying the value of the closest node) if there is a discontinuity in the extrapolation stencil. The results obtained with that technique are satisfactory and robust, with a weight design that, unlike those defined in [45], is dimensionless and scale independent.

From the experiments, it can be concluded that the WLS-GAW technique is better than WLS-UW without having to be scaled artificially ("magnetize" the weights to 1) in order to obtain a less diffusive profile. We have seen that the results of WLS-GAW are better than WLS-UW even for quite negative λ values.

On the other hand, we have presented an alternative method based on the Lax-Wendroff and Cauchy-Kowalewski procedure through approximations of the derivatives of the flux. The scheme is, on the one hand, capable of dealing in a more general scenario, with less implementation and computational cost than the one originally proposed by Qiu and Shu in [39]; on the other hand, generates smooth data to approximate derivatives of degree two and higher, so that large terms are not propagated through these approximations of the derivatives, solving the issue of the original formulation.

This scheme is less costly than the original version in terms of the approximations of the fluxes. On the other hand, the control of fluctuations requires additional computational resources, mainly due to the computation of Jiang-Shu smoothness indicators. However, combined with the flux approximation technique yields a scheme with similar cost than the original version and generally better results.

6.2 Further work

Since we now have a fully developed boundary extrapolation strategy and a cheap time discretization procedure, our next purpose is to develop a parallelized AMR code [3], exploiting the main benefits of using this global scheme in the above terms. The extension of all the mentioned techniques to 3D is also under consideration.

We also plan to extend all these combination of techniques to more challenging and physically relevant problems, such as polydisperse sedimentation, transport in porous media, traffic models and problems with non-local fluxes.

Bibliography

[1] F. Aràndiga and R. Donat. Nonlinear multiscale decompositions: the approach of A. Harten. *Numer. Algorithms*, 23:175–216, 2000.

[2] F. Aràndiga, A. Baeza, A. M. Belda, and P. Mulet. Analysis of WENO schemes for full and global accuracy. *SIAM J. Numer. Anal.*, 49(2):893–915, 2011.

[3] A. Baeza and P. Mulet. Adaptive mesh refinement techniques for high-order shock capturing schemes for multi-dimensional hydrodynamic simulations. *Int. J. Numer. Meth. Fluids*, 52:455–471, 2006.

[4] A. Baeza, P. Mulet, and D. Zorío. High order boundary extrapolation technique for finite difference methods on complex domains with cartesian meshes. *J. Sci. Comput.*, 66:761–791, 2016.

[5] A. Baeza, P. Mulet, and D. Zorío. High order weighted extrapolation for boundary conditions for finite difference methods on complex domains with Cartesian meshes. *J. Sci. Comput.*, 2016.

[6] O. Boiron, G. Chiavassa, and R. Donat. A high-resolution penalization method for large Mach number flows in the presence of obstacles. *Computers & Fluids*, 38:703–714, 2009.

[7] M.H. Carpenter, D. Gottlieb, S. Abarbanel, and W.-S. Don. The theoretical accuracy of Runge-Kutta time discretizations for the initial boundary value problem: a study of the boundary error. *SIAM J. Sci. Comput.*, 16:1241–1252, 1995.

[8] A. J. Chorin and J. E. Marsden. *A mathematical introduction to fluid mechanics*. Springer, New York, 3^{rd} edition, 2000.

[9] C. M. Dafermos. Polygonal approximations of solutions of the initial value problem for a conservation law. *J. Math. Anal. Appl.*, 38:33–41, 1972.

[10] C. M. Dafermos. *Hyperbolic conservation laws in continuum physics.* Springer, 2000.

[11] R. Donat and A. Marquina. Capturing shock reflections: An improved flux formula. *J. Comput. Phys.*, 125:42–58, 1996.

[12] L. C. Evans. *Partial differential equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 1998.

[13] C. F. Faà di Bruno. Note sur un nouvelle formule de calcul différentiel. *Quart. J. Math.*, 1:359–360, 1857.

[14] S. K. Godunov. A finite difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics. *Matematicheskii Sbornik*, 47:271, 1959.

[15] S. Gottlieb and C.-W. Shu. Total variation diminishing Runge-Kutta schemes. *Math. Comp.*, 67(221):73–85, 1998.

[16] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, III. *J. Comput. Phys.*, 71(2):231–303, 1987.

[17] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, III. *J. Comput. Phys.*, 71(2):231–303, 1987.

[18] A. Harten and S. Osher. Uniformly high order accurate essentially non-oscillatory schemes, I. *SIAM J. Numer. Anal.*, 24(2):279–309, 1987.

[19] C. Hirsch. *Numerical computation of internal and external flows (volume 1): fundamentals of numerical discretization.* John Wiley & Sons, Inc., New York, NY, USA, 1988.

[20] C. Hirsch. *Numerical computation of internal and external flows (volume 2): computational methods for inviscid and viscous flow.* John Wiley & Sons, Inc., New York, NY, USA, 1988.

[21] H. Holden, L. Holden, and R. Høegh-Krohn. A numerical method for first order nonlinear scalar conservation laws in one dimension. *Comput. Math. Appl.*, 15(6-8):595–602, 1988. Hyperbolic partial differential equations. V.

[22] H. Holden and N. H. Risebro. *Front tracking for hyperbolic conservation laws*, volume 152 of *Applied Mathematical Sciences*. Springer, Heidelberg, second edition, 2015.

[23] L. Huang, C.-W. Shu, and M. Zhang. Numerical boundary conditions for the fast sweeping high order WENO methods for solving the Eikonal equation. *J. Comp. Math.*, 26:336–346, 2008.

[24] H. Hugoniot. Sur la propagation du movement dans les coprs et spécialement dans les gaz parfaits. *J. Ecole Polytechnique*, 57:3–97, 1887.

[25] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *J. Comput. Phys.*, 126(1):202–28, 1996.

[26] G.-S. Jiang and C.-W. Shu. Efficient implementation of Weighted ENO schemes. *J. Of Comput. Phys.*, 126:202–228, 1996.

[27] S. N. Kružkov. First order quasilinear equations with several independent variables. *Mat. Sb. (N.S.)*, 81 (123):228–255, 1970.

[28] L. D. Landau and E. M. Lifshitz. *Fluid mechanics*. Course of theoretical physics, vol. 6. Pergamon Press, Oxford, 2^{nd} edition, 1987.

[29] P. D. Lax. Shock waves and entropy. In E.A. Zarantonello, editor, *Contributions to nonlinear functional analysis*, pages 603–634. Academic Press, 1971.

[30] P. D. Lax. *Hyperbolic systems of conservation laws and the mathematical theory of shock waves*, volume 11 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics, 1973.

[31] P. D. Lax and B. Wendroff. Systems of conservation laws. *Comm. Pure Appl. Math.*, 13:217–237, 1960.

[32] R. J. LeVeque. *Numerical methods for conservation laws*. Birkhäuser Verlag, 1992.

[33] R. J. LeVeque. *Finite-volume methods for hyperbolic problems*. Cambridge University Press, 2004.

[34] T.-P. Liu. The entropy condition and the admissibility of shocks. *J. Math. Anal. Appl.*, 53:78–88, 1976.

[35] X-D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *J. Comput. Phys.*, 115:200–212, 1994.

[36] A. Marquina and P. Mulet. A flux-split algorithm applied to conservative models for multicomponent compressible flows. *J. Comput. Phys.*, 185:120–138, 2003.

[37] O. Oleinik. Discontinuous solutions of nonlinear differential equations. *Amer. Math. Soc. Transl. Ser. 2*, 26:95–172, 1957.

[38] S. Osher. Riemann solvers, the entropy condition, and difference approximations. *SIAM J. Numer. Anal.*, 21(2):217–235, 1984.

[39] J. Qiu and C.W. Shu. Finite difference WENO schemes with Lax-Wendroff-type time discretizations. *J. Sci. Comput.*, 24(6):2185–2198, 2003.

[40] W. J. M. Rankine. On the thermodynamic theory of waves of finite longitudinal disturbance. *Phil. Trans. Roy. Soc. London*, 160:277–288, 1870.

[41] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.*, 77(2):439–471, 1988.

[42] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. *J. Comput. Phys.*, 83(1):32–78, 1989.

[43] B. Sjogreen and N.A. Petersson. A cartesian embedded boundary method for hyperbolic conservation laws. *Commun. Comput. Phys.*, 2:1199–1219, 2007.

[44] E. Tadmor. Numerical viscosity and the entropy condition for conservative difference schemes. *Math. Comp.*, 43(168):369–381, 1984.

[45] S. Tan and C.-W. Shu. Inverse Lax-Wendroff procedure for numerical boundary conditions of conservation laws. *J Comput. Phys.*, 229:8144–8166, 2010.

[46] S. Tan, C. Wang, C.-W. Shu, and J. Ning. Efficient implementation of high order inverse Lax-Wendroff boundary treatment for conservation laws. *J. Comput. Phys.*, 231(6):2510–2527, 2012.

[47] E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics*. Springer-Verlag, third edition, 2009.

[48] B. van Leer. Towards the ultimate conservative finite difference scheme, V. A second order sequel to Godunov's method. *J. Comput. Phys.*, 32:101–136, 1979.

[49] B. Wendroff. The Riemann problem for materials with nonconvex equation of state. *J. Math. Anal. Appl.*, 38:454–466, 1972.

[50] G. B. Whitham. *Linear and nonlinear waves*. Pure and Applied Mathematics (New York). John Wiley & Sons, Inc., New York, 1999. Reprint of the 1974 original, A Wiley-Interscience Publication.

[51] P.R. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.*, 54:115–173, 1984.

[52] T. Xiong, M. Zhang, Y.-T. Zhang, and C.-W. Shu. Fast sweeping fifth order WENO scheme for static Hamilton–Jacobi equations with accurate boundary treatment. *J. Sci. Comput.*, 45:514–536, 2010.