TESI DOCTORAL

Maria del Carmen Martí Raga

NEW COMPUTATIONAL TECHNIQUES FOR FINITE-DIFFERENCE WEIGHTED ESSENTIALLY NON-OSCILLATORY SCHEMES AND RELATED PROBLEMS

Maria del Carmen Martí Raga

Director: Pep Mulet Mestre

Universitat de
València
Departament de
Matemàtica Aplicada

PhD Thesis

New computational techniques for finite-difference Weighted Essentially Non-Oscillatory schemes and related problems

Maria del Carmen Martí Raga

Advisor: Pep Mulet Mestre

Universitat de València
València, 2014.

New computational techniques for finite-difference Weighted Essentially Non-Oscillatory schemes and related problems

Memòria presentada per Maria del Carmen Martí Raga, Llicenciada en Matemàtiques; realitzada al departament de Matemàtica Aplicada de la Universitat de València sota la direcció de Pep Mulet Mestre, Professor Titular d'aquest departament, amb l'objectiu d'aspirar al Grau de Doctora en Matemàtiques.

València, 29 d'abril del 2014

Pep Mulet Mestre	Maria del Carmen Martí Raga
Director de la Memòria	Aspirant al grau de Doctor

DEPARTAMENT DE MATEMÀTICA APLICADA
FACULTAT DE MATEMÀTIQUES
UNIVERSITAT DE VALÈNCIA

Agraïments

En primer lloc agraïr al meu director Pep Mulet la seua dedicació i ajuda al llarg d'aquests anys. Sense el seu suport aquest treball d'investigació no hagués estat possible.

També a la resta de membres del Departament de Matemàtica Aplicada que m'han tractat com un membre més des del primer dia.

Als meus companys de despatx i/o investigació: Dioni, Maria, David, Francesco i Anna amb els que ha estat un plaer compartir experiències.

Al professor Guillaume Chiavassa, qui em va acollir com a casa durant els mesos que vaig passar a Marsella i em va guiar en una part de la meua investigació.

Gràcies als revisors d'aquesta tesi, pels seus comentaris i suggeriments.

Finalment, donar les gràcies a les persones més importants: els meus pares i la meua germana Teresa. Gràcies per estar al meu costat incondicionalment. Sempre.

València, 2014 *MCarmen*

Contents

Resum

Introducció

Els sistemes de lleis de conservació sorgeixen de forma natural en una gran varietat d'aplicacions, com per exemple en modelar el flux de vent al voltant d'un vehicle, el flux d'aigua al llarg d'un canal o la sedimentació de partícules disperses a un fluid viscós.

Com que en general no és possible obtenir solucions exactes d'aquests sistemes d'equacions, és necessari desenvolupar mètodes numèrics capaços d'aproximar aquestes solucions. El que ens agradaria és que aquests mètodes obtingueren els resultats de les simulacions el més aviat i amb la millor precisió possible, però la simulació numèrica de problemes físics modelats per sistemes de lleis de conservació és una tasca delicada, degut a la presència de discontinuïtats en les solucions. Si calculem solucions discontínues de lleis de conservació emprant mètodes estàndard desenvolupats sota la suposició de solucions suaus es poden obtenir solucion errònies.

Aquest fet ha motivat el desenvolupament d'esquemes d'alt ordre per a la captura de xocs o "High-Resolution Shock-Capturing" (HRSC) que constitueixen l'estat de l'art quan parlem de simulacions numèriques de problemes físics. L'objectiu d'aquests mètodes és obtenir solucions numèriques amb una alta resolució quan la solució és suau, mantenint els perfils afilats de les discontinuïtats, evitant l'aparició i desenvolupament d'oscil·lacions prop d'elles.

Els esquemes HRSC robustos i precisos normalment tenen un alt cost computacional, relacionat amb el fet de que incorporen tècniques molt sofisticades per al càlcul de solucions. En situacions d'interés pràctic, és important reduir aquest alt cost computacional, mantenint la precisió de les solucions numèriques.

En aquest treball hem desenvolupat diverses tècniques per a millorar els resultats numèrics i l'eficiència dels esquemes WENO en diferències finites incorporant algunes de les tècniques més avançades presents a la literatura. Alguns punts importants que hem estudiat són: l'anàlisi dels pesos proposats per als esquemes WENO, estudiant la relació de cadascun dels paràmetres presents amb la pèrdua de precisió que apareix prop de discontinuïtats i extrems, l'anàlisi de la partició de fluxos ("flux-splitting") de Lax-Friedrichs i estudi del comportament de les solucions quan emprem altres particions de fluxos, el desenvolupament d'un esquema adaptatiu ben balancejat "AMR Well-Balanced", desenvolupat per a preservar les solucions estacionàries d'aigua en repòs per a les equacions d'aigües poc profundes ("Shallow water equations").

La tesi està estructurada en 7 capítols. Al capítol 2 s'introdueixen els conceptes més bàsics sobre dinàmica de fluids, explicant amb deteniment els sistemes d'equacions emprats als experiments numèrics d'aquesta tesi: els models de sedimentació polidispersa, les equacions d'Euler i les equacions d'aigües poc profundes. Al capítol 3 es revisen alguns dels conceptes més importants sobre mètodes numèrics per a la dinàmica de fluids, descrivint l'aproximació en diferències finites de Shu-Osher [95] i el procediment de reconstrucció WENO [59, 78].

Al capítol 4 es presenten diferents pesos per al mètode WENO proposats en [19, 54, 104] per a obtenir nous esquemes WENO amb major resolució que l'esquema WENO clàssic [59, 78]. En particular, s'estudien el pesos proposats per Yamaleev i Carpenter en [104] i es demostra que amb aquests pesos el mètode WENO només obté precisió de primer ordre prop de discontinuïtats. Intentant resoldre aquest problema de precisió, es proposen nous pesos i algunes restriccions sobre els paràmetres presents a la definició dels pesos per a garantir ordre màxim de precisió prop dels extrems.

Al capítol 5 emprem els models de sedimentació polidispersa per a avaluar el rendiment dels esquemes WENO en diferències finites emprant les diferents definicions dels pesos estudiades al capítol anterior i una nova partició de fluxos anomenada HLL [52, 97] que empra menys viscositat numèrica per tractar d'estabilitzar les reconstruccions "upwind".

Al capítol 6, ens centrarem en les equacions que modelen el flux d'aigües poc profundes i descriurem un esquema ben balancejat per a la captura de xocs o "Well-balanced Shock-Capturing" (WBSC) que emprarem juntament amb la tècnica AMR ("Adaptive Mesh Refinement"), recordant les parts més importants que la componen. En segon lloc, descriurem les correccions necessàries per a obtenir un còdig WB-AMR i per finalitzar mostrarem diversos experiments que mantenen la nostra discussió. Finalment, les conclusions del treball s'exposen al capítol 7.

Equacions per a la dinàmica de fluids

Els sistemes de lleis de conservació hiperbòlics són sistemes d'equacions diferencials en derivades parcials (EDPs) dependents del temps que són d'especial interès en dinàmica de fluids degut a que la major part dels models de moviment de fluids estan representats per equacions d'aquest tipus. En la pràctica, les lleis de conservació estan representades per sistemes d'equacions en derivades parcials, que són equivalents a la formulació integral original per a solucions suaus.

Les lleis de conservació es poden escriure com un sistema de EDPs de la forma:

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

on $u:\mathbb{R}^d\times\mathbb{R}^+\longrightarrow\mathbb{R}^m$ és la solució de la llei de conservació, d és el nombre de dimensions espacials i $f^j:\mathbb{R}^m\longrightarrow\mathbb{R}^m$ són les funcions de flux, $j=1,\ldots,d$.

Per a resoldre un problema de Cauchy, és a dir, per a trobar l'estat del sistema per a un cert temps $t=T$ a partir de l'estat a temps inicial $t=0$, són necessàries les condicions inicials $u(x,0)=u_0(x), \quad x\in\mathbb{R}^d$. També són necessàries condicions de frontera quan considerem un domini fitat en $\mathbb{R}^d$.

El sistema d'equacions diferencials (1) es pot escriure en forma quasilineal de la següent manera:

$$\frac{\partial u}{\partial t}+\sum_{j=1}^{d}\frac{\partial f^j}{\partial u}\frac{\partial u}{\partial x_j}=0, \quad x\in\mathbb{R}^d, t\in\mathbb{R}^+.$$

on les matrius $A_j\equiv\frac{\partial f^j}{\partial u}$ s'anomenen matrius Jacobianes del sistema. Direm que el sistema (1) és hiperbòlic si qualsevol combinació lineal de les matrius Jacobianes A_j

$$\sum_{j=1}^{d}\alpha_j A_j, \quad (\alpha_j\in\mathbb{R})$$

és diagonalitzable amb valors propis reals.

Una solució clàssica del sistema (1) és una funció suau $u:\mathbb{R}^d\times\mathbb{R}^+\longrightarrow\mathbb{R}^m$ que verifica l'equació (1) punt a punt. Com hem dit abans, una característica fonamental dels sistemes del tipus (1) és que en general no tenen solucions clàssiques més enllà d'un interval finit de temps, inclús quan la condició inicial és una funció suau. Per a poder considerar solucions no suaus, es pot definir una formulació dèbil que involucra l'ús de menys derivades en u, requerint així menys suavitat.

Direm que una funció $u(x,t)$ és una solució dèbil de (1), amb condició inicial $u(x,0)$, si es compleix

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

per a tota funció $\phi\in C_0^1(\mathbb{R}^d\times\mathbb{R}^+)$, 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 solucions dèbils proporcionen una generalització adequada del concepte de solució clàssica per a sistemes de lleis de conservació hiperbòliques. És fàcil veure que les solucions fortes també ho són dèbils i que les solucions dèbils contínuament diferenciables són també solucions fortes.

La condició de Rankine-Hugoniot [58, 88] es pot deduir a partir de la definició de solució dèbil [28, 55, 56]. Aquesta condició caracteritza les solucions dèbils en termes del moviment de les discontinuïtats i dóna informació sobre el comportament de les variables conservades a través de les discontinuïtats.

Per a una llei de conservació qualsevol, les condicions de Rankine-Hugoniot es poden escriure com

$$[f]\cdot n=s[u]\cdot n, \tag{2}$$

on $f=(f^1,\dots f^d)$ és una matriu que conté 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 bot a través de la discontinuïtat en una variable.

Estructura característica

Cadascun del vectors columna r_p de la matriu R defineix un camp vectorial $r_p:\mathbb{R}^m\to\mathbb{R}^m, u\to r_p(u)$, anomenat p-camp característic. Direm que un camp característic definit pel vector propi r_p és genuïnament no lineal si

$$\nabla\lambda_p(u)\cdot r_p(u)\neq 0,\quad \forall u,$$

on $\nabla\lambda_p(u) = (\partial\lambda_p/\partial u_1, \ldots, \partial\lambda_p/\partial u_m)$ és el gradient de $\lambda_p(u)$. És important notar que per a sistemes lineals, els valors propis λ_p són constants respecte de u, aleshores $\partial\lambda_p/\partial u_i = 0, \quad \forall i = 1, \ldots, m$. Per tant, els camps genuïnament no lineals no poden aparèixer en sistemes lineals i són exclusius dels sistemes no lineals.

Altre tipus de camp interessant són els camps linealment degenerats, per als quals:

$$\nabla\lambda_p(u) \cdot r_p(u) = 0, \quad \forall u.$$

Aquests camps són una generalització dels camps característics d'un sistema lineal amb coeficients constants, on $\nabla\lambda_p = 0$.

Anem a estudiar breument a continuació la relació entre discontinuïtats i camps característics. Una discontinuïtat definida per $x = s(t)$, separant dos estats $u_L(t)$ i $u_R(t)$, direm que és un p-xoc, o una ona de xoc associada al p-camp característic $r_p(u)$, si

$$\lambda_p(u_L) \geq s'(t) \geq \lambda_p(u_R). \tag{3}$$

Aquesta condició s'anomena Condició d'Entropia de Lax [66].

Una p-discontinuïtat de contacte és un cas especial de p-ona de xoc on en (3) es compleixen les igualtats:

$$\lambda_p(u_L) = s'(t) = \lambda_p(u_R).$$

Les discontinuïtats de contacte són l'únic tipus de discontinuïtat associada als camps linealment degenerats. Aquest és l'únic tipus de discontinuïtat que pot aparèixer en la solució de sistemes lineals.

Les ones de rarefacció són un altre tipus d'ones que són típiques dels camps genuïnament no lineals, caracteritzades per la condició:

$$\lambda_p(u_L) < \lambda_p(u_R).$$

Els camps genuïnament no lineals poden presentar tant xocs com ones de rarefacció depenent, entre altres coses, del valor dels estats a la dreta i a l'esquerra de la discontinuïtat.

Equacions model

Finalment, anem a presentar a continuació alguns dels models d'equacions i sistemes de lleis de conservació hiperbòliques que anem a emprar en aquesta tesi, estudiant algunes de les seues propietats més importants descrites en aquesta secció.

L'equació d'advecció és el model més simple de llei de conservació que podem considerar. En una dimensió espacial es pot escriure com:

$$u_t + au_x = 0, \tag{4}$$

on $a \in \mathbb{R}$ és una constant.

Si tenim una condició inicial $u(x, 0) = u_0(x)$, aleshores la solució del problema de Cauchy corresponent és $u(x, t) = u_0(x - at)$. Aquesta solució representa el transport d'una pertorbació inicial donada, descrita per u_0, a través del flux a velocitat constant a, sense canviar de forma, moventse cap a l'esquerra si $a < 0$ o cap a la dreta si $a > 0$. L'únic tipus de discontinuïtat que es pot donar a les solucions d'aquesta equació és la discontinuïtat de contacte.

Altra equació important, en aquest cas no lineal, és l'equació de Burgers sense viscositat que queda definida per:

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

i que es pot escriure en forma quasi-lineal com

$$u_t + uu_x = 0.$$

Aquesta equació és similar a l'equació d'advecció però amb la particularitat de que la velocitat de propagació, donada per $f'(u) = u$, no és constant, sinó que depèn de la mateixa solució. Malgrat la resemblança, el comportament de la solució d'aquesta equació és completament diferent del de l'equació d'advecció. Ones de xoc i rarefaccions poden aparèixer de manera natural en la solució d'aquesta equació.

Els sistemes lineals representen una generalització a diverses variables de l'equació d'adveció (4). Un sistema lineal hiperbòlic és un cas particular de l'EDP (1) en el que la funció flux $f(u)$ depèn linealment de u, i per tant es pot escriure com $f(u) = Au$, on A és una matriu $\mathbb{R}^m \times \mathbb{R}^m$ amb coeficients constants. Per tant, el sistema es pot escriure com:

$$u_t + Au_x = 0. \tag{5}$$

Sabem que si el sistema és hiperbòlic aleshores la matriu A és diagonalitzable amb valors propis reals, i es pot expressar com $A = R\Lambda R^{-1}$, amb $\Lambda = diag(\lambda_1, \ldots, \lambda_m)$, $\lambda_p \in \mathbb{R}$, i $R = [r_1, \ldots, r_m]$, $r_p \in \mathbb{R}^m$.

Emprant aquesta informació podem definir un canvi de base donat per la matriu R de la següent forma: $v = R^{-1}u$. Aplicant aquest canvi de base a l'equació (5), aquesta es pot reescriure com

$$v_t + \Lambda v_x = 0. \tag{6}$$

Aquest canvi de base produeix un nou sistema lineal que és diagonal i que es pot desacoblar com m equacions d'advecció, la solució de les quals és coneguda. Donada la condició inicial $u(x,0) = u_0(x)$ per a (5), la solució v del sistema d'equacions (6) ve donada per:

$$v_p(x,t) = (v_0)_p(x - \lambda_p t),$$

on $(v_0)_p$ és la component p-èssima de $v_0 = R^{-1}u_0$. Aplicant el canvi de base invers obtenim la solució general del sistema lineal (5):

$$u(x,t) = \sum_{p=1}^{m} v_p(x - \lambda_p t, 0) r_p.$$

Els sistemes de lleis de conservació no lineals es poden definir per

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

on $u : \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R}^m$ i $f : \mathbb{R}^m \longrightarrow \mathbb{R}^m$, i es poden escriure en forma quasilineal com

$$u_t + A(u)u_x = 0,$$

on $f'(u) = A(u)$ és la matriu Jacobiana $m \times m$ del sistema, les entrades de la qual no són constants respecte de u.

Les equacions d'Euler són un sistema de lleis de conservació hiperbòliques no lineals que governa la dinàmica de fluids compressibles, com gasos o líquids a altes pressions. Les equacions d'Euler en 1D es poden escriure com:

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

on ρ és la densitat, v^x és la velocitat, ρv^x és el moment, E és l'energia i p és la pressió, les quals satisfan la relació

$$E = \frac{1}{2}\rho||v||_2^2 + \frac{p}{\gamma - 1}.$$

on γ és una constant que depèn del gas particular que estem considerant.

La matriu Jacobiana d'aquest sistema d'equacions és diagonalitzable amb valors propis

$$\lambda_1 = v^x - c \quad \lambda_2 = v^x, \quad \lambda_3 = v^x + c,$$

on el paràmetre c, anomenat velocitat local del so, ve donat per a gasos ideals politròpics per

$$c = \sqrt{\frac{\gamma p}{\rho}}$$

A la solució d'aquest sistema podem trobar tant ones de xoc, com discontinuïtats de contacte o ones de rarefacció.

Els models de sedimentació polidispersa són altre exemple de sistemes de lleis de conservació no lineals.

Una suspensió polidispersa és una mescla composta per partícules sòlides menudes que pertanyen a M espècies de partícules distintes i que estan disperses en un fluid viscós. Considerarem que totes les partícules tenen la mateixa densitat i que, si anomenem D_i al diàmetre de les partícules de l'espècie i, aquestes estan ordenades de manera que $D_1 > D_2 > \cdots > D_M$. Aleshores, si ϕ_i denota el volum de concentració de partícules i v_i la velocitat de fase de cadascuna de les espècies i, l'equació de continuïtat de cadascuna de les espècies es pot escriure com

$$\partial_t \phi_i + \partial_x(\phi_i v_i) = 0, \quad i = 1, \dots, M,$$

on t és el temps i x és la profunditat a la qual es troben les partícules en el fluid. Suposarem que les velocitats $v_1, \dots, v_M$ venen donades com a funcions del vector de concentracions locals $\Phi := \Phi(x,t) := (\phi_1(x,t), \dots, \phi_M(x,t))^{\mathrm{T}}$ (hipòtesi cinemàtica). Aleshores obtenim un sistema de lleis de conservació no lineal, fortament acoblat, del tipus:

$$\Phi_t + f(\Phi)_x = 0, \quad f_i(\Phi) := \phi_i v_i(\Phi), \quad i = 1, \dots, M.$$

Un dels models de velocitat més utilitzats comunament per a la sedimentació polidispersa és el model de Masliyah-Lockett-Bassoon (MLB) [79, 81]. En aquest model, per a partícules amb la mateixa densitat, les velocitats $v_1(\Phi), \dots, v_M(\Phi)$ venen donades per

$$v_i(\Phi) = \frac{(\varrho_{\mathrm{s}} - \varrho_{\mathrm{f}}) g D_1^2}{18\mu_{\mathrm{f}}}(1-\phi)V(\phi)\big(d_i^2 - (\phi_1 d_1^2 + \cdots + \phi_M d_M^2)\big),$$

on ϱ_{s} i ϱ_{f} són les densitats dels sòlids i del fluid respectivament, g és l'acceleració de la gravetat, μ_{f} és la viscositat del fluid, $d_i = D_i/D_1$ són els diàmetres normalitzats de les partícules $i = 1 \dots M$, i V és una funció empírica que ha de satisfer $V(0) = 1$, $V(\phi_{\max}) = 0$, $V'(\phi) \leq 0$ per a $\phi \in [0, \phi_{\max}]$.

En [18, 23, 36] es demostra que el model MLB és estrictament hiperbòlic sempre que $\phi_i > 0 \; \forall i = 1, \dots, M$, i $\phi := \sum_{i=1}^{M} \phi_i < \phi_{\max}$, on el paràmetre $\phi_{\max} \in (0,1]$ és una constant donada de concentració màxima de sòlid. L'anàlisi dut a terme en [23] proporciona fites per als valors propis $\lambda_i = \lambda_i(\Phi)$ de la matriu Jacobiana $f'(\Phi)$ que es troben entrellaçats amb les velocitats $v_1, \ldots, v_M$ de la següent forma

$$M_1(\Phi) < \lambda_M(\Phi) < v_M(\Phi) < \lambda_{M-1}(\Phi) < v_{M-1}(\Phi) < \cdots < \lambda_1(\Phi) < v_1(\Phi) \tag{7}$$

on la fita inferior ve donada per

$$M_1(\Phi) = v_1(0)\Big(d_M^2 V(\Phi) + \big((1-\phi)V'(\phi) - 2V(\phi)\big)(d_1^2\phi_1 + \cdots + d_M^2\phi_M)\Big).$$

Aquesta propietat d'entrellaçat és important per als esquemes numèrics, ja que els valors propis poden ser calculats emprant un mètode per a calcular arrels adequat. Les fites per als valors propis també són molt importants per a la implementació numèrica com veurem més avant.

Finalment, les equacions d'aigües poc profundes modelen la propagació d'alteracions en l'aigua i altres fluids incompressibles, sempre que la profunditat del fluid siga menuda comparada amb la longitud d'ona de l'alteració. Les equacions d'aigües poc profundes en dos dimensions representen la conservació de la massa i del moment en un domini en dos dimensions, i es poden escriure com:

$$\begin{pmatrix} h \\ q^x \\ q^y \end{pmatrix} + \begin{pmatrix} q^x \\ \frac{(q^x)^2}{h} + \frac{gh^2}{2} \\ \frac{q^x q^y}{h} \end{pmatrix}_x + \begin{pmatrix} q^y \\ \frac{q^x q^y}{h} \\ \frac{(q^y)^2}{h} + \frac{gh^2}{2} \end{pmatrix}_y = \begin{pmatrix} 0 \\ -ghz_x \\ -ghz_y \end{pmatrix},$$

on h és la profunditat de l'aigua, q^x i q^y són les dues components del moment i z representa la topografia del fons. Els valors i vectors propis d'aquest sistema es poden calcular explícitament.

Mètodes numèrics per a la dinàmica de fluids

Després de veure algunes de les característiques principals dels sistemes de lleis de conservació hiperbòlics, en aquesta secció anem a descriure alguns dels conceptes i resultats bàsics relacionats amb els mètodes numèrics per a aquests tipus de sistemes de lleis de conservació.

El primer pas per a resoldre numèricament EDPs és remplaçar el problema continu, representat per les EDPs, per una discretització del mateix. Considerem un problema escalar de valors inicials en una dimensió:

$$\begin{cases} u_t + f(u)_x = 0, \quad x \in I, \quad t \in \mathbb{R}^+, \\ u(x,0) = u_0(x), \end{cases} \tag{8}$$

on $u, f : \mathbb{R} \longrightarrow \mathbb{R}$ i $I \subset \mathbb{R}$ és un interval tancat de la recta real, que en aquest cas, per simplificar la notació, considerarem $I = [0,1]$.

Per a discretitzar un interval de la recta real, definim una malla, o xarxa, de la següent forma: considerem un nombre sencer positiu N i definim un conjunt discret de punts $\{x_j\}_{0 \leq j < N}$ complint que $x_j = (j + 1/2)\Delta x$, amb $\Delta x = \frac{1}{N}$. A partir dels nodes x_j podem definir les cel·les c_j com

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

Una xarxa pot quedar definida, depenent del context, com el conjunt de nodes $\{x_j\}_{0 \leq j < N}$ o el conjunt de cel·les $\{c_j\}_{0 \leq j < N}$.

Els punts de la discretització temporal $\{t^n\}_{0 \leq n < M}$ es poden definir de la mateixa forma com $t^n = n\Delta t$, amb $\Delta t = \frac{1}{M}$ i M un nombre sencer positiu.

Denotarem per $U^n = \{U_j^n\}_{0 \leq j < N}$ a les aproximacions de la solució exacta $u(x_j, t^n)$ de (8) calculades puntualment als nodes x_j.

Emprant les dades inicials $u_0(x)$, podem definir U^0 com el vector d'aproximacions U_j^0 quan $t = 0$. Emprant un procediment d'evolució temporal podem construir les aproximacions U^{n+1} a partir de les aproximacions en temps anteriors $U^n, U^{n-1}, \ldots, U^{n-r}$ amb $r \in \mathbb{N}$, $r \leq n$. En el nostre cas, només considerarem mètodes numèrics explícits d'un pas, on $r = 0$, que construeixen U^{n+1} només a partir de U^n i que anem a expressar de la següent forma

$$U^{n+1} = \mathcal{H}_{\Delta t}(U^n),$$

on el subíndex Δt indica que el mètode també depèn del pas de temps Δt.

Convergència

L'objectiu dels mètodes numèrics és calcular aproximacions precises a la solució exacta de l'equació (8), per tant, el que esperem és que el mètode numèric siga convergent, és a dir, que la solució numèrica U_j^n s'aproxime a la solució exacta de l'equació diferencial $u_j^n = u\left(\left(j + \frac{1}{2}\right)\Delta x, n\Delta t\right)$ per a qualsevol punt x_j i temps t^n fixes quan Δx i Δt tendeixen a zero, és a dir quan refinem la malla. Per a mesurar si les aproximacions obtingudes amb el mètode numèric s'aproximen o no a la solució exacta de l'EDP, emprarem normes. Així, direm que un mètode és convergent, per a una norma particular $||\cdot||$, si

$$\lim_{\Delta t\to 0,\Delta x\to 0}||U_j^n-u_j^n||=0$$

per a qualsevol valor fixat de x_j i t^n.

Com que en general és quasi impossible demostrar si un mètode numèric és convergent emprant la definició de convergència, normalment emprarem els conceptes de consistència, estabilitat i el Teorema de Lax per a demostrar la convergència d'un mètode numèric.

La consistència estudia el comportament d'un mètode numèric localment, és a dir, en un únic pas de temps. Si definim l'error local de trucament, aquell que mesura l'error produït quan apliquem un únic pas de temps del mètode numèric, com

$$L^n_{\Delta t}=\frac{1}{\Delta t}\left(\mathcal{H}_{\Delta t}(U^n)-u^{n+1}\right),$$

aleshores tenim que un mètode és d'ordre p si $L_{\Delta t}(\cdot,t)=\mathcal{O}(\Delta t^p)$. Si $p\geq 1$, el nostre mètode numèric és consistent.

D'altra banda, direm que un mètode és estable si quan fem xicotetes pertorbacions en les condicions inicials $u(x,0)$, aquestes no s'amplifiquen amb el pas del temps, és a dir, aquestes pertorbacions es mantenen menudes en $u(x,t^n)$ quan $n\to\infty$. El teorema de Lax [68], ens permet relacionar aquests tres conceptes, ja que ens diu que donat un mètode d'un pas lineal consistent per a un problema de Cauchy lineal ben posat, aleshores l'estabilitat és condició necessària i suficient per a la convergència.

Mètodes numèrics

Hi ha una gran varietat de mètodes en diferències que es poden emprar per a calcular aproximacions a la solució de lleis de conservació. Molts d'aquests mètodes es basen en la substitució de les derivades parcials que apareixen en (8) per aproximacions en diferències finites apropiades. Utilitzant diferents aproximacions en diferències finites es poden desenvolupar un gran nombre de possibles esquemes en diferències finites, cadascun dels quals tindrà diferents propietats en termes de precisió, estabilitat o error.

Però aquests mètodes tenen un punt feble: si alguna singularitat apareix en la solució $u(x,t)$ aleshores les diferències finites no poden aproximar amb precisió les derivades parcials que apareixen a les EDPs. Quan treballem amb solucions discontínues, pot aparèixer més d'una solució i el mètode pot no convergir a la correcta, és més, pot arribar a convergir a una funció que no és solució dèbil de l'EDP. Alguns exemples d'aquests problemes es poden trobar, per exemple, en [70].

Per a resoldre aquest inconvenient, i garantir que els mètodes numèrics no convergeixen a funcions que no siguen solució dèbil de l'EDP, emprarem mètodes conservatius. Direm que un mètode numèric és conservatiu si es pot escriure de la següent forma:

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

on la funció $\hat{f} : \mathbb{R}^{p+q+1} \to \mathbb{R}$ s'anomena flux numèric i $p, q \in \mathbb{N}, \quad p, q \geq 0$.

El Teorema de Lax-Wendroff [69] garanteix que si un mètode conservatiu convergeix a una funció $u(x,t)$ quan la malla es refinada, aleshores aquesta funció ha de ser necessàriament una solució dèbil de la llei de conservació.

Els mètodes dels que hem parlat fins ara són mètodes que estan totalment discretizats tant en temps com en espai. Una altra forma d'obtenir un mètode conservatiu és considerar el procés de discretització en dos passos: primer es discretitza només en espai, obtenint un sistema d'equacions diferencials ordinàries (EDOs) respecte del temps, anomenades "equacions semi-discretes". Si calculem l'aproximació espacial emprant una reconstrucció conservativa dels fluxos numèrics, aleshores podem escriure el sistema d'EDOs com:

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

on $\hat{f}_{j+\frac{1}{2}} = \hat{f}(U_{j-p+1}(t), \dots, U_{j+q}(t))$.

Per a resoldre aquest sistema d'EDOs s'ha d'emprar un resolvedor d'equacions diferencials ordinàries adequat. En aquest treball utilitzem un resolvedor Runge-Kutta TVD de tercer ordre, especialment desenvolupat per Shu i Osher en [94] per a resoldre aquest tipus de sistemes d'EDOs, i que té la forma:

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

on $\mathcal{D}(U_j^n) = \dfrac{\hat{f}_{j+\frac{1}{2}}(U^n) - \hat{f}_{j-\frac{1}{2}}(U^n)}{\Delta x}$, $\forall j, n$.

Esquemes WENO en diferències finites

Per finalitzar aquesta secció anem a descriure breument els elements constitutius del mètode numèric conservatiu d'alt ordre utilitzat per a resoldre sistemes hiperbòlics de lleis de conservació en aquest treball, compost per la formulació en diferències finites de Shu i Osher [94, 95], l'esquema de reconstrucció WENO d'ordre cinc [59] i l'integrador Runge-Kutta TVD de tercer ordre explicat prèviament.

Seguint la tècnica desenvolupada per Shu i Osher en [94, 95], podem obtenir la propietat conservativa de la discretització espacial de l'equació

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

definint implícitament la funció f com:

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

de manera que la derivada espacial de (1) es pot obtenir exactament mitjançant una fórmula en diferències finites als punts extrems de les cel·les c_j,

$$u_t + \frac{1}{h}\left(f\left(x+\frac{h}{2}\right) - f\left(x-\frac{h}{2}\right)\right) = 0.$$

Aleshores, si $\widehat{f}$ és una aproximació de f obtinguda emprant els valors puntuals de la funció F en un *stencil* al voltant del punt $x_{j+\frac{1}{2}}$ de manera que $f(x_{j+\frac{1}{2}}) = \widehat{f}(x_{j+\frac{1}{2}}) + d(x_{j+\frac{1}{2}})h^r + \mathcal{O}(h^{r+1})$, per a una funció Lipschitz d, podem discretitzar la derivada espacial $(F(u))_x(x_{j+\frac{1}{2}})$ de la següent forma:

$$(F(u))_x(x_{j+\frac{1}{2}}) = \frac{\widehat{f}(x_{j+\frac{1}{2}}) - \widehat{f}(x_{j-\frac{1}{2}})}{\triangle x} + \mathcal{O}(h^r).$$

on les aproximacions $\widehat{f}(x_{j\pm\frac{1}{2}})$ es calculen a partir de valors puntuals coneguts de F sobre la malla i un esquema de reconstrucció $\mathcal{R}$.

Un punt molt important quan calculem les reconstruccions és que s'ha de tenir en compte l'"upwinding", és a dir, la direcció de propagació de la informació en la xarxa en la que estem treballant, que ve donada pels signes dels valors propis de la matriu Jacobiana, de manera que les aproximacions $\hat{f}^n_{j+\frac{1}{2}}$ es calculen emprant reconstruccions d'alt ordre esbiaixades cap a l'esquerra $\mathcal{R}^{\pm}(\bar{f}_{j-s_1}, \ldots, \bar{f}_{j+s_2}, x)$ si el valor propi corresponent és major que 0 i reconstruccions esbiaixades cap a la dreta $\mathcal{R}^{\pm}(\bar{f}_{j-s_1+1}, \ldots, \bar{f}_{j+s_2+1}, x)$ si el valor propi corresponent és menor que 0.

Resumint, el càlcul dels fluxos numèrics emprant el procediment de Shu-Osher es pot esquematitzar de la següent forma:

Algorisme 1. *(Algorisme de Shu-Osher per a equacions escalars)*

```
*Definir* $\beta_{j+\frac{1}{2}} = \max_{u\in[U_j,U_{j+1}]} |f'(u)|$
**Si** $f'(u) \neq 0 \quad \forall u \in [U_j, U_{j+1}]$
  **si** $sign(f'(u)) > 0$
    $\hat{f}_{j+\frac{1}{2}} = \mathcal{R}^+(f_{j-s_1}, \ldots, f_{j+s_2}, x_{j+\frac{1}{2}})$
  **sinó**
    $\hat{f}_{j+\frac{1}{2}} = \mathcal{R}^-(f_{j-s_1+1}, \ldots, f_{j+s_2+1}, x_{j+\frac{1}{2}})$
  **fi**
**Sinó**
  $\hat{f}^+_{j+\frac{1}{2}} = \mathcal{R}^+(f^+_{j-s_1}, \ldots, f^+_{j+s_2}, x_{j+\frac{1}{2}})$
  $\hat{f}^-_{j+\frac{1}{2}} = \mathcal{R}^-(f^-_{j-s_1+1}, \ldots, f^-_{j+s_2+1}, x_{j+\frac{1}{2}})$
  $\hat{f}_{j+\frac{1}{2}} = \hat{f}^+_{j+\frac{1}{2}} + \hat{f}^-_{j+\frac{1}{2}}$
**fi**
```

on les funcions $f^\pm$ defineixen una *partició de fluxos* que verifica $f^+ + f^- = f$ i els valors propis λ^k satisfan $\pm\lambda^k\left((f^\pm(u))'\right) \geq 0$ ($f^\pm$ són fluxos *upwind*) per a $u \in [u_j, u_{j+1}]$.

L'esquema de reconstrucció que anem a emprar en aquesta tesi és el mètode "Weighted essentially non-oscillatory" (WENO), introduït per Liu, Osher i Chan en [78] com una millora del mètode ENO ("Essentially non-oscillatory"), desenvolupat per Harten et al. en [51], i posteriorment millorat per Jiang i Shu en [59].

Si denotem per $S_k, \quad k = 0, \ldots, r-1$ als r *stencils* candidats del mètode ENO

$$S_k = \{x_{j+k-r+1}, \ldots, x_{j+k}\}, \quad k = 0, \ldots, r-1.$$

i $p^r_k(x)$ a la reconstrucció polinòmica de f d'ordre $r-1$ definida en l'*stencil* S_k, satisfent $p^r_k(x_{j+\frac{1}{2}}) = f(x_{j+\frac{1}{2}}) + \mathcal{O}(h^r)$, aleshores una reconstrucció WENO de f esbiaixada cap a l'esquerra ve donada per la combinació convexa

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

on

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

Els pesos w_k $k = 0, \ldots, r-1$, s'han de definir amb l'objectiu d'obtenir màxim ordre de precisió $2r-1$ quan la funció f siga suau, i ordre r, com el mètode ENO, quan no ho siga. Liu, Osher i Chan en [78], van definir, per a $r = 2$, els pesos complint aquestes propietats com:

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

i C_k^r són els pesos òptims, $I_k = I_k(h)$ és l'indicador de suavitat de la funció f en el *stencil* S_k i ε és una constant positiva menuda, introduïda per a evitar que s'anul·le el denominador, però que com veurem més avant, té una gran influència en el càlcul d'aproximacions prop de punts crítics i discontinuïtats.

Jiang i Shu van definir en [59] l'indicador de suavitat de la funció f en el *stencil* S_k de la següent manera:

$$I_{j,k} = \sum_{l=1}^{r-1} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} h^{2l-1} (p_{j,k}^{(l)}(x))^2 dx, \tag{9}$$

amb el qual obtenien mètodes WENO amb ordre òptim $2r-1$ per a $r = 2, 3$.

Als experiments numèrics desenvolupats posteriorment emprarem el mètode WENO de cinquè ordre, amb $r = 3$, que denotarem per WENO5, i el mètode WENO d'ordre nou, per al qual $r = 5$ i que denotarem per WENO9.

Disseny de pesos per a esquemes WENO d'ordre màxim

L'esquema WENO que hem presentat en la secció anterior, amb els pesos definits per Liu et al. en [78] i els indicadors de suavitat proposats per Jiang i Shu en [59], està desenvolupat per a obtenir reconstruccions d'ordre màxim $2r-1$ quan la funció f és suau, i reconstruccions d'ordre r, com l'algorisme ENO, sempre que la funció no siga suau.

Però, com es mostra en [19, 54, 104], els pesos clàssics de l'esquema WENO d'ordre cinc no obtenen màxim ordre de convergència prop dels extrems suaus, on la primera derivada de la solució s'anul·la. Per a resoldre aquesta pèrdua de precisió, en [54], Henrick et al. defineixen un nou mètode WENO anomenat "mapped WENO". En aquest treball els autors es basen en els pesos de Jiang i Shu per a definir uns nous pesos que utilitzen els pesos de Jiang i Shu, $w_k^{(JS)}$, com a estimació inicial que es *mapetja* a un valor més precís utilitzant les funcions

$$g_k(w) = \frac{w(\overline{w}_k + \overline{w}_k^2 - 3\overline{w}_k w + w^2)}{\overline{w}_k^2 + w(1 - 2\overline{w}_k)},$$

on $\overline{w}_k \in (0,1)$ per a $k = 0,1,2$. Per tant, $\alpha_k = g_k(w_k^{(JS)})$ és una approximació més precisa dels pesos. Utilitzant aquestos pesos s'obté que el mètode és d'ordre cinc inclús prop dels punts crítics on $f' = 0$.

Altra aproximació es pot trobar en [19] on els autors construeixen nous pesos, emprant un nou indicador de suavitat de major ordre que l'indicador de suavitat de Jiang i Shu, per a l'esquema WENO d'ordre cinc, obtenint un nou esquema WENO amb menys dissipació i una resolució major que la dels esquemes WENO clàssics, però no aconsegueixen ordre de convergència màxim als punts crítics on les primeres tres derivades s'anul·len a la vegada.

Yamaleev i Carpenter proposen en [104, 105] nous pesos que obtenen una convergència dels pesos més ràpida i millor resolució prop de discontinuïtats fortes que els pesos proposats en [19], i estableixen algunes restriccions sobre els paràmetres dels pesos que garanteixen que l'esquema WENO tinga ordre màxim per a solucions suficientment suaus amb un nombre arbitrari de derivades que s'anul·len.

Els pesos que proposen Yamaleev i Carpenter en [104, 105] es poden escriure com:

$$w_k = \frac{\alpha_k}{\sum_{i=0}^{r-1} \alpha_i}, \quad \alpha_k = C_k \left(1 + \frac{\tau_{2r-1}}{I_k + \varepsilon}\right), \qquad k = 0, \ldots, r-1,$$

I_k és l'indicador de suavitat clàssic de Jiang i Shu, ε és un paràmetre menut positiu que pot dependre de h i la funció τ_{2r-1} està definida per:

$$\tau_{2r-1} = \left(V\langle x_{j-r+1}, \ldots, x_{j+r-1}\rangle\right)^2,$$

on $V\langle x_{j-r+1}, \ldots, x_{j+r-1}\rangle$ és la diferencia no dividida definida en tots els punts del *stencil*.

Yamaleev i Carpenter demostren que els esquemes WENO amb aquests pesos i paràmetre ε complint:

$$\varepsilon \geq \mathcal{O}\left(h^{3r-4}\right),$$

tenen ordre màxim siga quin siga el nombre de derivades nul·les de la solució.

En canvi, si analitzem amb més deteniment l'estructura d'aquests pesos, es pot demostrar que prop de discontinuïtats, quan la funció f és suau almenys en un dels *stencils* $S_k,\ k=0,\dots,r-1$, no obtenim l'ordre de precisió màxim que esperàvem.

Si anomenem $\mathcal{K}=\{k/f$ no és suau en $S_k\}$, tenim que

$$\alpha_k = C_k\left(1+\frac{\tau_{2r-1}}{I_k+\varepsilon}\right) = C_k\left(1+\frac{\mathcal{O}(1)}{\mathcal{O}(1)+\varepsilon}\right) = \mathcal{O}(1), \quad \text{si } k\in\mathcal{K},$$

mentre que

$$\alpha_k = C_k\left(1+\frac{\mathcal{O}(1)}{\mathcal{O}(h^2)+\varepsilon}\right) = \mathcal{O}(h^{-2}), \quad \text{si } k\notin\mathcal{K},$$

tenint en compte que com els nodes que defineixen τ_{2r-1} creuen una discontinuïtat aleshores $\tau_{2r-1}=\mathcal{O}(1)$ i, a més, $I_k=\mathcal{O}(1)$ si f no és suau en l'*stencil* S_k, mentre que $I_k=\mathcal{O}(h^2)$ si f és suau en S_k. Aleshores obtenim que

$$w_k = \frac{\alpha_k}{\sum_{i=0}^{r-1}\alpha_i} = \frac{\mathcal{O}(1)}{\mathcal{O}(h^{-2})} = \mathcal{O}(h^2) \quad \text{si } k\in\mathcal{K},$$

mentre que

$$w_k = \frac{\alpha_k}{\sum_{i=0}^{r-1}\alpha_i} = \frac{\mathcal{O}(h^{-2})}{\mathcal{O}(h^{-2})} = \mathcal{O}(1) \quad \text{si } k\notin\mathcal{K}.$$

Emprant que $\sum_{k=0}^{r-1} w_k = 1$, aleshores:

$$\begin{aligned}
f(x_{j+\frac{1}{2}}) - q(x_{j+\frac{1}{2}}) &= f(x_{j+\frac{1}{2}}) - \sum_{k=0}^{r-1} w_k p_k^r(x_{j+\frac{1}{2}})\\
&= \sum_{k=0}^{r-1} w_k\left(f(x_{j+\frac{1}{2}}) - p_k^r(x_{j+\frac{1}{2}})\right)\\
&= \sum_{k\notin\mathcal{K}} w_k\left(f(x_{j+\frac{1}{2}}) - p_k^r(x_{j+\frac{1}{2}})\right) + \sum_{k\in\mathcal{K}} w_k\left(f(x_{j+\frac{1}{2}}) - p_k^r(x_{j+\frac{1}{2}})\right)\\
&= \sum_{k\notin\mathcal{K}} \mathcal{O}(1)\mathcal{O}(h^r) + \sum_{k\in\mathcal{K}} \mathcal{O}(h^2)\mathcal{O}(1) = \mathcal{O}(h^2)
\end{aligned}$$

Per tant, l'ordre de precisió de les reconstruccions obtingudes amb el mètode WENO5 i els pesos definits per Yamaleev i Carpenter descendeix a 2 sempre que un *stencil*, però no tots ells, pot evitar la discontinuïtat. Aquesta precisió és menor que l'ordre de precisió r corresponent a l'esquema ENO quan $r>2$.

Per a resoldre aquest problema de precisió, proposem uns nous pesos, obtinguts modificant els pesos anteriors, amb els quals s'obté ordre màxim de precisió quan la funció és suau i es milloren els resultats obtinguts amb els pesos de Yamaleev i Carpenter, quan la funció no es suau. Aquests pesos estan definits de la manera següent:

$$w_k = \frac{\alpha_k}{\sum_{i=0}^{r-1} \alpha_i}, \quad \alpha_k = C_k \left(1 + \left(\frac{\tau_{2r-1}}{I_k + \varepsilon}\right)^{\mu}\right), \quad \mu = \left\lceil \frac{r}{2} \right\rceil, \quad k = 0, \dots, r-1,$$

on I_k és l'indicador de suavitat clàssic de Jiang i Shu, ε és un paràmetre positiu menut i la funció τ_{2r-1} és el quadrat de la diferència no dividida definida en tot l'*stencil* de $(2r-1)$ punts.

Emprant el mateix raonament que en [3, Proposició 3], podem definir algunes restriccions en l'ordre del paràmetre ε, menys restrictives que les obtingudes per Yamaleev i Carpenter per als seus pesos, per a garantir mètodes WENO d'ordre màxim.

Particions de flux esbiaixades per a esquemes WENO en diferències finites

Un dels majors inconvenients dels esquemes HRSC és el seu alt cost computacional, en gran part degut a que molts d'ells utilitzen la descomposició espectral de la matriu Jacobiana del sistema per a calcular les aproximacions numèriques mitjançant projeccions locals a camps característics. Les solucions numèriques que s'obtenen emprant la informació característica són normalment excel·lents en termes de resolució però l'esforç computacional necessari per a obtenir-les sol ser molt alt, especialment per a aquells problemes en els quals la informació espectral de la matriu Jacobiana no està disponible o és molt difícil d'obtenir.

En el cas que la descomposició espectral completa de la matriu Jacobiana $f'(\Phi)$ siga coneguda, és a dir, que coneguem els valors propis $\lambda_k(f'(\Phi))$ de $f'(\Phi)$ i els corresponents vectors propis normalitzats a la dreta, $r^k(\Phi)$, i a l'esquerra, $l^k(\Phi)$, per a tot $k = 1, \dots, M$, aleshores, podem calcular el flux numèric $\widehat{f}_{j+\frac{1}{2}}$ mitjançant un esquema *upwind* amb informació característica, que denotarem per SPEC, de la manera següent:

$$\begin{aligned} \widehat{f}_{j+\frac{1}{2}} = &\sum_{k=1}^{N} r^k \left(\mathcal{R}^+ \left(l^k \cdot f^+_{j-2}, \dots, l^k \cdot f^+_{j+2}; x_{j+\frac{1}{2}}\right)\right) \\ &+ \sum_{k=1}^{N} r^k \left(\mathcal{R}^- \left(l^k \cdot f^-_{j-1}, \dots, l^k \cdot f^-_{j+3}; x_{j+\frac{1}{2}}\right)\right), \end{aligned} \tag{10}$$

on $r^k = r^k(\Phi_{j+\frac{1}{2}})$, $l^k = l^k(\Phi_{j+\frac{1}{2}})$, $\Phi_{j+\frac{1}{2}} = \frac{1}{2}\big(\Phi_j + \Phi_{j+1}\big)$, $f_j^{\pm} := f^{\pm}(x_j)$, $\mathcal{R}^{\pm}$ són operadors reconstrucció *upwind* esbiaixats (reconstruccions WENO d'ordre 5 en el nostre cas) i les funcions $f^{\pm}$ defineixen una partició de fluxos que verifica que $f^+ + f^- = f$ i $\pm\lambda^k\left((f^{\pm}(\Phi))'\right) \geq 0$ ($f^{\pm}$ són fluxos *upwind*) on Φ està definit en:

$$\mathcal{D} = \begin{cases} \{\Phi_i / i = 1, \ldots, N\} & \text{per a particions de flux globals,} \\ \{\Phi_i / i = j-2, \ldots, j+3\} & \text{per a particions de flux locals.} \end{cases}$$

La partició de fluxos de Lax-Friedrichs està definida per $f^{\pm} = \frac{1}{2}(f(\Phi) \pm \alpha\Phi)$ on el paràmetre de viscositat numèrica α està definit com una fita superior del màxim de totes les velocitats característiques, en valor absolut, de la solució per a cada pas de temps:

$$\max\{|\lambda_k(f'(\Phi))| / k = 1, \ldots, M, \Phi \in \mathcal{D}\} \leq \alpha.$$

Per a intentar millorar l'eficiència d'aquests mètodes, s'han proposat diverses alternatives per a calcular solucions numèriques sense emprar informació característica. L'ús d'esquemes WENO en diferències finites per components es va introduir en [106]. Aquests esquemes es basen en els esquemes en diferències finites de Shu i Osher [95], que obtenen els fluxos numèrics en cada interfície de la cel·la mitjançant reconstruccions *upwind* esbiaixades de fluxos *upwind* dividits (aquells en els que la matriu Jacobiana té valors propis de cert signe).

En els mètodes per components [106], el valor del vector de flux numèric $\widehat{f}_{j+\frac{1}{2}}$ es calcula fent $l_l^k = r_l^k = \delta_{k,l}$ en l'equació (10), de manera que el flux numèric queda:

$$\widehat{f}_{j+\frac{1}{2},k} = \mathcal{R}^+\left(f^+_{j-2,k}, \ldots, f^+_{j+2,k}; x_{j+\frac{1}{2}}\right) + \mathcal{R}^-\left(f^-_{j-1,k}, \ldots, f^-_{j+3,k}; x_{j+\frac{1}{2}}\right).$$

El comportament oscil·latori dels esquemes per components i la excessiva difusió de les solucions numèriques obtingudes emprant una partició de fluxos Lax-Friedrichs global han sigut estudiades i reflectides en diversos treballs, com per exemple [24, 35].

Per tractar de millorar els problemes de difusió i alleujar el comportament oscil·latori obtingut quan emprem una partició de fluxos de Lax-Friedrichs, en aquest capítol es proposa l'ús d'una partició de fluxos basada en la possibilitat de triar asimètricament les velocitats d'ona de cadascun dels termes de la partició de fluxos i que utilitza menys viscositat numèrica per tractar d'estabilitzar les reconstruccions *upwind*. Aquesta partició de fluxos s'anomena HLL ja que va ser introduïda per primera vegada, com a resolvedor de Riemann, per Harten, Lax i van Leer en [52].

Si definim $F^{\pm}(\Phi) = f(\Phi) - \alpha_{\mp}\Phi$, aleshores una condició suficient per a que $f = \gamma F^{-} + (1-\gamma)F^{+}$ siga una partició de fluxos és que els valors propis $\lambda_k((F^{+}(\Phi))')$ i $\lambda_k((F^{-}(\Phi))')$ tinguen el signe corresponent per a tot $\Phi \in \mathcal{D}$ i $\gamma \in [0,1]$.

Podem calcular $\lambda_k((F^{+}(\Phi))')$ com

$$\lambda_k((F^{+}(\Phi))') = \lambda_k(f'(\Phi) - \alpha_{-}I) = \lambda_k(f'(\Phi)) - \alpha_{-}$$

Aleshores,

$$\lambda_k((F^{+}(\Phi))') = \lambda_k(f'(\Phi)) - \alpha_{-} \geq 0 \Leftrightarrow \lambda_k(f'(\Phi)) \geq \alpha_{-}$$

Anàlogament,

$$\lambda_k((F^{-}(\Phi))') = \lambda_k(f'(\Phi)) - \alpha_{+} \leq 0 \Leftrightarrow \lambda_k(f'(\Phi)) \leq \alpha_{+}$$

Per tant, $\lambda_k((F^{+}(\Phi))') \geq 0$ i $\lambda_k((F^{-}(\Phi))') \leq 0 \quad \forall k, \Phi \in \mathcal{D}$ si i només si

$$\alpha_{-} \leq \lambda_k(f'(\Phi)) \leq \alpha_{+}, \quad \forall \Phi \in \mathcal{D}, \quad \forall k = 1, \ldots, n.$$

Com es pot veure, per a que la condició *upwind* sobre $F^{\pm}$ es complisca, α_{+} i α_{-} han de ser una estimació del màxim i mínim respectivament de les velocitats característiques en $\mathcal{D}$.

Ara, si

$$f(\Phi) = \gamma F^{-}(\Phi) + (1-\gamma)F^{+}(\Phi)$$

s'ha de complir per a qualsevol Φ, aleshores

$$\begin{aligned} f(\Phi) &= \gamma(f(\Phi) - \alpha_{+}\Phi) + (1-\gamma)(f(\Phi) - \alpha_{-}\Phi) \\ &= f(\Phi) + (-\alpha_{-} + (\alpha_{-} - \alpha_{+})\gamma)\Phi \end{aligned}$$

i per tant $\gamma = \dfrac{\alpha_{-}}{\alpha_{-} - \alpha_{+}}$. Finalment, $0 \leq \gamma \leq 1$ si i només si $\alpha_{-} \leq 0 \leq \alpha_{+}$.

Resumint, podem definir la partició de fluxos HLL ([52, 97]) com:

$$f^{+} = \begin{cases} f & \alpha_{-} \geq 0, \\ 0 & \alpha_{+} \leq 0, \\ (1-\gamma)F^{+} & \alpha_{-} \leq 0 \leq \alpha_{+} \end{cases} \qquad f^{-} = \begin{cases} 0 & \alpha_{-} \geq 0, \\ f & \alpha_{+} \leq 0, \\ \gamma F^{-} & \alpha_{-} \leq 0 \leq \alpha_{+} \end{cases}$$

amb

$$\alpha_{-} \leq \lambda_k(f'(\Phi)) \leq \alpha_{+}, \quad \forall \Phi \in \mathcal{D}, \quad \forall k = 1, \ldots, M.$$

$$\max\{\lambda_k(f'(\Phi))/k = 1, \ldots, M, \Phi \in \mathcal{D}\} \leq \alpha_+$$

$$\alpha_- \leq \min\{\lambda_k(f'(\Phi))/k = 1, \ldots, M, \Phi \in \mathcal{D}\}$$

Es pot comprovar que la viscositat numèrica emprada en la partició de fluxos de Lax-Friedrichs és major que la que s'utilitza amb la partició de fluxos HLL.

Continuant amb el nostre interés per tractar de millorar els resultats obtinguts quan treballem amb mètodes en diferències finites, seguint el treball desenvolupat per Levy et al. en [74, 75], proposem l'ús d'una definició global dels indicadors de suavitat en la definició dels pesos de l'esquema WENO. En [74, 75] es detecta que el càlcul dels indicadors de suavitat del mètode WENO és un punt clau del comportament oscil·latori de les solucions numèriques, i es proposa una definició global dels indicadors de suavitat, vàlida per a totes les components del problema, definits com una mitjana dels indicadors de suavitat de Jiang i Shu (9):

$$GI_{j,k} = \frac{1}{M} \sum_{q=1}^{M} \frac{1}{||\Phi_q||_2} \left(\sum_{l=1}^{q-1} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} h^{2l-1} (p_{j,k,q}^{(l)}(x))^2 dx \right)$$

on M és el nombre d'equacions, i $p_{j,k,q}$ i $I_{j,k,q}$ són les reconstruccions polinòmiques i els indicadors de suavitat de Jiang i Shu, respectivament, de les dades $\{f_{j+k-2,q}, f_{j+k-1,q}, f_{j+k,q}\}$. El factor $||\Phi_q||_2$ és un factor d'escala, definit com la norma L^2 de les mitjanes en cel·la de la component r-èssima de Φ:

$$||\Phi_q||_2 = \left(\sum_{j=1}^{N} |\Phi_{j,q}|^2 h \right)^{\frac{1}{2}}.$$

Amb aquesta definició dels indicadors de suavitat el que s'intenta és que prop de discontinuïtats les reconstruccions actuen uniformement en totes les components.

Refinament de malles adaptatiu ben balancejat per a fluxos d'aigües poc fondes

Les equacions que modelen el comportament de les aigües poc fondes o "Shallow Water equations" (SWE) són un sistema de lleis de balanç hiperbòliques no lineals àmpliament utilitzades i que han rebut una gran atenció per part de la comunitat científica en els últims anys.

Quan el fons és pla, les SWE esdevenen un sistema homogeni de lleis de conservació. Les seues solucions poden desenvolupar discontinuïtats, inclús quan el flux inicial és suau, el que fa necessari l'ús d'esquemes HRSC. La presència d'un fons no pla fa que es tinguen que incloure termes font en el sistema relacionats amb la geometria del fons.

És ben conegut que una discretització simple del terme font pot conduir a l'aparicició d'oscil·lacions numèriques que poden arribar a arruïnar la solució real que necessitem calcular. Aquest comportament numèric apareix quan calculem solucions estacionàries, o quasi-estacionàries, per a les quals el balanç entre els fluxos convectius i el terme font associat al fons no es respecta pel mètode numèric. Els esquemes ben balancejats o "Well-balanced" (WB) [17, 47] estan específicament dissenyats per a mantenir aquest balanç, amb precisió màquina si és possible.

Els esquemes ben balancejats per a la captura de xocs o "Well-Balanced Shock-Capturing schemes" (WBSC) constitueixen l'estat de l'art en la simulació numèrica de fluxos d'aigües poc fondes. Aquests esquemes normalment tenen un alt cost computacional relacionat amb el fet de que incorporen *upwinding* mitjançant la informació característica, procediments de reconstrucció d'alt ordre i un tractament numèric sofisticat del terme font del fons. En situacions d'interés pràctic és altament necessari combinar els esquemes WBSC amb una tècnica adaptativa que puga disminuir l'alt temps computacional de les simulacions [15, 42, 57, 76, 62].

En aquest capítol analitzarem la tècnica AMR ("Adaptive Mesh Refinement") estructurada per blocs desenvolupada en [9] i l'esquema WBSC emprat per aquesta tècnica, identificant les parts que són potencialment responsables de la pèrdua de comportament WB. Els esquemes WBSC preserven exactament la solució estacionària d'"aigua en repòs", per a la qual $v^x = v^y = 0$ i $h + z = C$ (constant), on v^x, v^y són les components corresponents de la velocitat, h l'altura del fluid i z la topografia del fons. Però, l'aigua en repòs pot no ser exactament conservada si el mateix esquema s'utilitza en un marc multi-escala. L'objectiu d'aquest capítol és abordar el problema del *well-balancing* quan un esquema WBSC s'utilitza com a resolvedor inclòs en una tècnica AMR estructurada a blocs.

L'esquema WBSC que anem a emprar en aquest treball és el desenvolupat per Donat i Martínez-Gavara en [34, 80], que preserva exactament l'estat estacionari d'aigua en repòs. Per a descriure aquest esquema d'una manera més clara, anem a considerar les SWE en una dimensió:

$$\begin{cases} h_t + (hv)_x = 0 \\ (hv)_t + \left(hv^2 + \dfrac{gh^2}{2}\right)_x = -ghz_x \end{cases}, \tag{11}$$

amb $v = v^x$. Si emprem la notació:

$$u = \begin{bmatrix} h & hv \end{bmatrix}^T, \quad f(u) = \begin{bmatrix} hv & hv^2 + \frac{gh^2}{2} \end{bmatrix}^T, \quad s(x,u) = \begin{bmatrix} 0 & -ghz_x \end{bmatrix}^T,$$

el sistema (11) es pot escriure com:

$$u_t + f(u)_x = s(x,u)$$

que es pot reescriure de forma homogènia com:

$$u_t + g[u]_x = 0,$$

on el funcional g (que depèn de f i s) actua sobre $u = u(x,t)$ de la següent forma:

$$g[u](x,t) = f(u(x,t)) - \int_{x_0}^{x} s(r, u(r,t))\, dr.$$

El punt x_0 és un punt de referència en el domini computacional, per exemple considerarem $x_0 = 0$ quan el domini computacional siga $[0,1]$.

Emprant aquesta reformulació, en [34, 80] les autores proposen el següent esquema aplicat a la solució exacta $u(x,t)$:

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}(\mathcal{G}^n_{i+\frac{1}{2}} - \mathcal{G}^n_{i-\frac{1}{2}}) \tag{12}$$

on $\mathcal{G}_{i+\frac{1}{2}}$ són fluxos numèrics híbrids per a $g[u]$.

En [34, 80] es demostra que la diferència dels fluxos $\mathcal{G}^n_{i+\frac{1}{2}} - \mathcal{G}^n_{i-\frac{1}{2}}$ en (12) es pot reescriure com una suma de termes que contenen les quantitats $\Delta g^n_{i\pm\frac{1}{2}}$

$$\Delta g^n_{i+\frac{1}{2}} := g^n_{i+1} - g^n_i = f(u(x_{i+1}, t_n)) - f(u(x_i, t_n)) + b^n_{i,i+1},$$

on

$$b^n_{i,i+1} = -\int_{x_i}^{x_{i+1}} s(r, u(r, t_n)) dr. \tag{13}$$

Per tant, per a obtenir un mètode numèric totalment discretitzat necessitem aproximar les integrals (13) mitjançant alguna regla d'integració numèrica, que ens done una bona aproximació $\hat{b}^n_{i,i+1} \approx b^n_{i,i+1}$.

Per a les SWE, $\hat{b}^n_{i,i+1}$ es pot definir de manera que s'obtinga la conservació exacta de la solució estacionaria d'aigua en repòs emprant una definició apropiada de les integrals (13). Més detalls sobre aquest esquema estan disponibles, per exemple, a [7, 34, 80].

Degut al caràcter hiperbòlic dels sistemes de lleis de balanç, els errors numèrics que apareixen quan treballem amb xarxes uniformes no es troben uniformement distribuïts. Errors més grans apareixen prop de les discontinuïtats mentre que errors molt més menuts apareixen en regions suaus, per tant, els esquemes adaptatius que incorporen refinament només on errors grans apareixen, són apropiats i algunes vegades absolutament necessaris per a simulacions multi-dimensionals o quan necessitem obtenir alta precisió. En aquest treball anem a emprar l'esquema AMR proposat en [16] per a volums finits i posteriorment ampliat per diversos autors [10, 14, 87].

Els algorismes AMR estructurats en blocs calculen l'evolució temporal d'una representació multi-escala de la solució, que es basa en un sistema jeràrquic de malles $G_0, \dots, G_L$. Per simplicitat en la notació anem a considerar que el domini computacional és $\Omega = [0,1]^d$. La malla més grossa G_0 és una malla uniforme, mentre que en nivells de resolució més alts les cel·les computacionals s'obtenen mitjançant una subdivisió uniforme d'algunes de les cel·les de la malla del nivell de resolució més gros immediatament anterior.

Suposem que la malla més grossa s'obté subdividint, en cada dimensió, l'interval unitat en N_0 intervals, de manera que la malla més grossa ve donada per

$$c_j^0 = \prod_{k=1}^{d} [j_k h_0, (j_k+1)h_0], \quad j \in G_0 := \{1, \dots, N_0\}^d, \quad h_0 = \frac{1}{N_0}.$$

Si cada nivell refinat s'obté biseccionant cadascuna de les cel·les del nivell de resolució més gros immediatament anterior, aleshores les cel·les del nivell refinat l venen donades per:

$$c_j^l = \prod_{k=1}^{d} [j_k h_l, (j_k+1)h_l], \quad j \in G_l \subseteq \{1, \dots, N_l\}^d, \quad h_l = \frac{1}{N_l}, \quad N_l = 2^l N_0.$$

Per a un temps i un nivell de resolució donats, t i l respectivament, tenim una solució numèrica multi-escala $u_l^t = (u_{l,j}^t)_{j \in G_l^t}$, on G_l^t és la malla en el nivell de resolució l i temps t i $u_{l,j}^t$ és la dada associada a un punt $x_j^l \in c_j^l$ (podria ser al centre o a l'extrem d'una cel·la) a temps t.

Els blocs constitutius més importants de l'algorisme AMR són la integració, l'adaptació i la projecció, que descriurem breument a continuació. Una descripció més completa d'aquest algorisme es pot trobar en [10].

Integració

Per a avançar la solució multi-escala des de temps t fins a $t + \triangle t_0$, Δt_0 ha de ser un pas de temps adequat per a la xarxa més grossa, de manera que la condició CFL per a la xarxa G_0^t es satisfaci:

$$\Delta t_0 = \frac{Ch_0}{\max_{u\in U^t} |f'(u)|}, \quad 0 < C \leq 1,$$

on $U^t = (u_{l,j}^t)_{j\in G_l^t}, l = 0, \ldots, L$. El pas de temps corresponent per a l'evolució dels *patches* en G_l ve donat per $\Delta t_l = \Delta t_{l-1}/2 = \Delta t_0/2^l$, el que implica que la condició CFL equivalent es compleix automàticament per a a G_l, però també que el pas temporal per a G_0 correspon a 2^l passos temporals per a G_l. Les xarxes s'integren des de la més grossa fins a la més fina d'una forma seqüencial, seguint l'ordre establert per la següent condició: $t_{l'} \leq t_l \leq t_{l'} + \Delta t_l$, si $l \leq l'$.

Sabem que en el nivell de resolució l, G_l està composta per un conjunt de *patches* uniformes disjunts. Cadascun d'aquests *patches* en un nivell de resolució determinat s'ha d'envoltar per un nombre suficientment gran de *cel·les fantasma* (2 en el nostre codi), les quals s'han d'omplir amb informació apropiada del flux, necessària per a l'aplicació de l'esquema numèric en cada *patch*.

Per a la integració des de temps t fins a $t + \Delta t_l$, les dades contingudes a les cel·les fantasma s'obtenen mitjançant *interpolació espacial* de (u_{l-1}^t, G_{l-1}). D'altra banda, per a la integració des de temps $t + \Delta t_l$ a $t + 2\Delta t_l$, la informació a la frontera s'obté aplicant primer interpolació lineal en temps per a (u_{l-1}^t, G_{l-1}), $(u_{l-1}^{t+\Delta t_{l-1}}, G_{l-1})$, i després l'operador d'interpolació espacial usual.

Adaptació

Les xarxes corresponents a diversos nivells G_l, $1 \leq l \leq L$ han de ser construïdes tenint en compte les característiques del fluid en el temps concret en el que ens trobem. El principal objectiu en aquest procés és assegurar que les discontinuïtats que estan inicialment cobertes per una xarxa d'un determinat nivell de resolució, continuen cobertes en el mateix nivell de resolució per a temps posteriors.

D'altra banda el procés de refinament ha de ser capaç de detectar noves discontinuïtats generades només es formen. L'adaptació de cada nivell de refinament es realitza descartant la xarxa actual i creant una nova d'acord amb un criteri de refinament. El criteri de refinament es basa en afegir un llindar als errors d'interpolació i als gradients discrets (veure [8] per a una descripció amb més detall).

Una vegada es crea la nova xarxa, la solució en cadascuna de les cel·les s'actualitza copiant dades ja existents o emprant interpolació espacial a partir de les dades d'una xarxa d'un nivell de resolució més gros [10, 93].

Projecció i Interpolació

El traspàs d'informació entre les xarxes es duu a terme mitjançant dos operadors: la interpolació, que s'utilitza per a generar noves dades en un nivell de resolució donat (dades en les cel·les fantasma abans de la integració i noves dades després del procés de refinament) i la projecció, que s'utilitza per a obtenir consistència en les dades presents en distints nivells de resolució.

Si treballem amb mitjanes en cel·la, podríem considerar que les dades estan associades a punts $x_j^l = (j+1/2)h_l$. Com que $(x_{2j}^l + x_{2j+1}^l)/2 = x_j^{l-1}$, aleshores ens trobem en el marc de multiresolució en mitjanes en cel·la (veure [29, 50]). Dins d'aquest marc, per a cada funció $u(x)$ de tipus L^1, la relació entre les seues mitjanes en cel·la en nivells de resolució consecutius és $(u_{l,2j} + u_{l,2j+1})/2 = u_{l-1,j}$. Per tant, la definició canònica de l'operador projecció en el marc de mitjanes en cel·la (1D) és la següent: per a cada j tal que $2j \in G_l$ podem calcular

$$u_{l-1,j}^{t+\triangle t_{l-1}} \quad \leftarrow \quad [P(u_l^{t+2\Delta t_l})]_j = \frac{u_{l,2j}^{t+2\Delta t_l} + u_{l,2j+1}^{t+2\Delta t_l}}{2}$$

D'altra banda si treballem amb valors puntuals, podríem considerar les dades associades als punts $x_j^l = jh_l$, per tant ens trobem en un marc de valors puntuals [50], en el que $x_{2j}^l = x_j^{l-1}$. Així, en el marc de valors puntuals en 1D, l'operador projecció s'obté només copiant:

$$u_{l-1,j}^{t+\triangle t_{l-1}} \quad \leftarrow \quad [P(u_l^{t+2\triangle t_l})]_j = u_{l,2j}^{t+2\triangle t_l},$$

amb $x_j^{l-1} = x_{2j}^l$.

Una vegada ja coneguts els blocs constitutius més importants de l'algorisme AMR, el nostre objectiu és aconseguir que aquest preserve almenys una classe de solucions estacionàries. Basant-nos en la descripció que hem fet, sembla que siga necessari exigir que totes les components de l'algorisme (l'esquema WBSC però també la interpolació i la projecció) han de preservar els estats estacionaris seleccionats. Recordem que en el pas adaptatiu es creen nous valors de la solució numèrica interpolant a partir de nivells de resolució més baixos. Òbviament si un estat estacionari, com és l'aigua en repòs, s'ha de mantenir, aquests nous valors han de complir amb les condicions de l'aigua en repòs. A més, també es produeixen nous valors numèrics mitjançant interpolació espacial i espacial-temporal a les cel·les fantasma, i aquests nous valors produïts també han de complir amb les condicions d'estat estacionari que volem conservar.

A continuació examinarem les condicions necessàries que s'han d'imposar en els operadors de predicció i d'interpolació per a assegurar la conservació de l'estat estacionari. Per simplicitat, realitzarem la descripció en 1D.

Suposem que l'esquema que estem emprant manté exactament almenys l'estat estacionari d'aigua en repòs, com l'esquema WBSC. Aleshores, per cada pas de temps de l'evolució temporal, tenim que, per a un *patch* donat, si $i, j \in G_l$

$$h^t_{l,j} + z_{l,j} = h^t_{l,i} + z_{l,i} = C \quad \rightarrow \quad h^{t+\Delta t_l}_{l,j} + z_{l,j} = h^{t+\Delta t_l}_{l,i} + z_{l,i}, \tag{14}$$

on $z^l = (z_{l,j})_{j \in G_l}$ és una discretització apropiada del fons en el nivell de resolució l-èssim.

Aleshores l'operador projecció respecta el *well-balancing* si i nomes si

$$[P(h_l^{t+2\triangle t_l})]_j + z_{l-1,j} = [P(h_l^{t+2\triangle t_l})]_i + z_{l-1,i} \tag{15}$$

amb $i, j \in G_{l-1}$.

L'operador interpolació inclòs en l'algorisme AMR està construït emprant tècniques interpolatòries polinòmiques a trossos. En general, sempre s'utilitza en el context següent: siguen u_{l-1} les dades del nivell de resolució $l-1$ conegudes, aleshores es construeix una funció polinòmica a trossos per a generar noves dades mitjançant l'avaluació d'un polinomi, específicament construït per a complir amb els requeriments del marc multi-escala considerat, és a dir

$$\mathcal{I}(u_{l-1}, x^l_k) = p_j(x^l_k)$$

on $p_j(x)$ és el tros polinòmic corresponent a la cel·la computacional j-èssima, que és la cel·la del nivell $l-1$ que conté a x^l_k.

Considerarem, per exemple, la interpolació espacial utilitzada per a omplir les dades dels nous *patches* creats en el pas d'adaptació de l'esquema AMR, i suposem que hem utilitzat l'esquema WBSC que manté exactament els estats estacionaris d'aigua en repòs per a determinar la solució en temps t de manera que les dades disponibles en el nivell de resolució $l-1$ satisfan

$$h_{l-1,i} + z_{l-1,i} = h_{l-1,j} + z_{l-1,j} = C, \qquad q_{l-1,j} = 0, \quad i, j \in G_{l-1},$$

i

$$h_{l,i} + z_{l,i} = h_{l,j} + z_{l,j} = C, \qquad q_{l,j} = 0, \quad i, j \in G_l.$$

Per a assegurar que les condicions d'aigua en repòs es satisfan per a les dades generades mitjançant el procés d'interpolació, el que proposem en aquest capítol és aplicar la tècnica interpolatòria a les dades obtingudes de les variables d'equilibri per a l'estat estacionari d'aigua en repòs,

$$V(x,[h,q]) = [h+z(x),q].$$

Per tant, per a solucions d'aigua en repòs, $V_{l-1} = [h_{l-1}+z_{l-1}, q_{l-1}] = [C,0]$, emprant qualsevol tècnica interpolatòria polinòmica a trossos que preserve constants tindrem que

$$\mathcal{I}(V_{l-1}, x_j^l) = [C,0].$$

Aleshores, la interpolació espacial s'implementa de la següent manera

$$\hat{u}_{l,j}^t = [h_{l,j}^t, q_{l,j}^t] = \begin{cases} \mathcal{I}(V_{l-1}^t, x_j^l) - [z_{l,j}, 0] & \textbf{si } j \in \hat{G}_l^t \setminus G_l^t, \\ u_{l,j}^t & \textbf{si } j \in G_l^t, \end{cases}$$

on $\hat{G}_l^t$ és la xarxa adaptada resultant de G_l^t.

Per tant, per a preservar les solucions estacionàries d'aigua en repòs, l'operador interpolació involucrat en la transferència de dades entre diversos nivells de resolució ha d'actuar sobre les variables d'equilibri per a l'estat estacionari d'aigua en repòs: $V = [h+z, q]$.

Cal remarcar finalment que si els operadors interpolació i/o projecció no compleixen amb aquests requeriments, l'algorisme AMR no preservarà solucions estacionàries en el mateix sentit que l'esquema WBSC.

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.

A drawback of these schemes is that most of them use the spectral decomposition of the Jacobian matrix of the system to compute the numerical approximations by local projections to characteristic fields. The numerical solutions obtained are often excellent in terms of resolution, but the computational effort needed may be too high for some problems, especially those for which the spectral information of the flux Jacobian matrix is not available or is quite difficult to obtain.

In order to reduce the computational cost, we can use component-wise finite-difference WENO schemes, based on Shu-Osher's finite-difference schemes, which compute the numerical fluxes at each cell interface by upwind-biased reconstructions of split upwind fluxes, avoiding the use of the characteristic information, but, unfortunately, they tend to yield results that are too diffusive and oscillatory.

In an attempt to improve the results obtained when using a component-wise finite-difference WENO scheme, in this work we analyze different strategies as using different split upwind fluxes, the use of a high-order reconstruction method with a control of the oscillations or the use of adaptivity, in order to speed up computing times. We make extensive testing to compare the performance of several schemes and support our discussion.

1

Introduction

1.1 Motivation

Systems of conservation laws naturally arise in many applications, including aerodynamics, 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.

As it is not generally possible to derive exact solutions to these systems of equations, hence there is a need to devise and study numerical methods to compute approximated solutions. We wish to obtain 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.

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.

1.1.1 High-resolution shock-capturing schemes

Low-order methods are faster and easier to implement, but provide less accurate solutions than high-resolution methods, that compute more accurate numerical approximations, are at least second-order accurate on smooth solutions and yet give well-resolved non-oscillatory discontinuities, but with a higher computational cost per computational cell.

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, typically second, third or even higher order, 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, high-order reconstruction procedures and, in the particular case of the shallow water equations, a sophisticated numerical treatment of the bathymetry source term, as we will see in chapter 6.

In situations of practical interest, it is highly desirable to reduce this high computational cost, while maintaining the accuracy of the numerical solutions. Different ways to achieve it are for example, avoiding the use of the characteristic information of the system, using a component-wise approach of these schemes, or combining the scheme with an adaptive technique.

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.

The discretization of the computational domain itself imposes a limit in the flow features that can be resolved. The numerical solution within a cell is often interpreted as an approximation to the average or point-value of the true solution in that cell, which means that no method can resolve phenomena whose scale is smaller than the mesh size.

The difference between numerical methods can be interpreted in terms of their relative ability to get the information of the solution contained in a single computational cell. High-order methods give better results than low-order methods asymptotically because they are able to better resolve the flow in a single cell, but to properly resolve small scale features it is a necessary condition for the grid size to be smaller than the scale of the phenomena to be solved.

To summarize, the optimal method would be a high-order method applied on a very fine computational grid, but the computational requirements of such a method would be, by far, out of reach with today's technology in a reasonable time, both in storage and computational power requirements.

1.1.2 Adaptive Mesh Refinement

Accurate approximations of the exact solution of the equations can be obtained wherever the solution has enough smoothness using a relatively coarse mesh and low-order methods. Most of the difficulties associated with the numerical solution of hyperbolic conservation laws come from the lack of smoothness of the solution in some regions of the computational domain. Fine grids are particularly helpful only in these parts of the solution which have non-smooth structure or where the solution is rapidly changing. This idea led researchers to develop a variety of techniques in order to reduce the computational cost of the overall algorithm, mainly based on the use of non-uniform grids. These algorithms use a grid with cells of variable size, trying to use cells of smaller size in some regions of interest, maintaining cells of bigger size in other regions where the solution is smooth. These grids are often difficult to manipulate in more than one space dimension, because the solution at a cell depends on the solution at some neighborhood around it. The use of cells of mixed size makes difficult the computation of the solution at the next time step because of the variable number of neighbors with non-uniform sizes and relative locations.

Adaptive Mesh Refinement (AMR) [13, 14, 16, 87] adds a new feature: temporal refinement. The goal of the AMR technique is to perform as few cell updates as possible, instead of reducing the number of cells, exploiting that cells of different sizes can be advanced in time with different time steps by splitting the cells into different grids with uniform grid size, that are integrated according to their corresponding time steps.

The main idea of the algorithm that we use in this work, developed by A. Baeza in [6], is to use a hierarchical set of Cartesian, uniform meshes that occupy different resolution levels. At the coarsest level there is a set of coarse mesh patches covering the whole domain. Mesh patches at some resolution level are obtained by the sub-division of groups of immediately coarser cells according to a suitable refinement criterion. By repeating this sub-division procedure one can cover the regions of interest with mesh patches so that the non-smooth structure of the solution can be resolved with the desired resolution. The grids at different resolution levels coexist, and some mesh connectivity information is needed to connect the solutions at different resolution levels. Provided the connectivity information, each mesh patch can be viewed in isolation and can be integrated independently. The presence of discontinuities at a small part of the domain does not restrict the time step than can be used at the coarse grid. Note that, on the other hand, there is some redundancy in the solution, since grids that correspond to different resolutions can refer to the same spatial location.

1.2 Previous work

Weighted essentially non-oscillatory (WENO) finite-difference 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 [78] as an improvement of ENO (essentially non-oscillatory) schemes, originally introduced and developed in [51, 53]. 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 [59], 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 [78].

There are a lot of works that analyze the main parts of WENO schemes, as the definition of the weights, the smoothness indicators or the role of the parameter ε in the loss of accuracy near discontinuities and extrema (see, e. g, [3, 11, 19, 39, 43, 54, 74, 104]).

The other basic ingredient of WENO finite-difference schemes is the use of the upwinding when computing the numerical flux function. The sophisticated design of the numerical flux function, that incorporates upwinding through characteristic information that needs to be computed at each cell boundary in the computational domain, tends to be fairly expensive. To speed up computing times, different strategies have been proposed as Adaptive Mesh Refinement (AMR) [6, 9, 10, 13, 86] or avoiding the use of characteristic information when computing the numerical fluxes [52, 61, 74, 83, 106].

In the case of the numerical simulation of shallow water flows it has been studied that to accurately represent discontinuous behavior, known to occur due to the non-linear hyperbolic nature of the shallow water system, and, at the same time, numerically maintain stationary solutions it is necessary the use of well-balanced shock-capturing (WBSC) schemes [17, 34, 47, 48, 73, 80].

1.3 Scope of the work

In this work we develop some techniques to improve the accuracy of the numerical results obtained with finite-difference WENO schemes, but also the efficiency of those schemes. Some points of interest investigated in this work are:

- We derive new weights for the WENO scheme and get some constraints on some parameters present in their definition to guarantee maximal order for sufficiently smooth solutions with an arbitrary number of vanishing derivatives. Although the computational times do not diminish with the use of these new weights, the numerical solutions turn to be less oscillatory and slightly more accurate than those obtained using Yamaleev and Carpenter's weights [104, 105] and also Jiang and Shu's weights [59].

- We introduce an alternative flux-splitting to the usual Lax-Friedrichs flux-splitting. The use of this flux-splitting leads to more accurate numerical solutions, especially near discontinuities, where the use of this flux-splitting reduces the dissipation of the numerical solutions.

- We combine the block structured AMR technique developed in [9] with a well-balanced scheme introduced in [34, 80] to develop a combined AMR-WBSC scheme. We show that in order for the combined AMR-WBSC scheme to maintain its well-balanced character it is necessary to implement well-balanced interpolatory techniques in the transfer operators involved in the multi-level structure. It is shown that the new AMR-WBSC scheme is more efficient than usual WBSC schemes and that it preserves the "water at rest" stationary solutions as the underlying WBSC in [34, 80] does.

1.4 Organization of the text

The text is organized as follows: In chapter 2 we recall the basic concepts and ideas of fluid dynamics, focusing on the model equations used in this work: polydisperse sedimentation models, Euler equations and shallow water equations. In chapter 3 we introduce the basics of numerical methods for fluid dynamics and describe Shu-Osher's finite-difference approach and the weighted essentially non-oscillatory (WENO) reconstruction procedure.

In chapter 4 we review the WENO reconstruction techniques obtained using the new weights proposed in [19, 54, 104] to define new WENO methods with better resolution than the classical WENO method [59, 78]. We analyze the weights developed by Yamaleev and Carpenter in [104], showing that WENO schemes with these weights achieve only first-order accuracy near discontinuities. In section 4.3 we propose new weights to solve those accuracy problems and we get some constraints on the parameter ε to guarantee that the new WENO scheme has maximal order for sufficiently smooth solutions with an arbitrary number of vanishing derivatives. Furthermore, in section 4.4, we present numerical experiments that support our theoretical results. This chapter is based on "F. Aràndiga, M.C. Martí and P. Mulet", *Weights design for maximal order WENO schemes*, to appear in *Journal of Scientific Computing*.

In chapter 5 we perform a brief exposition on characteristic based and component-wise finite-difference WENO schemes, introducing the HLL flux-splitting and a global definition of the indicators of smoothness as an instrument to alleviate the oscillations and the excessive diffusion obtained by the numerical solutions computed with component-wise finite-difference WENO schemes. In section 5.4 we perform some numerical experiments on standard tests of polydisperse sedimentation to illustrate and compare the performance of several finite difference WENO schemes. This chapter is based on "P. Mulet and M.C. Martí", *Some techniques for improving the resolution of finite difference component-wise WENO schemes for polydisperse sedimentation models*, *Applied Numerical Mathematics*, 78: 1-13, 2014.

Chapter 6 is organized as follows: first of all we briefly recall the underlying WBSC scheme used by the block structured AMR technique and the main ingredients of this technique, identifying those which are potentially responsible of the WB loss. Later on, we describe the necessary corrections to obtain a WB-AMR code and we show several numerical experiments that support our discussion. This chapter is based on "R. Donat, M.C. Martí, A. Martínez-Gavara and P. Mulet", *Well-Balanced Adaptive Mesh Refinement for shallow water flows*, *J. Comput. Phys.*, 254: 937-953, 2014.

Finally some conclusions and future research lines to be followed from this work are pointed out in chapter 7.

2

Fluid dynamics equations

In this chapter we review some basic facts about hyperbolic conservation laws, focusing on fluid dynamics equations. We will review the basic properties of some model equations used in the numerical experiments and their solutions, with the goal of getting information that has to be taken into account when building numerical schemes for their solution.

There are many sources of information about hyperbolic conservation laws and fluid mechanics as the book of Landau and Lifshitz [63] or the work of Lax [67], or more recent works, such as the books of Batchelor [12], Chorin and Marsden [28] and Dafermos [31].

2.1 Hyperbolic conservation laws

Hyperbolic systems of conservation laws are time-dependent systems of partial differential equations of special interest in fluid dynamics since the most important models of fluid motion are represented by equations of this type. In physics, a conservation law is obtained when postulating that a particular measurable property, as mass, linear momentum or energy, of an isolated physical system does not change as the system evolves in time (see [45]).

In practice conservation laws are represented by systems of partial differential equations, that are equivalent to the original integral formulation for smooth solutions.

The numerical simulation of physical problems modeled by systems of conservation laws is considerably delicate, due to the presence of discontinuities in the solution. This is one of the main reasons why particular numerical methods for hyperbolic conservation laws have to be developed. In the last years, an enormous amount of literature on numerical methods especially designed for hyperbolic conservation laws has been produced, see e.g. [2, 72, 98].

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

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

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 $f^j : \mathbb{R}^m \longrightarrow \mathbb{R}^m$ are the flux functions, $j = 1, \ldots, d$.

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 defined in $u : \mathbb{R} \times \mathbb{R}^+ \longrightarrow \mathbb{R}$ and the flux function satisfying $f : \mathbb{R} \longrightarrow \mathbb{R}$.

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 $t = 0$. Boundary conditions have to be also specified when considering a bounded domain in $\mathbb{R}^d$.

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

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

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

The integral form is more general than the differential form (2.1). In fact, the integral form implies the differential form, but the reciprocal is only true for smooth functions. In practice the solution u, in general, is not smooth and only the integral form is valid in this case. As we will see in section 2.2, a mixed formulation, where the differential form is used wherever u is smooth, and additional conditions are given for the zones where discontinuities appear, can be used.

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

$$\frac{\partial u}{\partial t} + \sum_{j=1}^{d} (f^j)'(u) \frac{\partial u}{\partial x_j} = 0, \quad x \in \mathbb{R}^d, t \in \mathbb{R}^+.$$

The matrices

$$(f^j)'(u) \equiv A_j$$

are called the Jacobian matrices of the system. System (2.1) is said to be hyperbolic if any linear combination of the Jacobian matrices A_j

$$\sum_{j=1}^{d} \alpha_j A_j, \quad (\alpha_j \in \mathbb{R})$$

has real eigenvalues and a complete set of eigenvectors. The system is said to be strictly hyperbolic if all the eigenvalues of the Jacobian matrix are distinct.

For any u the Jacobian matrices can be diagonalized as:

$$A_j = R_j \Lambda_j R_j^{-1},$$

where Λ_j is a diagonal matrix whose entries are the eigenvalues of the Jacobian matrix A_j,

$$\Lambda_j = diag(\lambda_1^j, \dots, \lambda_m^j) = \begin{pmatrix} \lambda_1^j & \dots & 0 \\ 0 & \dots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \dots & \lambda_m^j \end{pmatrix}, \tag{2.3}$$

and R_j is the matrix whose column vectors are the corresponding right eigenvectors of A_j,

$$R_j = [r_1^j|, \cdots, |r_m^j] \tag{2.4}$$

satisfying $A_j r_i^j = \lambda_j r_i^j \quad \forall i = 1, \dots, m$.

Hyperbolicity is a requirement for well-posedness. The solution of simple hyperbolic linear problems (e.g. Riemann problems) is constituted by m simple waves moving independently (see section 2.3.2). For the existence of such solutions it is necessary for the system to be hyperbolic (see [72] for an easy proof). For non-linear systems the above argument can be applied at least locally, so hyperbolicity is also necessary for non-linear systems. For a non-linear system in $\mathbb{R}^d$ the necessity of hyperbolicity can be seen when considering an initial value problem for the system written in quasi-linear form, and considering an initial data that varies only in a direction given by $\alpha = (\alpha_1, \dots, \alpha_d) \in \mathbb{R}^d$:

$$\begin{cases} \dfrac{\partial u}{\partial t} + \displaystyle\sum_{j=1}^{d} A_j(u) \dfrac{\partial u}{\partial x_j} = 0, \\ \\ u(x,0) = u_0(\alpha \cdot x), \quad \alpha = (\alpha_1, \dots, \alpha_d) \in \mathbb{R}^d. \end{cases} \tag{2.5}$$

2.2 Properties of hyperbolic conservation laws

In this section we study some important qualitative properties of hyperbolic systems of conservation laws as the development of discontinuous solutions, even if smooth initial data is provided, the concept of weak solution or the spectral structure of such systems, that helps in the development of numerical methods to approximate its solution.

2.2.1 Characteristics

In section 2.2.3 we will show how to exploit the possibility of diagonalizing the Jacobian matrix in a more general context. In this section we only aim to show that hyperbolic equations can develop discontinuities in their solutions by means of simple examples, and how these discontinuities can be treated using the spectral information contained in the Jacobian matrices.

Consider a Cauchy problem for a one-dimensional hyperbolic scalar equation of the form

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

or in quasi-linear form:

$$\begin{cases} u_t + f'(u)u_x = 0, \quad x \in \mathbb{R}, \quad t \in \mathbb{R}^+, \\ u(x,0) = u_0(x). \end{cases} \tag{2.6}$$

If $x(t)$ is a parameterized curve in the $x-t$ plane satisfying the ordinary differential equation

$$x'(t) = f'(u(x(t),t)), \tag{2.7}$$

it is easy to see that for such a curve there holds

$$\frac{d}{dt}u(x(t),t) = u_t + u_x x'(t) = u_t + f'(u)u_x = 0,$$

i.e., the solution u is constant along the curve $x(t)$ as time varies and, by (2.7), so is $x'(t)$. Such a curve is called a characteristic curve of the equation (2.6). The characteristic curves are hence given by $x(t) = f'(u)t+C$. For scalar equations, the characteristics are straight lines in the $x-t$ space, with slopes given by $f'(u)$. Roughly speaking, characteristics are the curves in the $x-t$ space that carry information.

The value of a smooth solution at a given point can be obtained from the initial data by tracing back a characteristic that passes through the point until time $t = 0$. But, since characteristic curves can intersect in $x-t$ space, at a point where two different characteristics intersect the solution would take two different values, so there would appear a *shock wave*, a jump discontinuity that propagates in time.

If no characteristics departing from $t = 0$ pass through a given point, then the solution at that point cannot be defined by means of characteristics and some information, that was not present in the initial data, has to be incorporated to build a suitable solution. In gas dynamics, this situation corresponds to the formation of an expansion wave, where the gas is being rarefied, and is therefore commonly called a rarefaction wave.

For linear equations the slope of the characteristics is constant, and thus they are parallel. In this case, the formation of a shock or rarefaction wave is not possible. Contact discontinuities are typical of linear equations with jump discontinuities in the initial data and are characterized by the propagation of the data with constant speed.

The three kind of phenomena described above (shocks, rarefactions, and contacts) represent, in a simplified form, the main typical features of the solution of hyperbolic systems of conservation laws. In section 2.2.3 we will extend more formally all these basic and intuitive ideas to hyperbolic non-linear systems.

2.2.2 Weak solutions and Rankine-Hugoniot conditions

A classical solution of (2.6) is a smooth function $u : \mathbb{R} \times \mathbb{R}^+ \longrightarrow \mathbb{R}$ that satisfies the equation (2.6) point-wise. As pointed out in the previous section, an essential feature of this problem is that there do not exist, in general, classical solutions of (2.6) beyond some finite time interval, even when the initial condition u_0 is a very smooth function.

In order to be able to consider non-smooth solutions, we could relax the classical concept of solution using the integral form of the equation, more general than the differential form which is obtained from the integral form by means of smoothness assumptions that do not hold in general, to obtain a weak formulation that involves fewer derivatives on u, hence requiring less smoothness.

Definition 1. *A function $u(x,t)$ is a weak solution of (2.1) with given initial data $u(x,0)$ if*

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

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.

The Rankine-Hugoniot condition [58, 88], whose derivation can be found for example in [28, 55, 56], 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:

$$[f] \cdot n = s[u] \cdot n, \tag{2.8}$$

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 gives simply:

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

where u_L and u_R are 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. [28].

However, weak solutions are often not unique (see e.g. [70]), and there are *entropy conditions* proposed to single out a unique weak solution, known as entropy solution: Lax's E-condition [65], defined in (2.11) below, Oleinik's generalization [85], Wendroff's condition [101] or Liu's condition [77].

2.2.3 Characteristic structure

As we have seen in section 2.2.1, characteristics play an essential role in the theory of first-order nonlinear PDE. The propagation of information along characteristics is particular to hyperbolic systems and it is used as a design mechanism for numerical methods and as a way to understand the behavior of the solutions. In this section we extend the ideas presented in section 2.2.1 for scalar equations to hyperbolic systems. More complete studies can be found in [31].

For simplicity we will restrict the study to one-dimensional problems,

$$u_t + f(u)_x = 0. \tag{2.9}$$

We know that if the system (2.9) is hyperbolic, we can decompose the Jacobian matrix as $f'(u) = R\Lambda R^{-1}$, with Λ and R as in (2.3) and (2.4) respectively. Let us describe now admissible types of discontinuities, depending on the properties of the characteristic structure of $f'(u)$, in the flow solution and their properties.

For a constant-coefficient linear system of conservation laws the characteristic information is sufficient to completely solve the system. However, non-linear systems cannot be solved by the same method but, with an analogous analysis we can obtain qualitative information about the solution structure that allows to tackle the numerical solution of the system in a more convenient way.

Each column vector r_p of R defines a vector field $r_p : \mathbb{R}^m \to \mathbb{R}^m, u \to r_p(u)$, called p-th characteristic field.

Definition 2. *Given a hyperbolic system of conservation laws $u_t + f(u)_x = 0$, with $\{\lambda_p(u)\}_{p=1}^m$ the eigenvalues of the Jacobian matrix $f'(u)$, we say that a curve $x = x(t)$ is a characteristic curve of the system if it is a solution of the ordinary differential equation:*

$$\frac{dx}{dt} = \lambda_p(u(x,t))$$

for some $p, 1 \leq p \leq m$.

Note that only strictly hyperbolic systems have m different characteristic curves.

If we interpret characteristic curves in the $x - t$ plane, we can see that, for linear systems for which the eigenvalues $\{\lambda_p\}_{p=1}^m$ do not depend on u, they are straight lines and the solution of the system is constant along them. For non-linear systems characteristics are no longer straight lines, nor the solution is constant along them but we can suppose that, for small times, the behavior of the non-linear system can be imitated by that of a linear system, coming from some suitable linearization.

We present next some types of characteristic fields of interest. The first type are genuinely non-linear fields. A characteristic field defined by an eigenvector $r_p(u)$ is called genuinely non-linear if

$$\nabla\lambda_p(u) \cdot r_p(u) \neq 0, \quad \forall u,$$

where $\nabla\lambda_p(u) = (\partial\lambda_p/\partial u_1, \ldots, \partial\lambda_p/\partial u_m)$ is the gradient of $\lambda_p(u)$. Note that for linear systems the eigenvalues λ_p are constant with respect to u, hence $\partial\lambda_p/\partial u_i = 0, \quad \forall i = 1, \ldots, m$, so genuinely non-linear fields cannot appear in linear systems and are particular of non-linear systems.

Another interesting type of characteristic fields are linearly degenerate fields, for which

$$\nabla\lambda_p(u)\cdot r_p(u) = 0, \quad \forall u. \tag{2.10}$$

In linearly degenerate fields $\lambda(u)$ remains therefore constant along integral curves of $r_p(u)$ as u varies, due to (2.10). These fields are a generalization of the characteristic fields of a constant-coefficient linear system, where $\nabla\lambda_p = 0$.

Roughly speaking, the behavior of the solution with respect to a linearly degenerate field is similar to that of a linear system, whereas a genuinely non-linear field implies types of discontinuous solutions that can never appear in a linear system. So, in certain cases, the presence or not of an specific type of discontinuity can be determined from the characteristic structure of the Jacobian matrix, more precisely, from the particular types of characteristic fields.

Let us study here briefly the relationship between discontinuities and characteristic fields.

A discontinuity defined by $x = s(t)$, separating two states $u_L(t)$ and $u_R(t)$, is said to be a p-shock, or a shock wave associated to the p-th characteristic field if

$$\lambda_p(u_L) \geq s'(t) \geq \lambda_p(u_R). \tag{2.11}$$

Condition (2.11) is called Lax's E-condition [66]. It is a particular case of the entropy conditions mentioned in Section 2.2.2.

A p-contact discontinuity is an special case of a p-shock wave, where (2.11) holds with equalities, i.e.

$$\lambda_p(u_L) = s'(t) = \lambda_p(u_R). \tag{2.12}$$

In gas dynamics, a contact discontinuity represents the separation of two zones with different density, but in pressure equilibrium, whereas shock waves represent a discontinuity arising from an abrupt pressure change, resulting in a compression of the medium.

Rarefaction waves are a kind of waves that are typical of genuinely non-linear fields. A rarefaction wave does not involve discontinuities in the conserved variables and in gas dynamics represents the situation in which the fluid is expanding and there is a zone where the fluid is being rarefied. Rarefactions are characterized by the condition

$$\lambda_p(u_L) < \lambda_p(u_R).$$

Analyzing the relationships between the different types of waves introduced above and the different kinds of characteristic fields, we can see that: in linearly degenerate fields corresponding to single eigenvalues, if two states u_L and u_R lie in the same integral curve and compose a jump discontinuity, then by (2.10) these two states propagate with the same velocity, i. e. $\lambda_p(u_L) = \lambda_p(u_R)$ holds, forcing (2.12) to hold. Therefore discontinuities associated to these fields can only be contact discontinuities. These are the only type of discontinuities that can appear in the solution of linear systems. On the other hand genuinely non-linear fields can host both shocks and rarefaction waves, depending on the left and right states and the kind of monotonicity of the variation of $\lambda_p(u)$.

2.3 Model equations

In this section we introduce the main model equations used in this memoir and we recall some of their main features. The model equations that we consider are: the advection equation, Burgers' equation, linear hyperbolic systems and the Euler equations, in one and two dimensions, the polydisperse sedimentation models and the shallow water equations, as a model of non-linear systems of conservation laws. All these models will be used for the validation of our results.

2.3.1 Scalar hyperbolic equations

First of all, we consider the advection equation and Burgers' equation, which represent two of the most studied examples of hyperbolic scalar equations. Many of the difficulties encountered with systems of equations are already encountered in those scalar equations.

Advection equation

The advection equation is the simplest model of a conservation law. In one space dimension it is written as:

$$u_t + au_x = 0, \tag{2.13}$$

where $a \in \mathbb{R}$ is a constant.

The advection equation governs the motion of a (conserved) quantity, with density u, in a fluid as it is advected with constant velocity a. Advection with space or time-dependent velocities will not be considered here.

If an initial condition $u(x,0) = u_0(x)$ is given, the solution of the corresponding Cauchy problem is $u(x,t) = u_0(x - at)$. This solution represents the transport of a given perturbation described by u_0 through the flow at constant speed a, without changing shape, moving towards the left if $a < 0$ and to the right if $a > 0$. Note that even if u_0 is not continuous $u(x,t) = u_0(x - at)$ is still a weak solution of (2.13), and such a situation is a simple case of a contact discontinuity propagating with constant velocity.

For the advection equation, the characteristic curves are curves in the $x - t$ plane satisfying the ordinary differential equation

$$\begin{cases} x'(t) = a \\ x(0) = x_0. \end{cases}$$

which solutions are the straight lines $x - at = x_0$. If we differentiate $u(x,t)$ along one of these curves we confirm that u is constant along these characteristics, as we expected.

Inviscid Burgers' equation

In [25], Burgers studied the equation

$$u_t + \left(\frac{u^2}{2}\right)_x = \epsilon u_{xx}$$

which includes a viscous term ϵu_{xx}, with $\epsilon > 0$. This is one of the simplest models that include the non-linear and viscous effects of fluid dynamics.

The inviscid Burgers' equation is defined by dropping this viscous term ϵu_{xx}:

$$u_t + \left(\frac{u^2}{2}\right)_x = 0,$$

and it can be written in quasi-linear form as

$$u_t + u u_x = 0.$$

This equation is similar to the advection equation but with the particularity that the speed of propagation, given by $f'(u) = u$, is no longer constant, but depends on the solution itself. Despite of this resemblance, the behavior of the solution of this equation is completely different from the advection equation. Here u is not simply advected as time evolves, but can also be compressed or rarefied. Shocks and rarefaction waves typically appear in the solution of this equation.

2.3.2 Linear hyperbolic systems

Linear systems represent a generalization to several variables of the scalar advection equation (2.13). In this section we will study the main properties of linear hyperbolic systems, focusing on how we can compute the solution of a linear hyperbolic system through a change of variable using our knowledge of the solution of the scalar advection equation.

A linear hyperbolic system is a particular case of the PDE (2.1) where the flux function $f(u)$ depends linearly on u, hence it can be written as $f(u) = Au$, where A is an $\mathbb{R}^m \times \mathbb{R}^m$ constant-coefficient matrix. Then, for this case, the equation can be written as

$$u_t + Au_x = 0. \tag{2.14}$$

As we have said in section 2.1, if the system is hyperbolic, the matrix A has m real eigenvalues $\lambda_1, \dots, \lambda_m$ and m linearly independent (right) eigenvectors $r_1, \dots, r_m$. This is equivalent to saying that the matrix A is diagonalizable with real eigenvalues, i.e., it can be expressed as $A = R\Lambda R^{-1}$, where $\Lambda = diag(\lambda_1, \dots, \lambda_m)$, with $\lambda_p \in \mathbb{R}$ and $R = [r_1, \dots, r_m]$, $r_p \in \mathbb{R}^m$.

Using all these information, we can introduce a change of basis given by the matrix R. The variables u, when expressed in the basis given by R, are called the characteristic variables, as stated in the next definition.

Definition 3. *Given a hyperbolic linear system, with matrix $A = R\Lambda R^{-1}$, the characteristic variables $v = [v_1, \dots, v_m]^T$ of the system are defined by*

$$v = R^{-1}u.$$

If we apply this change of basis to the equation (2.14), we can rewrite it as:

$$\begin{aligned}
R^{-1}(u_t &+ Au_x) = 0 \\
R^{-1}u_t &+ R^{-1}Au_x = 0 \\
R^{-1}u_t &+ R^{-1}(R\Lambda R^{-1})u_x = 0 \\
R^{-1}u_t &+ \Lambda R^{-1}u_x = 0 \\
(R^{-1}u)_t &+ \Lambda(R^{-1}u)_x = 0 \\
v_t &+ \Lambda v_x = 0 \qquad (2.15)
\end{aligned}$$

$$\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix}_t + \begin{bmatrix} \lambda_1 & 0 & \dots & \dots & 0 \\ 0 & \lambda_2 & 0 & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & 0 \\ 0 & \dots & 0 & 0 & \lambda_m \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix}_x = 0 \qquad (2.16)$$

The equation (2.15) is called the characteristic form of the linear system (2.14).

This change of basis produces a new linear system that is diagonal, and can hence be solved as m (decoupled) advection equations, whose solution is known, see section 2.3.1. Each row in (2.16) reads as

$$\frac{\partial v_p}{\partial t} + \lambda_p \frac{\partial v_p}{\partial x} = 0,$$

which is nothing but an advection equation with constant velocity λ_p. Given initial data $u(x, 0) = u_0(x)$ for (2.14), the solution v of (2.15) is given by

$$v_p(x, t) = (v_0)_p(x - \lambda_p t),$$

where $(v_0)_p$ is the p-th component of $v_0 = R^{-1}u_0$. By applying the inverse change of basis to the solution of the diagonal system one obtains the general solution of the linear system (2.14) as:

$$u = Rw,$$

or in expanded form:

$$u(x, t) = \sum_{p=1}^{m} v_p(x - \lambda_p t, 0) r_p.$$

2.3.3 Non-linear hyperbolic systems

Non-linear hyperbolic systems of conservation laws are defined as

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

where $u : \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R}^m$ and $f : \mathbb{R}^m \longrightarrow \mathbb{R}^m$ and that can be written in quasi-linear form as

$$u_t + A(u)u_x = 0,$$

where $f'(u) = A(u)$ is the $m \times m$ Jacobian matrix of the system whose entries are not constant with respect to u.

Within this section we introduce the model equations that will be used as test problems and reference models throughout the text: the one and two dimensional Euler equations, the polydisperse sedimentation models and the shallow water equations.

Euler equations

The Euler equations are a system of non-linear hyperbolic conservation laws that govern the dynamics of a compressible fluids, such as gases or liquids at high pressures, for which the effects of body forces, viscous stresses and heat flux are neglected.

The two-dimensional Euler equations can be written as:

$$u_t + f(u)_x + g(u)_y = 0, \tag{2.17}$$

with

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

where ρ denotes the mass density, v^x and v^y are the Cartesian components of the velocity vector v, ρv^x and ρv^y are the Cartesian components of the momentum, E is energy and p is pressure. Physically, these conserved variables result naturally from the application of the fundamental laws of conservation of mass, linear momentum and energy in the fluid as it evolves, and represent a simplified model for the Navier-Stokes equations, which are the most complete model used up to now for the simulation of fluid dynamics.

Discarding the first two terms in the left hand side of (2.17) and canceling out the third row of u and $f(u)$, we obtain the one dimensional version of the Euler equations:

$$\begin{bmatrix} \rho \\ \rho v^x \\ E \end{bmatrix}_t + \begin{bmatrix} \rho v^x \\ \rho (v^x)^2 + p \\ v^x(E+p) \end{bmatrix}_x = 0. \tag{2.18}$$

The Euler equations are insufficient to completely describe the physical processes involved. There are more unknowns than equations and thus, to close the system, we need to specify an additional relation joining all these variables. This relation is called Equation Of State (EOS), and depends on the type of fluid under consideration.

Total energy can be decomposed into kinetic energy plus internal energy as follows:

$$E = \frac{1}{2}\rho ||v||_2^2 + \rho e, \tag{2.19}$$

where e denotes the specific internal energy. Kinetic energy is due to the advection of the flow, whereas internal energy is the result of other forms of energy. We assume that internal energy is a known function of pressure and density, $e = e(p, \rho)$, expressed as

$$e = \frac{p}{\rho(\gamma - 1)}, \tag{2.20}$$

which is the equation of state for polytropic ideal gases that we are going to use in this work. The constant $\gamma > 1$ depends on the particular gas under consideration.

Substituting (2.20) into (2.19) gives

$$E = \frac{1}{2}\rho ||v||_2^2 + \frac{p}{\gamma - 1}. \tag{2.21}$$

The one-dimensional version of the Euler equations has a diagonalizable Jacobian matrix that can be written as:

$$f'(u) = \begin{bmatrix} 0 & 1 & 0 \\ \frac{1}{2}(\gamma-3)(v^x)^2 & (3-\gamma)v^x & \gamma - 1 \\ \frac{1}{2}(\gamma-2)(v^x)^3 - \frac{c^2 v^x}{\gamma - 1} & \frac{3-2\gamma}{2}(v^x)^2 + \frac{c^2}{\gamma-1} & \gamma v^x \end{bmatrix},$$

with eigenvalues

$$\lambda_1 = v^x - c \quad \lambda_2 = v^x, \quad \lambda_3 = v^x + c,$$

and computable matrices of right and left eigenvectors. The parameter c is called the sound velocity of the gas and, for ideal polytropic gases, can be written as

$$c = \sqrt{\frac{\gamma p}{\rho}}$$

In 2D, the eigenstructure of the Jacobian of the fluxes can be similarly obtained in closed form.

Polydisperse sedimentation models

One dimensional multispecies kinematic flow models have received a great deal of attention in recent years because, even in the simplest 1D setting, they are of use in engineering applications such as traffic flow modeling or the controlled sedimentation of polydisperse suspensions of small particles.

Polydisperse suspensions are mixtures composed of small solid particles belonging to M different species, that vary in size or density, and which are dispersed in a viscous fluid. We will only consider particles of the same density, so each species can be seen as a size class. If D_i denotes the diameter of the i-th species, we assume the species to be ordered so that $D_1 > D_2 > \cdots > D_M$. We use ϕ_i to denote the volume fraction of particle species i and v_i for the phase velocity of species i. Then the continuity equations of the M species are

$$\partial_t \phi_i + \partial_x(\phi_i v_i) = 0, \quad i = 1, \ldots, M,$$

where t is time and x is depth. The velocities $v_1, \ldots, v_M$ are assumed to be given functions of the vector $\Phi := \Phi(x,t) := (\phi_1(x,t), \ldots, \phi_M(x,t))^{\mathrm{T}}$ of local concentrations (kinematic assumption). This yields non-linear, strongly coupled systems of conservation laws of the type

$$\Phi_t + f(\Phi)_x = 0, \quad f_i(\Phi) := \phi_i v_i(\Phi), \quad i = 1, \ldots, M. \tag{2.22}$$

We seek solutions $\Phi = \Phi(x,t)$ such that $\phi_i \geq 0 \quad \forall i = 1, \ldots, M$, and $\phi := \sum_{i=1}^{M} \phi_i \leq \phi_{\max}$, where the parameter $\phi_{\max} \in (0,1]$ stands for a given maximum solids concentration. For the typical application of batch settling of a suspension in a column of height L, (2.22) is defined for $(x,t) \in (0,L) \times (0,T)$ and zero-flux boundary conditions

$$f|_{x=0} = f|_{x=L} = 0$$

are prescribed.

Several choices of v_i ("models") or equivalently, of the fluxes f_i, as functions of Φ, and depending on the vector of normalized particle sizes $d := (d_1, \dots, d_M)^T$, where $d_i := D_i/D_1$ for $i = 1, \dots, M$, have been proposed in the literature. One of the most commonly used velocity models for polydisperse sedimentation is the Masliyah-Lockett-Bassoon (MLB) model [79, 81]. This model arises from the continuity and linear momentum balance equations for the solid species and the fluid through suitable constitutive assumptions and simplifications (cf. [18]). In this model, for particles that have the same density, the velocities $v_1(\Phi)$, ..., $v_M(\Phi)$ are given by

$$v_i(\Phi) = \frac{(\varrho_\mathrm{s} - \varrho_\mathrm{f}) g D_1^2}{18\mu_\mathrm{f}} (1-\phi)V(\phi)\big(d_i^2 - (\phi_1 d_1^2 + \cdots + \phi_M d_M^2)\big),$$

where ϱ_s and ϱ_f are the solid and fluid densities, g is the acceleration of gravity, μ_f is the fluid viscosity and V is an empirical hindered settling function assumed to satisfy

$$V(0) = 1, \quad V(\phi_{\max}) = 0, \quad V'(\phi) \leq 0 \quad \text{for } \phi \in [0, \phi_{\max}].$$

A standard choice for $V(\phi)$ is given by Richardson-Zaki's hindered settling function [90]:

$$V(\phi) = \begin{cases} (1-\phi)^{n_\mathrm{RZ}-2} & 0 < \phi < \phi_{\max} \\ 0 & \text{otherwise,} \end{cases}$$

with $n_\mathrm{RZ} > 2$.

It can be seen in [18, 23, 36] that the MLB model is strictly hyperbolic whenever $\phi_i > 0, \quad \forall i = 1, \dots, M$, and $\phi < \phi_{\max}$. In contrast to the development carried out in [18, 107], the analysis developed in [23, 36] does not involve the direct computation of the eigenpolynomial of the Jacobian matrix, but it obtains quite directly a rational function that characterizes the eigenvalues of the Jacobian matrix. The key structural property of this model, which led to these results, consists in that the fluxes f_i do not depend on each of the M components of Φ in an individual way, but only on a small number $m << M$ ($m = 2$ for the MLB model) of scalar functions of Φ. Therefore, the Jacobian $f'(\Phi)$ of the flux vector of (2.22) is a rank-m perturbation of a diagonal matrix and the eigenvalues of a rank-m perturbation of a diagonal matrix can be characterized as the roots of the so-called secular equation [1]. The analysis is based on a rational function, $R(\lambda)$, that satisfies

$$det(f'(\Phi) - \lambda I) = R(\lambda) \prod_{i=1}^{N} (v_i - \lambda)$$

for a fixed vector Φ, under appropriate circumstances. For (2.22), $R(\lambda)$ is of the form

$$R(\lambda) = \sum_{i=1}^{N} \frac{\gamma_i}{v_i - \lambda},$$

and its coefficients γ_i, $i = 1, \ldots, M$, can be calculated with acceptable effort for moderate values of m. The key result is that if these coefficients are of the same sign, then the existence of M different eigenvalues of $f'(\Phi)$ is ensured.

The analysis of [23] also provides sharp bounds of the eigenvalues of $f'(\Phi)$. The eigenvalues $\lambda_i = \lambda_i(\Phi)$ of the Jacobian $f'(\Phi)$ can be localized since they interlace with $v_1, \ldots, v_M$ as

$$M_1(\Phi) < \lambda_M(\Phi) < v_M(\Phi) < \lambda_{M-1}(\Phi) < v_{M-1}(\Phi) < \cdots < \lambda_1(\Phi) < v_1(\Phi) \tag{2.23}$$

where the lower bound is given by

$$M_1(\Phi) = v_1(0)\Big(d_M^2 V(\Phi) + \big((1-\phi)V'(\phi) - 2V(\phi)\big)(d_1^2\phi_1 + \cdots + d_M^2\phi_M)\Big).$$

The right and left eigenvectors of $f'(\Phi)$, denoted by $\mathbf{x} = (x_1, \ldots, x_M)^T$ and $\mathbf{y} = (y_1, \ldots, y_M)^T$, respectively, that correspond to a root λ of the secular equation are

$$x_i = \frac{1}{v_i - \lambda}\left[b_{i,1}\sum_{k=1}^{M}\frac{a_{k,1}b_{k,2}}{v_k - \lambda} - b_{i,2}\left(1 + \sum_{k=1}^{M}\frac{a_{k,1}b_{k,1}}{v_k - \lambda}\right)\right], \quad i = 1, \ldots, M$$

$$y_i = \frac{1}{v_i - \lambda}\left[a_{i,1}\sum_{k=1}^{M}\frac{b_{k,1}a_{k,2}}{v_k - \lambda} - a_{i,2}\left(1 + \sum_{k=1}^{M}\frac{b_{k,1}a_{k,1}}{v_k - \lambda}\right)\right], \quad i = 1, \ldots, M$$

where

$$\begin{aligned} &b_{i,1} = \phi_i d_i^2 V'(\phi), && b_{i,2} = -\phi_i, \\ &a_{i,1} = 1, && a_{i,2} = V'(\phi)\left(d_1^2\phi_1 + \ldots + d_M^2\phi_M\right) + d_i^2 V(\phi). \end{aligned}$$

The interlacing property is important for numerical schemes, since the actual eigenvalues may be computed conveniently by a root finder. The bounds for the eigenvalues, i.e, the characteristic speeds of the system, are also important for numerical purposes as we will see later on in this thesis.

This information eventually permits us to numerically calculate the eigenvalues and corresponding eigenvectors of $f'(\Phi)$ with acceptable effort. The full spectral decomposition of $f'(\Phi)$, which can be numerically computed at each cell interface thanks to the analysis in [23], can be used in order to obtain characteristic-based WENO schemes, for which, the WENO reconstruction procedure is applied to the local characteristic variables and fluxes at each cell-interface.

Shallow water equations

The shallow water equations model the propagation of disturbances in water and other incompressible fluids. The underlying assumption is that the depth of the fluid is small compared to the wave length of the disturbance. The equations are derived from the principles of conservation of mass and conservation of momentum. The independent variables are time, t, and two space coordinates, x and y. The dependent variables are the fluid height or depth, h, and the two-dimensional fluid velocity field, $v = (v^x, v^y)$. With the proper choice of units, the conserved quantities are mass, which is proportional to h, and momentum, which is proportional to $q = (q^x, q^y) := (hv^x, hv^y)$. The force acting on the fluid is gravity, represented by the gravitational constant g.

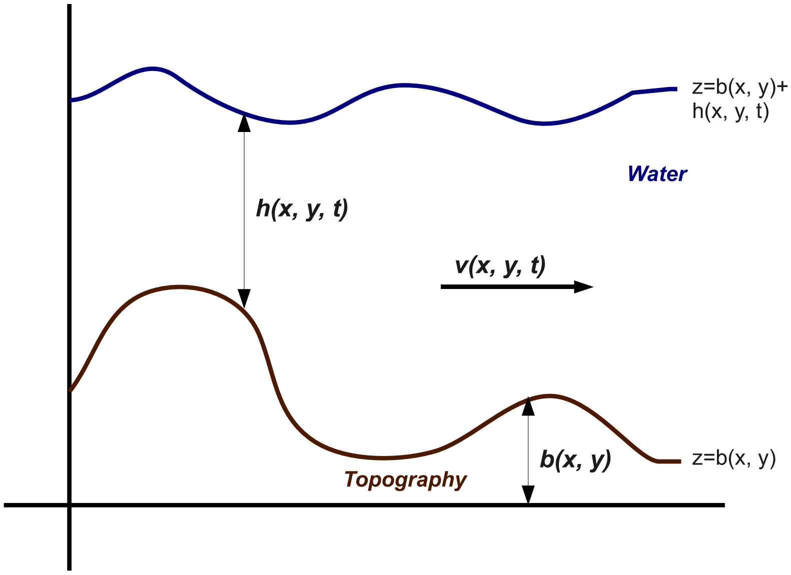


Figure 2.1: *Flow with free surface under gravity, for a fixed section y.*

The two-dimensional shallow water equations represent mass and momentum conservation in two-dimensional domains. They are derived by depth averaging the Navier-Stokes equations, neglecting diffusion of momentum by viscous and turbulent effects and not including wind effects or Coriolis force terms. Ignoring also friction losses, the source term is only due to the geometry of the bottom topography or bathymetry, and the resulting system of equations becomes

$$u_t + f_1(u)_x + f_2(u)_y = s(x, u)$$

or using coordinates,

$$\begin{pmatrix} h \\ q^x \\ q^y \end{pmatrix} + \begin{pmatrix} q^x \\ \frac{(q^x)^2}{h} + \frac{gh^2}{2} \\ \frac{q^x q^y}{h} \end{pmatrix}_x + \begin{pmatrix} q^y \\ \frac{q^x q^y}{h} \\ \frac{(q^y)^2}{h} + \frac{gh^2}{2} \end{pmatrix}_y = \begin{pmatrix} 0 \\ -ghz_x \\ -ghz_y \end{pmatrix}, \qquad (2.24)$$

where z denotes the bottom topography. This is a two-dimensional hyperbolic system of conservation laws with source terms. The corresponding eigenvalues (characteristic velocities) of the Jacobian matrices of the flux components f_1 and f_2 are:

$$\lambda_1^{(1)} = v^x - c \quad \lambda_2^{(1)} = v^x \quad \lambda_3^{(1)} = v^x + c$$

$$\lambda_1^{(2)} = v^y - c \quad \lambda_2^{(2)} = v^y \quad \lambda_3^{(2)} = v^y + c$$

where $c = \sqrt{gh}$ is the sound velocity.

3

Numerical methods for fluid dynamics

After recalling the main features of hyperbolic systems of conservation laws in chapter 2, pointing out that such equations are in general impossible to solve analytically, except in some trivial cases, like the linear advection equation presented in section 2.3.1, in this section we are going to revise the basic concepts and results related to numerical methods for hyperbolic systems of conservation laws, paying special attention to those that will be employed later on in this thesis.

Numerical methods aim to obtain a discrete approximation of the exact solution, which is often sufficient for practical applications. For simplicity, we will center our description in one-dimensional scalar equations, with some notions on the application to one-dimensional systems. We refer the reader to the basic textbooks of LeVeque [70, 72] and Toro [98] for a more detailed description of numerical solution of hyperbolic PDEs.

3.1 Discretization

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,

$$\begin{cases} u_t + f(u)_x = 0, \quad x \in \mathbb{R}, \quad , t \in \mathbb{R}^+, \\ u(x,0) = u_0(x), \end{cases} \tag{3.1}$$

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 by:

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

It is clear that each cell is a subinterval whose center is x_j. For convenience, we will use discrete grids indexed as:

$$x_j = \left(j + \frac{1}{2}\right)\Delta x.$$

We will use non-integer indexes to indicate points that do not correspond to nodes. For example the point $x_{j+\frac{1}{2}}$ represents the point $(j+1)\Delta x$.

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 (3.1).

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 N and M and define a set of nodes $\{x_j\}_{0\leq j<N}$ given by $x_j = (j+1/2)\Delta x$, with $\Delta x = \frac{1}{N}$. The points in time $\{t^n\}_{0\leq n<M}$ can be defined by $t^n = n\Delta t$, with $\Delta t = \frac{1}{M}$.

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. Using the same convention as in the 1D case for the indices, we have that

$$x_i = \left(i + \frac{1}{2}\right)\Delta x, \quad y_j = \left(j + \frac{1}{2}\right)\Delta y.$$

The Cartesian product of $\{x_i\}$ and $\{y_j\}$ defines nodes in 2D by

$$(x_i, y_j) = \left(\left(i + \frac{1}{2}\right)\Delta x, \left(j + \frac{1}{2}\right)\Delta y\right),$$

and 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 $c_{i,j}$.

In practice, the discretization is performed on a bounded subset of $\mathbb{R}\times\mathbb{R}$ (a rectangle) and, as in the 1D case, we can take positive numbers N_x and N_y and define

$$x_i = \left(i + \tfrac{1}{2}\right)\Delta x, \quad 0 \leq i < N_x,$$

$$y_j = \left(j + \tfrac{1}{2}\right)\Delta y, \quad 0 \leq j < N_y,$$

with $\Delta x = \dfrac{1}{N_x}$ and $\Delta y = \dfrac{1}{N_y}$.

3.2 Convergence

From the initial data $u_0(x)$, we can define data U^0, which is the vector of approximations U_j^0 at time $t = 0$. Using a time-marching procedure we can construct the approximation U^1 from U^0, then U^2 from U^1 (and possibly also U^0) and so on. In general, we can construct U^{n+1} from $U^n, U^{n-1}, \dots, U^{n-r}$ with $r \in \mathbb{N}$, $r \leq n$. In this section, we will restrict to one-step explicit time-marching numerical methods, which construct U^{n+1} only from U^n and that can be expressed as

$$U^{n+1} = \mathcal{H}_{\Delta t}(U^n).$$

Note that the value U_j^{n+1} at a particular point j typically depends on a small number of values from the vector U^n (stencil).

Convergence is a condition on the numerical solution that states that the numerical solution U_j^n should approach the exact solution $u_j^n = u(x_j, t^n) = u\left(\left(j + \frac{1}{2}\right)\Delta x, n\Delta t\right)$ of the differential equation at any point x_j and time t^n when Δx and Δt tend to zero, i.e when the mesh is refined, x_j and t^n being fixed. To measure how well the approximations obtained using a numerical method approximate the exact solution of the PDE, we use norms.

We say that a method is convergent in some particular norm $||\cdot||$ if

$$\lim_{\Delta t \to 0, \Delta x \to 0} ||U_j^n - u_j^n|| = 0$$

for any fixed value of x_j and t^n.

Note that the concept of convergence is strongly dependent on the norm. It could happen that some numerical methods converge in one norm but not in another. Often used norms are the discrete L^p norms

$$||u||_p = \left(\Delta x \sum_{j \in \mathbb{Z}} |u_j|^p\right)^{\frac{1}{p}},$$

and the discrete L^∞ norm

$$||u||_\infty = \max_{j \in \mathbb{Z}} |u_j|.$$

In this work we will use L^1, L^2 and L^∞ norms.

It is generally hard to show that a given numerical method is convergent in a given norm using the definition of convergence. The way in which one usually studies convergence is through the concepts of consistency and stability, making use of the Lax equivalence theorem.

Consistency studies how a numerical scheme behaves locally, i.e. in a single time step. To prove consistency we are going to define first the local truncation error, which is the error produced by a single application of the numerical method.

Given a one-step numerical method $U^{n+1} = \mathcal{H}_{\Delta t}(U^n)$ we can define the local truncation error as:

$$L^n_{\Delta t} = \frac{1}{\Delta t}\left(u^{n+1} - \mathcal{H}_{\Delta t}(u^n)\right).$$

where $u^n = (u(x_j, t_n))_j$ are the values of the exact solution of the PDE on the grid at $t = t_n$.

We say that the order of the method is p if

$$L_{\Delta t}(\cdot, t) = \mathcal{O}(\Delta t^p).$$

If $p \geq 1$ then the method is said to be consistent.

While consistency is a condition on the numerical scheme, stability states a condition on the numerical solution. The stability condition can be formulated by the requirement that any component of the initial solution $u(x, 0)$ should not be amplified without bound, at fixed values of t^n, in particular for $n \to \infty$, with $n\Delta t$ fixed.

A necessary condition for stability is the Courant-Friedrichs-Lewy (CFL) condition, stated by Courant, Friedrichs and Lewy in [30]. In their work, the authors recognized that a necessary stability condition for any numerical method is that the domain of dependence of the finite-difference method should include the domain of dependence of the PDE, at least in the limit as the grid is refined.

The numerical domain of dependence for a particular method, $\mathcal{D}_k(\overline{x}, \overline{t})$, is the set of points x for which the initial data $u_0(x)$ could possibly affect the numerical solution at $(\overline{x}, \overline{t})$, while the domain of dependence of the point $(\overline{x}, \overline{t})$, $\mathcal{D}(\overline{x}, \overline{t})$, is similarly defined as the set of points corresponding to time $t = 0$ that completely determine the solution of problem (3.1) at $(\overline{x}, \overline{t})$. So, the CFL condition simply states that the numerical method has to be able to take into account the information coming from any point that is actually influencing the solution at the next time step and that it must be used in such a way that the information has a chance to propagate at the correct physical speeds.

The concepts of stability, consistency and convergence are related by the Equivalence Theorem of Lax [68], a proof of which can be found in [91].

Theorem 1. *For a consistent one step linear scheme for a Cauchy problem of a well-posed linear PDE, stability is a necessary and sufficient condition to convergence.*

More general results relating consistency, stability and convergence can be found in [68] as well.

3.3 Numerical methods

3.3.1 Elementary methods

There are a huge variety of time-marching difference methods that can be used to compute approximations to the solution of conservation laws. Most of them are based on the substitution of the partial derivatives present in equation (2.1) by suitable finite-difference approximations. This is probably the simplest method to apply, but it requires the mesh to be set up in a structured way.

Let us consider, for simplicity, the scalar advection equation in one space dimension

$$u_t + au_x = 0, \quad x \in \mathbb{R}, \quad , t \in \mathbb{R}^+, \quad a \in \mathbb{R}. \tag{3.2}$$

For example, if we substitute the time derivative u_t in (3.2) by a first-order forward-in-time approximation and the spatial derivative u_x by a second-order central approximation, with the notation introduced in section 3.1, we obtain an explicit method that can be written in the form:

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + a\frac{U_{j+1}^n - U_{j-1}^n}{2\Delta x} = 0,$$

or

$$U_j^{n+1} = U_j^n - a\frac{\Delta t}{2\Delta x}\left(U_{j+1}^n - U_{j-1}^n\right)$$

This is an explicit scheme, since the discretized equation contains only one unknown at level $n+1$.

As the time derivative has been substituted by a first-order approximation and the space derivative has been approximated by a second-order finite-difference approximation, consequently, the method is globally first order accurate.

First-order methods give poor accuracy in smooth regions of the flow and suffer from diffusion: shocks tend to be heavily smeared and poorly resolved on the mesh.

Using different finite-difference approximations we can devise a unlimited number of possible finite-difference schemes. These schemes will have different properties, in terms of accuracy, stability or error properties. For instance, the classical methods of Lax and Wendroff [64], based on Taylor series expansion, or Beam and Warming [100], which is a one-sided version of Lax-Wendroff scheme, are second (or higher) order accurate elementary methods in both time and space. In general, these methods are not efficient when the solution is not smooth. They typically show spurious oscillations near the discontinuities which do not decrease as the grid size does. In most cases, it is due to the lack of numerical dissipation in the solution.

3.3.2 Conservative methods

If some singularity is present in the flow solution $u(x,t)$, then finite-differences can not approximate accurately the partial derivatives present in the PDEs. The methods described in section 3.3.1 are based on the assumption that finite-differences can approximate accurately the partial derivatives but this is only true in the points where the flow solution $u(x,t)$ is smooth with respect to (x,t).

When we deal with discontinuous solutions, as mentioned in section 2.2, 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 [70]. 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.

Definition 4. *A numerical method is said to be conservative if it can be written in the form*

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x}\left(\hat{f}(U_{j-p+1}^n, \dots, U_{j+q}^n) - \hat{f}(U_{j-p}^n, \dots, U_{j+q-1}^n)\right), \tag{3.3}$$

where the function $\hat{f} : \mathbb{R}^{p+q+1} \to \mathbb{R}$ 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 (3.3) 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 5. *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,*

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

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 converge to some function $u(x,t)$ as the grid is refined, then this function will be a weak solution of the conservation law:

Theorem 2. *(Lax-Wendroff, [69]) Consider a sequence of grids indexed by $k = 1, 2, \ldots$ 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 numerical solution obtained by a conservative numerical method, consistent with (2.1), on the k-th grid. If $U_k(x,t)$ converges to a function $u(x,t)$ as $k \to \infty$, then u is a weak solution of the conservation law.

The original definition of convergence stated in the theorem can be found in the work of Lax and Wendroff [69], but it has been relaxed and extended to more general grids, see e.g. [38, 60].

3.3.3 High-resolution conservative methods

The term “high-resolution” is applied to methods whose objective is to achieve high-order resolution, typically second or even higher order in smooth parts of the solution, while giving well-resolved non-oscillatory approximations near discontinuities.

Godunov’s method [44] is a first order accurate method based on the computation of Cauchy problems located in each cell interface, assuming that the solution is constant at each side of the interface and taking the cell-average values of the numerical solution corresponding to the cells at the left and right of the interface as initial data.

In Godunov’s method, for a given time step t^n we find, for each j, the exact solution at time t^{n+1} of (3.1) with initial data given by the Riemann problem

$$u(x,t^n)=\begin{cases} U_j^n & \text{if } x_{j-\frac{1}{2}}<x\leq x_{j+\frac{1}{2}}\\ U_{j+1}^n & \text{if } x_{j+\frac{1}{2}}<x\leq x_{j+\frac{3}{2}}\end{cases}$$

Solutions of Riemann problems corresponding to adjacent cell interfaces will not interact for short enough time, due to the finite speed of propagation of information along characteristics. Once these Riemann problems are solved, the solution is averaged on each cell to raise a new Riemann problem for the next time step.

The idea of solving Riemann problems forward in time is at the basis of modern high-resolution shock-capturing methods. A common practice to construct 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 [99]).

From the numerical solution at a given time step one reconstructs, by a certain interpolation or approximation procedure, two values $U_{j+\frac{1}{2}}^L$ and $U_{j+\frac{1}{2}}^R$ at each interface. Then the Riemann problem with initial data

$$u(x,t^n)=\begin{cases} U_{j+\frac{1}{2}}^L & \text{if } x_{j-\frac{1}{2}}<x\leq x_{j+\frac{1}{2}},\\ U_{j+\frac{1}{2}}^R & \text{if } x_{j+\frac{1}{2}}<x\leq x_{j+\frac{3}{2}},\end{cases}$$

is solved.

To achieve higher order some techniques have been developed as the essentially non-oscillatory (ENO) methods, introduced by Harten, Enguist, Osher and Chakravarthy in [51] and the weighted essentially non-oscillatory (WENO) methods [59, 78], explained in more detail in section 3.4.2.

3.3.4 Semi-discrete formulation

The methods previously presented have all been fully discrete methods, discretized in both space and time. Let us now consider the discretization process in two stages: 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

$$\frac{dU_j(t)}{dt} + \mathcal{D}(U(t))_j = 0, \quad \forall j, \tag{3.4}$$

where $\mathcal{D}(U(t))_j$ is some approximation of the spatial derivative $f(u)_x(x_j, t)$. This approach of reducing a PDE to a system of ODEs is known as the “method of lines”.

If we compute the spatial approximation using a conservative reconstruction of the numerical fluxes, we can rewrite the ODE system (3.4) as:

$$\frac{dU_j(t)}{dt} + \frac{\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}}{\Delta x} = 0, \quad \forall j, \tag{3.5}$$

where $\hat{f}_{j+\frac{1}{2}} = \hat{f}(U_{j-p+1}(t), \dots, U_{j+q}(t))$.

After that, we solve the system of ordinary differential equations (3.5) using an ODE solver. The ODE solver that is used in the computations performed in this thesis is a TVD Runge-Kutta method developed by Shu and Osher in [94]. This solver belongs to a class of ODE solvers especially designed to solve this kind of ODE systems. 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 [94, 95]). Specifically, in this work we use the third-order version:

$$\begin{cases} U^{(1)} = U^n - \Delta t \mathcal{D}(U^n), \\ U^{(2)} = \dfrac{3}{4}U^n + \dfrac{1}{4}U^{(1)} - \dfrac{1}{4}\Delta t \mathcal{D}(U^{(1)}), \\ U^{n+1} = \dfrac{1}{3}U^n + \dfrac{2}{3}U^{(2)} - \dfrac{2}{3}\Delta t \mathcal{D}(U^{(2)}). \end{cases} \tag{3.6}$$

If we use this TVD Runge-Kutta method as a ODE solver together with spatial operators that lead to ODE's of the form (3.5), then we obtain conservative schemes that can be expressed in the conservative form (3.3). For example, if we expand (3.6) 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

$$\begin{aligned} U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x} \Bigg[& \left(\frac{1}{6}\hat{f}_{j+\frac{1}{2}}(U^n) + \frac{1}{6}\hat{f}_{j+\frac{1}{2}}(U^{(1)}) + \frac{2}{3}\hat{f}_{j+\frac{1}{2}}(U^{(2)}) \right) \\ & - \left(\frac{1}{6}\hat{f}_{j-\frac{1}{2}}(U^n) + \frac{1}{6}\hat{f}_{j-\frac{1}{2}}(U^{(1)}) + \frac{2}{3}\hat{f}_{j-\frac{1}{2}}(U^{(2)}) \right) \Bigg], \end{aligned} \tag{3.7}$$

Since $U^{(1)}$ and $U^{(2)}$ are obtained from U^n we can write (3.7) 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

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

3.4 Finite-difference WENO schemes

3.4.1 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 [95]. The basic idea that makes possible Shu-Osher's approach is stated in the following lemma:

Lemma 1. *If the functions G, f satisfy*

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

then

$$G(x)_x = \frac{f\left(x+\frac{\Delta x}{2}\right) - f\left(x-\frac{\Delta x}{2}\right)}{\Delta x}.$$

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 f as:

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

so that the spatial derivative in

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

(we warn the reader for this slight change of notation) is exactly obtained by a conservative finite-difference formula at the cell boundaries,

$$u_t(x,t) + \frac{1}{h}\left(f\left(x+\frac{h}{2},t\right) - f\left(x-\frac{h}{2},t\right)\right) = 0.$$

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

$$(F(u))_x(x_{j+\frac{1}{2}}) = \frac{\widehat{f}(x_{j+\frac{1}{2}}) - \widehat{f}(x_{j-\frac{1}{2}})}{\triangle x} + \mathcal{O}(h^r).$$

We denote as $\mathcal{R}(f_{j-s_1}, \ldots, f_{j+s_2}, x)$ the generic local reconstruction procedure for $f(x)$ from its cell-averages $\{\bar{f}_{j-s_1}, \ldots, \bar{f}_{j+s_2}\}$ defined on the interval $[x_{j-s_1-\frac{1}{2}}, x_{j+s_2+\frac{1}{2}}]$, 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 $p>0$:

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

  where the total variation of a differentiable function $h(x)$ in an interval I is defined as

  $$TV(\varphi)=\int_I|\varphi'(x)|dx.$$

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 $sign(f'(u)) > 0$
    $\hat{f}_{j+\frac{1}{2}} = \mathcal{R}^+(f_{j-s_1}, \dots, f_{j+s_2}, x_{j+\frac{1}{2}})$
  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 [94] use a local Lax-Friedrichs (LLF) flux-splitting version of the ENO algorithms.

The generalization of this algorithm to systems of equations uses local characteristic decompositions of the flux Jacobians and projections of the state variables and fluxes onto characteristic fields.

3.4.2 WENO reconstruction method

Essentially Non Oscillatory (ENO) schemes were introduced by Harten et al. in [51]. For these schemes, a given cell interface reconstruction is obtained by selecting 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 chosen stencil was selected according to the smoothness of the numerical solution on it, in such a way that 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. The selection procedure is computationally expensive, since it involves a lot of conditional branches. ENO schemes are potentially inefficient since a large amount of information is simply discarded.

Weighted Essentially Non Oscillatory (WENO) reconstructions, introduced by Liu, Osher and Chan in [78], are based on the idea of increasing the order of accuracy of the method (at least 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 [78], 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 an 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 [59] to achieve fifth-order reconstructions from third-order ENO reconstructions, i.e. an order of $2r-1$ when $r=3$. These smoothness indicators are used by Balsara and Shu in [11], resp. by Gerolymos et al. [43], to obtain $(2r-1)$-th order accurate reconstructions from the basic r-th order ENO reconstructions by using symbolic calculus for each $4 \leq r \leq 6$ and $7 \leq r \leq 9$ respectively.

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

In the ENO algorithm [51], if we assume a left bias, an approximation to the value $f\left(\frac{h}{2}\right)$ 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

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

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

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

if f is smooth in the corresponding stencil.

Among all the possible reconstructions the ENO algorithm selects one, using divided differences as smoothness indicators, choosing the stencil which produces the smallest divided differences, in an attempt for producing less oscillatory interpolants, see [4, 51] for further details. When using stencils of r nodes, ENO reconstructions provide an order of accuracy of r, except in those subintervals containing singularities. The polynomial reconstruction $p_k^r(x_{j+\frac{1}{2}})$ would be the approximation of the numerical flux computed by the ENO algorithm if the stencil S_k had been chosen in the stencil selection procedure.

Weighted ENO reconstructions appeared in [78] as an improvement upon ENO reconstructions. In [78], 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. In the most favorable case, where f is smooth in all stencils, a $(2r-1)$-th order reconstruction

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

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

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

$$q(x_{j+\frac{1}{2}}) = \sum_{k=0}^{r-1} w_k p_k^r(x_{j+\frac{1}{2}}), \tag{3.8}$$

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:

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

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 [78], we first note that for $r \geq 2$, coefficients C_k^r, called optimal weights, can be computed such that:

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

where,

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

In [3], 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:

$$w_k = C_k + \mathcal{O}(h^m), \qquad k = 0, \dots, r, \tag{3.9}$$

with $m \leq r - 1$. Then, there holds (see [3], [78]) that

$$f(x_{j+\frac{1}{2}}) - q(x_{j+\frac{1}{2}}) = \mathcal{O}(h^{r+m}), \tag{3.10}$$

and, if $m = r - 1$ in (3.9), then the approximation (3.10) 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 [78] as follows:

$$w_k = \frac{\alpha_k}{\sum_{i=0}^{r-1} \alpha_i}, \quad \alpha_k = \frac{C^r_k}{(\varepsilon + I_k)^p}, \quad k = 0, \dots, r-1, \tag{3.11}$$

where $p \in \mathbb{N}$, C^r_k 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.

In the original WENO paper [78] the authors used an smoothness indicator based on the undivided differences of the function f. With this indicator, an increase of one order of accuracy was obtained upon the ENO reconstruction. Jiang and Shu defined in [59] the following smoothness indicator:

$$I_k = \sum_{l=1}^{r-1} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} h^{2l-1} (p_k^{(l)}(x))^2 dx, \tag{3.12}$$

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 $p_k(x)$.

We denote by JS-WENO the WENO schemes obtained with weights (3.11) and Jiang and Shu's smoothness indicators (3.12).

In [3], the authors give an explicit formulae for the optimal weights C_k^r:

$$C_k^r = \frac{\binom{r-1}{k}\binom{r}{k}}{\binom{2r-1}{r}},$$

and for the polynomials $p_k^r(x)$:

$$p_k^r(x_{j+\frac{1}{2}}) = \sum_{l=1}^{r} \bar{f}_{j+k-r+l} a_{k,l}^r,$$

$$a_{k,l}^r = \begin{cases} -\binom{r}{k}^{-1} \displaystyle\sum_{s=k+l-r}^{k} \frac{(-1)^s}{s} \binom{r}{k-s}, & k+l-r > 0, \\ \\ -\binom{r}{k}^{-1} \displaystyle\sum_{s=r-k-l+1}^{r-k} \frac{(-1)^s}{s} \binom{r}{k+s}, & k+l-r \leq 0, \end{cases}$$

for $k = 0, \dots, r-1$.

The optimal weights for $r = 2, 3, 4, 5$ obtained using this explicit formulae, are displayed in Table 3.1.

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 3.1: *Optimal weights for* $r = 2, 3, 4, 5$.

In [3], 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 [78] (given by (3.11)), the choice of ε being proportional to h^2 is optimal.

4

Weights design for maximal-order WENO schemes

4.1 Motivation

As mentioned before, weighted essentially non-oscillatory (WENO) technique, proposed by Liu, Osher and Chan in [78] and improved by Jiang and Shu in [59], uses a convex combination of the different polynomial reconstruction candidates of the ENO method, instead of selecting one of them. A weight, which depends on the smoothness of the function on the corresponding stencil, is assigned to each polynomial reconstruction, so that polynomials corresponding to singularity-crossing stencils should have a negligible contribution to the convex combination. Jiang and Shu

[59] presented an smoothness measurement of the reconstructed function which is more efficient than that proposed by Liu et al. Using the indicator of smoothness introduced by Liu et al. in [78], an $r-$th order ENO scheme can be converted into an $(r+1)-$th order WENO scheme, whereas Jiang and Shu's smoothness indicators provide the maximal order, $2r-1$, which can be attained with $2r-1$ data.

But, as has been shown in [19, 54, 104], the classical weight functions for the fifth-order WENO scheme fail to provide the maximal order of convergence near smooth extrema, where the first derivative of the solution becomes zero. Recently, new approaches have been proposed to solve this problem. In [54] a simple modification of the original scheme is found to be sufficient to give maximal-order convergence even near critical points. In [19] new weights are built for the fifth-order WENO scheme, providing new WENO schemes with less dissipation and higher resolution than classical WENO schemes. In [104], Yamaleev and Carpenter propose new weights to provide faster error convergence than those presented in [19], and find some constraints on the weights parameters to guarantee that the WENO scheme has maximal order for sufficiently smooth solutions with an arbitrary number of vanishing derivatives.

In this chapter we analyze the structure of the new weights proposed in [104] and we prove that near discontinuities the scheme drops to first order, instead of achieving the same order as the classical ENO scheme does. We also study the role of the parameter ε appearing in the definition of the weights to avoid null denominators and its relationship with the loss of accuracy near discontinuities and extrema of the reconstructed functions and we also find some constraints on this parameter in order to guarantee maximal order of accuracy in smooth regions, even at extrema. Finally, we solve these accuracy problems by deriving new weights from those developed in [104] and getting some constraints on the parameter ε to guarantee that the new WENO scheme has maximal order for sufficient smooth solutions with an arbitrary number of vanishing derivatives. Furthermore, we present some numerical experiments that support our theoretical results.

4.2 Maximal-order WENO schemes

In [54], it was detected that the classical fifth-order WENO scheme (obtained with $r=3$), called JS-WENO5, achieves only third order of accuracy at critical points of smooth solutions. In [3] Aràndiga et al. prove that if some requirements on the smoothness indicators are stated, then we obtain the desired accuracy features for the WENO interpolation. They prove that when the function to be reconstructed is smooth at all the stencils (we say that a function is smooth at an stencil when it is so at some interval containing it) and some requirements on the smoothness indicators are met, then the JS-WENO scheme has maximal order $2r-1$, regardless of neighboring extrema. But, if the function is smooth at one, but not all, of the stencils, then order r is achieved, as ENO reconstructions have.

In order to solve this loss of accuracy at extrema, Henrick et al. define a new WENO method called mapped WENO in [54]. In their work, instead of formulating a new indicator of smoothness to obtain fifth-order accurate schemes near critical points, they define new weights using Jiang and Shu's weights, denoted as $w_k^{(JS)}$, as an initial guess which is mapped to a more accurate value by using the functions

$$g_k(w) = \frac{w(\overline{w}_k + \overline{w}_k^2 - 3\overline{w}_k w + w^2)}{\overline{w}_k^2 + w(1-2\overline{w}_k)},$$

where $\overline{w}_k \in (0,1)$ for $k = 0,1,2$. Then, a more accurate approximation of the weights is given by $\alpha_k = g_k(w_k^{(JS)})$. By using these mapped weights, they obtain that the method is fifth-order accurate even near critical points where $f' = 0$.

Another attempt to get maximal order of accuracy for the WENO5 scheme can be found in [19], where Borges et al. devise a new smoothness indicator of higher order than Jiang and Shu's smoothness indicator and build new non-oscillatory weights, providing a new WENO scheme for $r = 3$, called WENO-Z, with less dissipation and higher resolution than the classical WENO5. The novel idea is to use the whole 5-points stencil to devise a new smoothness indicator of higher order than the classical smoothness indicators I_k. They define τ_5 as the absolute difference between I_0 and I_2 at x_i, namely $\tau_5 = |I_0 - I_2|$, and then define the new smoothness indicators I_k^Z as

$$I_k^Z = \frac{I_k + \varepsilon}{I_k + \tau_5 + \varepsilon}, \quad k = 0,1,2.$$

and the new WENO weights w_k^Z as

$$w_k^Z = \frac{\alpha_k^Z}{\sum_{l=0}^{2} \alpha_l^Z}, \quad \alpha_k^Z = C_k \left(1 + \frac{\tau_5}{I_k + \varepsilon}\right), \quad k = 0,1,2.$$

All I_k^Z are smaller than unity and they are all close to 1 at smooth parts of the solution. In fact, they are the normalization of the classical smoothness indicators I_k by the higher order information contained in τ_5.

Although these new weight functions recover the fifth order of convergence of the WENO scheme near smooth extrema, the problem persists if the first- and second-order derivatives vanish simultaneously. An attempt to resolve this loss of accuracy is presented by Borges et al. in [19], where the authors propose to modify the definition of α_k^Z as

$$\alpha_k^Z = C_k \left(1 + \left(\frac{\tau_5}{I_k + \varepsilon}\right)^2\right), \quad k = 0, 1, 2.$$

This proposed modification provides only a partial remedy for the problem; the same degeneration in the order of convergence occurs if at least the first three derivatives become equal to zero.

To fully resolve this problem, Yamaleev and Carpenter proposed in [104, 105] new weights providing faster weight convergence and better resolution near strong discontinuities. They presented these weights in an Energy Stable context that is out of the scope of the present work. The schemes with these weights will be called YC-WENO henceforth. The proposed weights are:

$$w_k = \frac{\alpha_k}{\sum_{i=0}^{r-1} \alpha_i}, \tag{4.1}$$

where

$$\alpha_k = C_k \left(1 + \frac{\tau_{2r-1}}{I_k + \varepsilon}\right), \qquad k = 0, \dots, r-1, \tag{4.2}$$

I_k is the classical Jiang and Shu's smoothness indicator (3.12), ε is a small positive parameter that can depend on h and the quantity τ_{2r-1} is defined by:

$$\tau_{2r-1} = \left(V\langle x_{j-r+1}, \dots, x_{j+r-1}\rangle\right)^2, \tag{4.3}$$

where $V\langle x_{j-r+1}, \dots, x_{j+r-1}\rangle$ is the undivided difference defined on the entire $(2r-1)$-point stencil. For example, the expressions of τ_5 and τ_9, used in WENO5 and WENO9 schemes respectively, are given by

$$\tau_5 = (f_{j-2} - 4f_{j-1} + 6f_j - 4f_{j+1} + f_{j+2})^2 \tag{4.4}$$

$$\tau_9 = (f_{j-4} - 8f_{j-3} + 28f_{j-2} - 56f_{j-1} + 70f_j - 56f_{j+1} + 28f_{j+2} - 8f_{j+3} + f_{j+4})^2$$

Note that these weights are similar to those proposed by Borges et al. in [19] for the fifth-order WENO scheme. The key difference between

Yamaleev and Carpenter's weights and those developed in [19] is the choice of the quantity τ in (4.2).

The fifth WENO scheme with the weights given by Eqs. (4.1)-(4.4) are design-order accurate for smooth solutions, including points at which the first and second-order derivatives of the solution vanish simultaneously. However, if all derivatives up to the third order are equal to zero and no constraint is imposed on the parameter ε, then the fifth-order WENO scheme locally become only third-order accurate.

Yamaleev and Carpenter show that WENO schemes with these weights and parameter ε satisfying:

$$\varepsilon \geq \mathcal{O}\left(h^{3r-4}\right), \tag{4.5}$$

have maximal order regardless of the number of vanishing derivatives of the solution. As the parameter ε is user-defined, this condition can always be satisfied.

We are going to study now the order of accuracy of the reconstructions obtained using the WENO scheme with Yamaleev and Carpenter's weights near discontinuities when the function f is not smooth but it is smooth at least in one of the stencils $S_k,\ k=0,\dots,r-1$. We know that:

- if f is not smooth at the stencil S_k then the smoothness indicator of f in S_k satisfies $I_k \nrightarrow 0,$ when $h \to 0$, i.e, $I_k = \mathcal{O}(1)$,
- whereas if f is smooth at the stencil S_k, then $I_k = \mathcal{O}(h^2)$.

Since the nodes in the undivided difference that defines τ_{2r-1} cross a discontinuity, then $\tau_{2r-1} = \mathcal{O}(1)$. Denoting $\mathcal{K} = \{k/f \text{ not smooth at } S_k\}$, we obtain that:

$$\alpha_k = C_k\left(1+\frac{\tau_{2r-1}}{I_k+\varepsilon}\right) = C_k\left(1+\frac{\mathcal{O}(1)}{\mathcal{O}(1)+\varepsilon}\right) = \mathcal{O}(1), \quad \text{if } k\in\mathcal{K},$$

whereas

$$\alpha_k = C_k\left(1+\frac{\tau_{2r-1}}{I_k+\varepsilon}\right) = C_k\left(1+\frac{\mathcal{O}(1)}{\mathcal{O}(h^2)+\varepsilon}\right) = \mathcal{O}(h^{-2}), \quad \text{if } k\notin\mathcal{K}.$$

This yields

$$\sum_{i=0}^{r-1}\alpha_i = \sum_{k\in\mathcal{K}}\alpha_i + \sum_{k\notin\mathcal{K}}\alpha_i = \mathcal{O}(1)+\mathcal{O}(h^{-2}) = \mathcal{O}(h^{-2}),$$

and

$$w_k = \frac{\alpha_k}{\sum_{i=0}^{r-1}\alpha_i} = \frac{\mathcal{O}(1)}{\mathcal{O}(h^{-2})} = \mathcal{O}(h^2) \quad \text{if } k\in\mathcal{K},$$

whereas

$$w_k = \frac{\alpha_k}{\sum_{i=0}^{r-1} \alpha_i} = \frac{\mathcal{O}(h^{-2})}{\mathcal{O}(h^{-2})} = \mathcal{O}(1) \quad \text{if } k \notin \mathcal{K}.$$

Using that $\sum_{k=0}^{r-1} w_k = 1$, we then deduce:

$$\begin{aligned} f(x_{j+\frac{1}{2}}) - q(x_{j+\frac{1}{2}}) &= f(x_{j+\frac{1}{2}}) - \sum_{k=0}^{r-1} w_k p_k^r(x_{j+\frac{1}{2}}) \\ &= \sum_{k=0}^{r-1} w_k \left(f(x_{j+\frac{1}{2}}) - p_k^r(x_{j+\frac{1}{2}}) \right) \\ &= \sum_{k\notin\mathcal{K}} w_k \left(f(x_{j+\frac{1}{2}}) - p_k^r(x_{j+\frac{1}{2}}) \right) + \sum_{k\in\mathcal{K}} w_k \left(f(x_{j+\frac{1}{2}}) - p_k^r(x_{j+\frac{1}{2}}) \right) \\ &= \sum_{k\notin\mathcal{K}} \mathcal{O}(1)\mathcal{O}(h^r) + \sum_{k\in\mathcal{K}} \mathcal{O}(h^2)\mathcal{O}(1) = \mathcal{O}(h^2) \end{aligned}$$

As we can see the order of accuracy of the reconstructions obtained using the YC-WENO scheme drops to 2 wherever one stencil, but not all of them, can avoid a discontinuity. This order of accuracy is worse than the order of accuracy r of the corresponding ENO scheme when $r > 2$. Furthermore, it is expected that the order of approximation to the derivative will drop to 1 when performing finite differences of the reconstructions, as can be seen in the following experiment, in which we test the performance of the YC-WENO5 reconstruction near discontinuities. This experiment appears in [3], where the authors test the performance of JS-WENO5 reconstructions.

Test 4.1.

We consider the discontinuous function $f(x) = x^3 + cos(x) + H(x)$ where $H(x) = 0$ if $x \leq 0.5$, $H(x) = 1$ if $x > 0.5$, and uniform grids on $[-1, 1]$ with $N = 25 \cdot 2^i,\ i = 0, \ldots, 7$. We compute the errors of the approximations of $f'(x_{j\pm1})$ provided by the YC-WENO5 reconstructions with $r = 3$ at the points $x_{j\pm1}$, where x_{j-1} is at the left part of the discontinuity and x_{j+1} is at the right part of the discontinuity, $0.5 \in [x_j, x_{j+1})$.

In this experiment we use $\varepsilon = 10^{-100} \approx 0$ and $\varepsilon = h^2$ and respectively display in Tables 4.1 and 4.2 the errors $e_{j\pm1}$ for the corresponding h. We also display the experimentally observed convergent rates $cr_{j\pm1}(h) = \log_2 \left(e_{j\pm1}(h)/e_{j\pm1}(h/2) \right)$ to reveal that the convergence rate of the YC-WENO5 reconstructions drops to 1 when using divided differences, as we expected, while the convergence rate of the JS-WENO5 reconstructions is 2, as can be seen in [3]. The numerical results in

Section 4.4 will show that this order loss may be reflected as oscillations near some discontinuities.

h	e_{j-1}	cr_{j-1}	e_{j+1}	cr_{j+1}
8.000e-02	-2.9427e-03		-1.6680e-03	
4.000e-02	2.9280e-03	7.2249e-03	-1.0310e-03	6.9407e-01
2.000e-02	3.1714e-03	-1.1520e-01	-6.8904e-04	5.8138e-01
1.000e-02	1.7614e-03	8.4840e-01	-3.3584e-04	1.0368e+00
5.000e-03	9.2775e-04	9.2491e-01	-1.6846e-04	9.9537e-01
2.500e-03	4.7607e-04	9.6256e-01	-8.4671e-05	9.9247e-01
1.250e-03	2.4114e-04	9.8130e-01	-4.2481e-05	9.9505e-01
6.250e-04	1.2135e-04	9.9070e-01	-2.1281e-05	9.9725e-01

Table 4.1: *Results of accuracy test 4.1 with* $\varepsilon = 10^{-100}$ *for YC-WENO5.*

h	e_{j-1}	cr_{j-1}	e_{j+1}	cr_{j+1}
8.000e-02	1.9614e-01		-3.8327e-02	
4.000e-02	1.0758e-01	8.6647e-01	-1.9529e-02	9.7274e-01
2.000e-02	5.6185e-02	9.3715e-01	-9.9767e-03	9.6898e-01
1.000e-02	2.8386e-02	9.8501e-01	-4.9889e-03	9.9984e-01
5.000e-03	1.4260e-02	9.9321e-01	-2.4969e-03	9.9858e-01
2.500e-03	7.1465e-03	9.9667e-01	-1.2493e-03	9.9902e-01
1.250e-03	3.5772e-03	9.9841e-01	-6.2493e-04	9.9935e-01
6.250e-04	1.7895e-03	9.9927e-01	-3.1253e-04	9.9970e-01

Table 4.2: *Results of accuracy test 4.1 with* $\varepsilon = h^2$ *for YC-WENO5.*

4.3 New weights for maximal-order WENO schemes

The analytical and numerical results obtained in Section 4.2 show that YC-WENO schemes achieve maximal order of accuracy when the function is smooth but the results could be improved near discontinuities. To solve these accuracy problems we propose new WENO weights, based on Yamaleev and Carpenter's weights, that yield maximal order of accuracy when the function is smooth and provide higher order of accuracy than the order of accuracy provided using Yamaleev and Carpenter's weights, when the function is not smooth. As we will see, these weights also reduce the spurious oscillations that appear near discontinuities.

The new WENO weights that we propose, named AMM-WENO henceforth, are defined by:

$$w_k = \frac{\alpha_k}{\sum_{i=0}^{r-1}\alpha_i}, \quad \alpha_k = C_k\left(1+\left(\frac{\tau_{2r-1}}{I_k+\varepsilon}\right)^{\mu}\right), \tag{4.6}$$

$$\mu = \left\lceil \frac{r}{2} \right\rceil, \quad k = 0, \dots, r-1, \tag{4.7}$$

where I_k is the classical Jiang and Shu's smoothness indicator, ε is a small positive parameter and the quantity τ_{2r-1} is the square of the undivided difference defined on the entire $(2r-1)-$point stencil. It is worth noticing that $\mu = 1$ gives Yamaleev and Carpenter's weights and that our choice yields $2\mu \geq r$.

The notation $\lceil \cdot \rceil$ denotes the ceiling function which maps a real number to the next integer.

With an analysis as in [3, Proposition 3], we can state some constraints on the order of the parameter ε to guarantee maximal-order WENO methods when we use the modified weights proposed above. These constraints are less restrictive than Yamaleev and Carpenter's restriction (4.5).

Proposition 1. *Let $\varepsilon = Kh^q$ with $K > 0$, $q \in \mathbb{N}$,*

$$q \leq 4r - 4 - \frac{r}{\mu}, \tag{4.8}$$

and $\mu = \left\lceil \frac{r}{2} \right\rceil$. The WENO reconstruction of f is defined by:

$$q(x) = \sum_{k=0}^{r-1} w_k p_k^r(x), \tag{4.9}$$

where

$$w_k = \frac{\alpha_k}{\sum_{i=0}^{r-1}\alpha_i} \quad k = 0, \dots, r-1 \text{ and } \alpha_k = C_k^r\left(1+\left(\frac{\tau_{2r-1}}{\varepsilon+I_k}\right)^{\mu}\right).$$

Then:

1. *At regions where f is smooth:*

$$w_k = C_k^r(1+\mathcal{O}(h^r)), \qquad 0 \leq k < r,$$

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

for a locally Lipschitz function d*.*

2. *If f is not smooth but it is smooth at least in one of the stencils* $S_k,\ k = 0,\dots,r-1$*, then:*

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

Proof. Let us suppose that the function f is smooth, then $\tau_{2r-1} = \mathcal{O}\left(h^{2(2r-2)}\right)$. With this we have:

$$\begin{aligned}\alpha_k &= C_k^r\left(1+\left(\frac{\tau_{2r-1}}{\varepsilon+I_k}\right)^{\mu}\right) \leq C_k^r\left(1+\left(\frac{\tau_{2r-1}}{\varepsilon}\right)^{\mu}\right) = \\ &= C_k^r\left(1+\left(\frac{\mathcal{O}(h^{2(2r-2)})}{Kh^q}\right)^{\mu}\right) = C_k^r\left(1+\mathcal{O}\left(h^{(4r-4-q)\mu}\right)\right).\end{aligned}$$

To get $(4r-4-q)\mu \geq r$, we deduce that $q \leq 4r-4-\dfrac{r}{\mu}$.

It follows that if q satisfies this bound then $\alpha_k = C_k^r\left(1+\mathcal{O}(h^r)\right)$, therefore

$$\sum_{i=0}^{r-1}\alpha_i = \sum_{i=0}^{r-1} C_k^r + \mathcal{O}\left(h^r\right) = 1+\mathcal{O}\left(h^r\right),$$

$$w_k = \frac{\alpha_k}{\sum_{i=0}^{r-1}\alpha_i} = \frac{C_k^r\left(1+\mathcal{O}(h^r)\right)}{1+\mathcal{O}(h^r)} = C_k^r\left(1+\mathcal{O}(h^r)\right). \tag{4.10}$$

Now if F is a primitive of f and P is an interpolator of F at $\{x_{j-r+\frac{1}{2}},\dots,x_{j+r-\frac{1}{2}}\}$, then $P' = p_{r-1}^{2r}$ and we can deduce that

$$f(x_{j+\frac{1}{2}}) - p_{r-1}^{2r}(x_{j+\frac{1}{2}}) = bf^{(2r-1)}(x_{j+\frac{1}{2}})h^{2r-1} + \mathcal{O}(h^{2r}), \quad b \neq 0. \tag{4.11}$$

We know that the $r-$th order polynomial reconstruction defined on the stencil S_k, $p_k^r(x)$, satisfies

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

Using this, (4.9), (4.10), (4.11) and $\sum\limits_{k=0}^{r-1}(C_k^r - w_k) = 0$, we get:

$$
\begin{aligned}
&f(x_{j+\frac{1}{2}}) - q(x_{j+\frac{1}{2}}) = f(x_{j+\frac{1}{2}}) - p_{r-1}^{2r-1}(x_{j+\frac{1}{2}}) + p_{r-1}^{2r-1}(x_{j+\frac{1}{2}}) - q(x_{j+\frac{1}{2}})\\
&= bf^{(2r-1)}(x_{j+\frac{1}{2}})h^{2r-1} + \mathcal{O}(h^{2r}) + \sum_{k=0}^{r-1}(C_k^r - w_k)\, p_k^r(x_{j+\frac{1}{2}})\\
&= bf^{(2r-1)}(x_{j+\frac{1}{2}})h^{2r-1} + \mathcal{O}(h^{2r}) + \sum_{k=0}^{r-1}(C_k^r - w_k)\left(p_k^r(x_{j+\frac{1}{2}}) - f(x_{j+\frac{1}{2}})\right)\\
&= bf^{(2r-1)}(x_{j+\frac{1}{2}})h^{2r-1} + \mathcal{O}(h^{2r}) + \sum_{k=0}^{r-1}(C_k^r - C_k^r(1+\mathcal{O}(h^r)))\left(p_k^r(x_{j+\frac{1}{2}}) - f(x_{j+\frac{1}{2}})\right)\\
&= bf^{(2r-1)}(x_{j+\frac{1}{2}})h^{2r-1} + \mathcal{O}(h^{2r}) + \sum_{k=0}^{r-1}(C_k^r - C_k^r - C_k^r\mathcal{O}(h^r))\left(p_k^r(x_{j+\frac{1}{2}}) - f(x_{j+\frac{1}{2}})\right)\\
&= bf^{(2r-1)}(x_{j+\frac{1}{2}})h^{2r-1} + \mathcal{O}(h^{2r}) - \sum_{k=0}^{r-1}C_k^r\mathcal{O}(h^r)\mathcal{O}(h^r)\\
&= bf^{(2r-1)}(x_{j+\frac{1}{2}})h^{2r-1} + \mathcal{O}(h^{2r})
\end{aligned}
$$

with $bf^{(2r-1)}$ a locally Lipschitz function.

Let us now suppose that f has a discontinuity at some, but not all, of the stencils $S_k,\ k = 0,\dots,r-1$. Since $I_k \nrightarrow 0$, when $h \to 0$ for the stencils where f has a discontinuity, whereas $I_k = \mathcal{O}(h^2)$ otherwise, with a similar analysis to the one conducted in the previous section, we deduce that, if f has a discontinuity on S_k then:

$$\alpha_k = C_k\left(1 + \left(\frac{\tau_{2r-1}}{I_k + \varepsilon}\right)^{\mu}\right) = C_k\left(1 + \frac{\mathcal{O}(1)}{\mathcal{O}(1) + \varepsilon}\right)^{\mu} = \mathcal{O}(1),$$

and

$$\alpha_k = C_k\left(1 + \left(\frac{\tau_{2r-1}}{I_k + \varepsilon}\right)^{\mu}\right) = C_k\left(1 + \frac{\mathcal{O}(1)}{\mathcal{O}(h^2) + \varepsilon}\right)^{\mu} = \mathcal{O}(h^{-2\mu}),$$

if f is smooth at S_k.

Therefore, $\sum\limits_{i=0}^{r-1}\alpha_i = \mathcal{O}(h^{-2\mu})$ and:

$$
\begin{aligned}
w_k &= \frac{\alpha_k}{\sum_{i=0}^{r-1}\alpha_i} = \frac{\mathcal{O}(1)}{\mathcal{O}(h^{-2\mu})} = \mathcal{O}(h^{2\mu}) \text{ if } f \text{ has a discontinuity in } S_k,\\
w_k &= \frac{\alpha_k}{\sum_{i=0}^{r-1}\alpha_i} = \frac{\mathcal{O}(h^{-2\mu})}{\mathcal{O}(h^{-2\mu})} = \mathcal{O}(1) \text{ if } f \text{ is smooth in } S_k.
\end{aligned}
$$

If we denote $\mathcal{K} = \{k/f \text{ not smooth in } S_k\}$, and using that $\sum_{k=0}^{r-1} w_k = 1$, then

$$\begin{aligned}
f(x_{j+\frac{1}{2}}) - q(x_{j+\frac{1}{2}}) &= f(x_{j+\frac{1}{2}}) - \sum_{k=0}^{r-1} w_k p_k^r(x_{j+\frac{1}{2}}) \\
&= \sum_{k=0}^{r-1} w_k \left(f(x_{j+\frac{1}{2}}) - p_k^r(x_{j+\frac{1}{2}}) \right) \\
&= \sum_{k\notin\mathcal{K}} w_k \left(f(x_{j+\frac{1}{2}}) - p_k^r(x_{j+\frac{1}{2}}) \right) + \sum_{k\in\mathcal{K}} w_k \left(f(x_{j+\frac{1}{2}}) - p_k^r(x_{j+\frac{1}{2}}) \right) \\
&= \sum_{k\notin\mathcal{K}} \mathcal{O}(1)\mathcal{O}(h^r) + \sum_{k\in\mathcal{K}} \mathcal{O}(h^{2\mu})\mathcal{O}(1) \\
&= \mathcal{O}(h^r)
\end{aligned}$$

since $\mu = \left\lceil \frac{r}{2} \right\rceil$ implies $2\mu \geq r$. □

Note 1. *The parameter ε in Proposition 1 is a dimensional quantity. For finite-difference WENO schemes K should be chosen proportional to the square of some reference flux value, so that the scheme would not be affected by changes of units.*

4.4 Numerical experiments

In this section we illustrate numerically the theoretical results obtained previously. We will also see that Yamaleev and Carpenter's weights may produce oscillations near discontinuities that our proposed weights seem to reduce.

The parameter ε is chosen in order to satisfy both condition (4.5) proposed by Yamaleev and Carpenter for their weights and condition (4.8) stated to guarantee maximal-order WENO methods when using the modified AMM weights. For WENO5 scheme we know that $r = 3$, so, if $\varepsilon = Kh^q$ then we have:

- For YC weights:

$$\varepsilon \geq \mathcal{O}(h^{3r-4}) = \mathcal{O}(h^5) \Rightarrow q \leq 5$$

- For AMM weights:

$$q \leq 4r - 4 - \frac{r}{\mu} = 4r - 4 - \frac{r}{\lceil \frac{r}{2} \rceil} = 12 - 4 - \frac{3}{2} = 6.5 \Rightarrow q \leq 6.5$$

In order to satisfy both conditions in our experiments with WENO5 scheme we choose $\varepsilon = h^2$ or $\varepsilon = h^5$. For WENO9 scheme we usually use $\varepsilon = h^2$ or $\varepsilon = h^{11}$ that satisfy both conditions (4.5) and (4.8) for $r = 5$.

For a given scheme we denote by $e(h)$ the ∞-norm of the error and by $cr(h)$ the convergence rate $cr(h) = \log_2(e(h)/e(h/2))$ deduced from the experiments. We use the notation $e_{j\pm1}$ and $cr_{j\pm1}(h)$ to denote the errors and the deduced convergence rates at the points right before and after the discontinuity respectively.

4.4.1 One-dimensional tests

Test 4.2.

We compute approximations up to $t = 1$ with $N = 25 \cdot 2^l$, $l = 0, \ldots, 6$ and $CFL = 0.5$ to the solution of the linear advection equation

$$u_t + u_x = 0, \qquad 0 < x < 1,$$

with periodic boundary conditions and initial condition given in [104, page 4264] as

$$u_0(x) = \begin{cases} z^{18} - 14z^{16} + 69z^{14} - 175z^{12} + 259z^{10} & \\ -231z^8 + 119z^6 - 29z^4 + 1, & \text{for } |z| \leq 1; \\ 0, & \text{otherwise.} \end{cases}$$

where $z = 5(x - 0.5)$ and $0 \leq x \leq 1$. It consists in a $\mathcal{C}^6$ function with three critical points: $x = 0.5$ with order 3 (i.e., $f'(0.5) = f''(0.5) = f'''(0.5) = 0,\ f^{(4)}(0.5) \neq 0$), $x = 0.3$ and $x = 0.7$ of order 6.

We use WENO5 reconstruction scheme, i.e. $r = 3$, so $\mu = \left\lceil \frac{3}{2} \right\rceil = 2$ is used for our proposed AMM weights. We display in Table 4.3 the results for $\varepsilon = h^2$ and in Table 4.4 the results for $\varepsilon = h^5$.

Since the ODE solver is third order accurate, we take $\Delta t = C \Delta x^{5/3}$, with $C = 0.5 \cdot 50^{2/3}$ to ensure $\Delta t^3 = \mathcal{O}(\Delta x^5)$ and $\Delta t / \Delta x \leq 0.5$ for the sizes N considered in the experiment. With this choice, the error introduced by

	JS-WENO5		YC-WENO5		AMM-WENO5	
h	$e(h)$	$cr(h)$	$e(h)$	$cr(h)$	$e(h)$	$cr(h)$
1/25	4.92e-01		4.49e-01		4.95e-01	
1/50	7.47e-02	2.72	6.84e-02	2.71	6.88e-02	2.85
1/100	5.99e-03	3.64	6.52e-03	3.39	6.52e-03	3.40
1/200	3.94e-04	3.93	2.28e-04	4.84	2.28e-04	4.84
1/400	1.96e-05	4.33	7.18e-06	4.99	7.18e-06	4.99
1/800	7.48e-07	4.71	2.25e-07	5.00	2.25e-07	5.00
1/1600	2.40e-08	4.96	7.04e-09	5.00	7.04e-09	5.00

Table 4.3: *Results of Test 4.2 with* $\varepsilon = h^2$.

	JS-WENO5		YC-WENO5		AMM-WENO5	
h	$e(h)$	$cr(h)$	$e(h)$	$cr(h)$	$e(h)$	$cr(h)$
1/25	5.13e-01		4.75e-01		4.91e-01	
1/50	2.84e-01	0.85	6.50e-02	2.87	1.42e-01	1.79
1/100	6.22e-03	5.51	6.64e-03	3.29	6.73e-03	4.40
1/200	1.32e-03	2.23	2.88e-04	4.53	2.61e-04	4.69
1/400	1.53e-04	3.11	1.55e-05	4.22	1.28e-05	4.35
1/800	1.69e-05	3.18	9.86e-07	3.97	7.62e-07	4.08
1/1600	2.06e-06	3.04	7.04e-08	3.81	5.20e-08	3.87

Table 4.4: *Results of Test 4.2 with* $\varepsilon = h^5$.

the ODE solver has an order of accuracy not less than the spatial order of accuracy.

As can be seen in Table 4.3 for $\varepsilon = h^2$ the errors for YC-WENO5 and AMM-WENO5 are similar and slightly smaller than the errors for JS-WENO5 and the convergence rate is 5 for all three reconstructions. For $\varepsilon = h^5$ and small h the results in Table 4.4 are slightly better for AMM-WENO5 than for YC-WENO5 and both quite better than JS-WENO5. The convergence rate is more difficult to deduce in this case for YC-WENO5 and AMM-WENO5, since $cr(h)$ does not seem to converge to 5, probably due to round-off errors when adding $h^5 \approx 10^{-15}$ for $h \approx 10^{-3}$. The convergence rate of JS-WENO5 approaches 3, thus revealing an order loss at smooth extrema.

Test 4.3.

We use the same setup as in Section 4.2 to test the performance of the AMM-WENO5 reconstruction when using divided differences to approximate derivatives. In Tables 4.5 and 4.6 we show the results of this test for the AMM-WENO5 scheme with $\varepsilon = 10^{-100}$ and $\varepsilon = h^2$ respectively. The columns corresponding to the deduced convergence rates $cr_{j\pm1}(h)$ reveal that the convergence rate of the divided differences of AMM-WENO5 reconstructions is 2, the same convergence rate that we obtained with the original JS-WENO5 method, whereas the convergence rate achieved with divided differences of YC-WENO5 reconstruction is 1 in this case (see Tables 4.1 and 4.2).

h	e_{j-1}	cr_{j-1}	e_{j+1}	cr_{j+1}
8.000e-02	-5.5575e-03		4.6570e-03	2.0164
4.000e-02	-2.8473e-03	0.9648	1.1511e-03	1.9972
2.000e-02	-7.3601e-04	1.9518	2.8833e-04	2.0001
1.000e-02	-1.8480e-04	1.9938	7.2078e-05	2.0005
5.000e-03	-4.6252e-05	1.9984	1.8013e-05	2.0005
2.500e-03	-1.1567e-05	1.9995	4.5018e-06	2.0003
1.250e-03	-2.8922e-06	1.9998	1.1252e-06	2.0002
6.250e-04	-7.2311e-07	1.9999	2.8126e-07	

Table 4.5: *Results of Test 4.3 with* $\varepsilon = 10^{-100}$ *for AMM-WENO5.*

h	e_{j-1}	e_{j-1}/h^2	e_{j+1}	e_{j+1}/h^2
8.000e-02	-1.0883e-02		4.4575e-03	
4.000e-02	-2.8393e-03	1.9385	1.1361e-03	1.9721
2.000e-02	-7.2572e-04	1.9681	2.8650e-04	1.9875
1.000e-02	-1.8328e-04	1.9854	7.1827e-05	1.9959
5.000e-03	-4.6051e-05	1.9927	1.7979e-05	1.9982
2.500e-03	-1.1542e-05	1.9963	4.4972e-06	1.9992
1.250e-03	-2.8890e-06	1.9983	1.1246e-06	1.9996
6.250e-04	-7.2270e-07	1.9991	2.8119e-07	1.9998

Table 4.6: *Results of Test 4.3 with* $\varepsilon = h^2$ *for AMM-WENO5.*

Test 4.4.

We use the same setup as in Section 4.2 for $f(x) = x^4 + \dfrac{x^2}{2} + \cos(x)$. In this experiment we want to test the order of accuracy of the reconstructions computed using the AMM-WENO5 ($r = 3$) method when we choose the value of the parameter ε depending on the condition (4.8) being satisfied or not. Note that $f'(0) = f''(0) = f'''(0) = 0$ but $f^{(iv)}(0) \neq 0$, so that the order of the critical point $x_{j+\frac{1}{2}} = 0$ is $s = 3$ and the order of the smoothness indicator is $I_k = \mathcal{O}(h^{2s+2}) = \mathcal{O}(h^8)$.

As we have seen, for $r = 3$, condition (4.8) leads to:

$$q \leq 4 \cdot 3 - 4 - \frac{3}{\lceil \frac{3}{2} \rceil} = 6.5. \tag{4.12}$$

For testing the accuracy order we choose $\varepsilon = h^2, h^4, h^6$, which satisfy condition (4.12), and $\varepsilon = h^8, h^{10}$ which do not satisfy this condition.

As we can see in Table 4.7, for the choices of ε that satisfy the condition (4.12) the method has maximal order of accuracy $2r-1$ but the order of accuracy is smaller for the choices that do not satisfy the condition.

h	$\varepsilon = h^2$	$\varepsilon = h^4$	$\varepsilon = h^6$	$\varepsilon = h^8$	$\varepsilon = h^{10}$
1.000e-01	4.9973	8.2845	2.1971	3.0001	3.0024
5.000e-02	4.9993	5.01797	5.1690	3.00004	3.0006
2.500e-02	4.9998	4.99984	7.2594	3	3.0001
1.250e-02	5	5	8.33170	3	3
6.250e-03	5	5	8.7003	3	3
3.125e-03	5	5	7.8990	3	3
1.562e-03	5	5	5.7840	3	3
7.812e-04	5	5	5.0672	3	3
3.906e-04	5	5	5.0043	3	3
1.953e-04	5	5	5.00027	3	3
9.765e-05	5	5	5	3	3

Table 4.7: *Estimated orders* $\log_2(e(h_i)/e(h_{i+1}))$, $i = 0.\dots,19$ *for test 4.4 computed with different values of* ε.

Test 4.5.

We compute the approximations of the solution of the linear advection equation

$$u_t + u_x = 0, \qquad 0 < x < 1,$$

with initial conditions: $u_0(x) = \begin{cases} 5, & x \leq 0.2; \\ 3, & x > 0.2. \end{cases}$

We compute the approximations up to $t = 0.5$ with $CFL = 0.5$, $\varepsilon = h^2$ and $\mu = \lceil r/2 \rceil$, that is, $\mu = 2$ for WENO5. We use the third-order TVD Runge-Kutta scheme described in (3.6), and proposed in [94], to solve the semi-discretized problem.

We observe in Figure 4.2 that the AMM-WENO5 scheme performs better than the YC-WENO5 scheme, especially near discontinuities where this scheme presents some oscillations which are reduced when using our modified weights. In addition, the approximation to the exact solution is better when using our modified weights than when Yamaleev and Carpenter's weights or Jiang and Shu's weights are used. We can also appreciate that the oscillations obtained near the discontinuity that appear with Yamaleev and Carpenter's weights are reduced when using our modified weights. The same behavior can be deduced when we use WENO9 scheme, as it could be seen in Figure 4.3.

In Figure 4.4 we show the results of this experiment when using the AMM-WENO9 reconstruction scheme with different exponents $\mu = 1, 2, 3, 4$ (note that the case $\mu = 1$ corresponds to YC-WENO9 scheme). As it can be seen, the AMM-WENO9 scheme with $\mu = 3$ performs better than the rest of the options, especially near discontinuities, where $\mu = 1$ gives an oscillatory behavior.

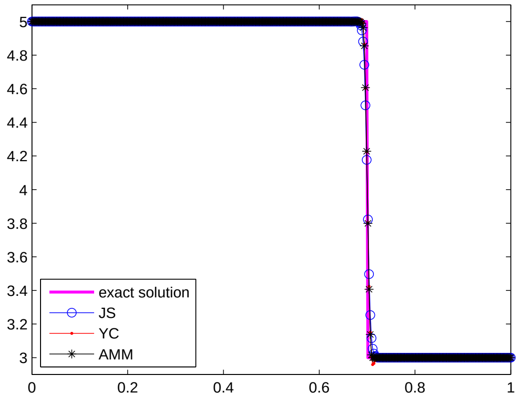


Figure 4.1: *Numerical solution of test 4.5 computed with WENO5 scheme and* $N = 400$ *nodes,* $t = 0.5$*.*

Test 4.6.

We consider Sod's problem [96] which consists in the one-dimensional Euler equations of gas dynamics (2.18), assuming the equation of state (2.21), with $\gamma = 1.4$. The initial conditions are given by:

$$(\rho, u, p) = \begin{cases} (1, 0, 1), & \text{if } x < 0.5; \\ (0.125, 0, 0.1), & \text{if } x > 0.5, \end{cases}$$

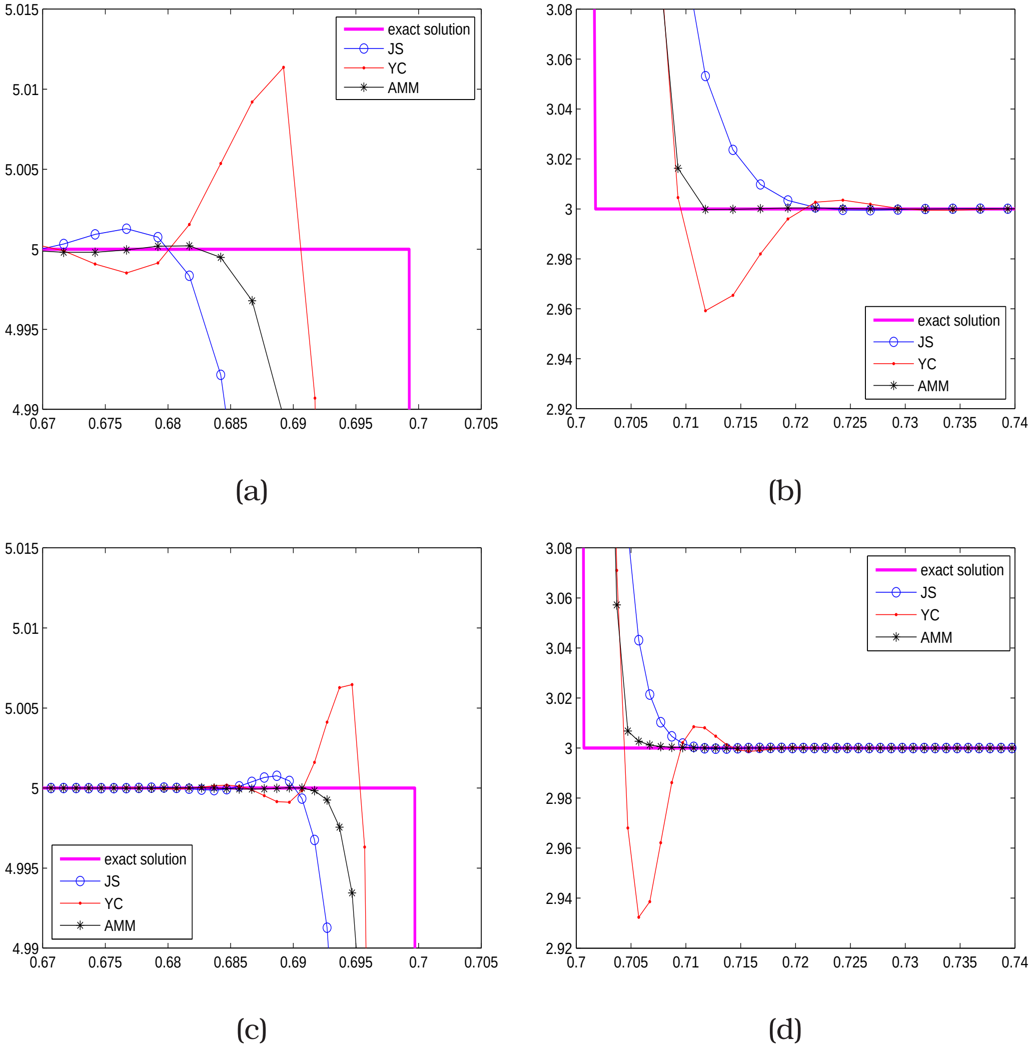
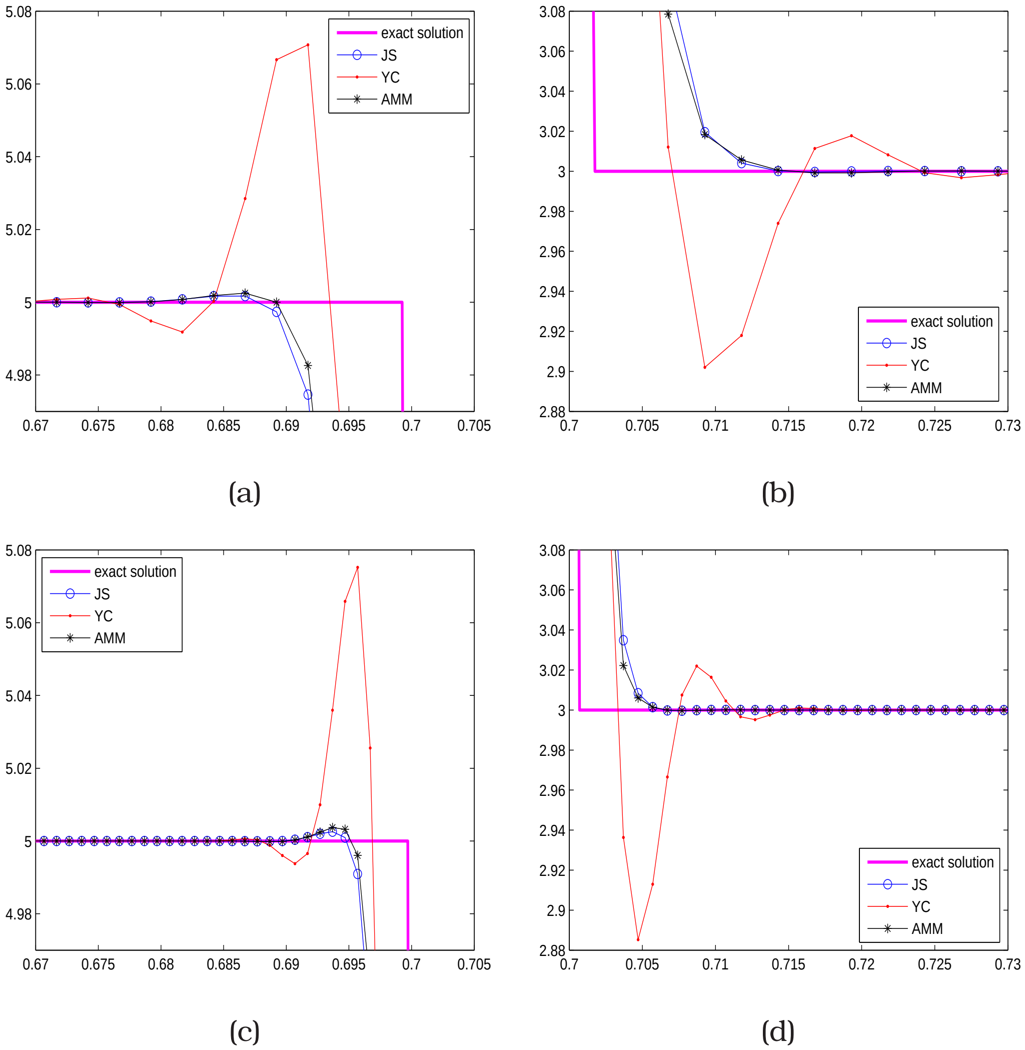


Figure 4.2: *Enlarged views of the numerical solution of test 4.5 computed with WENO5 scheme with* $N = 400$ *((a) and (b)) and* $N = 1000$ *((c) and (d)) nodes for the final time* $t = 0.5$.



(a)

(b)

(c)

(d)

Figure 4.3: *Enlarged views of the numerical solution of test 4.5 computed with WENO9 scheme with* $N = 400$ *((a) and (b)) and* $N = 1000$ *((c) and (d)) nodes for the final time* $t = 0.5$.

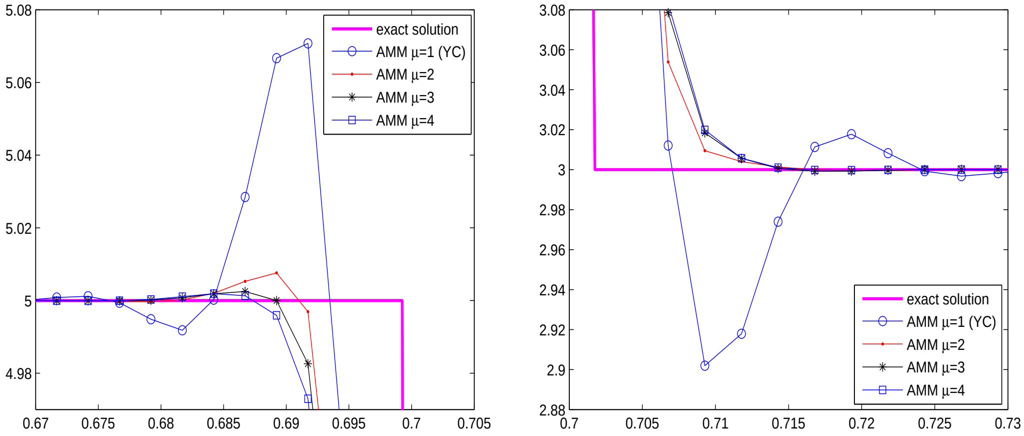


Figure 4.4: *Enlarged views of the numerical solution of test 4.5 computed with AMM-WENO9 using different values for the exponent μ present in the definition of the AMM weights and $N = 400$ nodes.*

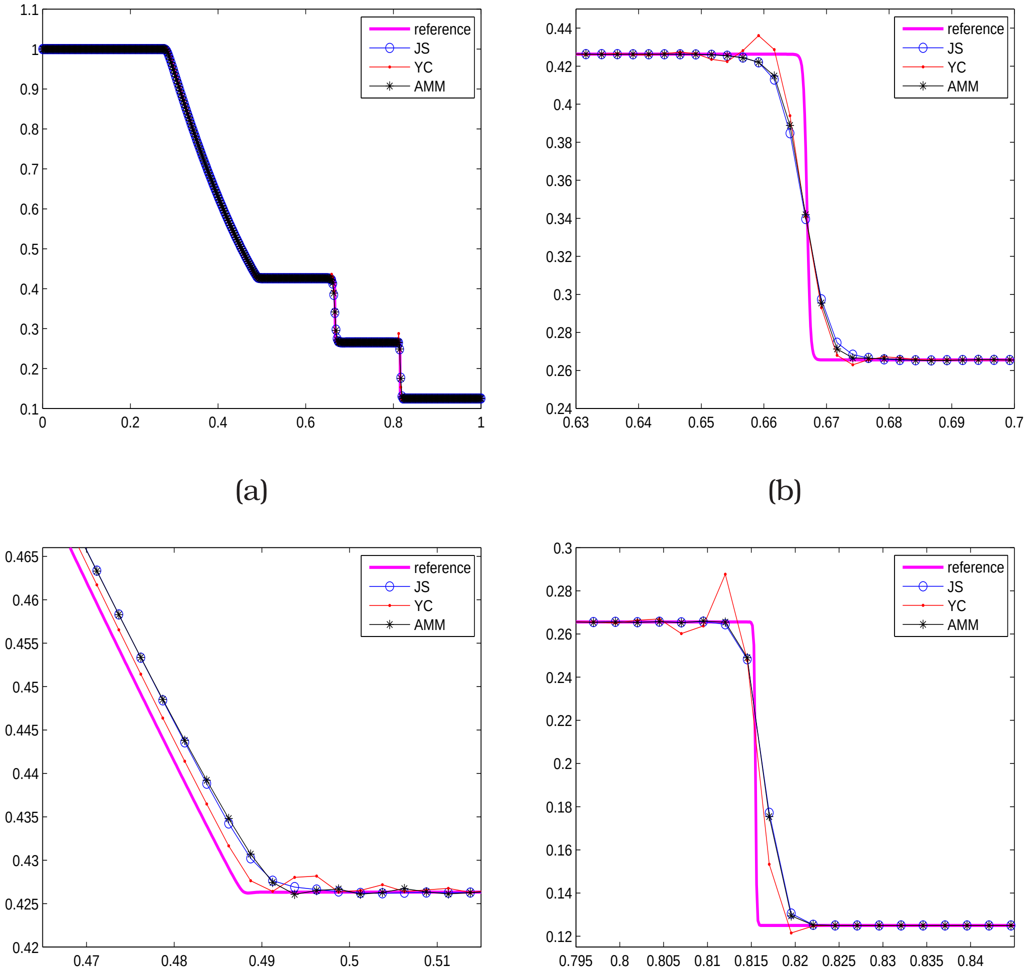
and outflow boundary conditions are used.

The numerical scheme follows a *method of lines* approach. The spatial discretization is obtained by Shu-Osher's finite-difference strategy [94], with the appropriate WENO reconstruction applied to characteristic fluxes obtained by Donat-Marquina's [33] two-Jacobians local characteristic projections and local Lax-Friedrichs flux-splittings. This spatially-discretized problem is solved by the third-order TVD Runge-Kutta scheme (3.6). We use this numerical scheme, and its two-dimensional extension, for all subsequent tests that require the solution of the Euler equations.

In Figures 4.5 and 4.6 we compare the solutions obtained with JS-WENO9, YC-WENO9 and AMM-WENO9 schemes at $t = 0.18$, with $CFL = 0.5$, $\varepsilon = h^2$ and $\mu = \lceil r/2 \rceil$, that is, $\mu = 2$ for WENO5, for $N = 400$ and 4000 cells. As can be deduced from these figures, the solutions obtained with the YC-WENO9 scheme show oscillations near discontinuities that do not seem to disappear upon mesh refinement. It can be further observed that the resolution of AMM-WENO9 at singularities is slightly better than the resolution of JS-WENO9.

Test 4.7.

We consider here the simulation of the interaction of a shock and an entropy wave, [95], with the one-dimensional Euler equations of gas dynamics (2.18).



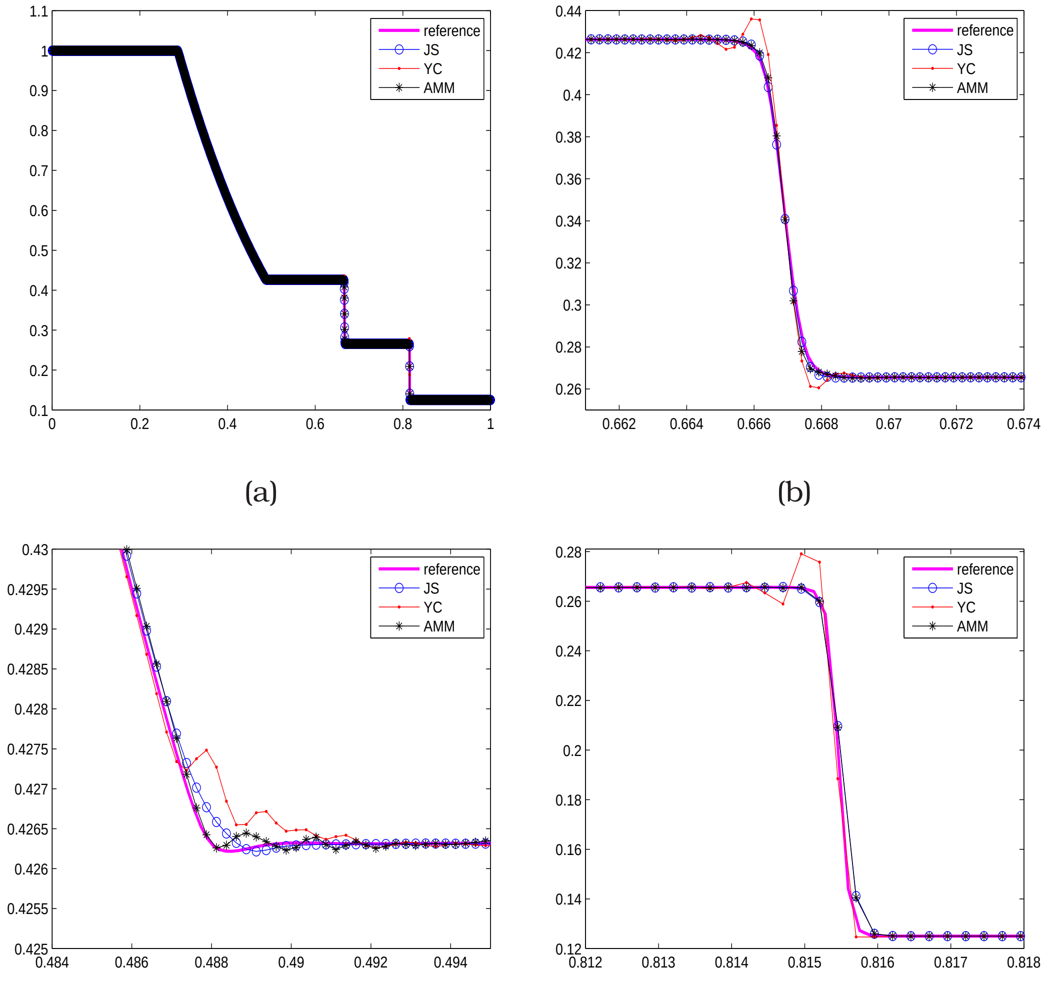
(a)

(b)

(c)

(d)

Figure 4.5: *Results of test 4.6 with WENO9: (a) Plot of the density with* $N = 400$ *cells. (b), (c) and (d) are enlarged views of (a).*



(a) (b)

(c) (d)

Figure 4.6: *Results of test 4.6 with WENO9: (a) Plot of the density with* $N = 4000$ *cells. (b), (c) and (d) are enlarged views of (a).*

The initial conditions are set by a Mach 3 shock interacting with a perturbed density field:

$$(\rho, u, p) = \begin{cases} (\frac{27}{7}, \frac{4\sqrt{35}}{9}, \frac{31}{3}) & \text{if } x < -4 \\ \\ (1 + \frac{1}{5}\sin(5x), 0, 1) & \text{if } x \geq -4 \end{cases}$$

on the domain $[-5, 5]$ with homogeneous Neumann boundary conditions at the left part of the domain and Dirichlet boundary conditions at the right part of the domain.

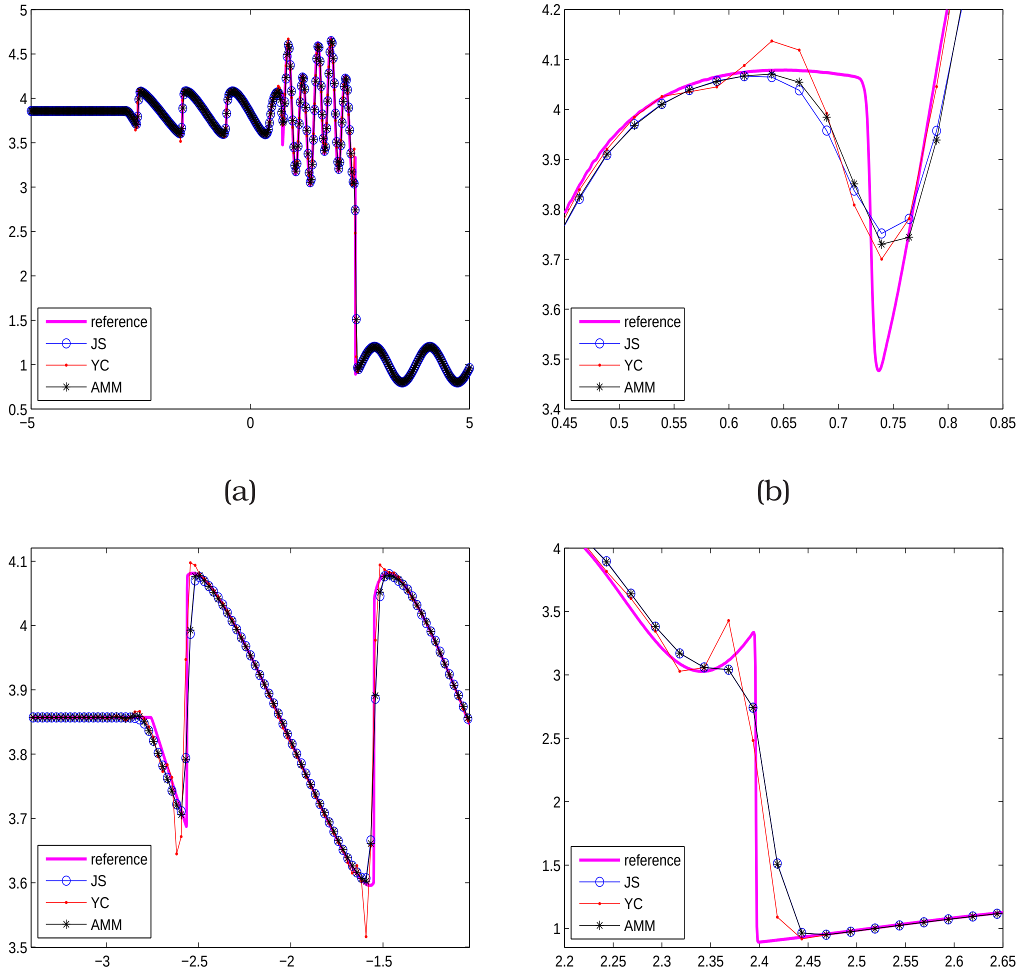
In Figures 4.7 and 4.8 we compare the results obtained using JS-WENO9, YC-WENO9 and AMM-WENO9 schemes at $t = 1.8$ with $N = 400$ and 4000 cells and $\varepsilon = h^2$. A reference solution is obtained with the JS-WENO9 scheme and $N = 12800$ cells. As can be seen, the approximations obtained using Yamaleev and Carpenter's weights show some oscillations in the vicinity of the strong shock near $x = 2.4$. These oscillations do not seem to diminish upon mesh refinement. One can also observe that our modified weights yield slightly better results than Jiang and Shu's weights, especially for $N = 400$.

4.4.2 Two-dimensional tests

Test 4.8.

In this test a double Mach reflection of a strong shock is simulated with the 2D Euler equations of gas dynamics (see [102]). The problem involves a Mach 10 shock in air ($\gamma = 1.4$) which makes a 60^o angle with a reflecting wall. The computational domain has been rotated by -30^o, so that the reflecting wall is located at the bottom, beginning at $x = 0.25$. The domain is then a rectangle 4 units long and 1 unit high, starting at $x = 0, y = 0$. Initially the shock extends from the point $x = 0.25$ at the bottom of the computational domain to the top boundary. Post-shock conditions are assigned at the boundaries located to the left of the shock; the air ahead of the shock is left undisturbed and has density 1.4 and pressure 1. Outflow conditions are applied at the right end of the domain, and the values on the top boundary to the right of the shock are those of undisturbed air.

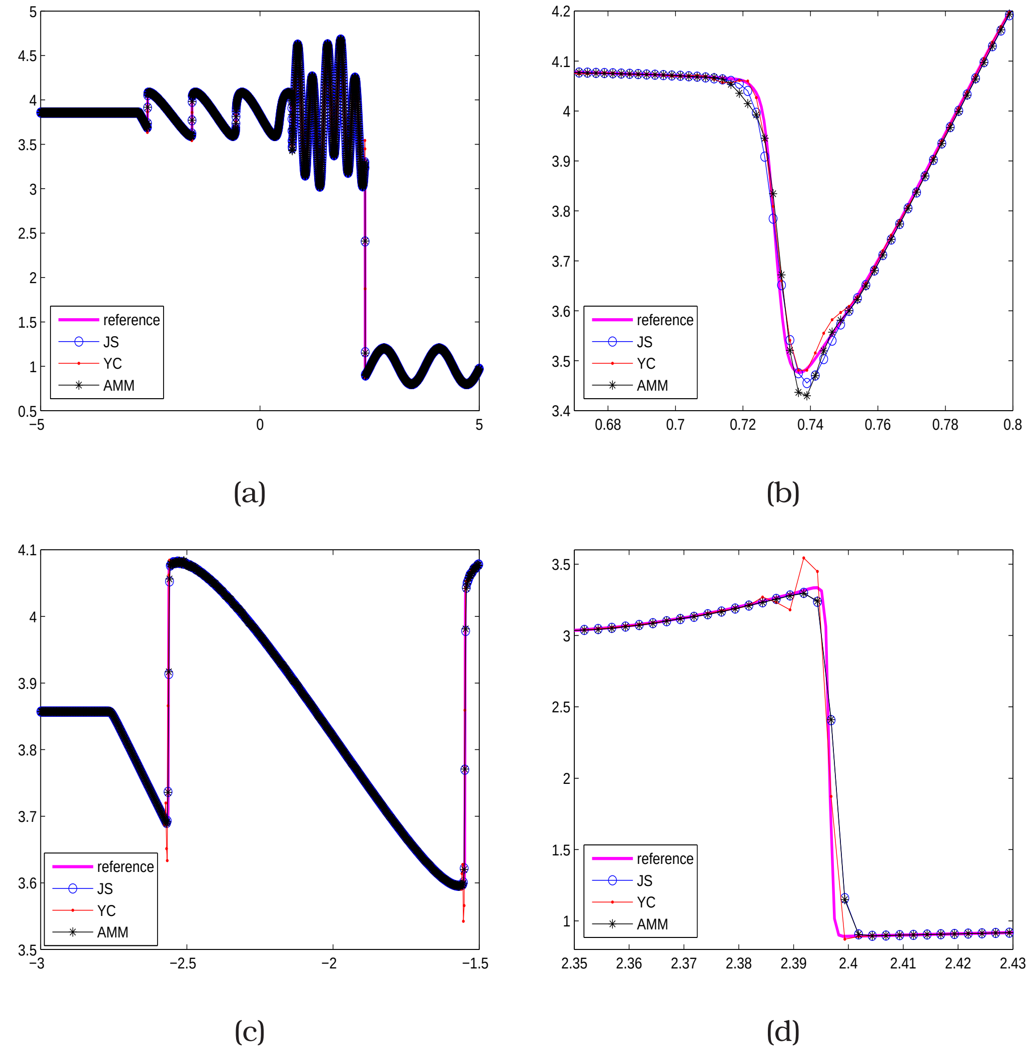
We run the experiment with $CFL = 0.25$, $t = 0.2$ and $\varepsilon = h^2$ for resolutions 1024×256 and 2048×512 and display an enlarged view of the results in Figures 4.9 and 4.10, respectively. As can be observed, the results



(a) (b)

(c) (d)

Figure 4.7: *Results of test 4.7 with WENO9: (a) Approximations of the density with* $N = 400$ *nodes. (b), (c) and (d) are enlarged views of the most interesting parts of (a).*



(a)

(b)

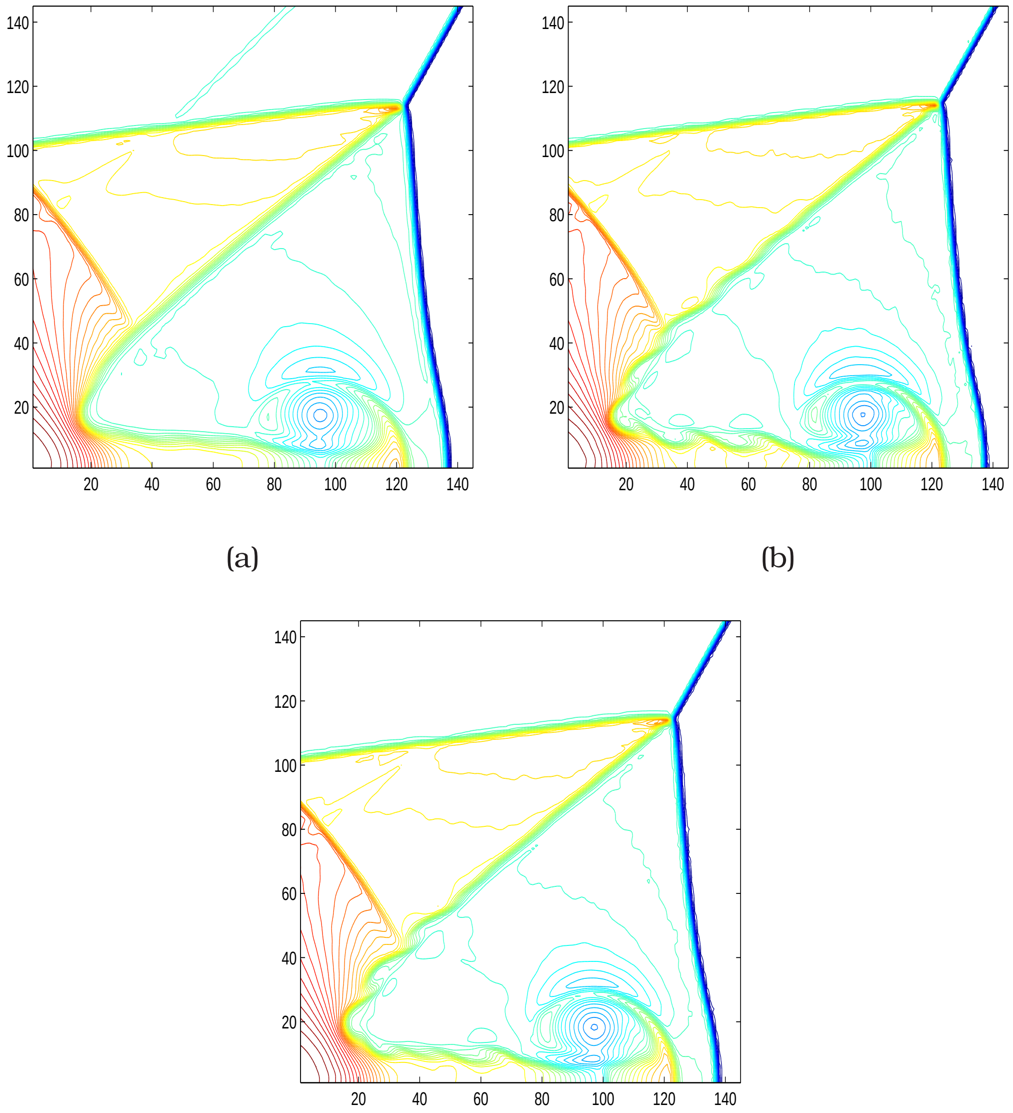
(c)

(d)

Figure 4.8: *Results of test 4.7 with WENO9: (a) Approximations of the density with* $N = 4000$ *nodes. (b), (c) and (d) are enlarged views of the most interesting parts of (a).*

obtained with YC-WENO5 and AMM-WENO5 have more vorticity developed at the contact lines, thus indicating that they introduce a smaller amount of numerical dissipation than JS-WENO5. The results obtained with AMM-WENO5 seem to have some more vorticity than those obtained with YC-WENO5 scheme.

In Figure 4.12, we show the results obtained when we use $\varepsilon = h^5$ for a resolution of 2048×512 cells. As it could be seen the vorticity diminishes when we decrease the value of the parameter ε.

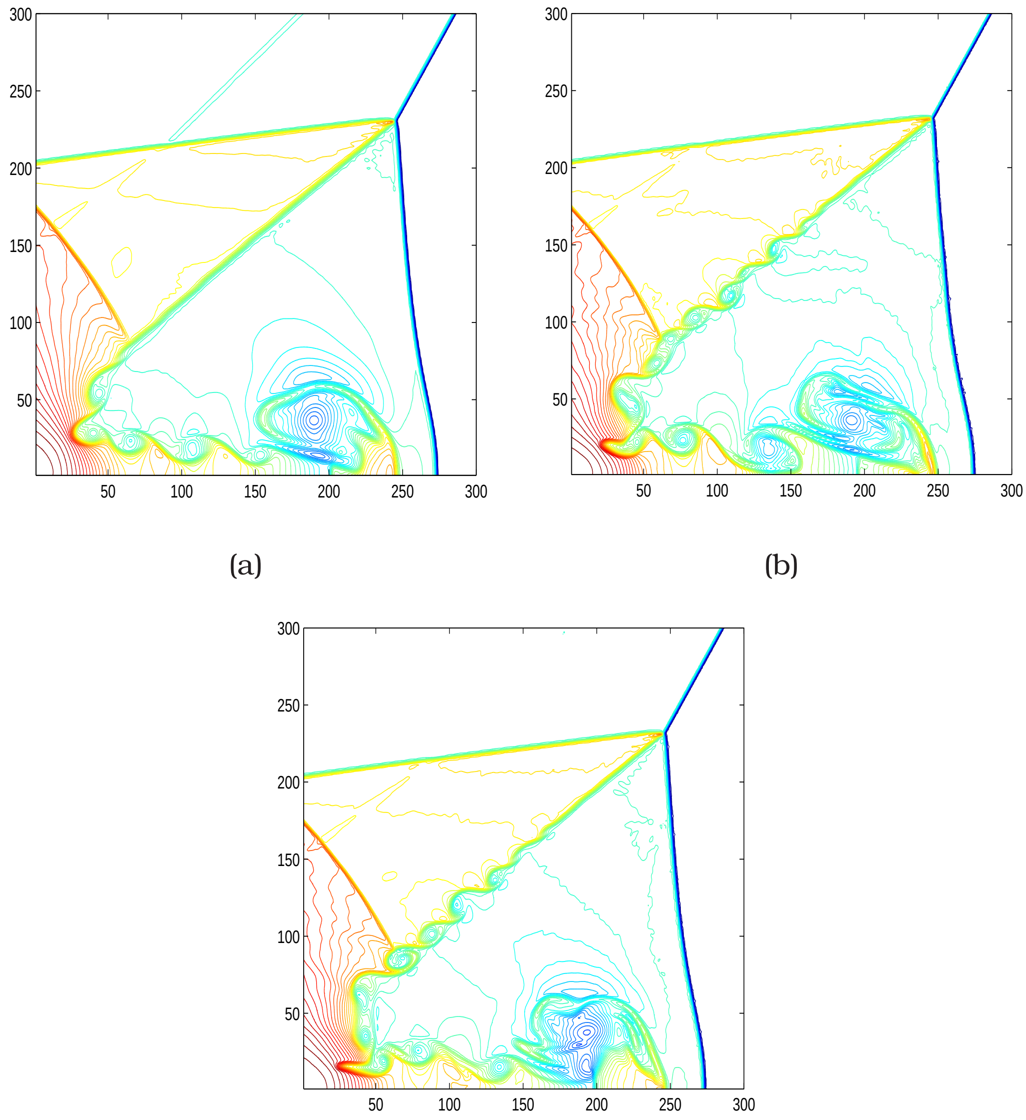


(a)

(b)

(c)

Figure 4.9: *Results of test 4.8 for a grid of* 1024×256 *cells and* $\varepsilon = h^2$*: 50 contour lines of the density obtained with (a) JS-WENO5, (b) YC-WENO5 and (c) AMM-WENO5 are shown, respectively.*

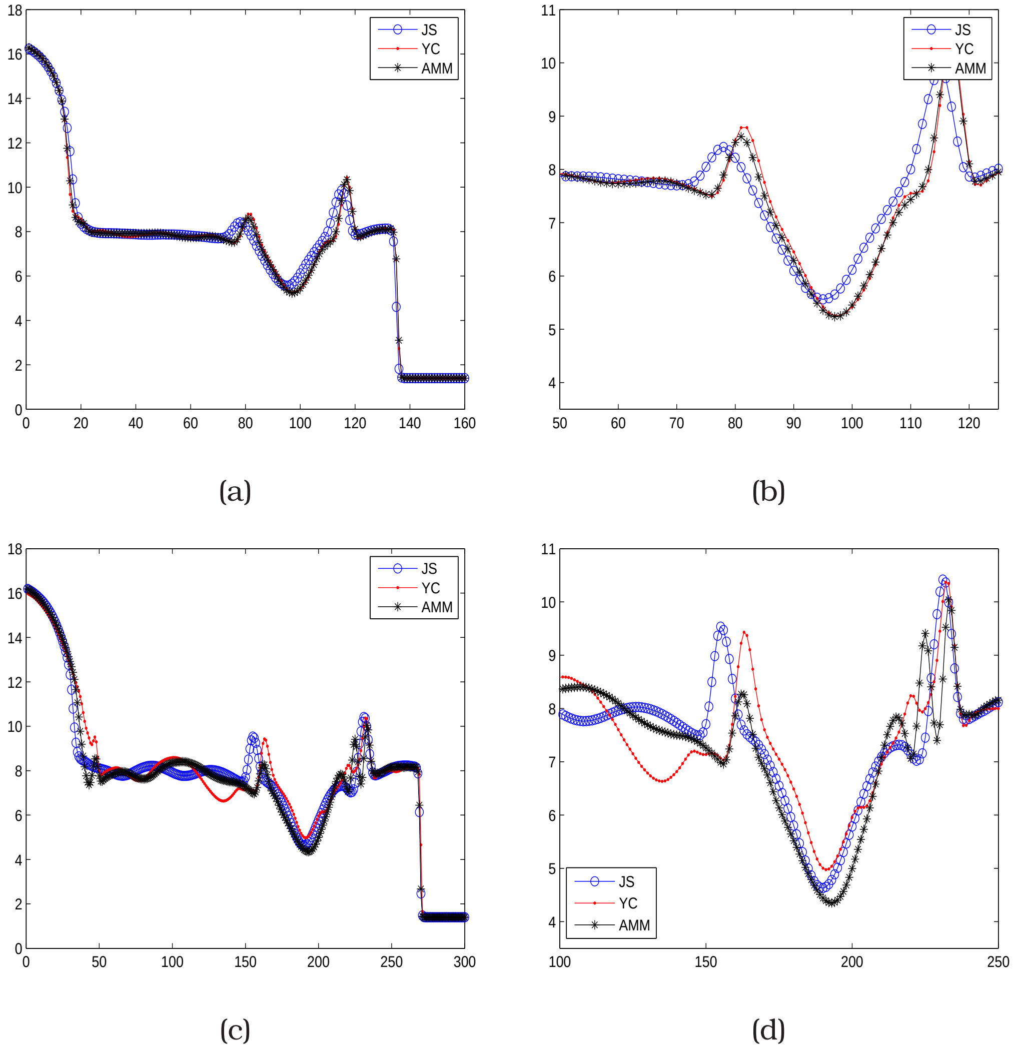


(a)

(b)

(c)

Figure 4.10: *Results of test 4.8 for a grid of* 2048×512 *cells and* $\varepsilon = h^2$*: 50 contour lines of the density obtained with (a) JS-WENO5, (b) YC-WENO5 and (c) AMM-WENO5 are shown, respectively.*



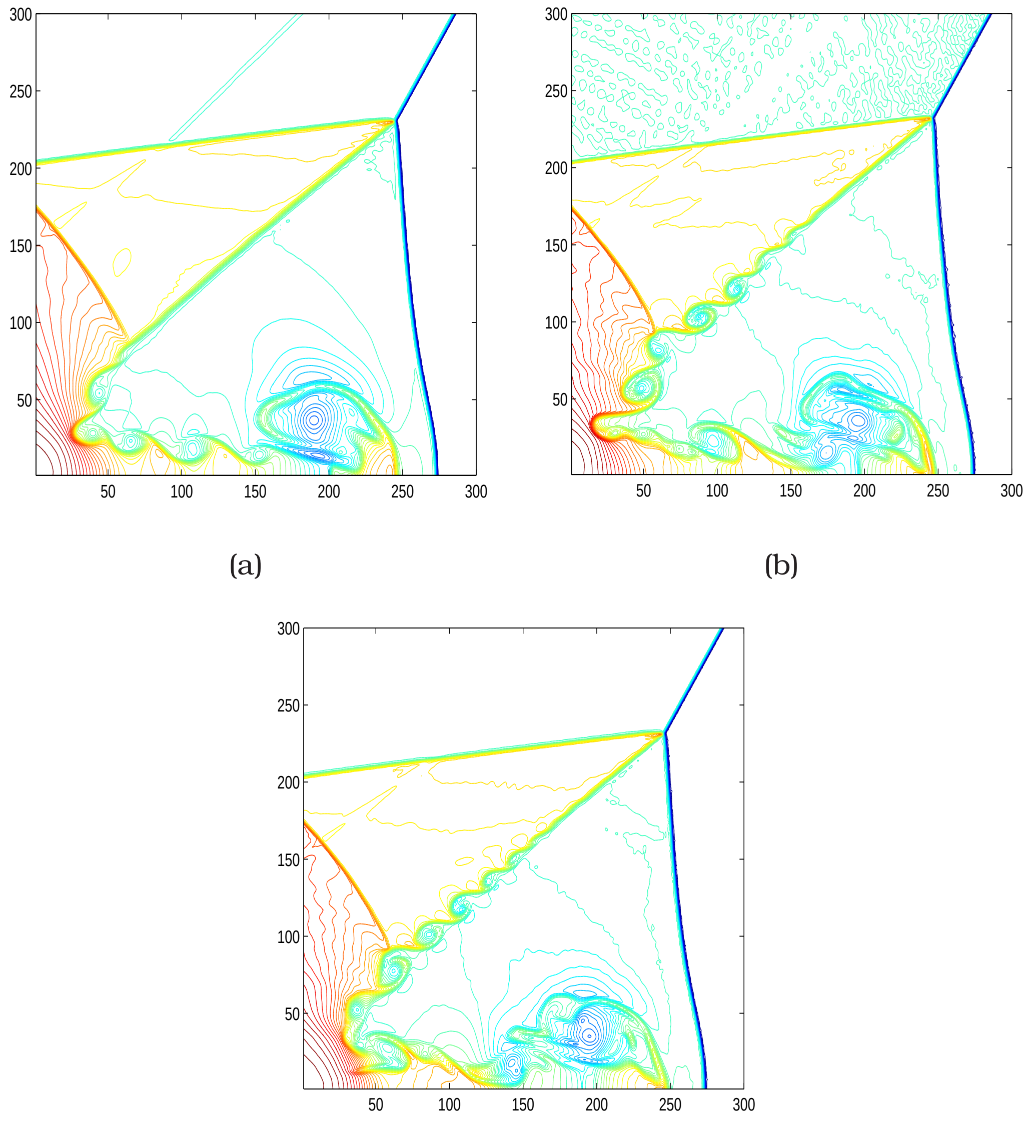
(a)

(b)

(c)

(d)

Figure 4.11: *(a) and (c) display sections of the 50 contour lines of the density obtained with JS-WENO5, YC-WENO5 and AMM-WENO5 at pixel height* 18 *(a) and* 36 *(c) of test 4.8 for a grid of* 1024×256 *(a) and* 2048×512 *(c) cells; (b) and (d) are zooms of (a) and (c) respectively.*



(a)

(b)

(c)

Figure 4.12: *Results of test 4.8 for a grid of* 2048×512 *cells and* $\varepsilon = h^5$*: 50 contour lines of the density obtained with (a) JS-WENO5, (b) YC-WENO5 and (c) AMM-WENO5 are shown, respectively.*

5

Biased flux-splittings for finite-difference WENO schemes

5.1 Introduction

One of the main drawbacks of high-resolution shock-capturing schemes is that most of them use the spectral decomposition of the Jacobian matrix of the system to compute the numerical approximations by local projections to characteristic fields. A typical computation involves an eigenvalue-eigenvector decomposition of the flux Jacobian matrix and approximations of the values of the numerical solution at each cell interface. These approximations are obtained by high-order upwind-biased reconstructions (i.e., cell-average interpolators whose stencils have more points at the upwind side of the points where they are evaluated). The numerical solutions obtained are often excellent in terms of resolution power, but the computational effort needed may be too high for some problems, especially those for which the spectral information of the flux Jacobian matrix is not available or is quite difficult to obtain.

Some authors have proposed schemes that do not use characteristic projections. Amongst them we can cite Nessyahu-Tadmor and Kurganov-Tadmor central schemes [61, 83] or Central WENO schemes [74]. The use of component-wise finite-difference WENO schemes was introduced in [106]. It is based on Shu-Osher's finite-difference schemes [95], which obtain the numerical fluxes at each cell interface by upwind-biased reconstructions of split upwind fluxes (those whose Jacobian matrix has eigenvalues of a definite sign).

Component-wise schemes use a global Lax-Friedrichs flux-splitting for each component of the flux function, computing at a given time level a numerical viscosity as the maximum of the characteristic speeds of the solution at that time. Since the resulting schemes tend to be too diffusive, in order to reduce those diffusive effects associated to the global choice of the viscosity coefficient, a local Lax-Friedrichs approach was proposed in [94], which consists in computing that viscosity coefficient locally, not over all the domain, but only on a neighborhood of the cell interface at which the numerical flux is computed.

In an attempt to improve the results obtained when using a Lax-Friedrichs flux-splitting, by reducing the numerical viscosity, we propose to use a flux-splitting, named HLL flux-splitting, first introduced by Harten, Lax and van Leer in [52] as a Riemann solver. This flux-splitting is based on a possibly asymmetric choice of wave speeds of each of the two terms of the flux-splitting.

The other key issue of the HRSC schemes is the use of a high-order reconstruction method with a control of the oscillations. In this work we use the fifth-order WENO reconstructions [59, 78]. As it has been mentioned in [51, 59], the numerical solutions obtained using a component-wise scheme and a WENO5 reconstruction method present some spurious oscillations whose amplitude does not decrease as the grid is refined.

In order to diminish those spurious oscillations when using component-wise schemes, we are going to study another strategy based on the work of Levy et al. (see [74, 75]) and the use of a global definition of the smoothness indicators in the definition of the weights of the WENO5 scheme. In this work we compare the results obtained using a global definition of the smoothness indicators to prove that, in some cases, when using these weights, we reduce the oscillatory behavior while maintaining the high resolution of the scheme.

In this chapter we recall first different flux-splitting functions used in high-resolution shock-capturing schemes, introducing the HLL flux-splitting. Second, we perform a brief exposition of an alternative to usual WENO reconstruction scheme and finally we perform some numerical experiments on standard tests of polydisperse sedimentation to illustrate and compare the performance of several schemes.

5.2 Characteristic based and Component-wise schemes with flux-splitting

In the case that the full spectral decomposition of the Jacobian matrix $f'(\Phi)$ is known, we denote, for $k = 1, \dots, M$, $\lambda_k(f'(\Phi))$ the eigenvalues of $f'(\Phi)$ and by $r^k(\Phi)$ and $l^k(\Phi)$ the corresponding normalized right and left eigenvectors respectively. Then, the numerical flux $\widehat{f}_{j+\frac{1}{2}}$ computed using an upwind characteristic-wise scheme (denoted in this work as SPEC scheme) can be written as:

$$\begin{aligned}\widehat{f}_{j+\frac{1}{2}} = &\sum_{k=1}^{N} r^k \left(\mathcal{R}^+ \left(l^k \cdot f^+_{j-2}, \dots, l^k \cdot f^+_{j+2}; x_{j+\frac{1}{2}}\right)\right) \\ &+ \sum_{k=1}^{N} r^k \left(\mathcal{R}^- \left(l^k \cdot f^-_{j-1}, \dots, l^k \cdot f^-_{j+3}; x_{j+\frac{1}{2}}\right)\right),\end{aligned} \tag{5.1}$$

where $r^k = r^k(\Phi_{j+\frac{1}{2}})$, $l^k = l^k(\Phi_{j+\frac{1}{2}})$, $\Phi_{j+\frac{1}{2}} = \frac{1}{2}\big(\Phi_j + \Phi_{j+1}\big)$, $f^\pm_j := f^\pm(x_j)$, $\mathcal{R}^\pm$ are upwind biased reconstruction operators (WENO reconstructions in our case) and the functions $f^\pm$ define a flux-splitting that satisfies $f^+ + f^- = f$ and $\pm\lambda^k\left((f^\pm(\Phi))'\right) \geq 0$ ($f^\pm$ are upwind fluxes) for Φ in some relevant range $\mathcal{D}$:

$$\mathcal{D} = \begin{cases} \{\Phi_i / i = 1, \dots, N\} & \text{for global flux-splittings,} \\ \{\Phi_i / i = j-2, \dots, j+3\} & \text{for local flux-splittings.} \end{cases}$$

The Lax-Friedrichs flux-splitting is given by $f^\pm = \frac{1}{2}(f(\Phi) \pm \alpha\Phi)$ with α satisfying:

$$\max\{|\lambda_k(f'(\Phi))| / k = 1, \dots, M, \Phi \in \mathcal{D}\} \leq \alpha.$$

In this work we are going to use SPECINT scheme introduced in [24]. This scheme differs from SPEC scheme in the way in which the parameter α is computed. As we have said in local SPEC schemes (SPEC-LLF), the viscosity coefficient α is computed as

$$\alpha^k_{j+\frac{1}{2}} = \max\left\{|\lambda_k(\Phi_j)|, |\lambda_k(\Phi_{j+1})|\right\}$$

In [24], the authors show that, when they apply this scheme to the polydisperse sedimentation problems, the numerical solutions computed present some spurious oscillations which do not disappear upon mesh refinement. In order to improve the results, the authors propose a new approximation to the parameter α calculated based on the interlacing property (2.23) as:

$$\alpha^k_{j+\frac{1}{2}} = \max_{\Phi\in\Gamma_j}\left\{|v_{k-1}(\Phi)|, |v_k(\Phi)|\right\}$$

where Γ_j denotes the straight line joining Φ_j and Φ_{j+1}.

As previously mentioned, the lack of information on the spectral decomposition of the Jacobian matrix of some problems or the high computational cost needed to obtain it may prevent the use of these fairly sophisticated high-resolution shock-capturing schemes. To tackle this shortcoming, a component-wise approach for these schemes was developed in [106]. For these schemes, the value of the numerical flux vector $\widehat{f}_{j+\frac{1}{2}}$ is computed by setting $l^k_l = r^k_l = \delta_{k,l}$ in (5.1). The numerical flux then reads as:

$$\widehat{f}_{j+\frac{1}{2},k} = \mathcal{R}^+\left(f^+_{j-2,k}, \dots, f^+_{j+2,k}; x_{j+\frac{1}{2}}\right) + \mathcal{R}^-\left(f^-_{j-1,k}, \dots, f^-_{j+3,k}; x_{j+\frac{1}{2}}\right).$$

The oscillatory behavior of the component-wise schemes obtained from global Lax-Friedrich flux splittings has been observed in the literature [35]. Also, as it could be seen for example in [24, 35], that scheme tends to be quite diffusive due to the global prescription of numerical viscosity. In this chapter we explore the possibility of using local LF flux-splittings for alleviating the oscillations and the excessive diffusion.

5.2.1 HLL flux-splitting

Another approach to address these issues is the use of a flux-splitting that uses less numerical viscosity to stabilize the upwind reconstructions. If one defines $F^{\pm}(\Phi) = f(\Phi) - \alpha_{\mp}\Phi$ then a sufficient condition for

$$f = \gamma F^- + (1-\gamma)F^+,$$

to be a flux-splitting is that the eigenvalues $\lambda_k((F^+(\Phi))')$ and $\lambda_k((F^-(\Phi))')$ have the corresponding sign for all $\Phi \in \mathcal{D}$ and $\gamma \in [0,1]$.

We can compute $\lambda_k((F^+(\Phi))')$ as

$$\lambda_k((F^+(\Phi))') = \lambda_k(f'(\Phi) - \alpha_- I) = \lambda_k(f'(\Phi)) - \alpha_-$$

So,

$$\lambda_k((F^+(\Phi))') = \lambda_k(f'(\Phi)) - \alpha_- \geq 0 \Leftrightarrow \lambda_k(f'(\Phi)) \geq \alpha_-$$

Analogously,

$$\lambda_k((F^-(\Phi))') = \lambda_k(f'(\Phi)) - \alpha_+ \leq 0 \Leftrightarrow \lambda_k(f'(\Phi)) \leq \alpha_+$$

Therefore, $\lambda_k((F^+(\Phi))') \geq 0$ and $\lambda_k((F^-(\Phi))') \leq 0 \quad \forall k, \Phi \in \mathcal{D}$ if and only if

$$\alpha_- \leq \lambda_k(f'(\Phi)) \leq \alpha_+, \quad \forall \Phi \in \mathcal{D}, \quad \forall k = 1, \ldots, n.$$

As we see here, $\alpha_\pm$ should be estimates of the extremal characteristic velocities in $\mathcal{D}$ for the upwind condition on $F^\pm$ to hold.

Now, if

$$f(\Phi) = \gamma F^-(\Phi) + (1-\gamma)F^+(\Phi)$$

should hold for any Φ, then

$$\begin{aligned} f(\Phi) &= \gamma(f(\Phi) - \alpha_+\Phi) + (1-\gamma)(f(\Phi) - \alpha_-\Phi) \\ &= f(\Phi) + (-\alpha_- + (\alpha_- - \alpha_+)\gamma)\Phi \end{aligned}$$

yields $\gamma = \dfrac{\alpha_-}{\alpha_- - \alpha_+}$. Therefore $0 \leq \gamma \leq 1$ if and only if $\alpha_- \leq 0 \leq \alpha_+$.

We define the HLL flux-splitting [52, 97] as:

$$f^+ = \begin{cases} f & \alpha_- \geq 0, \\ 0 & \alpha_+ \leq 0, \\ (1-\gamma)F^+ & \alpha_- \leq 0 \leq \alpha_+ \end{cases} \qquad f^- = \begin{cases} 0 & \alpha_- \geq 0, \\ f & \alpha_+ \leq 0, \\ \gamma F^- & \alpha_- \leq 0 \leq \alpha_+ \end{cases}$$

with

$$\alpha_- \leq \lambda_k(f'(\Phi)) \leq \alpha_+, \quad \forall \Phi \in \mathcal{D}, \quad \forall k = 1, \ldots, M.$$

$$\max\{\lambda_k(f'(\Phi))/k = 1, \ldots, M, \Phi \in \mathcal{D}\} \leq \alpha_+$$

$$\alpha_- \leq \min\{\lambda_k(f'(\Phi))/k = 1, \ldots, M, \Phi \in \mathcal{D}\}$$

This flux-splitting was similarly proposed in [97] for the Euler equations. We term it the HLL flux-splitting since a first order version of the scheme would be equivalent to the HLL scheme proposed in [52], based on an approximate Riemann solver with two wave speeds.

It is worth pointing out that the numerical viscosity used in a Lax-Friedrichs flux-splitting would be larger than that used for the HLL flux-splitting. For instance, if $\alpha_- \leq 0 \leq \alpha_+$ and $-\alpha_- < \alpha_+$, then the LF flux-splitting would read as:

$$f^+ = \frac{1}{2}(f + \alpha^+\Phi), \quad f^- = \frac{1}{2}(f - \alpha^+\Phi),$$

and would satisfy:

$$\begin{aligned}\min_{k,\Phi} \lambda_k((f^+)') &= \min_{k,\Phi} \lambda_k((\frac{1}{2}(f + \alpha^+\Phi))') = \min_{k,\Phi} \lambda_k(\frac{1}{2}(f' + \alpha^+)) \\ &= \frac{1}{2}\left(\left(\min_{k,\Phi} \lambda_k(f')\right) + \alpha^+\right) = \frac{1}{2}(\alpha^- + \alpha^+) > 0,\end{aligned}$$

$$\begin{aligned}\max_{k,\Phi} \lambda_k((f^-)') &= \max_{k,\Phi} \lambda_k((\frac{1}{2}(f - \alpha^+\Phi))') = \max_{k,\Phi} \lambda_k(\frac{1}{2}(f' - \alpha^+)) \\ &= \frac{1}{2}\left(\left(\max_{k,\Phi} \lambda_k(f')\right) - \alpha^+\right) = \frac{1}{2}(\alpha^+ - \alpha^+) = 0,\end{aligned}$$

whereas these extrema are 0 for the HLL flux-splitting:

$$\begin{aligned}\min_{k,\Phi} \lambda_k((f^+)') &= \min_{k,\Phi} \lambda_k(((1-\gamma)(f - \alpha^-\Phi))') = \min_{k,\Phi} \lambda_k((1-\gamma)(f' - \alpha^-)) \\ &= (1-\gamma)\left(\left(\min_{k,\Phi} \lambda_k(f')\right) - \alpha^-\right) = (1-\gamma)(\alpha^- - \alpha^-) = 0,\end{aligned}$$

$$\begin{aligned}\max_{k,\Phi} \lambda_k((f^-)') &= \max_{k,\Phi} \lambda_k((\gamma(f - \alpha^+\Phi))') = \max_{k,\Phi} \lambda_k(\gamma(f' - \alpha^+)) \\ &= \gamma\left(\left(\max_{k,\Phi} \lambda_k(f')\right) - \alpha^+\right) = \gamma(\alpha^+ - \alpha^+) = 0.\end{aligned}$$

5.3 Centered WENO5 reconstruction scheme

In [74], Levy et al. propose a central WENO reconstruction scheme for which no characteristic decomposition is required and the upwinding is replaced by a straight-forward centered computation of the quantities involved, but retaining the non-oscillatory properties of the WENO methodology.

In their numerical results they noticed the oscillatory behavior of the central WENO schemes they proposed and found that the computation of the smoothness indicators is a crucial issue. Their numerical results suggested that one way to improve the resolution of the scheme near the discontinuities could be that all the components are sensitive to the presence of a discontinuity by means of their smoothness indicators. They propose then to define a global smoothness indicator, valid for all the components of the system, as an average of the different smoothness indicators defined in (3.12):

$$\begin{aligned} GI_{j,k} &= \frac{1}{M}\sum_{q=1}^{M}\frac{1}{||\Phi_q||_2}\left(\sum_{l=1}^{r-1}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} h^{2l-1}(p_{j,k,q}^{(l)}(x))^2dx\right) \\ &= \frac{1}{M}\sum_{q=1}^{M}\frac{1}{||\Phi_q||_2}I_{j,k,q} \end{aligned} \tag{5.2}$$

where M is the number of equations, and $p_{j,k,q}$ and $I_{j,k,q}$ are the polynomial reconstruction and the Jiang and Shu's smoothness indicator of the data $\{f_{j+k-2,q}, f_{j+k-1,q}, f_{j+k,q}\}$, respectively. The quantity $||\Phi_q||_2$ is a scaling factor, and it is defined as the L^2 norm of the cell averages of the $r-$th component of Φ, namely:

$$||\Phi_q||_2 = \left(\sum_{j=1}^{N}|\Phi_{j,q}|^2h\right)^{\frac{1}{2}},$$

In [75], the authors made an 2D extension of these schemes. The correspondingly modified reconstruction (3.8), obtained with the global smoothness indicators GI in (5.2), will be termed G-WENO5.

5.4 Numerical experiments for polydisperse sedimentation models

The numerical scheme that we use in this section use a varying time step Δt computed as:

$$\Delta t = \frac{0.5h}{C},$$

where C is an estimate of the maximal characteristic velocity of the approximated solution at the given time step. For the SPECINT scheme the estimate is based on the computed eigenvalues. For the COMP schemes, we use the bounds on the eigenvalues quoted in (2.23).

Since the edges of the spatial domain $[0, L]$ are the cell interfaces $x_{\frac{1}{2}} = 0$, $x_{N+\frac{1}{2}} = L$, our implementation for the zero-flux boundary conditions is as follows:

$$\hat{f}_{\frac{1}{2}} = \hat{f}_{N+\frac{1}{2}} = 0.$$

This ensures conservation of each species throughout the time evolution.

In the following experiments for polydisperse sedimentation models we work with normalized depth, consequently, the spatial coordinate x varies between $x = 0$ (surface of the suspension) and $x = 1$ (bottom of the settling column).

The L^1-error for an approximation $(\phi_{j,k})$, $j = 1, \dots, N$, $k = 1, \dots, M$ to the solution at the cell centers x_j and given time t, $(\phi_k(x_j, t))$, is computed as

$$\frac{1}{N}\sum_{j=1}^{N}\sum_{k=1}^{M}|\phi^{\mathrm{ref}}_{j,k} - \phi_{j,k}|$$

where $(\phi^{\mathrm{ref}}_{j,k})$ is a reference solution computed at a fairly high resolution and interpolated at the coarse cell centers. In all the experiments, the reference solution is computed by the SPECINT scheme.

We will compare the different techniques at hand for improving the resolution of component-wise finite-difference WENO schemes. The basic schemes will be named after the flux-splitting, with a prefix L or G, depending on the character of the flux-splitting: LLF, LHLL, GLF, GHLL. If global smoothness indicators are used, the corresponding scheme will bear an additional G- prefix.

Test 5.1.

We consider this standard test case, proposed by Greenspan and Ungarish in [49] and solved numerically in [22, 24], defined by an initially homogeneous suspension in a column of height $L = 0.3m$ with four different species of particles with $D_1 = 4.96 \cdot 10^{-4}m$ and different normalized sizes $d_1 = 1$, $d_2 = D_2/D_1 = 0.8$, $d_3 = D_3/D_1 = 0.6$ and $d_4 = D_4/D_1 = 0.4$ and same density $\varrho_s = 2790kg/m^3$. The initial concentrations of the particles are $\phi_i^0 = 0.05$ for all $i = 1, \dots, 4$, the Richardson-Zaki exponent is $n_{\mathrm{RZ}} = 4.7$ and the maximum total concentration is $\phi_{max} = 0.68$. The density and viscosity of the fluid are $\varrho_f = 1208kg/m^3$ and $\mu_f = 0.02416kg/(s \cdot m)$, respectively.

In Figure 5.1 the reference solution, computed with SPECINT with $N = 6400$ cells and $\varepsilon = h^5$, is displayed. In Figures 5.2 and 5.3 we display some enlarged views of the numerical approximations of ϕ_1 and ϕ_3 for all the numerical schemes that we consider in this comparison. The conclusions about the qualitative behavior of the approximations that we could draw from inspection of the other components would be similar to those obtained for our choice.

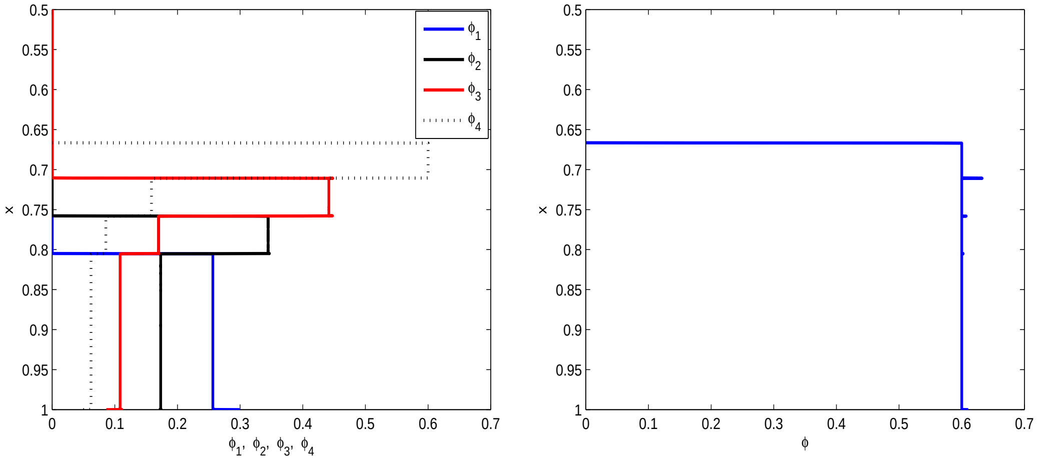
It can be seen throughout the pictures in Figures 5.2 and 5.3 that: the LF-based schemes are more diffusive than their HLL-based counterparts; schemes that use local flux-splittings are less oscillatory in smooth regions than their corresponding global flux-splitting schemes, but may present stronger oscillations near sharp discontinuities; the G-WENO5 reconstructions may help in reducing oscillations.

To perform quantitative assessments, in Table 5.1 and Figure 5.4 we show the approximate L^1-errors and the CPU times for this test. We also show the results for the SPECINT scheme. We have run each of the schemes for $N = 100, 200, 400, 800, 1600$ and recorded its CPU time for the execution and approximate L^1-error. Each symbol in a given graphic corresponds to a number N of cells.

As could be expected from the previous comments, global schemes should be less accurate than local schemes and schemes that use G-WENO5 reconstructions might be more accurate than those using WENO5 reconstructions. But when considering CPU times, it should be taken into account that: the local computations of extremal characteristic speeds has a higher computational cost than their global computation; the G-WENO5 reconstruction might be slightly faster than the WENO5 reconstruction, for the former requires less divisions (an arithmetical operation that may take about 20 times more CPU time than sums or products) to compute the weights than the latter.

From Figure 5.4 we deduce that the LHLL is the most accurate of the component-wise schemes, closely followed by the LLF scheme. Compared with the GLF scheme, those take about 10 times less computational time to achieve a given error level. If we bear in mind the considerations in the previous paragraph, the computational time of each of the component-wise schemes is comparable for moderate resolutions ($N \geq 400$, say). The conclusions that can be drawn from this figure are that local flux-splittings are more efficient than global ones (i.e., they take less CPU time for a given accuracy), that the HLL flux-splitting yields more efficiency than the LF flux-splitting, but the contribution of the G-WENO5 reconstruction to enhance the efficiency is not clear.

The SPECINT scheme for a given N has an accuracy roughly comparable to LHLL with $4N$ cells for about half the cost. The SPECINT scheme is therefore more efficient than LHLL, but this one is much more competitive with respect to SPECINT than the GLF scheme.

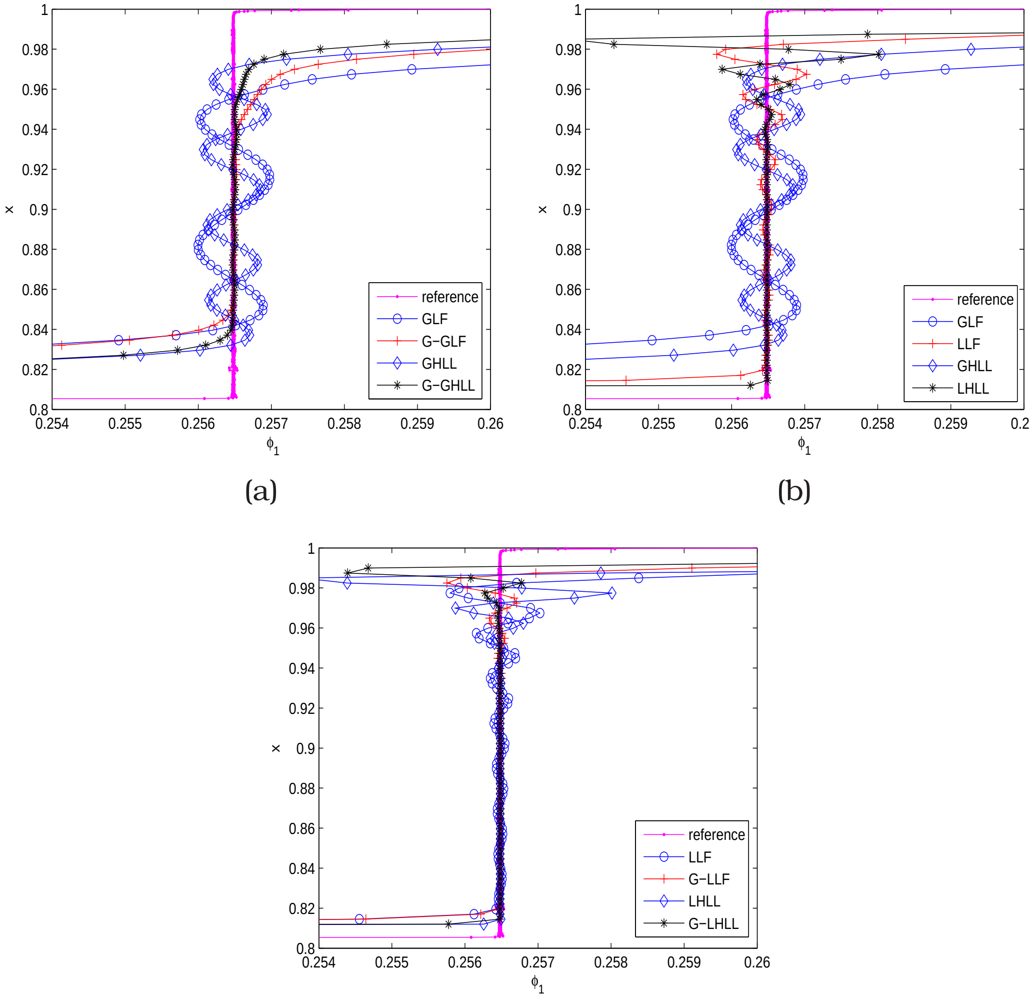


(a) (b)

Figure 5.1: *Enlarged views of the reference solution for* $\phi_1, \ldots, \phi_4$ *(a) and* $\phi = \sum_i \phi_i$ *(b) for Test 5.1 computed by SPECINT scheme for* $t = 300$*s with N=6400 cells.*

Test 5.2.

We consider here an example based on experimental data from [92]. It consists on the batch settling of an initially homogeneous suspension with eleven different species, in a column of height $L = 0.935$ m. We consider the Richardson-Zaki exponent $n_{RZ} = 4.65$, the maximum total concentration $\phi_{max} = 0.641$ and that the density of solid particles is $\varrho_s = 2790 kg/m^3$. The initial concentrations ϕ_i^0, diameters D_i and normalized diameters $d_i = D_i/D_1$ of the particles are given in Table 5.2. The

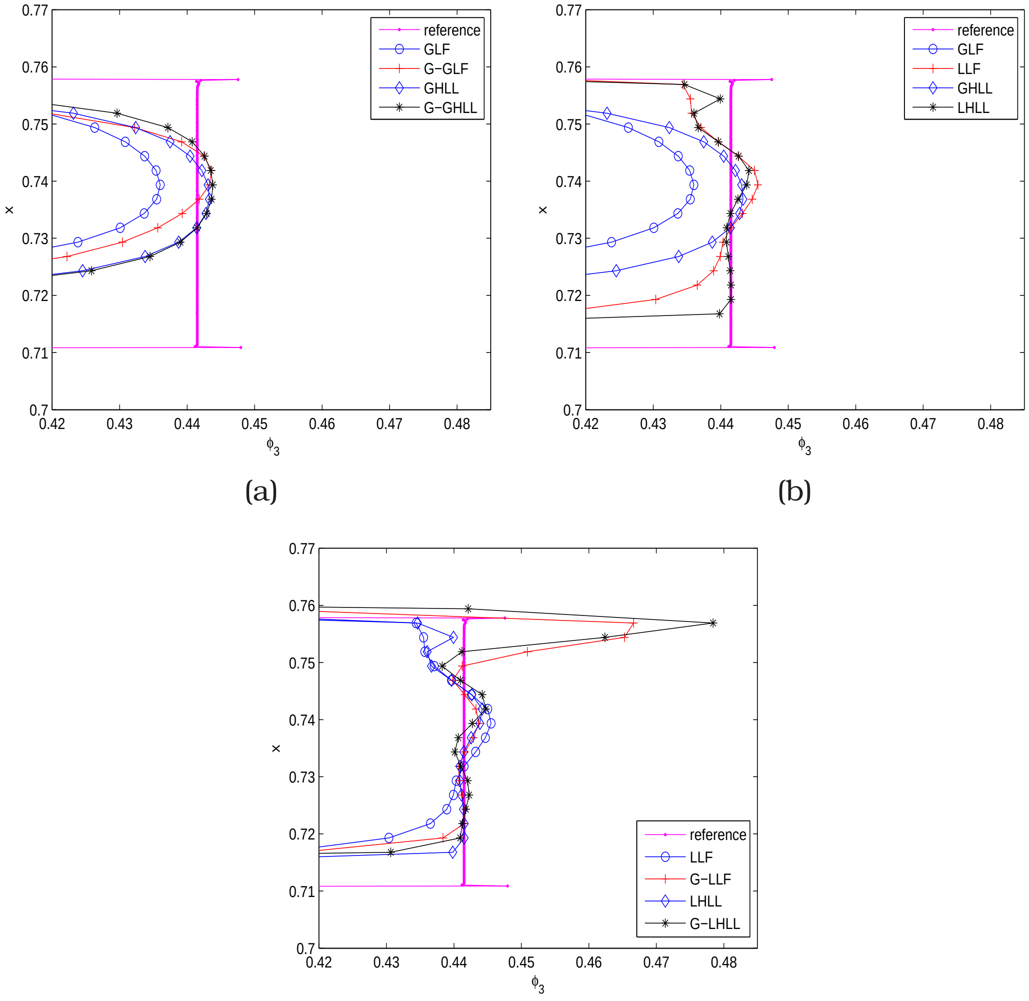


(a)

(b)

(c)

Figure 5.2: *Enlarged views of* ϕ_1 *for test 5.1 with* $N = 400$ *computed with all the versions of the component-wise scheme analyzed in this work.*



(a)

(b)

(c)

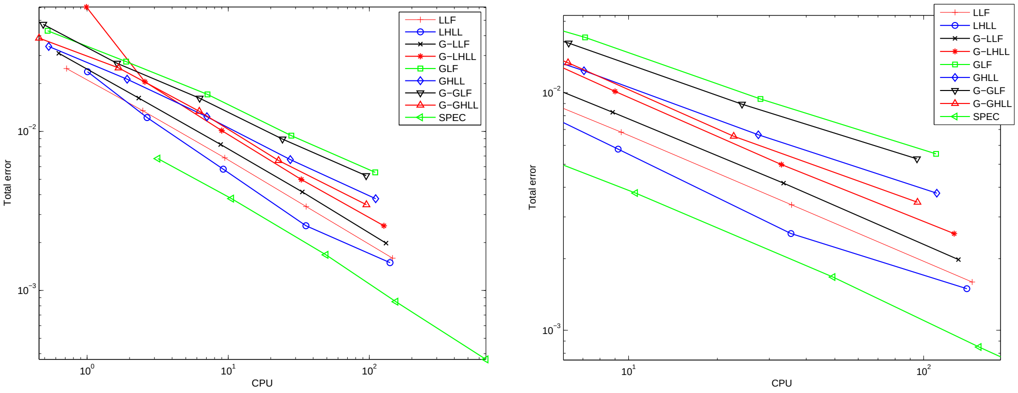
Figure 5.3: *Enlarged views of* ϕ_3 *for test 5.1 with* $N = 400$ *computed with all the versions of the component-wise scheme analyzed in this work.*

N	LLF		GLF		G-GLF	
	CPU	error	CPU	error	CPU	error
100	0.715	24.86	0.526	42.93	0.488	47.14
200	2.481	13.48	1.890	27.48	1.631	26.82
400	9.444	6.821	7.125	17.10	6.269	16.17
800	35.68	3.370	27.98	9.409	24.23	8.944
1600	145.8	1.595	110.1	5.528	94.90	5.266

N	LHLL		GHLL		G-GHLL	
	CPU	error	CPU	error	CPU	error
100	1.007	23.74	0.534	34.23	0.457	38.63
200	2.659	12.22	1.923	21.29	1.671	25.08
400	9.221	5.786	7.061	12.36	6.236	13.37
800	35.50	2.553	27.52	6.652	22.71	6.559
1600	140.0	1.495	110.7	3.774	95.24	3.458

N	G-LLF		G-LHLL		SPEC INT	
	CPU	error	CPU	error	CPU	error
100	0.630	31.00	0.988	60.62	3.171	6.749
200	2.321	16.21	2.570	20.50	10.55	3.781
400	8.835	8.274	8.999	10.12	49.26	1.675
800	33.49	4.161	32.95	4.982	154.0	0.850
1600	131.1	1.983	126.8	2.550	669.6	0.369

Table 5.1: *Approximate L^1-errors ($\times 10^{-3}$) and CPU times (seconds) for test 5.1.*



(a)

(b)

Figure 5.4: *(a): CPU-error comparison for test 5.1 with $t = 300$s.; (b): enlarged view of (a).*

characteristic of the fluid are those of the previous test. We show the results obtained at $t = 300$s with $\varepsilon = h^5$.

In Figure 5.5 the reference solution, computed with SPECINT with $N = 6400$ cells, is displayed. The appearance of very thin layers of sediment of the smaller particles at the top of the sedimentation vessel poses severe difficulties for the numerical schemes to capture them.

In Figures 5.6 and 5.7 we display some enlarged views of the numerical approximations of ϕ_5 and ϕ_{10} for all the numerical schemes under consideration. As in the previous test, it can be seen throughout the pictures that: the LF-based schemes are more diffusive than their HLL-based counterparts; schemes that use local flux-splittings are less diffusive than their corresponding global flux-splitting schemes; the G-WENO5 reconstructions do not seem to help in reducing oscillations near sharp gradients.

To get quantitative assessments, in Table 5.3 and Figure 5.8 we show the approximate L^1-errors and the CPU times for this test. We also show the results for the SPECINT scheme. We have run each of the schemes for $N = 100, 200, 400, 800, 1600$ and recorded its CPU time for the execution and approximate L^1-error.

From Figure 5.8 we deduce that the differences between the component-wise schemes are more reduced than in the previous test and that they are more efficient than the SPECINT scheme, which is penalized by the numerical solution of quite large eigenvalue/eigenvector problems. It is also deduced that the LHLL is the most accurate of the component-wise schemes, closely followed by the G-LHLL scheme. The G-GHLL scheme is slightly less accurate than those schemes, but saves some computational time, so, in this case, the G-GHLL scheme is the most efficient.

i	1	2	3	4	5	6
$\phi_i^0[10^{-3}]$	0.435	3.747	14.420	32.603	47.912	47.762
$D_i[10^{-5}]$	8.769	8.345	7.921	7.497	7.073	6.649
d_i	1.000	0.952	0.903	0.855	0.807	0.758
i	7	8	9	10	11	
$\phi_i^0[10^{-3}]$	32.663	15.104	4.511	0.783	0.060	
$D_i[10^{-5}]$	6.225	5.801	5.377	4.953	4.529	
d_i	0.710	0.662	0.613	0.565	0.516	

Table 5.2: *Test 5.2: initial concentrations ϕ_i^0, real and normalized diameters D_i and d_i.*

N	**LLF**		**GLF**		**G-GLF**	
	CPU	error	CPU	error	CPU	error
100	0.344	10.49	0.315	12.94	0.225	12.36
200	1.366	5.989	1.090	7.628	0.807	7.319
400	5.316	3.166	4.119	4.185	3.201	4.145
800	21.14	1.770	12.60	2.319	16.82	2.350
1600	84.84	0.832	66.92	1.166	51.67	1.130

N	**LHLL**		**GHLL**		**G-GHLL**	
	CPU	error	CPU	error	CPU	error
100	0.337	10.28	0.265	10.91	0.218	10.33
200	1.312	5.545	1.060	6.071	0.813	5.773
400	5.332	2.859	4.197	3.233	3.170	3.227
800	21.23	1.587	16.87	1.813	12.70	1.818
1600	84.93	0.720	67.26	0.857	51.62	0.851

N	**G-LLF**		**G-LHLL**		**SPEC INT**	
	CPU	error	CPU	error	CPU	error
100	0.300	10.23	0.316	10.08	6.606	9.582
200	1.142	5.834	1.140	5.311	25.32	4.931
400	4.538	3.230	4.493	2.896	100.6	2.434
800	18.59	1.816	17.54	1.634	401.3	1.368
1600	72.32	0.850	71.68	0.743	1507	0.594

Table 5.3: *Approximate L^1-errors ($\times 10^{-3}$) and CPU times (seconds) for test 5.2.*

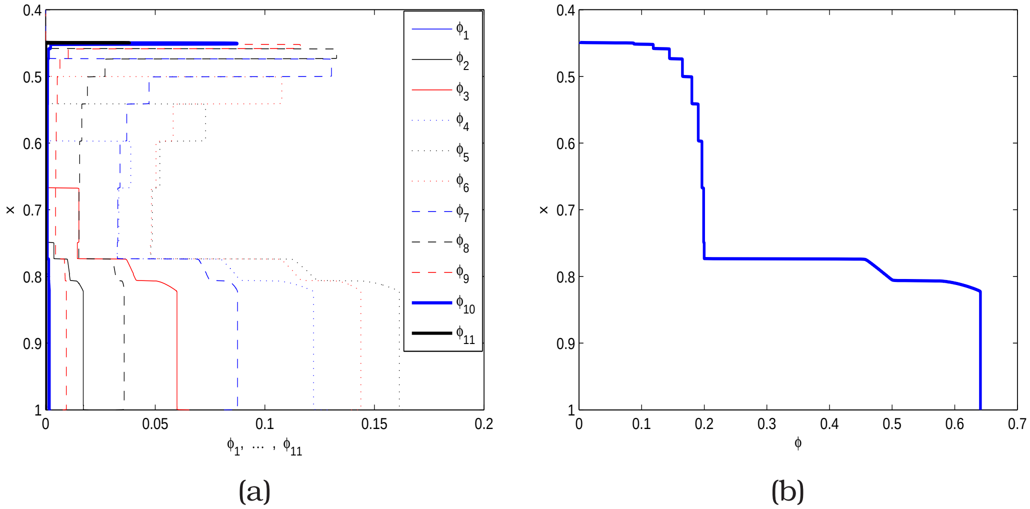


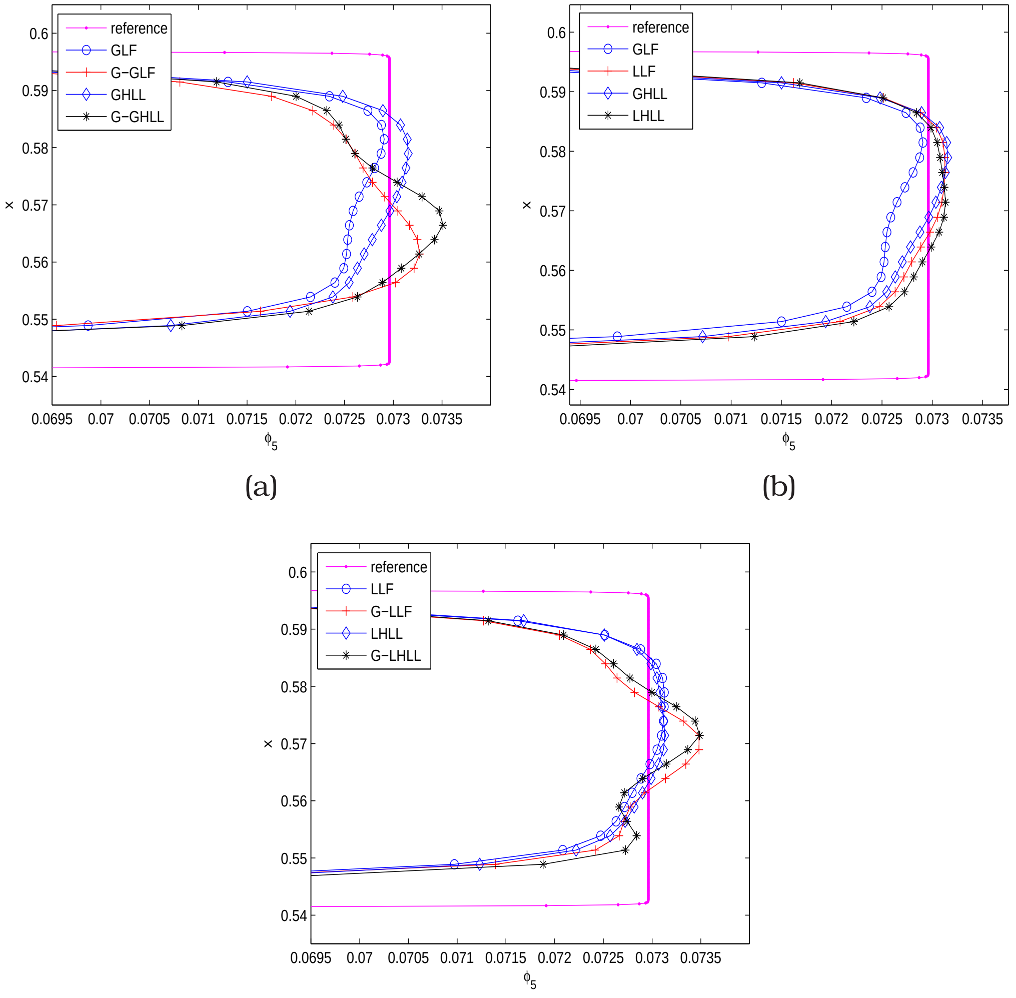
Figure 5.5: *Enlarged views of (a) reference solution $\phi_1, \dots, \phi_{11}$ and (b) reference solution $\phi = \sum_i \phi_i$ in test 5.2 computed by SPEC INT scheme with N=6400 cells and $t = 300s$.*

5.5 Further Numerical Experiments

In this section we repeat some of the experiments carried in this chapter and in chapter 4. We compute the numerical solutions using all the different techniques at hand for improving the resolution of finite-difference WENO schemes:

- the different weights' definitions for the WENO scheme that we have seen in chapter 4: Jiang and Shu's weights (JS) (3.11) - (3.12), Yamaleev and Carpenter's weights (YC) (4.1 - 4.4) and our proposed weights (AMM) (4.6) - (4.7),
- the flux-splitting methods used in this chapter: global and local Lax-Friedrichs flux-splitting (LF) and HLL flux-splitting (HLL).

As in the previous experiments, the basic schemes will be named after the flux-splitting, with a prefix L or G, depending on the character of the flux-splitting: LLF, LHLL, GLF, GHLL. Depending on the WENO scheme used, JS-WENO, YC-WENO or AMM-WENO, the corresponding scheme will bear an additional JS-, YC- or AMM- prefix respectively.

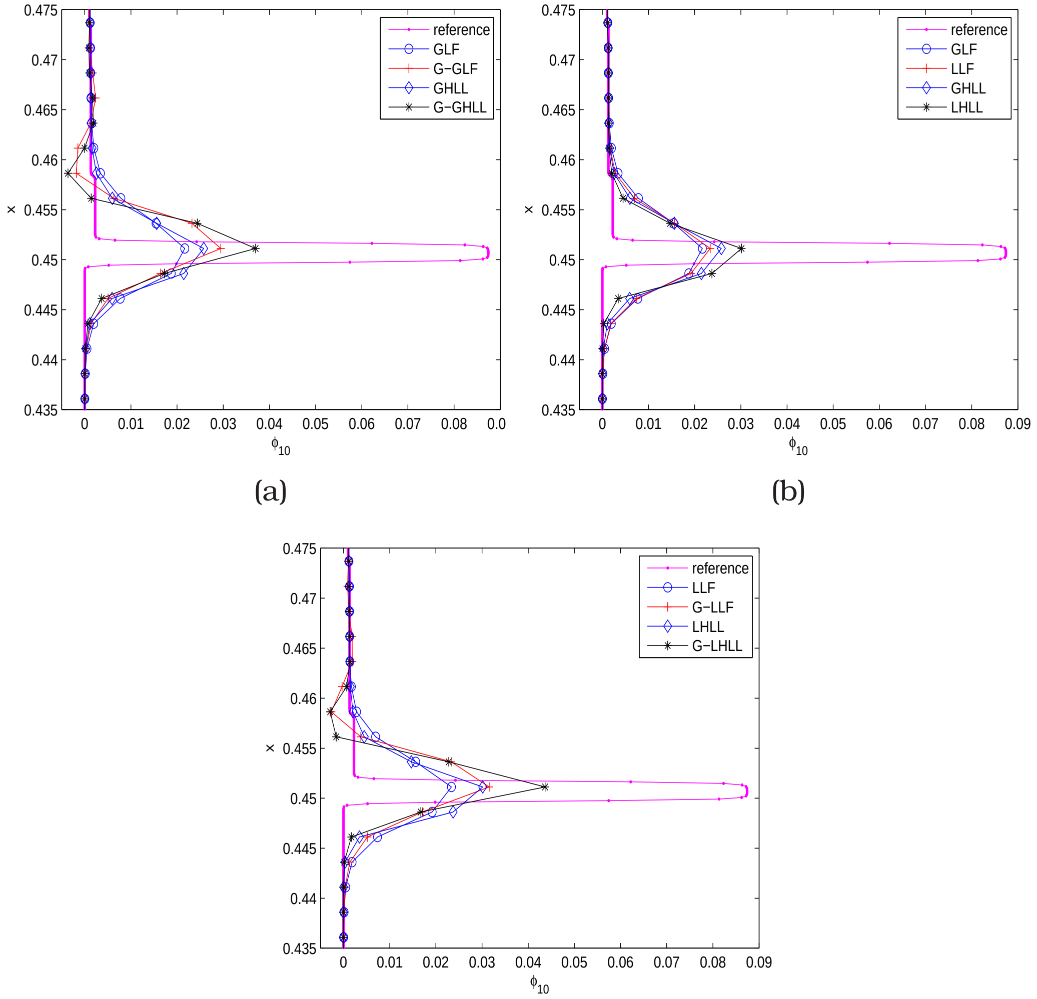


(a)

(b)

(c)

Figure 5.6: *Enlarged views of ϕ_5 for test 5.2 computed with $N = 400$ and all the versions of the component-wise scheme analyzed in this work.*

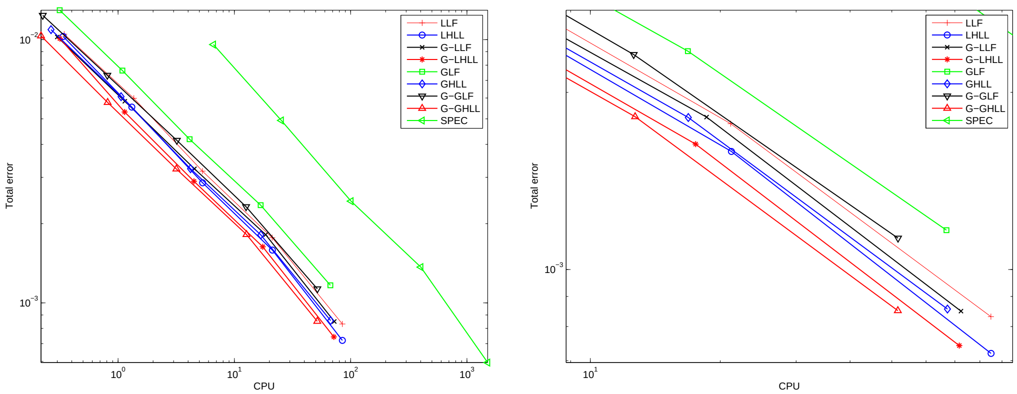


(a)

(b)

(c)

Figure 5.7: *Enlarged views of* ϕ_{10} *for test 5.2 computed with* $N = 400$ *and all the versions of the component-wise scheme analyzed in this work.*



(a) (b)

Figure 5.8: *(a): CPU-error comparison for test 5.2 for* $t = 300s$*.; (b): enlarged view of (a).*

5.5.1 One-dimensional tests

Test 5.3.

Let us consider first Sod's problem, explained in Test 4.6. In this section we show the results obtained using WENO5 reconstruction scheme with local LF and HLL flux-splitting with parameter $\varepsilon = h^2$. A reference solution is obtained with the JS-WENO5 scheme and $N = 6400$ cells.

As it could be seen in Figures 5.10 and 5.11 the use of the HLL flux-splitting increases the oscillatory behavior of the numerical solutions obtained with YC-WENO5 reconstruction scheme, especially near discontinuities. These oscillations do not seem to diminish with mesh refinement as it could be seen in Figure 5.11. With both flux-splittings the best results are obtained with AMM-WENO5 reconstruction scheme. It could be also noticed that the oscillations obtained with WENO5 reconstruction scheme are less strong than those obtained with WENO9 scheme (see chapter 4, Figures 4.5 and 4.6).

We also show in Figure 5.12 the results obtained using WENO9 reconstruction scheme and a local HLL flux-splitting. The use of the HLL flux-splitting does not seem to diminish the numerical oscillations obtained with YC-WENO9 scheme.

Test 5.4.

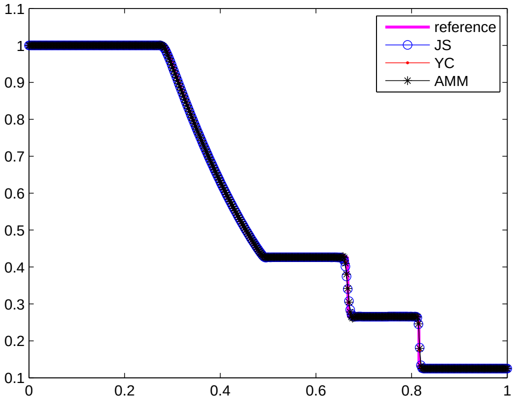


Figure 5.9: *Numerical solution of ρ in test 5.3 computed with $N = 400$ nodes.*

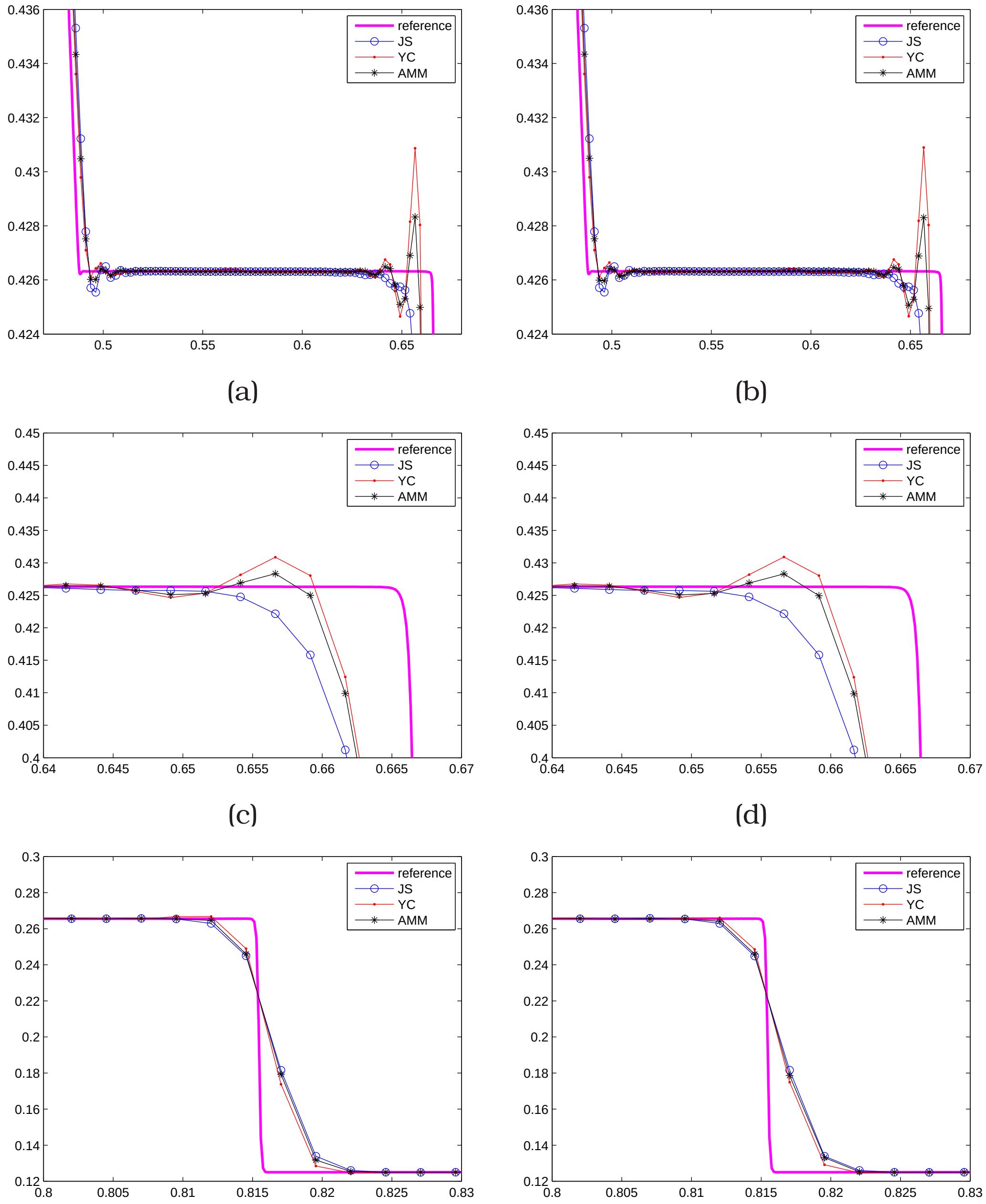
We analyze the polydisperse sedimentation test 5.1, defined by an initially homogeneous suspension in a column with four different species of particles with same density. We use $\varepsilon = h^5$ for WENO5 reconstruction scheme and $\varepsilon = h^{11}$ for WENO9 reconstruction scheme, regardless of the definition of the weights chosen.

In Figures 5.13 and 5.14 we show some enlarged views of the numerical solution of ϕ_2 and ϕ_4 computed with a global LF component-wise scheme and WENO5 reconstruction scheme. The conclusions about the qualitative behavior of the approximations that we could draw from inspection of the other components would be similar to those obtained for our choice.

It can be seen throughout those figures and Table 5.4 that the use of YC weights leads to more oscillations than the use of JS or AMM weights. In terms of accuracy, the results obtained using AMM weights are better than the results obtained with YC weights but they are not as good as the results obtained with JS weights.

In Figure 5.15 we show the numerical solutions for ϕ_4 obtained when using global HLL flux-splitting. It could be seen that the use of this flux-splitting diminishes the oscillatory behavior, improving the results of the AMM-WENO5 scheme. The results of the YC-WENO5 scheme are also improved but they are not as good as JS-WENO5 or AMM-WENO5 results.

In Figure 5.16 we show the approximations of ϕ_4 obtained when using a global LF and HLL flux-splitting and WENO9 reconstruction scheme.

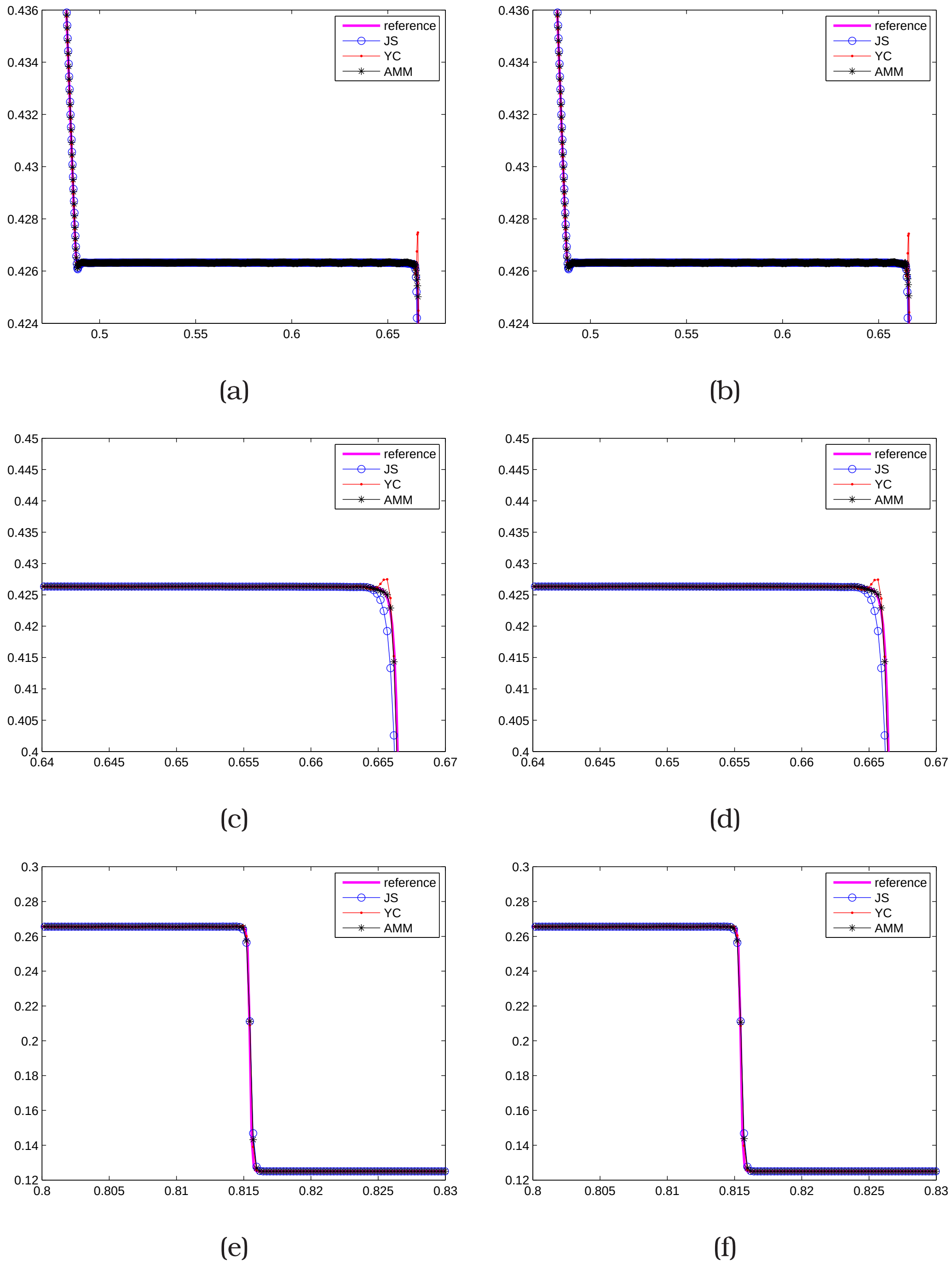


(a) (b)

(c) (d)

(e) (f)

Figure 5.10: *Enlarged views of interesting regions of the approximation of ρ for test 5.3 computed with $N = 400$ and with all the versions of the weights for the WENO5 scheme analyzed in this work and local Lax-Friedrichs ((a), (c) and (e)) and HLL ((b), (d) and (f)) characteristic based scheme. (c) and (d) are enlarged views of the discontinuity present at the right side on pictures (a) and (b) respectively.*

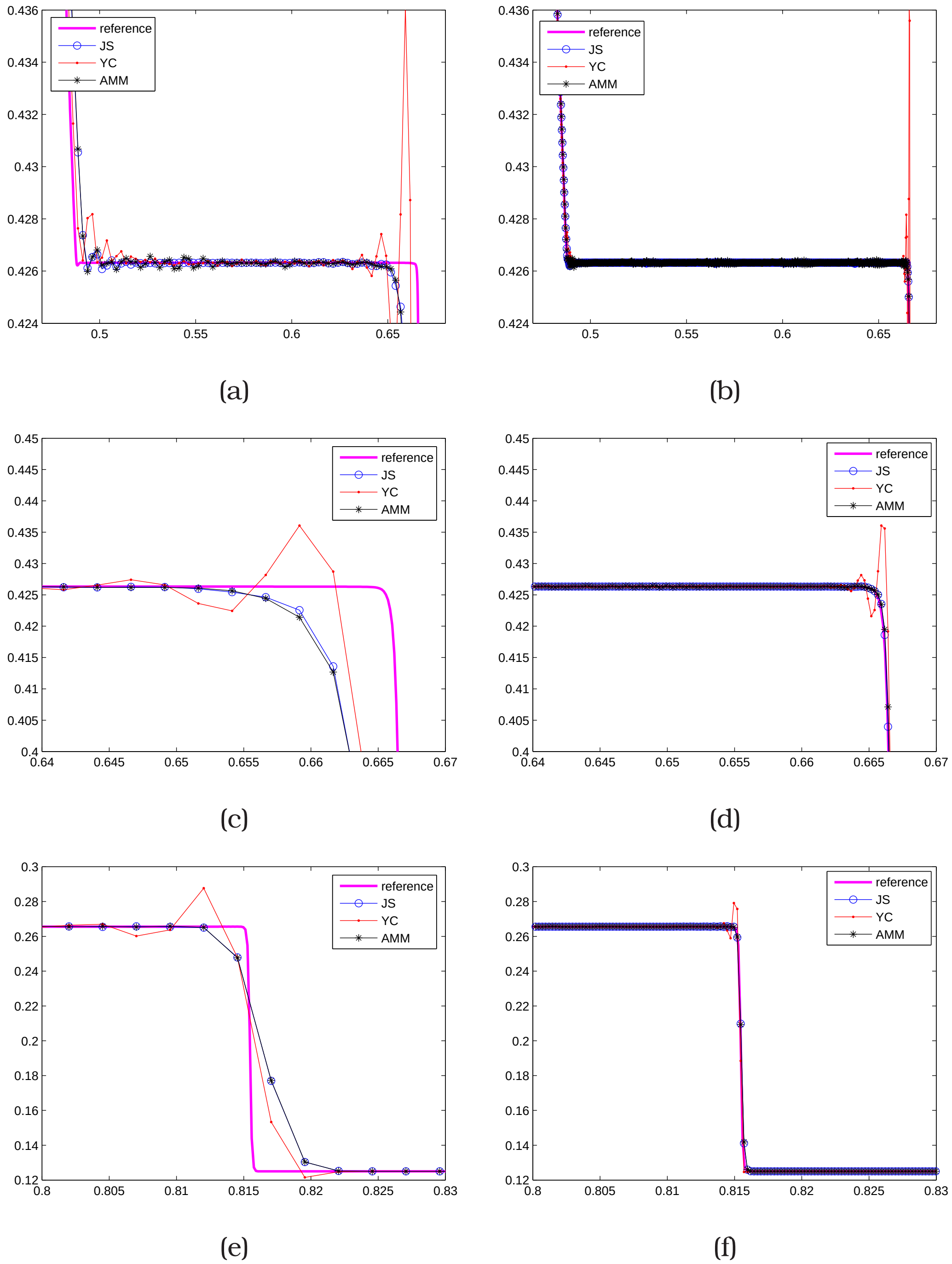


(a) (b)

(c) (d)

(e) (f)

Figure 5.11: *Enlarged views of interesting regions of the approximation of ρ for test 5.3 computed with $N = 4000$ and with all the versions of the weights for the WENO5 scheme analyzed in this work and local Lax-Friedrichs ((a), (c) and (e)) and HLL ((b), (d) and (f)) characteristic based scheme. (c) and (d) are enlarged views of the discontinuity present at the right side on pictures (a) and (b) respectively.*



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.12: *Enlarged views of interesting regions of the approximation of* ρ *for test 5.3 computed with* $N = 400$ *((a), (c) and (e)) and* $N = 4000$ *((b), (d) and (f)) and with all the versions of the weights for the WENO9 scheme analyzed in this work and local HLL characteristic based scheme. (c) and (d) are enlarged views of the discontinuity present at the right side on pictures (a) and (b) respectively.*

As it could be seen, in this case the numerical solutions obtained using YC weights present very strong oscillations near discontinuities and, as we show in Table 5.5, the accuracy of the scheme is worse than the accuracy obtained when using JS and AMM weights.

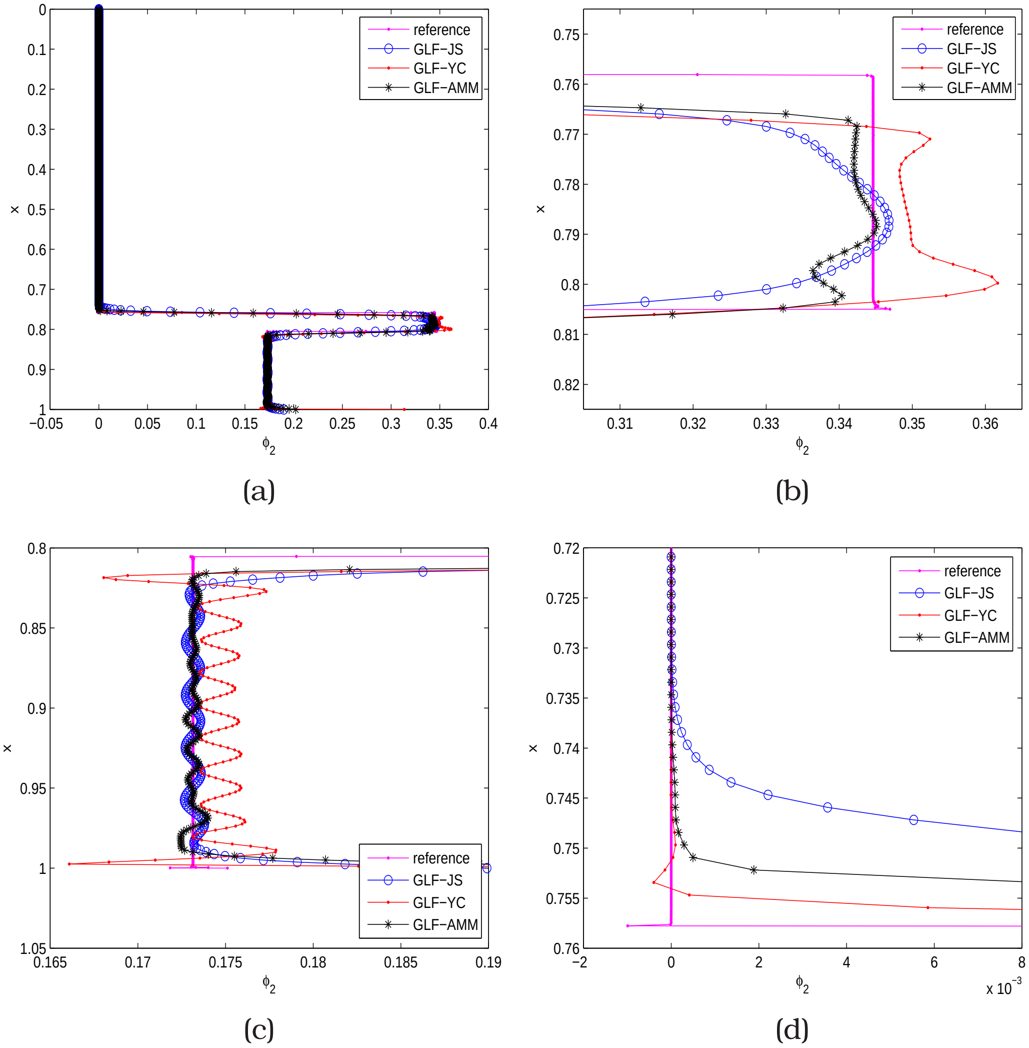
N	**LLF-JS**		**LHLL-JS**		**GLF-JS**		**GHLL-JS**	
	CPU	error	CPU	error	CPU	error	CPU	error
100	0.715	24.86	1.007	23.74	0.526	42.93	0.534	34.23
200	2.481	13.48	2.659	12.22	1.890	27.48	1.923	21.29
400	9.444	6.821	9.221	5.786	7.125	17.10	7.061	12.36
800	35.68	3.370	35.50	2.553	27.98	9.409	27.52	6.652
1600	145.8	1.595	140.0	1.495	110.1	5.528	110.7	3.774
N	**LLF-YC**		**LHLL-YC**		**GLF-YC**		**GHLL-YC**	
	CPU	error	CPU	error	CPU	error	CPU	error
100	0.855	26.63	1.033	31.71	0.559	55.75	0.731	36.82
200	3.432	14.71	3.208	17.40	1.964	43.21	2.914	24.23
400	11.86	8.364	11.17	8.319	10.30	31.26	8.340	13.75
800	46.94	4.848	41.39	4.386	30.17	20.04	28.14	7.708
1600	212.0	2.563	179.3	51.50	114.3	17.41	113.2	4.306
N	**LLF-AMM**		**LHLL-AMM**		**GLF-AMM**		**GHLL-AMM**	
	CPU	error	CPU	error	CPU	error	CPU	error
100	1.650	23.91	1.063	24.04	0.759	41.16	0.930	30.48
200	4.218	14.46	3.472	13.71	2.998	29.79	2.399	21.97
400	14.60	7.140	11.07	6.624	9.912	17.62	10.26	12.57
800	55.86	3.501	42.22	2.784	30.16	12.36	31.87	6.592
1600	234.9	1.656	170.6	1.623	132.7	9.062	143.7	3.892

Table 5.4: *Approximate L^1-errors ($\times 10^{-3}$) and CPU times (seconds) for test 5.4 using WENO5 reconstruction scheme.*

5.5.2 Two-dimensional tests

Test 5.5.

Finally, we show the results of test 4.8, the double Mach reflection of a strong shock simulated with the 2D Euler equations of gas dynamics, computed using two resolution grids of 1024×256 and 2048×512



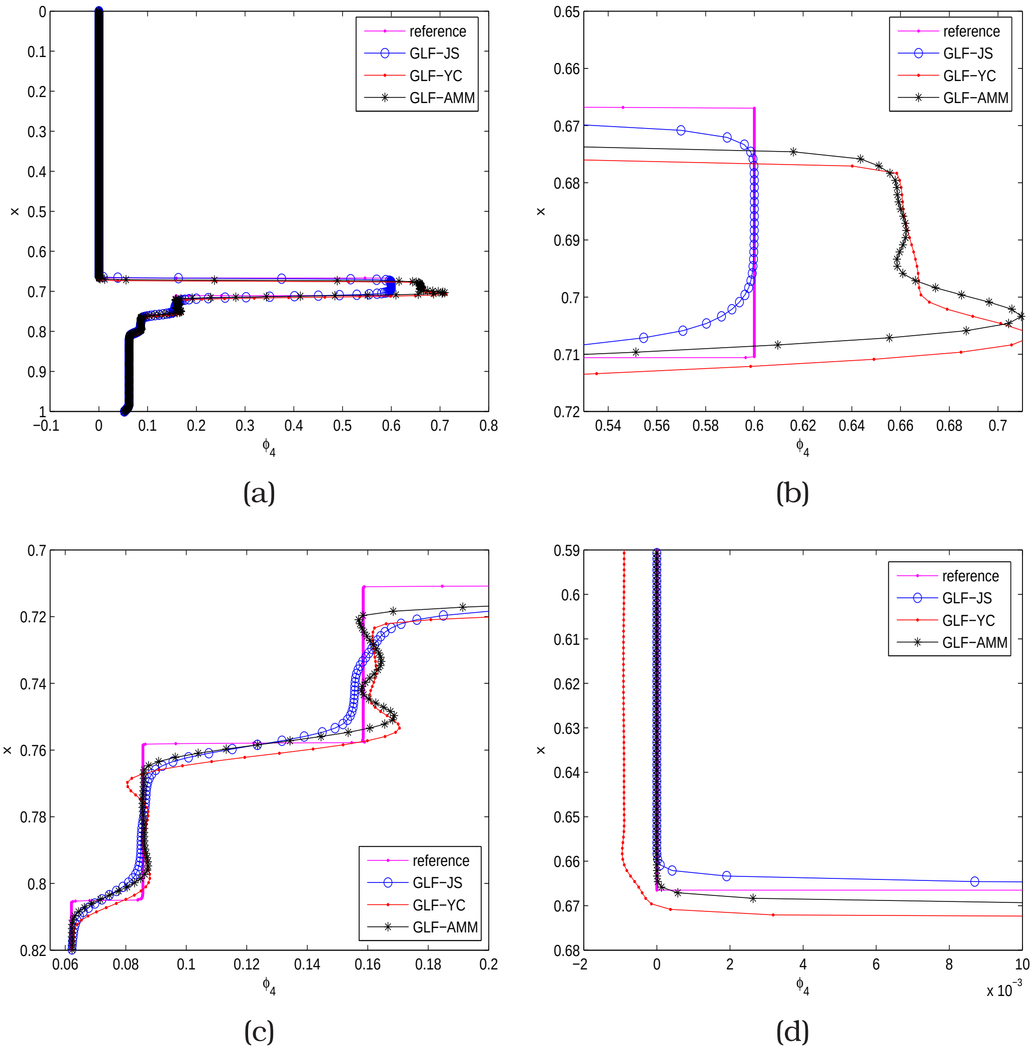
(a)

(b)

(c)

(d)

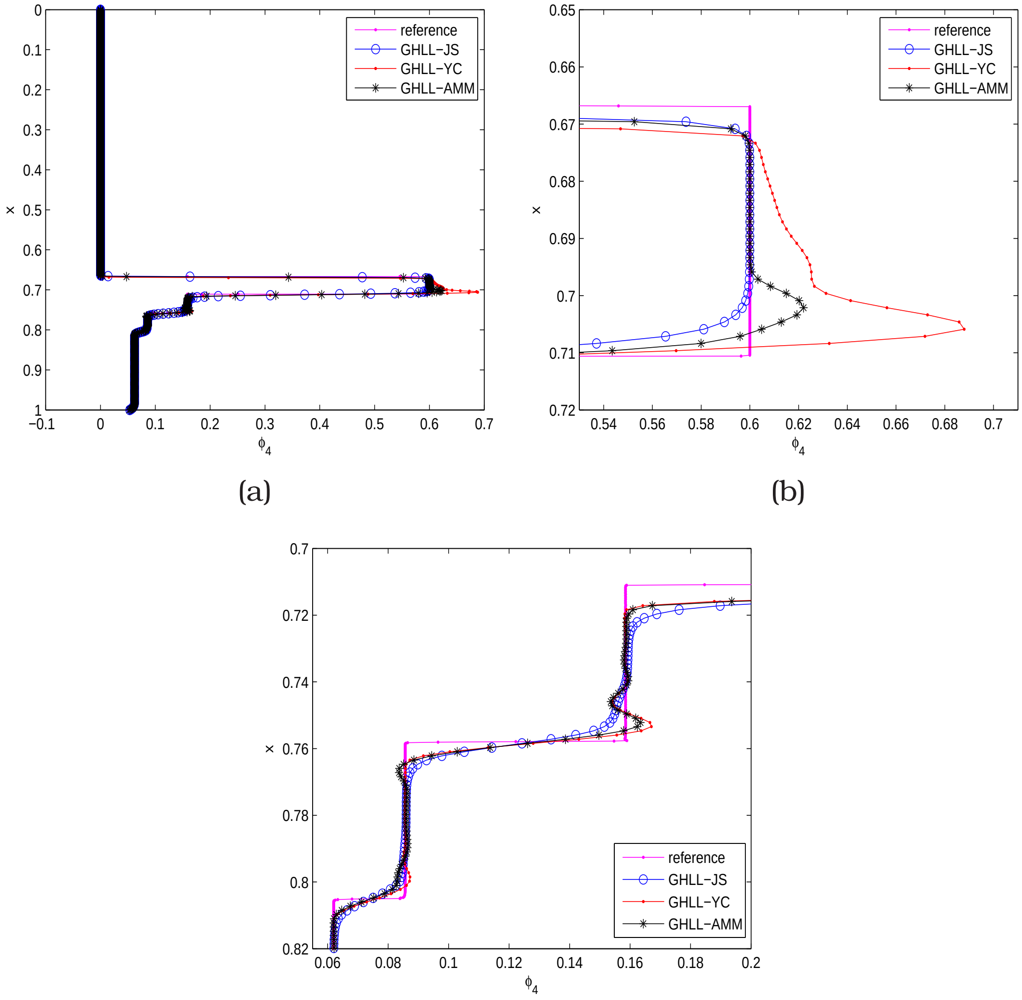
Figure 5.13: *Enlarged views of interesting regions of ϕ_2 for test 5.4 computed with $N = 800$ and with all the versions of the weights for the WENO5 scheme analyzed in this work and a global Lax-Friedrichs component-wise scheme.*



(a) (b)

(c) (d)

Figure 5.14: *Enlarged views of interesting regions of ϕ_4 for test 5.4 computed with $N = 800$ and with all the versions of the weights for the WENO5 scheme analyzed in this work and a global Lax-Friedrichs component-wise scheme.*



(a) (b)

(c)

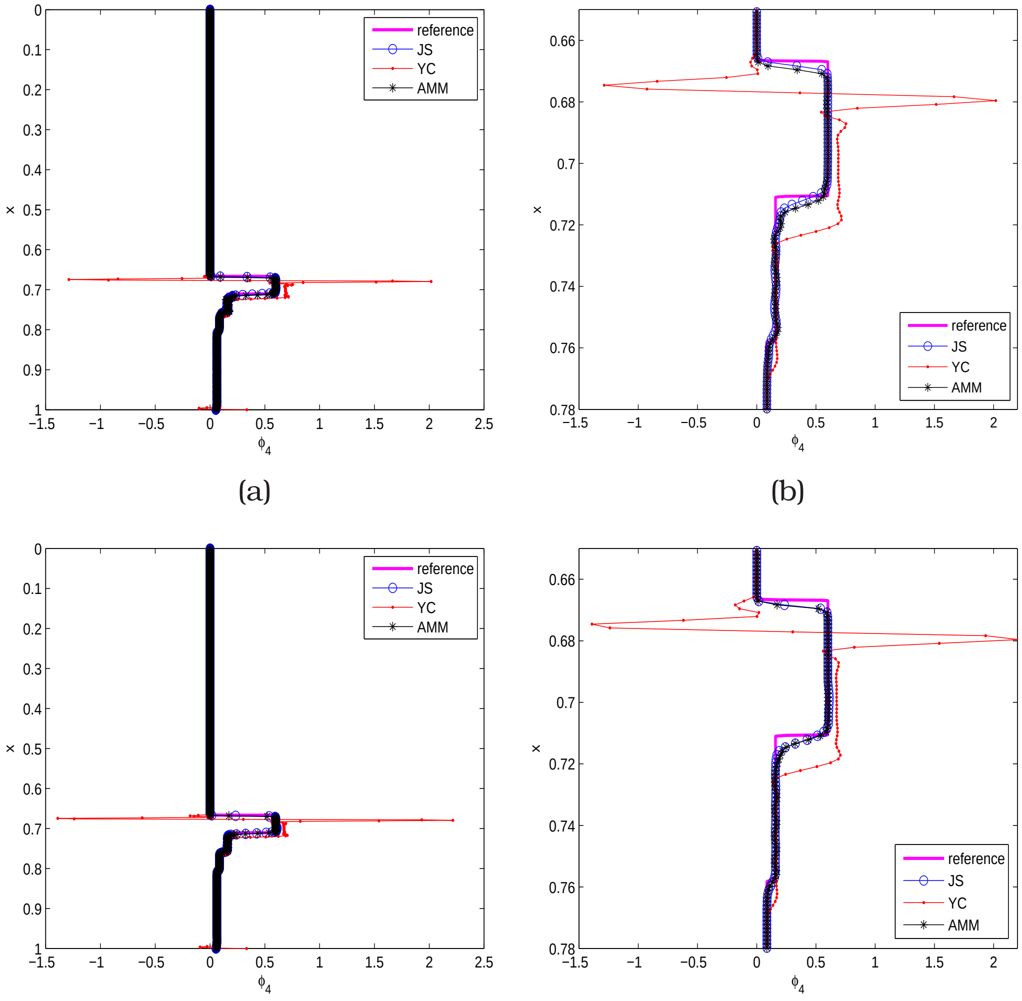
Figure 5.15: *Enlarged views of interesting regions of* ϕ_4 *for test 5.4 computed with* $N = 800$ *and with all the versions of the weights for the WENO5 scheme analyzed in this work and a global HLL flux-splitting component-wise scheme.*

N	**LLF-JS**		**LHLL-JS**		**GLF-JS**		**GHLL-JS**	
	CPU	error	CPU	error	CPU	error	CPU	error
100	3.516	30.08	2.832	33.83	1.576	38.04	1.226	31.62
200	44.75	14.93	7.578	50.59	18.83	21.97	10.93	20.50
400	102.9	8.341	31.97	55.70	26.01	12.78	21.61	11.66
800	188.3	4.889	100.6	54.98	131.9	7.268	71.08	6.725
1600	548.8	3.001	323.0	4.289	284.0	4.667	253.4	3.735

N	**LLF-YC**		**LHLL-YC**		**GLF-YC**		**GHLL-YC**	
	CPU	error	CPU	error	CPU	error	CPU	error
100	1.037	47.37	4.423	64.07	0.818	122.6	0.731	94.96
200	4.035	28.69	13.83	51.95	3.234	84.57	3.048	70.84
400	15.82	21.60	30.38	40.50	12.76	65.79	13.20	58.33
800	62.82	20.22	102.4	36.33	51.05	55.40	48.84	50.96
1600	315.8	18.33	579.0	33.74	199.4	18.83	193.6	47.30

N	**LLF-AMM**		**LHLL-AMM**		**GLF-AMM**		**GHLL-AMM**	
	CPU	error	CPU	error	CPU	error	CPU	error
100	3.509	30.27	2.688	33.65	1.523	37.69	1.517	31.60
200	24.90	15.02	7.326	50.65	17.80	25.56	6.643	21.43
400	59.44	7.739	30.63	55.64	24.48	16.38	20.09	12.86
800	177.6	4.542	94.59	54.95	90.81	11.99	66.93	7.776
1600	527.7	2.971	335.8	53.22	322.7	8.183	289.3	5.648

Table 5.5: *Approximate L^1-errors ($\times 10^{-3}$) and CPU times (seconds) for test 5.4 using WENO9 reconstruction scheme.*



(a) (b)

(c) (d)

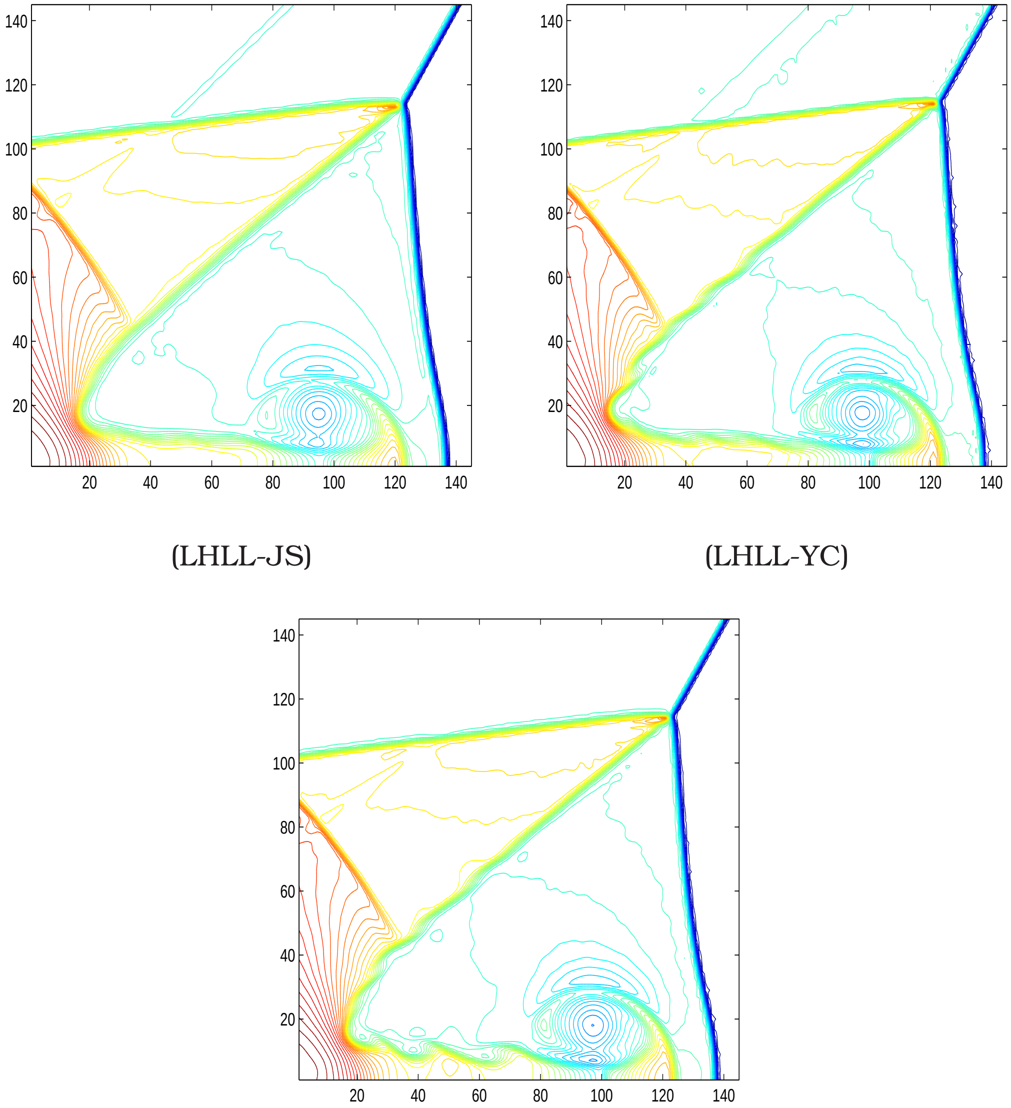
Figure 5.16: *Enlarged views of interesting regions of ϕ_4 for test 5.4 computed with $N = 800$ and with all the versions of the weights for the WENO9 scheme analyzed in this work and a global Lax-Friedrichs ((a) and (b)) and global HLL ((c) and (d)) component-wise scheme.*

nodes, a characteristic based scheme with local HLL flux splitting and JS-WENO5, YC-WENO5 or AMM-WENO5 reconstruction scheme.

If we compare the results obtained in Figures 5.17 and 5.19 with those showed in Figures 4.9 and 4.10 it can be seen that the results obtained with LLF-JS and LHLL-JS are very similar. When we use YC or AMM weights it can be seen that the numerical results obtained with both flux-splittings present some more vorticity than the LLF-JS numerical results. However, the results obtained with LLF flux-splitting seem to have some more vorticity than those obtained with LHLL flux-splitting.

In Figures 5.18 and 5.20 we show some sections to compare the results obtained with local LF and HLL flux-splitting and JS-WENO5, YC-WENO5 and AMM-WENO5 schemes with $\varepsilon = h^2$. When we use a LHLL flux-splitting the results obtained using JS-WENO5, YC-WENO5 and AMM-WENO5 schemes are very similar but the numerical solutions obtained with YC-WENO5 scheme show an stronger oscillatory behavior.

At last, in Figure 5.21 we show the results obtained with JS-WENO9 reconstruction scheme and local LF and HLL flux-splittings. It could be seen that the numerical solutions obtained with LHLL scheme present more or less the same vorticity than those computed with LLF scheme.

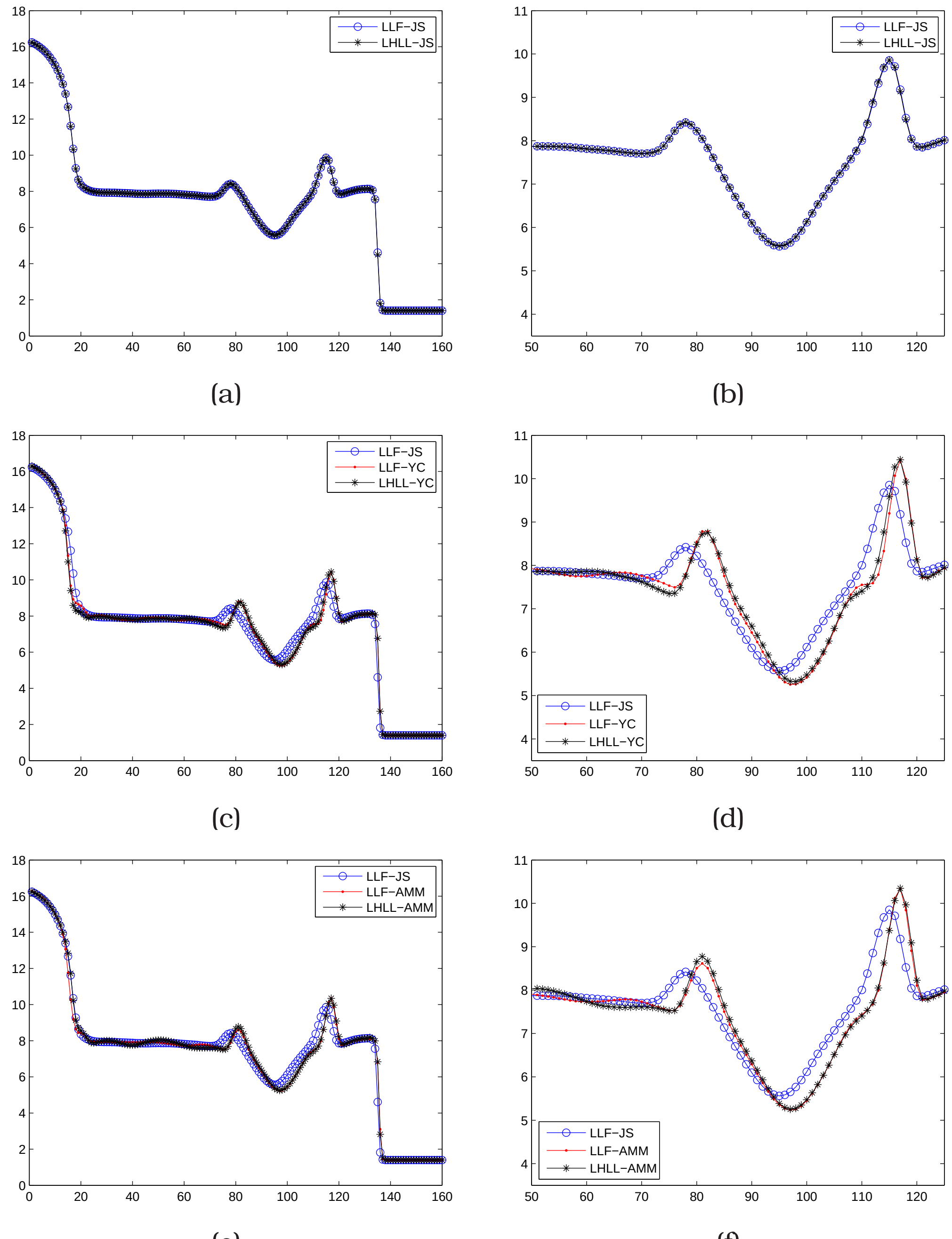


(LHLL-JS)

(LHLL-YC)

(LHLL-AMM)

Figure 5.17: *Results of test 5.5 for a grid of* 1024×256 *cells. We show 50 contour lines of the density obtained with a local LF and HLL flux-splitting and all the different definitions for the WENO weights studied in this work.*



(a)

(b)

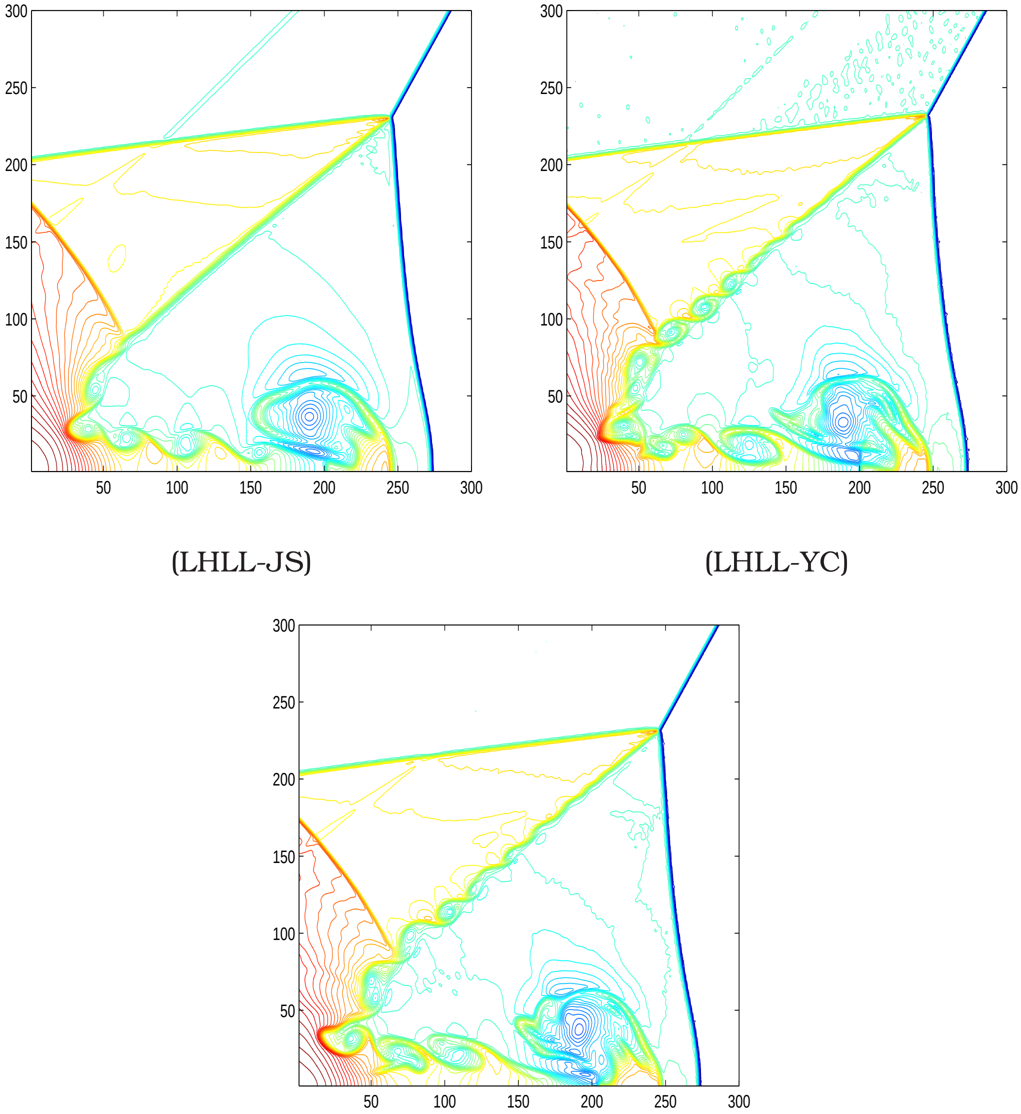
(c)

(d)

(e)

(f)

Figure 5.18: *Results of test 5.5 for a grid of* 1024×256 *cells. (a), (c) and (e) display sections of the* 50 *contour lines of the density obtained with a local LF and HLL flux-splitting and all the different definitions for the WENO weights studied in this work, at pixel height* 18*. (b), (d) and (f) are zooms of (a), (c) and (e) respectively.*

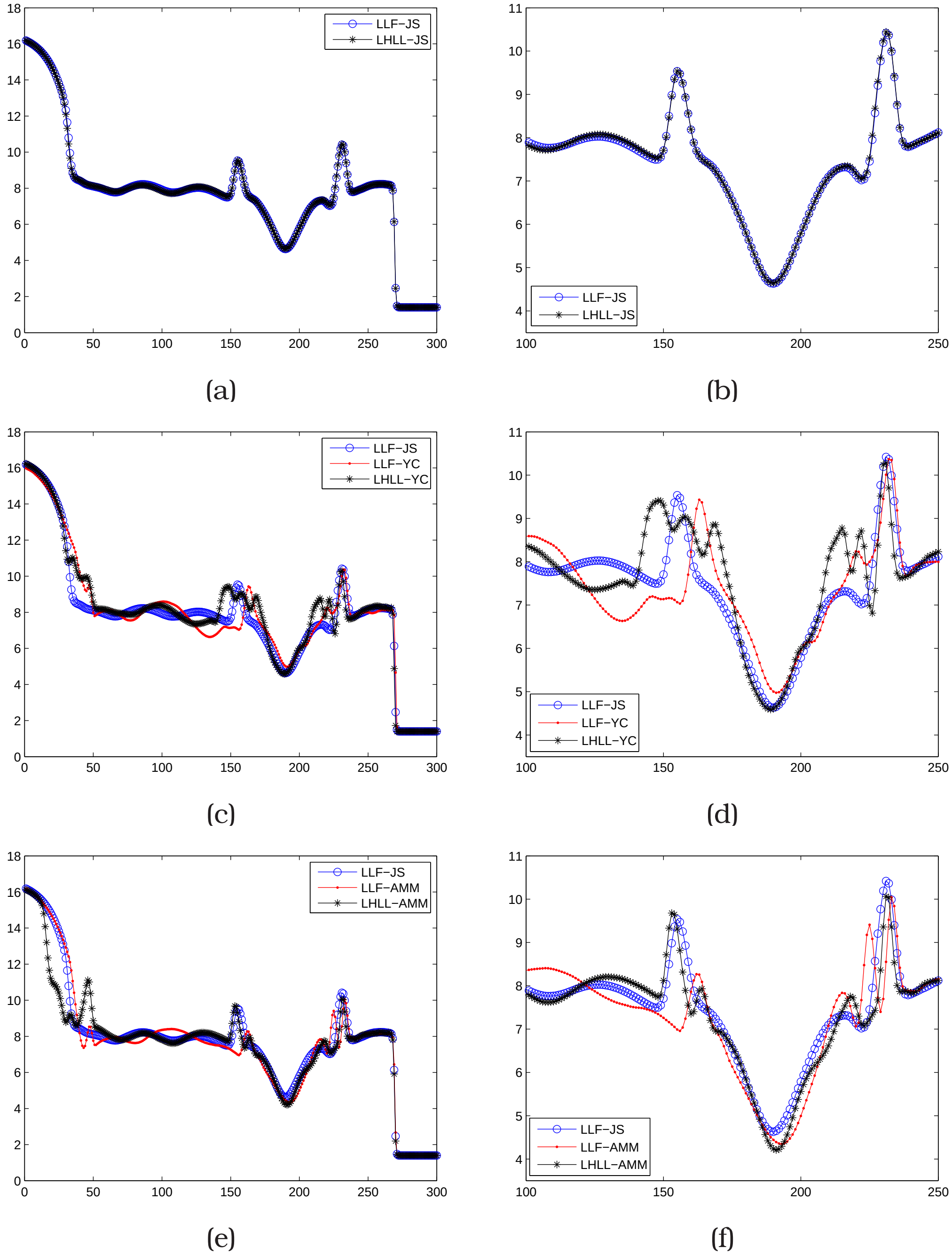


(LHLL-JS)

(LHLL-YC)

(LHLL-AMM)

Figure 5.19: *Results of test 5.5 for a grid of* 2048×512 *cells. We show 50 contour lines of the density obtained with a local LF and HLL flux-splitting and all the different definitions for the WENO weights studied in this work.*



(a)

(b)

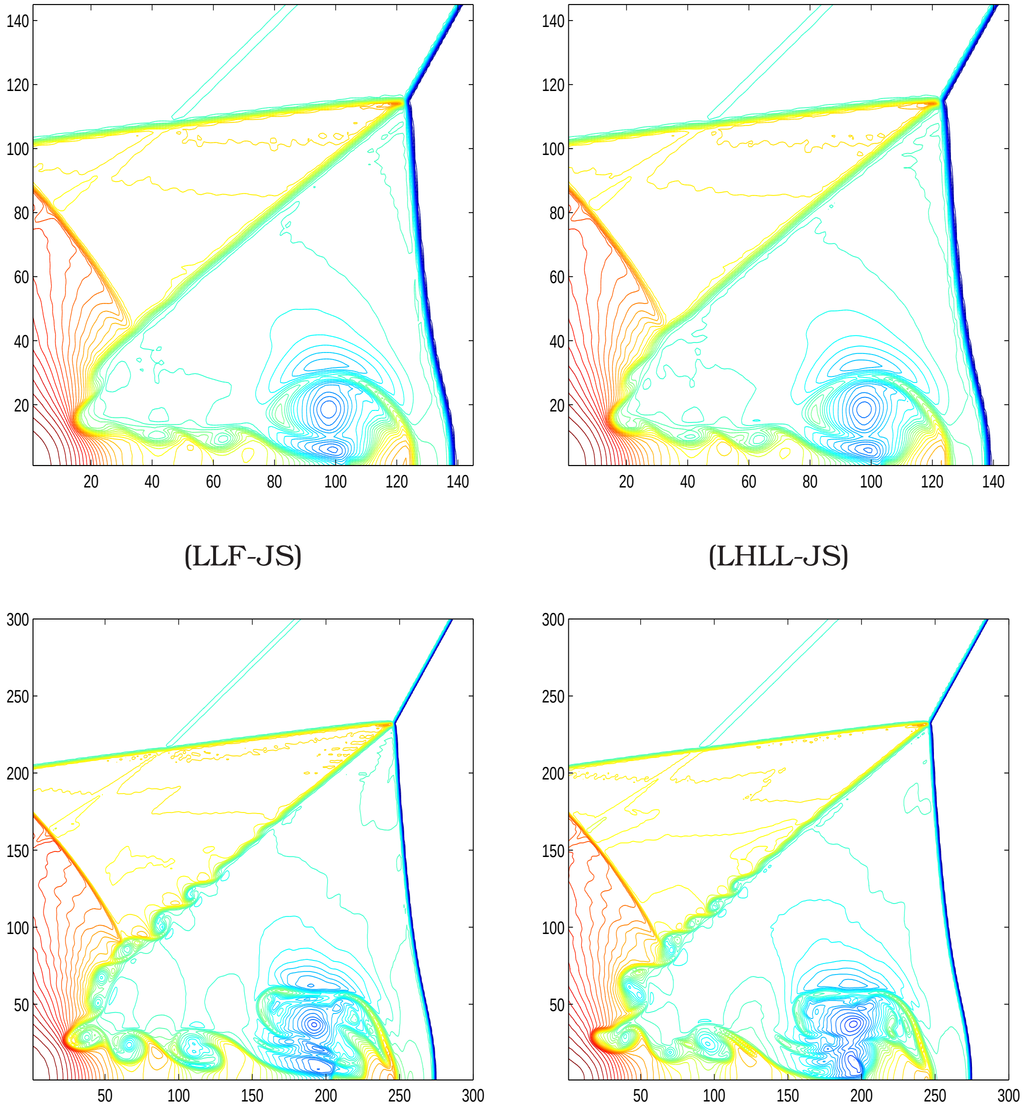
(c)

(d)

(e)

(f)

Figure 5.20: *Results of test 5.5 for a grid of* 2048×512 *cells. (a), (c) and (e) display sections of the* 50 *contour lines of the density obtained with a local LF and HLL flux-splitting and all the different definitions for the WENO weights studied in this work, at pixel height* 36*. (b), (d) and (f) are zooms of (a), (c) and (e) respectively.*



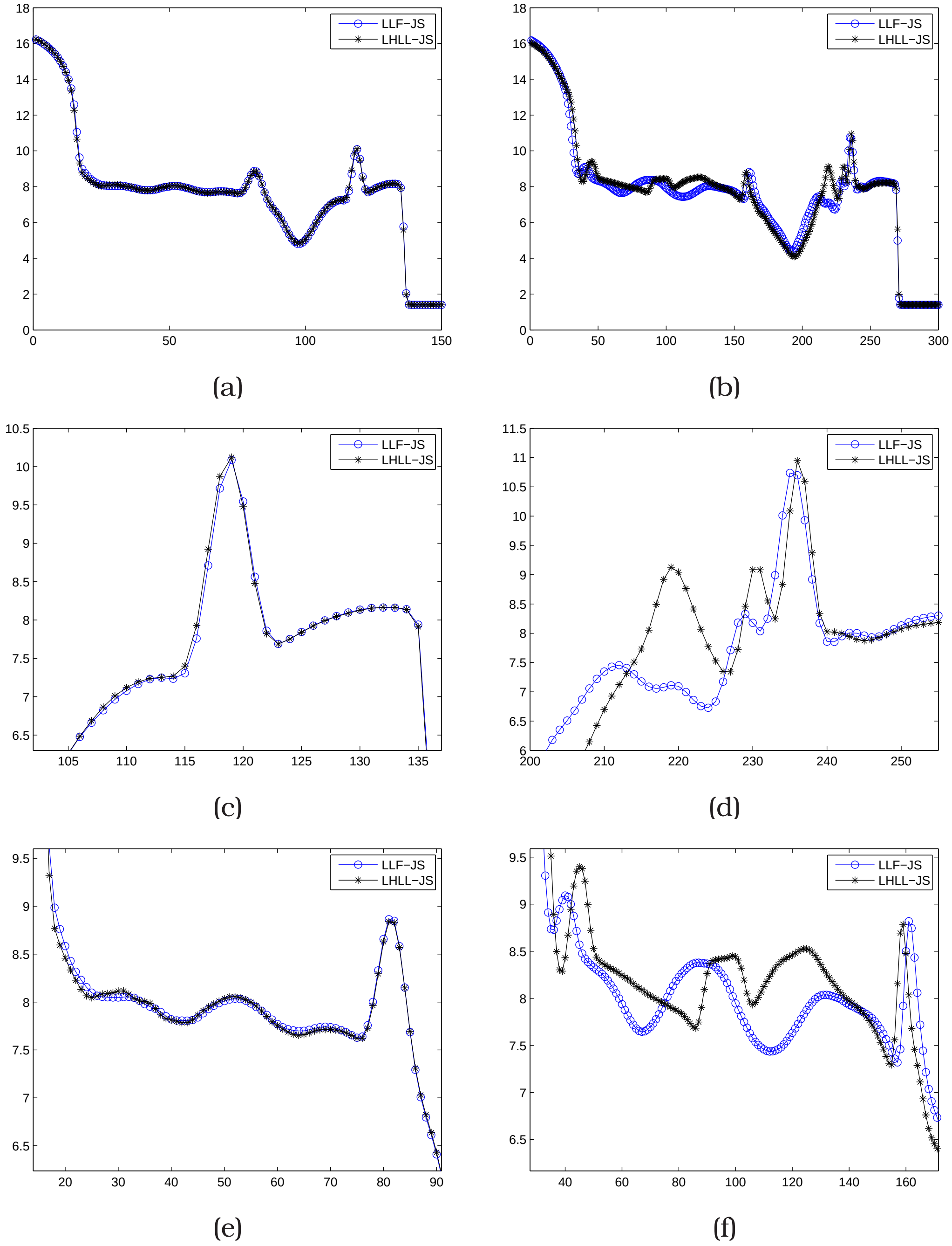
(LLF-JS)

(LHLL-JS)

(LLF-JS)

(LHLL-JS)

Figure 5.21: *Results of test 5.5 for a grid of* 1024×256 *((a) and (b)) and* 2048×512 *((c) and (d)) cells. We show 50 contour lines of the density obtained with a local LF and HLL flux-splitting and Jiang and Shu's WENO9 reconstruction scheme with* $\varepsilon = h^8$*.*



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.22: *Results of test 5.5 for a grid of* 2048×512 *cells. (a) and (b) display sections of the* 50 *contour lines of the density obtained with a local LF and HLL flux-splitting and Jiang and Shu's WENO9 reconstruction scheme, at pixel height* 18 *for a grid of* 1024×256 *(a), and at pixel height* 36 *for a grid of* 2048×512 *(b). (c), (d), (e) and (f) are zooms of (a) and (b) respectively.*

6

Well-Balanced Adaptive Mesh Refinement for shallow water flows

6.1 Introduction

The shallow water equations are a non-linear, hyperbolic, system of balance laws, which are obtained from the Navier-Stokes equations by depth averaging, after neglecting effects such as turbulence or shear stress. This system is widely used in many applications to model flows in river and coastal areas, and has received a lot of attention in the scientific community during the last ten to fifteen years. There has been a tremendous research effort towards the development of numerical techniques for the shallow water equations. This effort is due in part to the many modeling applications of shallow water flows, but also due to the specific difficulties in the numerical simulation of this system that make the problem attractive and challenging.

On a flat bathymetry, the shallow water equations become a homogeneous system of conservation laws. Their solutions may develop discontinuities, even when the initial flow is smooth, which requires the use of shock-capturing schemes in order to ensure a proper handling of discontinuities in numerical simulations concerning this system of equations.

The presence of a non-flat bathymetry leads to the inclusion of source terms in the system related to the bottom geometry. It is well-known that naive discretizations of these source terms may lead to spurious, numerical, oscillations that can obscure, or even ruin, the real solution that needs to be computed. This spurious numerical behavior occurs when computing stationary, or nearly stationary, solutions, for which the balance between the convective fluxes and the source terms associated to the bathymetry is not respected by the numerical scheme. Well-balanced schemes [17, 47] are specifically designed in order to maintain this balance, to machine accuracy if possible, and Well-balanced Shock-Capturing (WBSC) schemes constitute the state of the art in the numerical simulation of shallow water flows.

Robust and accurate WBSC schemes often have a high computational cost, which is related to the incorporation of upwinding through characteristic information that needs to be computed at each cell boundary in the computational domain, high-order reconstruction procedures, and a sophisticated numerical treatment of the bathymetry source term. In situations of practical interest, it is highly desirable to combine a WBSC scheme with an adaptive technique that can lower its high computational cost in 2D simulations [15, 42, 57, 62, 76].

The efficiency of an AMR algorithm is related to the reliability of the mesh adaption procedure, which is usually controlled by user-dependent thresholding parameters. Good efficiency factors are obtained when the thresholding parameter is relatively large, however, the use of an 'efficient' thresholding parameter might lead to spurious numerical behavior, akin to that observed when a non-Well-Balanced (NWB) numerical scheme is used on a uniform mesh, when computing stationary or nearly stationary solutions to the shallow water model, even if the underlying solver is a WBSC scheme.

In this chapter we analyze a block structured AMR technique developed in [9] and briefly recall the underlying WBSC scheme used by the block structured AMR technique, identifying those which are potentially responsible of the WB loss. We point out that, in addition to using a

WBSC scheme as the underlying scheme in the AMR process, it is necessary to implement Well-Balanced interpolatory techniques in the transfer operators involved in the multi-level grid structure in order for the combined AMR-WBSC scheme to maintain its well-balanced character. In section 6.4 we describe the necessary corrections to obtain a WB-AMR code and, in section 6.5, we show several numerical experiments that support our discussion.

6.2 Well-balanced schemes for shallow water flows

The shallow water system (2.24) admits stationary solutions, in which non-zero flux-gradients are exactly balanced by the source terms. Such solutions, along with their perturbations, are difficult to capture numerically because straightforward discretizations of the source term fail to preserve this balance. Computing these solutions is indeed a challenge and there is a large body of recent research concerning numerical techniques that incorporate the necessary balance in their discrete design (e.g. [20, 27, 40, 42, 80, 89, 103]). Such schemes are termed well-balanced (WB) schemes after the work of Leroux and collaborators [47, 48]. Bermúdez and Vázquez-Cendón, in an independent work [17], introduced the concept of the C*-property*. A scheme is said to satisfy the *exact* C-property if it preserves exactly the 'water at rest' stationary solution. Schemes that satisfy the exact C-property are WB for quiescent steady states.

All WBSC schemes preserve exactly the 'water at rest' stationary solution, for which $v^x = v^y = 0$ and $h + z = C$ (constant). However, as we shall see later on, the 'water at rest' might not be exactly preserved if the same scheme is used in a multi-scale framework. Our goal in this chapter is to address the issue of *well-balancing* when a WBSC scheme is used as the underlying solver within a block-structured AMR technique.

The numerical experiments in this chapter are carried out using a WBSC scheme developed in [34, 80], which preserves exactly the water at rest steady state. For the sake of completeness, we give next a brief description of the scheme for the simpler 1D shallow water model, which takes the form ($v = v^x$):

$$\begin{cases} h_t + (hv)_x = 0 \\ (hv)_t + \left(hv^2 + \dfrac{gh^2}{2}\right)_x = -ghz_x. \end{cases} \tag{6.1}$$

If we use the notation:

$$u = \begin{bmatrix} h & hv \end{bmatrix}^T, \quad f(u) = \begin{bmatrix} hv & hv^2 + \frac{gh^2}{2} \end{bmatrix}^T, \quad s(x,u) = \begin{bmatrix} 0 & -ghz_x \end{bmatrix}^T,$$

system (6.1) can be written as:

$$u_t + f(u)_x = s(x,u)$$

which, in turn, can be rewritten in the *homogeneous* form:

$$u_t + g[u]_x = 0,$$

where the functional g (dependent on f and s) acts on $u = u(x,t)$ as:

$$g[u](x,t) = f(u(x,t)) - \int_{x_0}^{x} s(r, u(r,t))\, dr.$$

Here x_0 is a reference point in the computational domain, e.g. $x_0 = 0$ when the latter is $[0,1]$. This reformulation, first proposed in [41], allows a 'unified treatment' of the flux and the source terms, so that upwind numerical methods for non-homogeneous conservation laws can be derived from well-established techniques for homogeneous conservation laws [26, 34, 41, 80].

In [34, 80] the authors proposed a Lax-Wendroff-type finite-differences discretization for $u_t + g[u]_x = 0$, which is hybridized with a first-order monotone scheme through flux-limiting techniques. The scheme applied to the exact solution $u(x,t)$ can be expressed as follows:

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}(\mathcal{G}^n_{i+\frac{1}{2}} - \mathcal{G}^n_{i-\frac{1}{2}}) \tag{6.2}$$

where $\mathcal{G}_{i+\frac{1}{2}}$ are hybrid numerical fluxes for $g[u]$. The scheme follows the finite-difference framework, so that its design makes use of the quantities

$$g_i^n := g[u](x_i, t_n) = f(u(x_i, t_n)) - \int_{x_0}^{x_i} s(r, u(r, t_n))dr.$$

It is shown in [34, 80] that the flux difference $\mathcal{G}^n_{i+\frac{1}{2}} - \mathcal{G}^n_{i-\frac{1}{2}}$ in (6.2) can be written as a sum of terms which contain the quantities $\Delta g^n_{i\pm\frac{1}{2}}$

$$\Delta g^n_{i+\frac{1}{2}} := g^n_{i+1} - g^n_i = f(u(x_{i+1}, t_n)) - f(u(x_i, t_n)) + b^n_{i,i+1},$$

where

$$b^n_{i,i+1} = -\int_{x_i}^{x_{i+1}} s(r, u(r, t_n))dr. \tag{6.3}$$

Hence, to get a fully discrete numerical method one needs to approximate the integral in (6.3) by some appropriate quadrature rule, which provides an approximation $\hat{b}^n_{i,i+1} \approx b^n_{i,i+1}$. Then,

$$\widehat{\Delta g^n_{i+\frac{1}{2}}} := f(u^n_{i+1}) - f(u^n_i) + \hat{b}^n_{i,i+1}$$

approximates $\Delta g^n_{i+\frac{1}{2}}$.

As observed in [34, 80], exact preservation of a stationary solution is obtained if the approximation $\hat{b}^n_{i,i+1} \approx b^n_{i,i+1}$ is exact for that solution. In fact, if one takes a stationary solution u that satisfies $f(u(x))_x = s(x, u(x))$ or, equivalently $g[u]_x = 0$, then $g^n_i = g[u](x_i, t_n)$ is constant $\forall i$.

If $\hat{b}^n_{i,i+1} = b^n_{i,i+1}$, this immediately gives that

$$\widehat{\Delta g^n_{i+\frac{1}{2}}} = \Delta g^n_{i+\frac{1}{2}} = g^n_{i+1} - g^n_i = 0, \ \forall i, n,$$

which implies that $u^{n+1}_i = u^n_i, \ \forall i, n$. Hence, the scheme preserves exactly the stationary solution $u(x)$ iff $\hat{b}^n_{i,i+1}(u(x)) = b^n_{i,i+1}(u(x))$.

For the shallow water equations, suitable $\hat{b}^n_{i,i+1}$ can be defined to get exact preservation of the water at rest stationary solution, via an appropriate definition of the integral in (6.3), see [7, 80]. The exactness of $\hat{b}^n_{i,i+1}$ relies heavily on the scheme being based on point-values. The resulting scheme follows the finite-difference framework and is formally second order accurate on smooth regions. The WB character of the scheme is a consequence of the hybridization procedure on $g[u]$ above, which leads to hybrid fluxes for the convective terms and a specific upwinding of the source terms compatible with it. More details on the scheme and its performance when applied to the shallow water equations can be found on the literature, e.g, [7, 34, 80].

6.3 Adaptivity: Block-structured AMR

The expected computational cost of explicit schemes for balance laws in d-dimensional simulations on uniform meshes is $\mathcal{O}(N^{d+1})$, with $N = 1/\Delta x$, and the storage requirements are $\mathcal{O}(N^d)$. The running time of a multidimensional simulation can, hence, be quite large for simulations on uniform meshes with Δx relatively small, which might be necessary in order to guarantee a certain, prescribed, accuracy in the simulation.

Because of the hyperbolic nature of the system of balance laws, numerical errors on uniform meshes are not uniformly distributed. Larger errors occur at discontinuities, whereas much smaller errors occur at smooth regions, hence adaptive schemes, that incorporate refinement only where higher errors occur, are appropriate, and often absolutely necessary, for multidimensional simulations and high precision needs. There are various approaches to achieve this goal [8, 10, 29, 82]. In this chapter we use the (block-structured) Adaptive Mesh Refinement framework, proposed in [16] for finite-volume schemes and extended by many authors (e.g. [10, 14, 87]), which we briefly review next.

Block-structured AMR algorithms compute the time evolution of a multi-scale representation of the solution, based on a hierarchical system of grids $G_0, \dots, G_L$. For simplicity of the exposition, we assume that the computational domain is $\Omega = [0, 1]^d$. The coarsest grid, G_0, is a uniform mesh, while at higher resolution levels, the computational cells are obtained from a uniform subdivision of some of the cells in the immediately coarser level. Specifically, assume that the coarsest grid is obtained by subdividing the unit interval in each dimension into N_0 subintervals, so that a coarse cell is given by

$$c_j^0 = \prod_{k=1}^{d} [j_k h_0, (j_k + 1)h_0], \quad j \in G_0 := \{1, \dots, N_0\}^d, \quad h_0 = \frac{1}{N_0}.$$

If each refinement level is obtained by bisecting each cell of the immediately coarser level, a cell at refinement level l is given by:

$$c_j^l = \prod_{k=1}^{d} [j_k h_l, (j_k + 1)h_l], \quad j \in G_l \subseteq \{1, \dots, N_l\}^d,$$

$$h_l = \frac{1}{N_l}, \quad N_l = 2^l N_0.$$

The *extent* of G_l, i.e. the union of the cells indexed by elements of G_l, is denoted by $\Omega_l(G_l)$:

$$\Omega_l(G_l) = \bigcup_{j \in G_l} c_j^l.$$

At the coarsest level, it is required that $\Omega_0(G_0) = \Omega$. At higher resolution levels, $\Omega_l(G_l)$ is formed by a set of disjoint uniform *patches* composed of cells at resolution l . Only *nested* grid hierarchies are considered, i.e., $\Omega_l(G_l) \subseteq \Omega_{l-1}(G_{l-1})$ for $1 \leq l \leq L$ is assumed to hold.

For the sake of illustration, we consider the 1D framework, with $\Omega = [0,1]$ as the computational domain. The coarsest mesh G_0 is given by a uniform partition of $[0,1]$, composed by N_0 subintervals of length $h_0 = 1/N_0$. A *mesh* G_l at resolution level l can be identified as a subset of the index set $\{0,\ldots,N_l\}$, where $N_l = 2^l N_0$. The cells at resolution level l are sub-intervals of length $h_0/2^l$. Figures 6.1 and 6.2 show samples of grid hierarchies that do and do not satisfy the nestedness requirement.

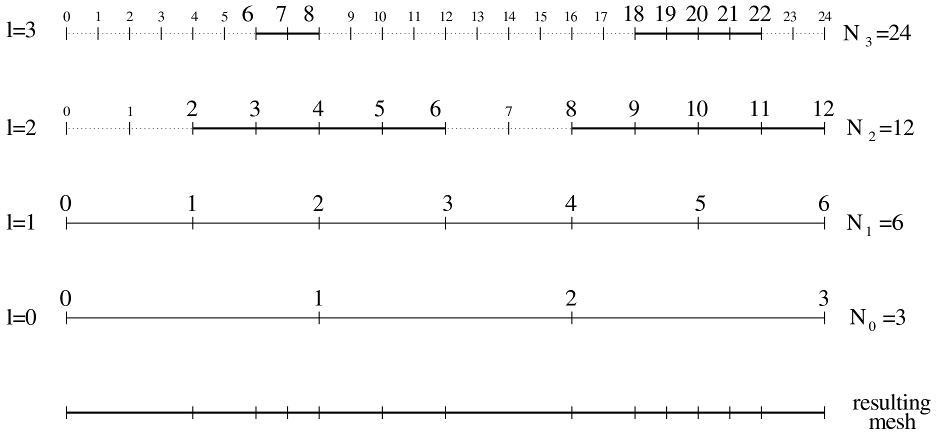


Figure 6.1: ***Nested mesh refinement.*** *$N_0 = 3$, $L = 3$. Every patch of cells is contained in a patch at the previous level.*

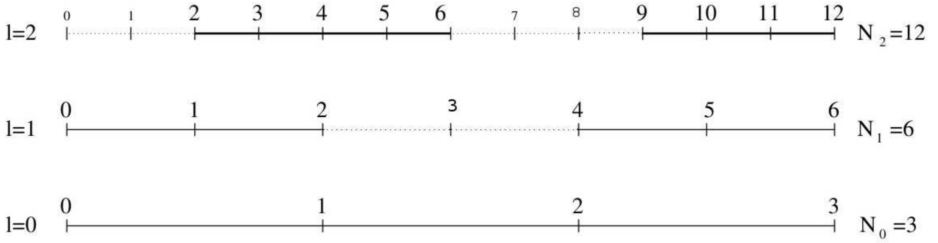


Figure 6.2: ***Non-nested mesh refinement.*** *$N_0 = 3$, $L = 2$. For $l = 2$ the first patch of cells is not contained in any patch at the previous level.*

At a given time, t, and resolution level, l, we have a multi-scale numerical solution $u_l^t = (u_{l,j}^t)_{j \in G_l^t}$, where G_l^t is the mesh at resolution level

l and time t and $u^t_{l,j}$ is the data attached to some point $x^l_j \in c^l_j$ (may be the center or an edge of the cell) at time t. The AMR algorithm specifies the time evolution of the multi-scale numerical solution and the associated hierarchical grid system. Each mesh on this system is dynamically updated so that the entire hierarchical structure adapts to the features of the associated multi-scale numerical solution at each step of a time-evolution procedure, from time $t = 0$ to time $t = T > 0$.

We briefly describe next the main building blocks of the AMR algorithm. We emphasize those aspects that are relevant for the analysis in this chapter. A more complete description of the algorithm can be found in [10].

6.3.1 Flow Integration

In order to advance the multi-scale solution from time t to time $t+\triangle t_0$, Δt_0 must be a suitable time step for the coarsest grid, so that the following CFL condition relevant for the grid G^t_0 is satisfied:

$$\Delta t_0 = \frac{C h_0}{\max_{u \in U^t} |f'(u)|}, \quad 0 < C \leq 1,$$

where $U^t = (u^t_{l,j})_{j \in G^t_l}, l = 0, \ldots, L$. As described in [10, 87], an adaptive time-stepping strategy must be used, in order to avoid unnecessary restrictions on the time step used on the coarsest grids (i.e. $C = O(1)$). Here, the corresponding time step for the evolution of patches in G_l is given by $\Delta t_l = \Delta t_{l-1}/2 = \Delta t_0/2^l$, which implies that the equivalent CFL condition holds automatically for G_l, but also that a time step for G_0 corresponds to 2^l time steps for G_l. The grids are integrated from coarse to fine in a sequential fashion, according to the order dictated by the following condition: $t_{l'} \leq t_l \leq t_{l'} + \Delta t_l$, if $l \leq l'$. For $L = 2$, the evolution sequence from $u^t = (u^t_0, u^t_1, u^t_2)$ to $u^{t+\Delta t_0} = (u_0^{t+\Delta t_0}, u_1^{t+2\Delta t_1}, u_2^{t+4\Delta t_l})$ would be computed in 5 steps, ordered as shown in Figure 6.3.

At resolution level l, G_l is composed by a set of uniform, disjoint patches, where a patch is

$$\prod_{k=1}^{d} \{m_k, m_{k+1}, \ldots, n_k\}.$$

Each patch at a given resolution level must have been surrounded by a sufficiently wide layer of *ghost cells* (2 cells in our code), which are given

$$\begin{array}{llllll}
\text{Step 1} & u_0^t & \rightarrow & u_0^{t+\Delta t_0} & & \\
\text{Step 2} & u_1^t & \rightarrow & u_1^{t+\Delta t_1} & & \\
\text{Step 3} & u_2^t & \rightarrow & u_2^{t+\Delta t_2} & \rightarrow & u_2^{t+2\Delta t_2} \\
\text{Step 4} & u_1^{t+\Delta t_1} & \rightarrow & u_1^{t+2\Delta t_1} & & \\
\text{Step 5} & u_2^{t+2\Delta t_2} & \rightarrow & u_2^{t+3\Delta t_2} & \rightarrow & u_2^{t+4\Delta t_2}
\end{array}$$

Figure 6.3: *Integration process for $L = 2$ from time t to time $t + \Delta t_0$.*

appropriate flow information prior to the application of the numerical scheme to the patch. Then, one step of the time evolution of a given patch at resolution level l can be done by a single call to the main solver, in this case a WBSC scheme.

For the integration from time t to $t + \Delta t_l$, the data at the ghost cells is obtained by *spatial interpolation* from (u_{l-1}^t, G_{l-1}). On the other hand, for the integration from time $t + \Delta t_l$ to $t + 2\Delta t_l$, the boundary data is obtained by applying first a linear interpolation in time from (u_{l-1}^t, G_{l-1}), $(u_{l-1}^{t+\Delta t_{l-1}}, G_{l-1})$, and then the usual spatial interpolation operator.

It should be noticed that this procedure leads to several numerical representations of the solution on areas covered by overlapping grids at consecutive resolution levels. In particular, since $2\Delta t_l = t_{l-1}$, once $(u_l^{t+2\Delta t_l}, G_l)$ is computed we have also some data coming from $(u_{l-1}^{t+\Delta t_{l-1}}, G_{l-1})$ filling the same region in space. At this point, a projection operator, to be discussed later on, needs to be applied in order to provide data that is consistent throughout the multiresolution hierarchy.

6.3.2 Adaptation

The grids corresponding to the various levels G_l, $1 \leq l \leq L$ have to be constructed according to the characteristics of the flow at the current time. The main goal of the process is to ensure that discontinuities that are initially covered by a grid at a given resolution level, continue being covered at the same resolution at later time, as long as the discontinuity persists. On the other hand, the refinement procedure should detect newly generated discontinuities as they are forming. The adaptation at each refinement level is performed by discarding the current grid and creating a new one according to specified refinement criteria. In this way, coarsening is not directly performed on refined areas, but implicitly obtained by not refining.

The refinement criteria are based on thresholding of interpolation errors and discrete gradients (see [8] for more details). A cell at level $l < L$, c_i^l, is selected for refinement if

$$\left|u_{l+1,j}^t - \mathcal{I}\left(u_l^t, x_j^{l+1}\right)\right| > \tau \cdot \max_{q<L,s} \left|u_{q+1,s}^t - \mathcal{I}\left(u_q^t, x_s^{q+1}\right)\right|,$$

for some $j \in G_{l+1}$ such that $j_k \in \{2i_k, 2i_k+1\}$, $k = 1, \ldots, d$ (i.e., $x_j^{l+1} \in c_i^l$), where the thresholding parameter on relative interpolation errors, denoted by τ in this work, is user/problem dependent.

Furthermore, we also include c_i^l, $l \leq L$, in the refinement list if the max-norm of the discrete gradient exceeds some large threshold (10 in our experiments), so that shock formation can be detected from steepened data. For the discrete gradient we use the approximation

$$\frac{\partial u}{\partial x_m}(x_i^l, t) \approx \frac{1}{h_l} \max\{|u_{l,i+e_m}^t - u_{l,i}^t|, |u_{l,i}^t - u_{l,i-e_m}^t|\},$$

where $e_{m,k} = \delta_{m,k}$.

Once a new grid-patch is constructed the solution on this patch is updated by copying from existing data, or by spatial interpolation from coarser grid data [10, 93].

6.3.3 Interpolation and projection

The transfer of information between grids is carried out by two operators: Interpolation, which is used in order to generate data at a given resolution level (ghost cell data prior to integration and new data, after refinement takes place) and Projection, which is used in order to enforce consistency between data at different resolution levels. The definition of the projection operator is related to the multi-scale framework used, which we briefly recall next in the 1D case.

The cell-average setting

We may consider the data to be attached to the points $x_j^l = (j+1/2)h_l$. Since $(x_{2j}^l + x_{2j+1}^l)/2 = x_j^{l-1}$, this corresponds to the so-called cell-average multiresolution setting (see [29, 50] and references therein). This setting is used when the underlying shock-capturing scheme is a finite-volume scheme, since the numerical solution is, then, naturally associated to the cell-averages of the true solution. For any L^1 function, $u(x)$, the relation between its cell-averages at consecutive resolution levels also satisfies $(u_{l,2j} + u_{l,2j+1})/2 = u_{l-1,j}$.

The canonical definition of the projection operator in the (1D) cell-average setting is as follows (see e.g. [50]): for each j such that $2j \in G_l$ we recompute

$$u_{l-1,j}^{t+\triangle t_{l-1}} \quad \leftarrow \quad [P(u_l^{t+2\Delta t_l})]_j = \frac{u_{l,2j}^{t+2\Delta t_l} + u_{l,2j+1}^{t+2\Delta t_l}}{2} \qquad (6.4)$$

The point-value setting

On the other hand, we may consider instead the data attached to the points $x_j^l = jh_l$, which corresponds to the point-value setting [50], since now $x_{2j}^l = x_j^{l-1}$. This setting is linked to finite-difference schemes, like the so-called Shu-Osher numerical schemes for hyperbolic conservation laws, whose numerical solutions are naturally interpreted as approximations to the point-values of the true solution.

In the (1D) point-value framework projection is just given by copying [50],

$$u_{l-1,j}^{t+\triangle t_{l-1}} \quad \leftarrow \quad [P(u_l^{t+2\triangle t_l})]_j = u_{l,2j}^{t+2\Delta t_l},$$

since $x_j^{l-1} = x_{2j}^l$.

6.4 Well-balanced AMR

Our goal is to obtain an adaptive mesh refinement algorithm that preserves at least a class of stationary solutions. Based on the above description, it seems necessary to require that its components (single grid solver, but also interpolation and projection), should also preserve the selected steady states. We recall that in the adaptation step, new values of the numerical solution are created by interpolation from a lower resolution level. Obviously, if a steady state, such as the 'water at rest', is to be maintained, these new values should comply with the 'water at rest' conditions. Also, numerical values are produced by space and space-time interpolation at ghost-cells, and the new values produced should also comply with the steady state conditions which we seek to preserve.

In fact, as we shall see in section 6.5, if the interpolation and/or projection operator do not comply with this requirement, the AMR algorithm will not preserve stationary solutions in the same sense as the basic WBSC scheme, which is a fact that is never explicitly mentioned in [57, 62], where adaptive techniques, in combination with WBSC schemes, are also applied to shallow water models. A similar approach as the one proposed in this chapter may be found in [73] in the cell-average framework. Since the numerical oscillations induced by a non-WB interpolation /projection operator are only observed near stationary solutions, the need for this requirement may have been unnoticed, in particular if only moving water experiments were performed.

We examine next the necessary conditions to be imposed on the prediction and interpolation operators in order to enforce preservation of the 'water at rest' stationary solution. As usual, and for the sake of clarity, the description will be carried out in the 1D framework.

Let us assume that we use a WBSC scheme that preserves exactly at least the 'water at rest' steady state as the basic solver. Then, at each step of the time evolution for a given patch, we have that whenever $i, j \in G_l$

$$h^t_{l,j} + z_{l,j} = h^t_{l,i} + z_{l,i} = C \quad \rightarrow \quad h^{t+\Delta t_l}_{l,j} + z_{l,j} = h^{t+\Delta t_l}_{l,i} + z_{l,i}, \tag{6.5}$$

where $z^l = (z_{l,j})_{j \in G_l}$ is an appropriate discretization of the bathymetry at the l-th level of resolution.

The projection operator preserves well-balancing iff whenever $i, j \in G_{l-1}$

$$[P(h_l^{t+2\triangle t_l})]_j + z_{l-1,j} = [P(h_l^{t+2\triangle t_l})]_i + z_{l-1,i} \tag{6.6}$$

6.4.1 Well-balanced Projection in the cell-average setting

As mentioned previously, the canonical definition of the projection operator in the cell-average setting is as follows: for each j such that $2j \in G_l$ we recompute

$$u^{t+\triangle t_{l-1}}_{l-1,j} \quad \leftarrow \quad [P(u_l^{t+2\triangle t_l})]_j = \frac{u^{t+2\Delta t_l}_{l,2j} + u^{t+2\Delta t_l}_{l,2j+1}}{2} \tag{6.7}$$

Hence, dropping the time for the sake of simplicity, relation (6.6) becomes

$$\frac{1}{2}(h_{l,2j+1} + h_{l,2j}) + z_{l-1,j} = \frac{1}{2}(h_{l,2j+3} + h_{l,2j+2}) + z_{l-1,j+1}$$

which, taking into account (6.5) is equivalent to

$$\frac{1}{2}\big(z_{l,2j+2} + z_{l,2j+3} - (z_{l,2j} + z_{l,2j+1})\big) = z_{l-1,j+1} - z_{l-1,j}$$

hence, the prediction operator in the cell-average setting can only be well-balanced if the discretization of the bathymetry along the different resolution levels follows the cell-average framework, i.e.

$$z_{l-1,j} = \frac{1}{2}(z_{l,2j} + z_{l,2j+1})$$

Remark 1. *The cell-average projection (6.4) is not well-balanced if the discretization of the bathymetry at each resolution level is obtained in a point-value manner, i.e. considering $z_{l,j} = z(x_j^l)$ when $x_j^l = (j + 1/2)h_l$, unless very special (e.g. linear) z are considered.*

Remark 2. *We note that the projection operator (6.7) maintains conservation in the homogeneous case (no source terms). For homogeneous conservation laws, the use of a conservative scheme at each resolution level ensures that the values obtained immediately after the application of a single integration step satisfy*

$$\sum_{j\in G_l} u_j^{t_l+\Delta t_l} = \sum_{j\in G_l} u_j^{t_l}$$

If $u_{l-1,j}^{t_l} = \frac{u_{l,2j}^{t_l}+u_{l,2j+1}^{t_l}}{2}$, then this consistency is maintained after application of the projection step, i.e.

$$\sum_{j\in G_{l-1}} u_j^{t_{l-1}+\Delta t_{l-1}} = \sum_{j\in G_{l-1}} u_j^{t_{l-1}}$$

see [8].

6.4.2 Well-balanced Projection in the point-value setting

In the point-value framework, the projection operator is just given by copying

$$u_{l-1,j}^{t+2\Delta t_l} \quad \leftarrow \quad [P(u_l^{t+2\triangle t_l})]_j = u_{l,2j}^{t+2\Delta t_l}.$$

In this framework, (6.6) becomes

$$h_{l,2j} + z_{l-1,j} = h_{l,2j+2} + z_{l-1,j+1}$$

which, taking into account (6.5) is equivalent to

$$z_{l,2j+2} - z_{l,2j} = z_{l-1,j+1} - z_{l-1,j}$$

Hence, the prediction operator in the point-value setting is well-balanced if the discretization of the bathymetry along the different resolution levels follows the point-value framework, i.e.

$$z_{l-1,j} = z_{l,2j},$$

which is ensured when using the following assignment:

$$z_{l,j} = z(x_{l,j}).$$

Remark 3. *We note that, in the homogeneous case, discrete conservation on coarser grids cannot be ensured for this projection operator. On the other hand, in the AMR context no adverse effects have been observed when this projection operator has been implemented [93]. Our own experience for balance laws supports this evidence.*

6.4.3 Well-balanced interpolation

The interpolation operator in the AMR algorithm is constructed using piecewise polynomial interpolatory techniques. Linear interpolation is used for the generation of ghost-cells by space-time interpolation, but higher order polynomial pieces might be used for space interpolation. In any case, the interpolation operator within the AMR algorithm is always used in the following general context: Data at level $l-1$ is known, say u_{l-1}, and a piecewise polynomial function is constructed in order to generate new data by evaluation of a polynomial, specifically constructed to comply with the requirements of the multi-scale framework considered, i.e.

$$\mathcal{I}(u_{l-1}, x_k^l) = p_j(x_k^l)$$

where $p_j(x)$ is the polynomial piece corresponding to the $j-$th computational cell, which is the cell at level $l-1$ that contains x_k^l.

Let us consider, for example, the space interpolation used when filling data at a newly created patch, in the adaptation step of the AMR algorithm, and assume that a WBSC scheme, which maintains exactly the water at rest steady state, has been used to determine the solution at time t so that (dropping the t superscript for simplicity) the data available at resolution level $l-1$ satisfies

$$h_{l-1,i} + z_{l-1,i} = h_{l-1,j} + z_{l-1,j} = C, \qquad q_{l-1,j} = 0, \quad i,j \in G_{l-1},$$

and

$$h_{l,i} + z_{l,i} = h_{l,j} + z_{l,j} = C, \qquad q_{l,j} = 0, \quad i,j \in G_l.$$

To ensure that the water at rest conditions hold for the data generated through the interpolation process, we propose to apply the interpolatory technique on the data obtained from the *equilibrium variables* for the water at rest steady-state,

$$V(x,[h,q]) = [h+z(x),q]$$

For 'water at rest' solutions, $V_{l-1} = [h_{l-1} + z_{l-1}, q_{l-1}] = [C,0]$, hence any piecewise polynomial interpolatory technique that preserves constants will lead to

$$\mathcal{I}(V_{l-1}, x_j^l) = [C,0].$$

Then, the space interpolation is implemented as follows

$$\hat{u}_{l,j}^t = [h_{l,j}^t, q_{l,j}^t] = \begin{cases} \mathcal{I}(V_{l-1}^t, x_j^l) - [z_{l,j},0] & \text{if } j \in \hat{G}_l^t \setminus G_l^t, \\ u_{l,j}^t & \text{if } j \in G_l^t, \end{cases}$$

where $\hat{G}_l^t$ is the adapted grid resulting from G_l^t. This well-balanced interpolation is related to hydrostatic reconstruction [5] (see also [21, 37] for other recent approaches).

In order to preserve the 'water at rest' stationary solution, the interpolation operator involved in the transfer of information between levels should act on the so-called *equilibrium variables* for the 'water at rest' steady state, $V = [h+z,q]$. For the one-dimensional shallow water equations, a general stationary solution $u(x)$ for which $f(u)_x = s(x,u)$ is characterized by the equilibrium variables:

$$V(x,\begin{bmatrix}h, & q\end{bmatrix}) = \begin{bmatrix}\frac{(q/h)^2}{2} + g(h+z(x)), & q\end{bmatrix}.$$

In order to preserve general stationary solutions a similar technique could be employed, i.e. the interpolation procedure should be applied on the equilibrium variables for the steady state to be preserved.

If we suppose that $V(x,\cdot)$ is bijective onto some relevant range then we could define an interpolator that preserves equilibrium variables by:

$$\widetilde{\mathcal{I}}((u_i);x) = V(x,\cdot)^{-1}(\mathcal{I}((V_i);x)), \quad V_i = V(x_i,u_i).$$

For the shallow water system, $V(x,\cdot)$ is not injective in general. Therefore, to preserve the maximum number of stationary solutions it is necessary to decide what is the regime of the stationary solution to be found (subcritical, transcritical or supercritical) in order to choose the adequate branch of the inverse $V(x,\cdot)^{-1}$ in the definition of the interpolation procedure. There are several works addressing this problem. For example, in [20] Bouchut and Morales only preserve the subcritical stationary solutions while Castro et al. in [32] and Noelle et al. in [84] introduce different techniques to locate singularities in the solution and choose the adequate branch of the inverse in each case.

The Well-Balanced interpolatory technique can be made positivity preserving by considering instead

$$\begin{aligned}\widetilde{\mathcal{I}}((u_i);x) =& P\Big(V(x,\cdot)^{-1}\big(\mathcal{I}((V_i);x)\big)\Big), \quad V_i = V(x_i,u_i), u_i \in \mathbb{R}^2\\ P(\big[h,q\big]) =& \big[\max(h,0), \quad q\big]\,.\end{aligned}$$

Thus, the proposed space interpolation should be implemented as follows

$$\widehat{u}^t_{l,j} = \begin{cases}\widetilde{\mathcal{I}}(u^t_{l-1},x^l_j) & \text{if } j \in \widehat{G}^t_l \setminus G^t_l,\\ u^t_{l,j} & \text{if } j \in G^t_l.\end{cases}$$

6.5 Numerical results

In this section we perform a series of numerical tests that intend to show the effects of incorporating a WB interpolatory technique in the transfer operators. The results in this section are obtained with an AMR code based on the code used in [8, 10]. Here we use a point-value-based grid hierarchy, instead of the cell-based grid hierarchy used in [8, 10]. The interpolation operator used for the transfer of information between levels is cubic in our experiments. A non-WB scheme results if the interpolatory technique is applied directly on the variables (h,q). Neumann boundary conditions are used at the domain boundary.

In this section the gravity acceleration is set to 9.812 and the CFL number is set to 0.6 for the one-dimensional simulations and to 0.4 in the two-dimensional setting.

6.5.1 Stationary and Quasi Stationary Flows

We consider first the case of steady-state and quasi-steady-state flow. The following tests demonstrate that the use of Well-Balanced interpolation operators is essential in order to maintain the exact C-property in the numerical solution computed with the AMR code.

Test 6.1. *Water at rest over an irregular topography*

The following test case was proposed in a workshop on dam-break wave simulation [46]. The initial data are a non-smooth bottom topography, tabulated in [46] and shown in Figure 6.4, and the water at rest at a level of 12 m. The boundary conditions are a water level of 12 m and no discharge. This data is taken as in [26], Section 4.1.1.

In Figure 6.4, we show the water height at $T = 200$ obtained with the WB-AMR code with $N_0 = 50$, $L = 7$ (i.e. eight levels with $N_7 = 6400$), for a threshold parameter $\tau = 10^{-2}$. The bottom topography and the grid patches active at each resolution level at the time of the simulation are also shown. Table 6.1 confirms that the steady state solution is maintained up to machine precision.

On the other hand, if the WB interpolation is not implemented in the transfer operators of the AMR code, numerical errors do occur. The effects of a rough thresholding parameter, τ, can readily be appreciated in Figure 6.5. The results in Table 6.1 and Figure 6.5 show that the loss of the exact C-property when using a non-WB interpolation in the transfer operators is analogous to that observed when using a high-order non-WB scheme on a similar mesh.

interp type	WB	NWB	WB	NWB	WB	NWB
L=5	7.1e-15	6.8e-3	7.1e-15	3.9e-4	8.8e-15	8.8e-15
L=7	2.8e-13	9.4e-3	1.6e-14	5.3e-4	1.2e-14	4.9e-5
thresholding	τ =1e-2		τ =1e-3		τ =1e-4	

Table 6.1: *Steady state over rough topography test ($N_0 = 50$). Errors $||h+z-12||_\infty$. For $\tau = 10^{-4}$, the adaptive patches cover the entire computational domain for $L = 5$, but not for $L = 7$.*

Test 6.2. *Two-Dimensional Steady Flow*

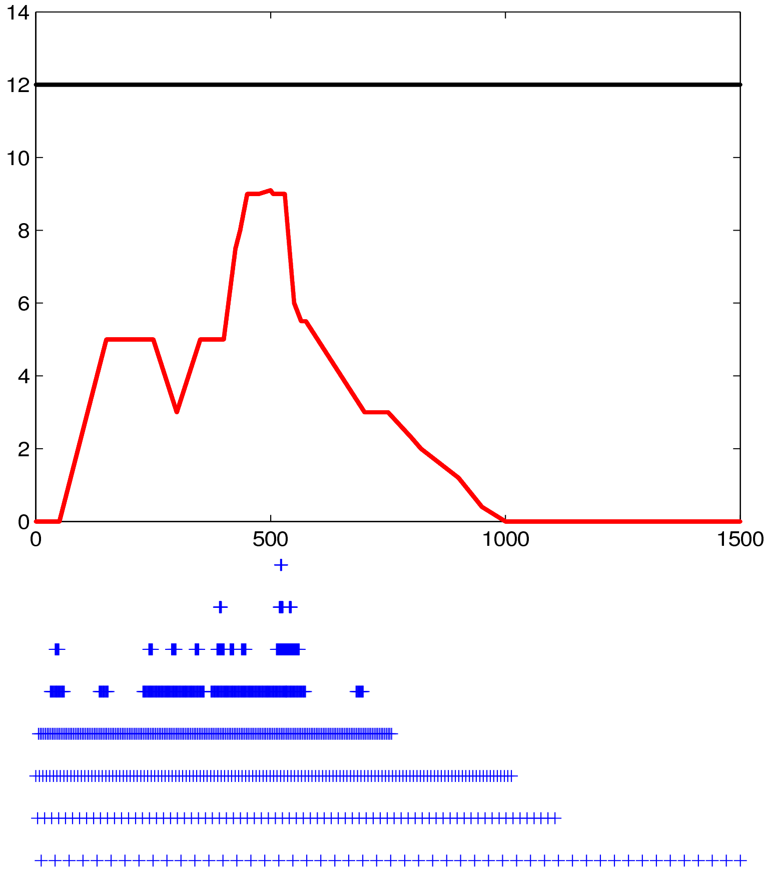
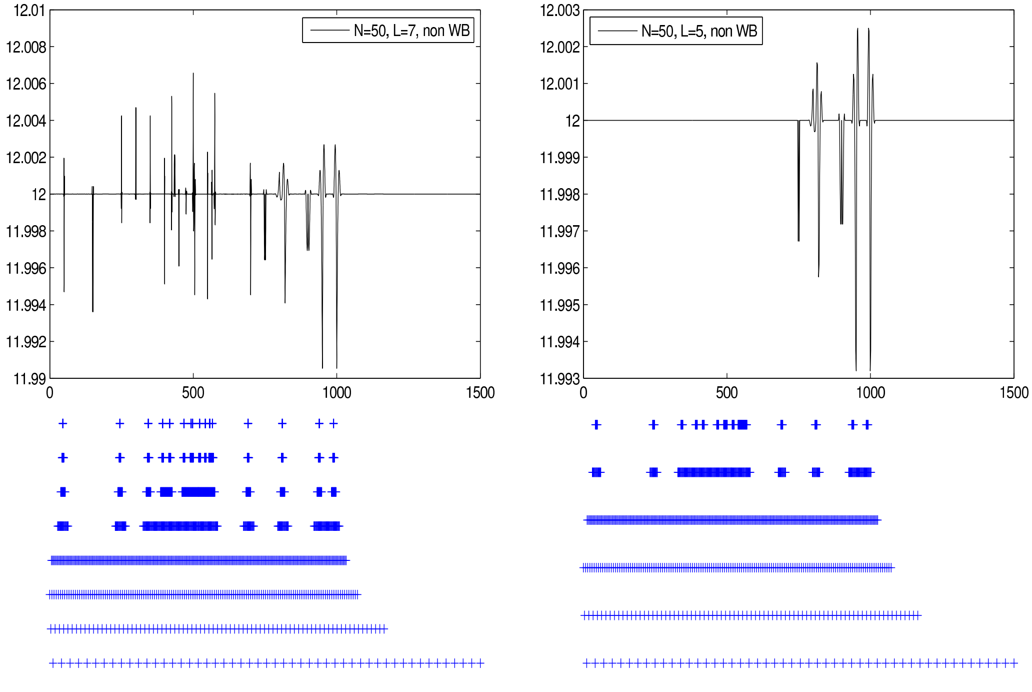


Figure 6.4: *Stationary flow over rough topography.* $T = 200$, $\tau = 10^{-2}$, $N_0 = 50$, $L = 7$, *(*$N_7 = 6400$*) with WB interpolation in the transfer operators.*



(a) (b)

Figure 6.5: *Same set-up and conditions as in Figure 6.4 with regular (non WB) interpolation in transfer operators of AMR code. (a)* $L = 7$ *and (b)* $L = 5$

To test the C-property in a 2D setting we consider a test proposed in [71].

The initial conditions correspond to water at rest with a total height of 1 and a smooth bottom topography displayed in Figure 6.6. The computational domain is the unit square and we have used $\tau = 10^{-1}$, $N_0 = 25$ and 4 levels ($L = 3$, $N_3 = 200$).

Table 6.2 shows the errors with respect to the steady state solution at $T = 1.0$. As in the previous example, the use of a non-WB interpolation in the transfer operators of the AMR code leads to the loss of the exact C-property. In Figures 6.6 and 6.7 we display the approximation to the 'water at rest' surface obtained using the AMR scheme with and without well-balanced interpolation respectively, in order to show the numerical oscillations present in the simulation without well-balanced interpolation.

interp type	WB	NWB	WB	NWB
$\|\|h+z-1\|\|_\infty$	6.38e-14	3.3e-2	6.04e-14	7.1e-3
$\|\|v^x\|\|_\infty$	2.0e-13	5.8e-2	1.4e-13	2.86e-2
$\|\|v^y\|\|_\infty$	3.0e-13	6.09e-2	2.3e-13	2.45e-2
thresholding	$\tau =$1e-1		$\tau =$1e-2	

Table 6.2: *2D-water at rest over smooth topography. Computational results at* $T = 1.0$. $N_{0,x} = N_{0,y} = 25$, $L = 3$.

Test 6.3. *Quasi Stationary Flow over smooth topography*

The following test, proposed by R. LeVeque in [71], has become a standard test for evaluating the capability of a numerical scheme to accurately compute small perturbations of 'water at rest' flows over non-flat topographies. The (smooth) bottom topography is given by the following function,

$$z(x) = \begin{cases} 0.25(\cos(\pi(x-0.5)/0.1)+1) & \text{if } |x-0.5| < 0.1, \\ 0 & \text{otherwise} \end{cases}$$

and the initial conditions are $q = 0$ and

$$h(x) = \begin{cases} 1 - z(x) + \varepsilon & \text{if } 0.1 < x < 0.2, \\ 1 - z(x) & \text{otherwise.} \end{cases}$$

Here, we shall consider $\varepsilon = 10^{-3}$.

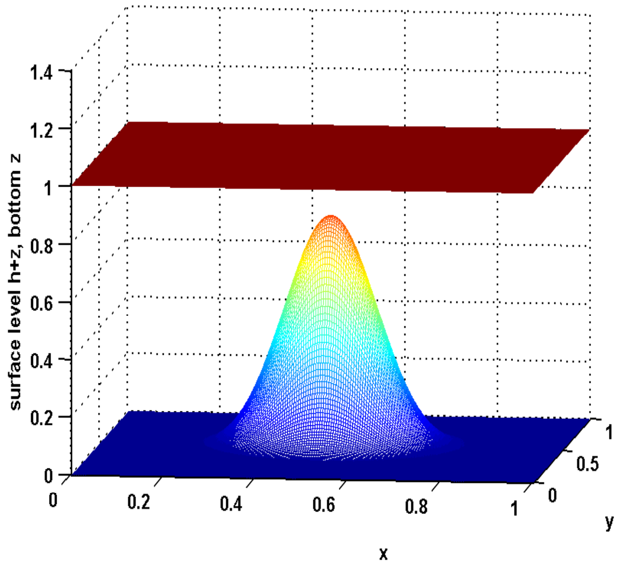


Figure 6.6: *Bottom topography and computed water-surface using the AMR scheme with WB interpolation in 2D 'water at rest' test.*

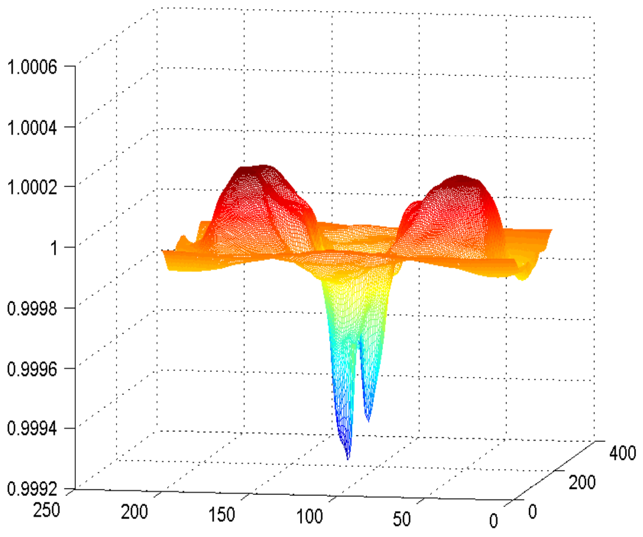
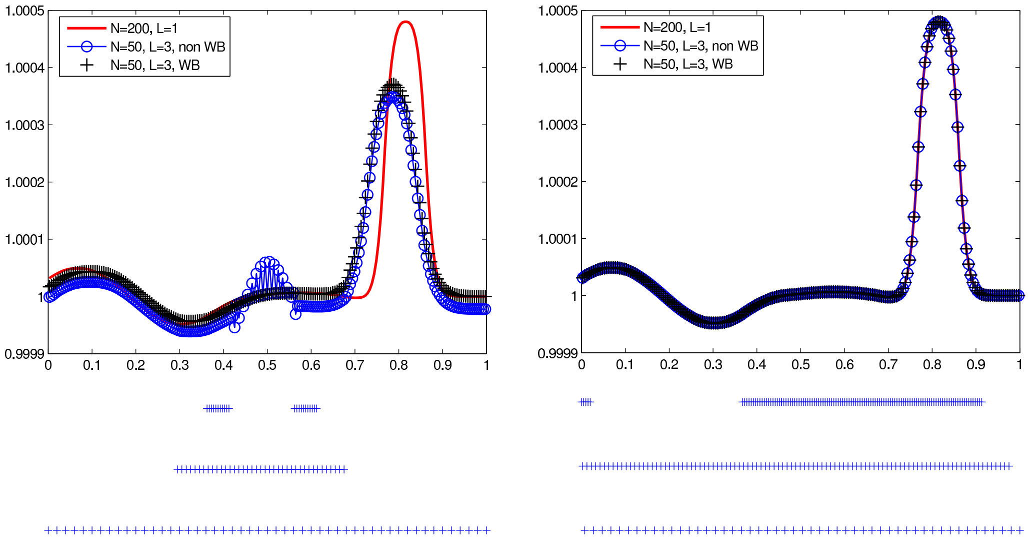


Figure 6.7: *Computed water-surface using the AMR scheme without WB interpolation.*

We consider first $N_0 = 50$ and three levels ($L = 2$, $N_2 = 200$), which is a rather coarse final mesh. In addition, we consider $\tau = 10^{-1}$, which is a refinement threshold much larger than the initial perturbation, and $\tau = 10^{-3}$, i.e. a threshold of the order of the perturbation itself. The numerical results at $T = 0.224$ are shown in Figure 6.8.

The simulation shows that the WB-AMR code produces an approximate solution that allows for a reasonable representation of the evolution of the initial perturbation when $\tau = 10^{-1}$. On the other hand, the numerical approximation obtained with the NWB-AMR code for this large value of the threshold parameter displays a numerical perturbation similar to those obtained with non-WB schemes. For $\tau = 10^{-3}$, the difference between the numerical solutions computed with the WB-AMR and NWB-AMR codes is well below 10^{-4}.



(a)

(b)

Figure 6.8: *Quasi Stationary Flow over smooth topography with $N_0 = 50$, $L = 2$ ($N_2 = 200$) and (a) $\tau = 10^{-1}$, (b) $\tau = 10^{-3}$. The grid hierarchies of the WB and non WB data are very similar, so, for simplicity, we have displayed their merging.*

Test 6.4. *Quasi Stationary Flow over rough topography*
With the same bottom topography as in Test 6.1, we consider now a

slight perturbation of the steady state $h + z = 12$, as follows

$$\eta(x) = h(x) + z(x) = \begin{cases} 12.01 & x \in [680, 720] \\ 12 & \text{otherwise} \end{cases}$$

and $v(x) \equiv 0$. For this simulation, we use $N_0 = 50$, $L = 5$, so that $N_5 = 800$, and $\tau = 10^{-2}$. In Figures 6.9 and 6.10 we compare the approximated solutions obtained with $N_0 = 800$, $L = 1$ (single-grid solution) with those obtained with the AMR code with and without WB interpolations in the transfer operators.

Again, the lack of WB interpolation in the transfer operators leads to oscillations, that are of the same order as the moving perturbations, hence displaying the typical behavior of a non-WB approximation.

6.5.2 Rapidly varying flow in 1D and 2D

The well-balancing of the transfer operators is not a crucial issue when computing numerical solutions of rapidly moving shallow water flow. This might explain why the issue of WB interpolation in the inter-level transfer operators has not been discussed in previous works. The following tests illustrate the performance of the AMR technique for rapidly moving flows. In these cases, there are no significant differences between the WB-AMR and non-WB-AMR results.

Test 6.5. *Dam break over a square bump bottom topography*

This test involves a rapidly varying flow over a discontinuous bottom topography, see e.g. [26] for details. The initial conditions are $q = 0$ and

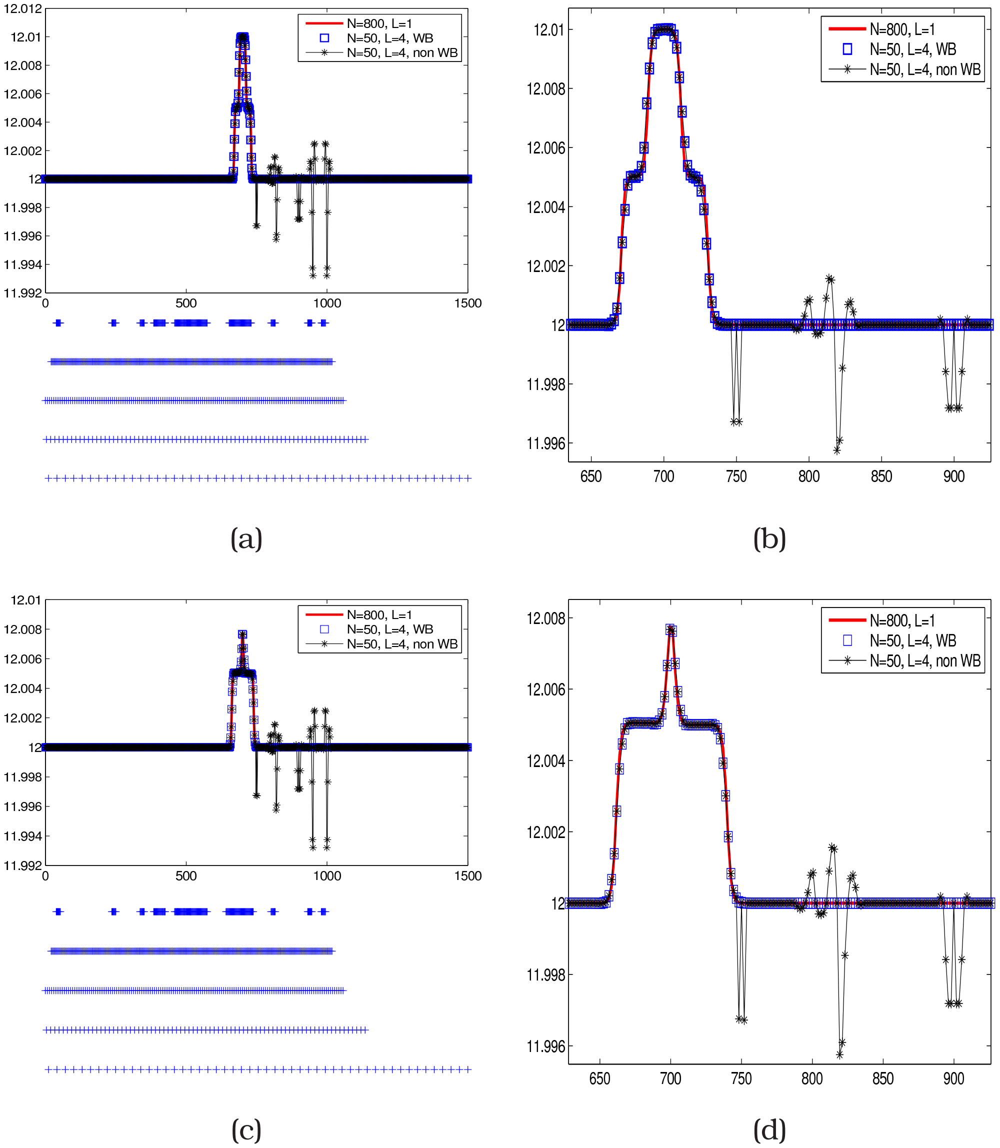
$$h(x) = \begin{cases} 20 - z(x), & x \leq 750 \\ 15 - z(x), & \text{otherwise} \end{cases}$$

The bottom topography is given as

$$z(x) = \begin{cases} 8, & |x - 750| \leq 1500/8 \\ 0, & \text{elsewhere}, \end{cases}$$

where $0 \leq x \leq 1500$.

In Figures 6.11 and 6.12 we display the computed water level at $T = 15$ and $T = 60$, together with the bottom topography. Figure 6.11 shows



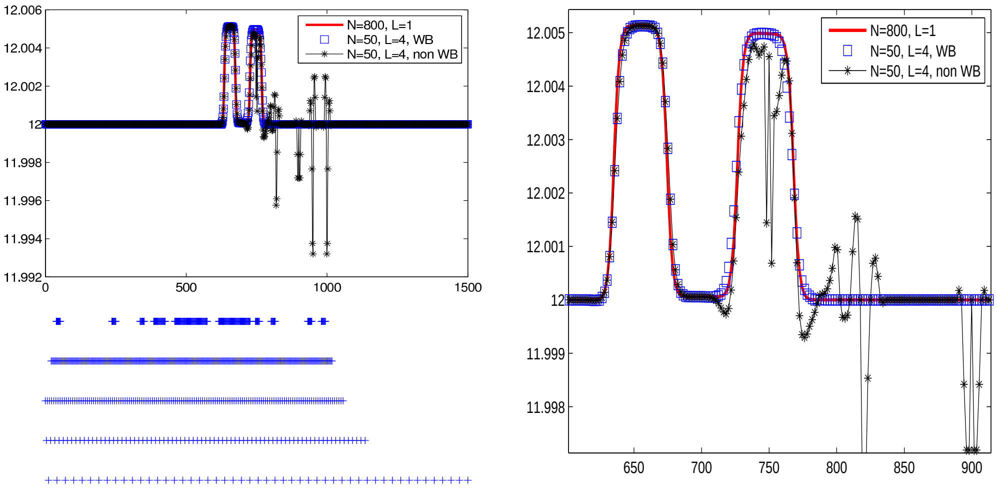
(a)

(b)

(c)

(d)

Figure 6.9: *Temporal evolution of the perturbation of the steady state in Figure 6.4.* $N_0 = 50$, $L = 4$, *(*$N_4 = 800$*)*, $\tau = 10^{-2}$, $T = 1$ *(a) and* $T = 2$ *(c). (b) and (d) are enlarged views of (a) and (c) respectively. The grid hierarchies of the WB and non WB data are very similar, so, for simplicity, we have displayed their merging.*



(a) (b)

Figure 6.10: *Temporal evolution of the perturbation of the steady state in Figure 6.4.* $N_0 = 50$, $L = 4$, *(*$N_4 = 800$*)*, $\tau = 10^{-2}$ *and* $T = 5$. *(b) is an enlarged view of (a). The grid hierarchies of the WB and non WB data are very similar, so, for simplicity, we have displayed their merging.*

also the grid patches active (at that time) at each resolution level. For this test, we have used $\tau = 10^{-2}$, $N_0 = 50$, and eight levels ($L = 7$, $N_7 = 6400$).

We readily observe that the AMR technique is able to identify correctly the discontinuities in the flow variables. In Table 6.3, we display the difference between the AMR solution and a reference solution computed by the single-grid algorithm on a very fine mesh. As expected, the error is lower than the chosen tolerance. The CPU speedup of the AMR computation is ≈ 17.36.

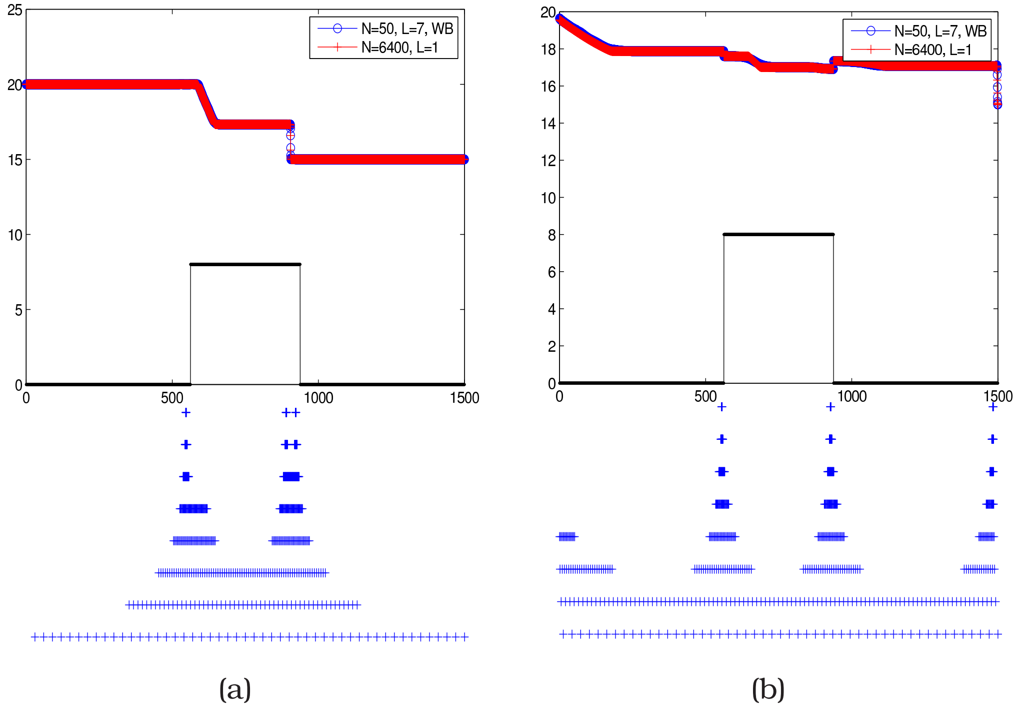


Figure 6.11: *Dam Break over a discontinuous topography with $N_0 = 50$, $L = 7$, $\tau = 10^{-2}$ and (a) $T = 15$ and (b) $T = 60s$. We display the water surface and multilevel grids structure.*

Test 6.6. *Circular Dam-Break Problem*

This test, proposed in [27], simulates a circular dam break problem over a non-flat topography. The domain is the square $[0, 2] \times [0, 2]$ with outflow boundary conditions.

In Figure 6.13 we display the numerical results for the WB-adaptive scheme at $T = 0.15$ and $T = 0.25$. In Figure 6.14 we show the multilevel

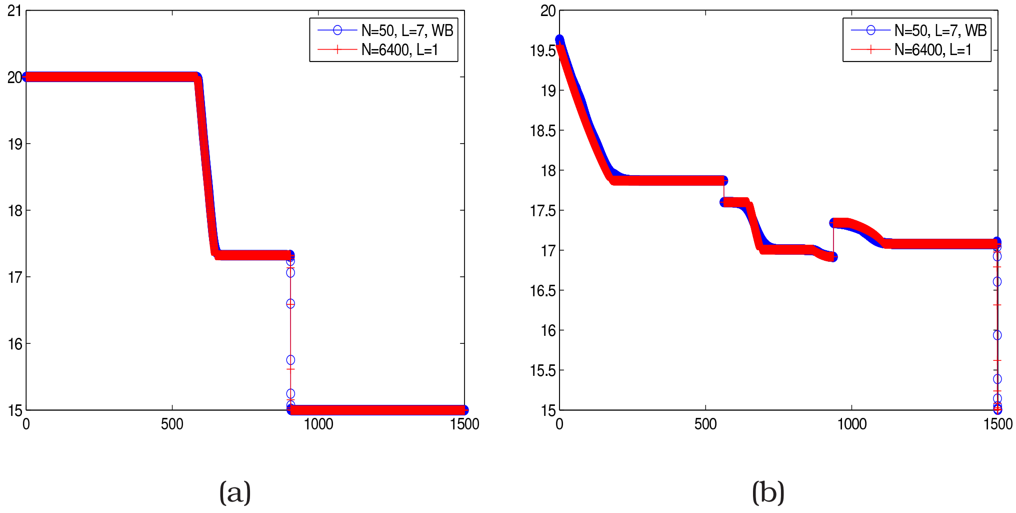
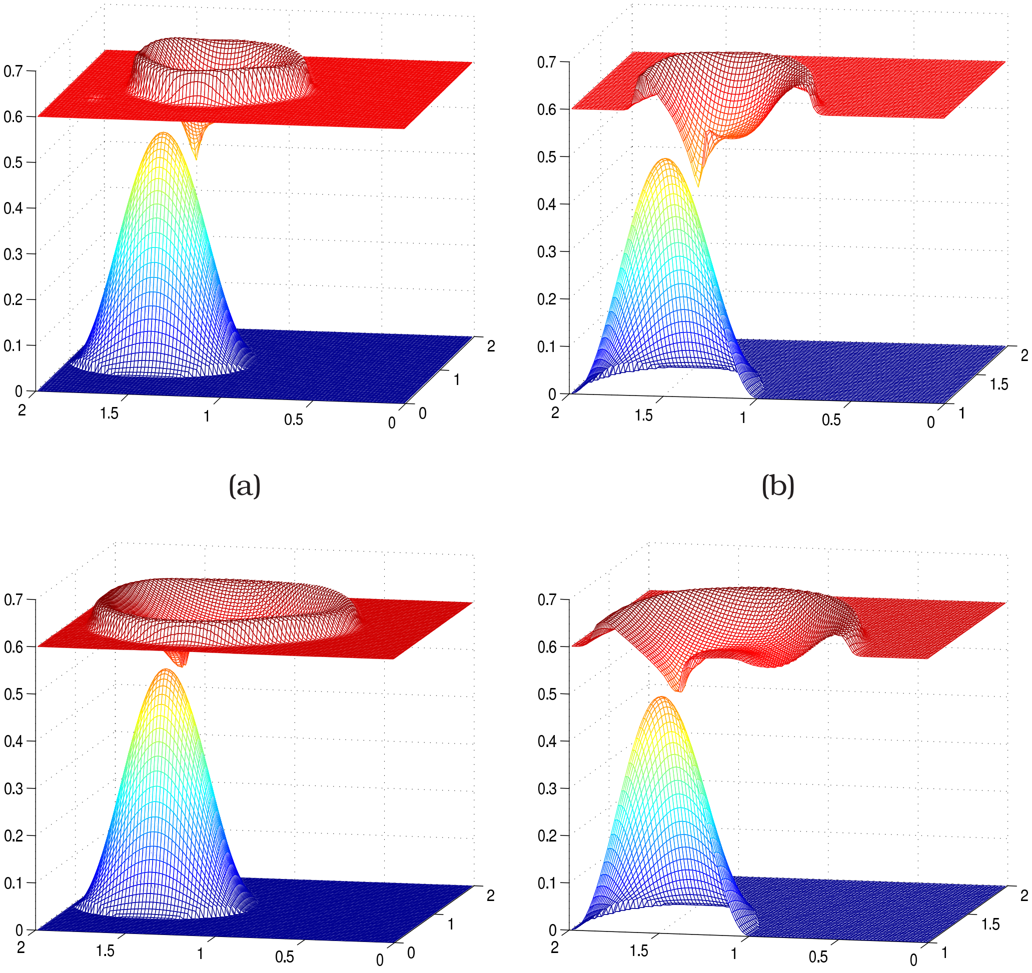


Figure 6.12: *Zoom of the approximations in Figure 6.11. (a)* $T = 15$ *and (b)* $T = 60s$*.*

interp type	WB	NWB
$\|\|h - h_{fixed}\|\|_1$	2.77e-3	2.77e-3
$\|\|v - v_{fixed}\|\|_1$	2.74e-3	2.74e-3

Table 6.3: *Dam break errors. In this case* (h_{fixed}, v_{fixed}) *is a reference solution computed with* $N = 12800$ *points.*

grid structure for a simulation with $L = 4$ and in Figure 6.15 a longitudinal section at $y = 1$ of h and a longitudinal section at $y = 1$ of $q_1 = u_x h$ at time $T = 0.15$, which allow for a direct comparison with [27, 80] . The CPU speedup when using $L = 2$ (Figure 6.13) is 3.26 while using $L = 4$ (Figure 6.14) it is 9.57. As mentioned before, there is no noticeable difference between the solutions computed with the WB-AMR code and those obtained with the non-WB-AMR code for this test.



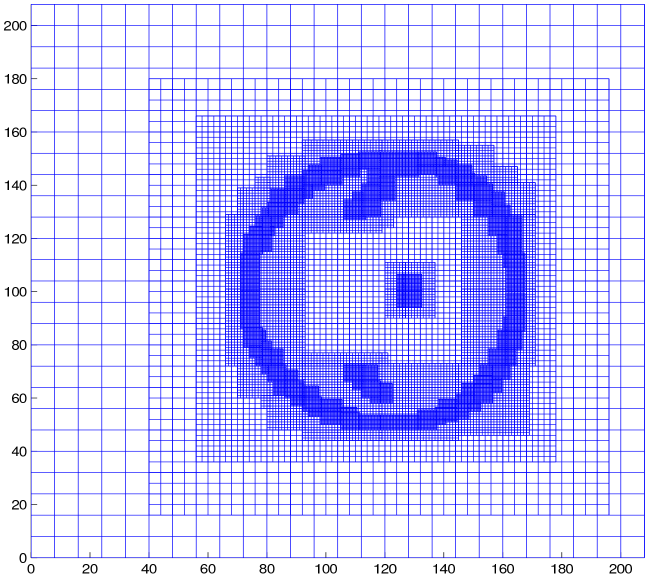
(a)

(b)

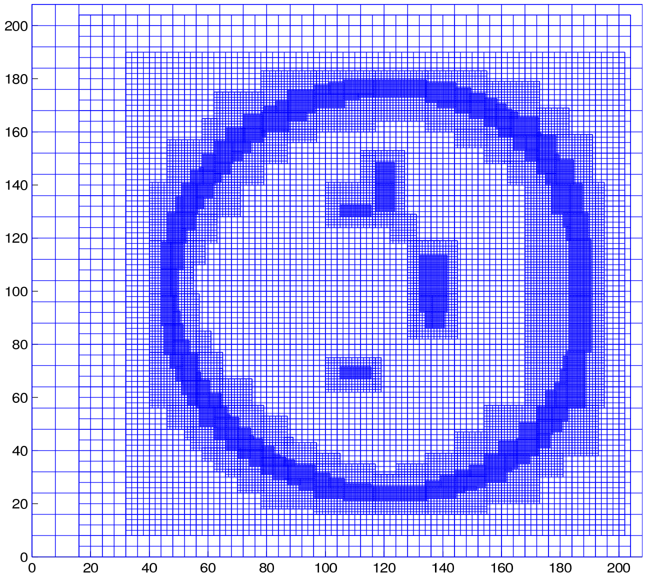
(c)

(d)

Figure 6.13: *Circular Dam-break problem with* $N_0 = 25$*,* $L = 2$*, (*$N_2 = 100$*),* $\tau = 10^{-1}$ *and (a)* $T = 0.15$ *(c)* $T = 0.25$*. (b) and (d) are slices of the channel at* $y = 1$

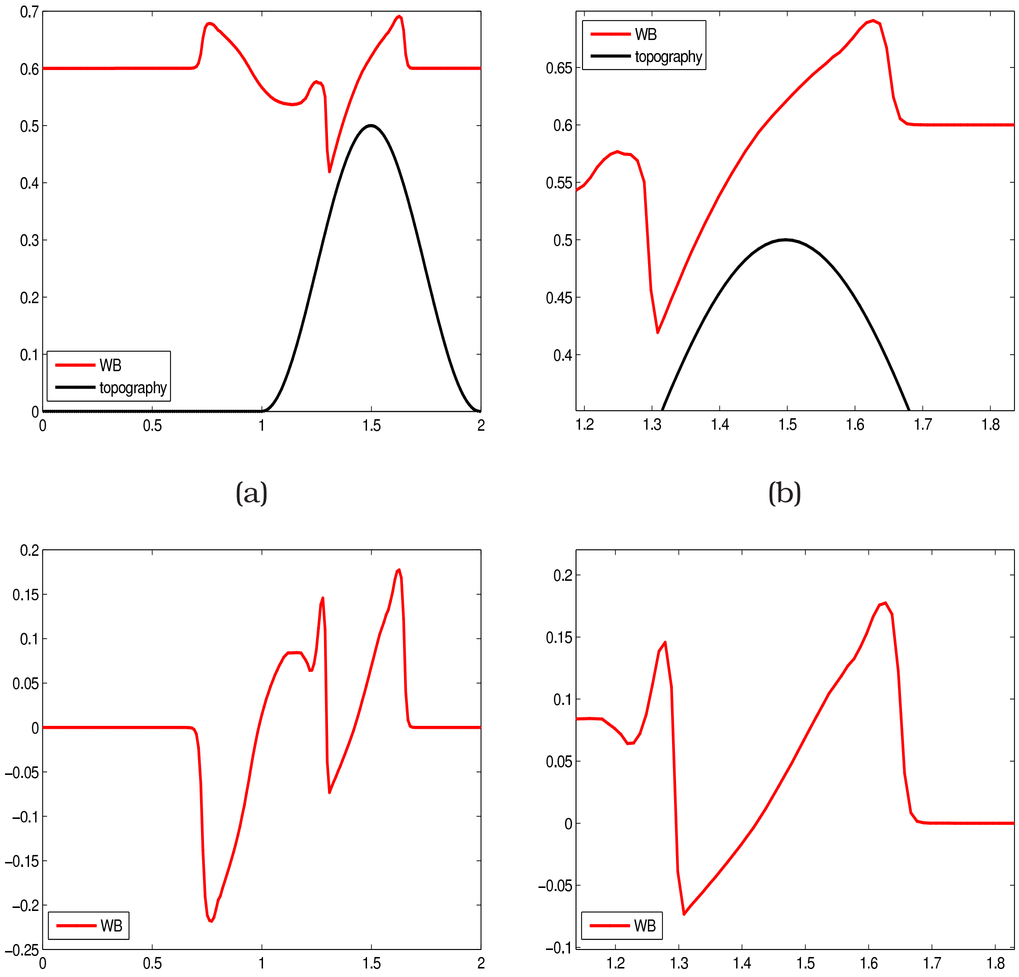


(a)



(b)

Figure 6.14: *Multi-level grid structure for the Circular Dam-break problem at times (a)* $T = 0.15$ *and (b)* $T = 0.25$*. Here* $L = 4$*, and* $\tau = 10^{-1}$.



(a) (b)

(c) (d)

Figure 6.15: *Circular Dam-break problem: (a) and (b) are a longitudinal section at* $y = 1$ *of* h. *(c) and (d) are a longitudinal section at* $y = 1$ *of* $q_1 = u_x h$ *at time* $T = 0.15$.

7

Conclusions and further work

7.1 Conclusions

In this work we have considered a high-order shock-capturing scheme built from Shu-Osher's conservative scheme, Lax-Friedrichs' flux-splitting method, a fifth-order WENO interpolatory technique and a third-order Runge-Kutta algorithm and we have introduced some improvements to some of its elements.

In first place, we have analyzed the Weighted ENO reconstructions proposed by Yamaleev and Carpenter in [104], showing that the WENO method with Yamaleev and Carpenter's weights has worse order of accuracy near discontinuities than the corresponding WENO method with Jiang and Shu's weights, fact that is reflected in the numerical experiments as some oscillations that may appear near discontinuities.

To alleviate the problem of loss of accuracy at extrema while retaining maximal-order weights near discontinuities, we have proposed a new set of weights based on Yamaleev and Carpenter's weights. We have proved that, near discontinuities, the order of accuracy with these weights is better than the order of accuracy achieved with Yamaleev and Carpenter's weights and that the oscillations obtained in the approximated solutions using Yamaleev and Carpenter's weights diminish considerably when we use the newly proposed weights.

Secondly, we have presented a comparative study of different strategies to reduce diffusion and spurious oscillations when using HRSC component-wise finite-difference WENO schemes for polydisperse sedimentation problems. On the one hand we have analyzed two flux-splitting methods: the commonly used Lax- Friedrichs' flux-splitting and the HLL flux-splitting, which allows an asymmetric choice of the wave speeds of each of the two terms of the flux-splitting. We have tested the algorithm with several experiments and we have seen that a local HLL flux-splitting improves the results obtained with global Lax-Friedrichs flux-splittings.

On the other hand we have studied different weight's design for the fifth-order WENO scheme to reduce spurious oscillations caused by the reconstruction method. We have focused our study on the global weights defined by Levy, Puppo and Russo in [74, 75] and we have seen that when using these weights instead of Jiang and Shu's weights the spurious oscillations may be reduced in some cases.

Finally, we have studied Adaptive Mesh Refinement algorithms applied to realistic simulations involving shallow water flows. We have combined the HRSC scheme with the AMR technique, developed by Berger et al., and we have seen how these techniques can be merged together to build up a highly efficient numerical method.

We have shown that, even when the underlying scheme is well-balanced, the numerical solution obtained when implementing block-structured AMR techniques for shallow water flows, will fail to satisfy the exact C-property if the operators that are in charge of transferring information between levels are not well-balanced themselves.

We have pointed out some of the difficulties for getting finite-volume well-balanced adaptive mesh refinement schemes for the shallow water equation, and we have presented a technique for obtaining point-value-based adaptive mesh refinement schemes for shallow water flow which are well-balanced for water at rest solutions, provided the underlying scheme is so. Our technique is based on interpolating the equilibrium variables, instead of the state variables, as in the original block-structured AMR technique [9].

We have performed a series of numerical tests, taking as the underlying well-balanced scheme the hybrid second-order scheme described in [34, 80], that confirm that the proposed AMR technique is able to preserve, up to machine accuracy, water-at-rest steady state solutions of the shallow water equations in 1D and 2D.

7.2 Further work

As future research, we are working on improving the efficiency of the HRSC scheme for polydisperse sedimentation models. We know that when solving hyperbolic systems of conservation laws, non-smooth structures might appear spontaneously and evolve in time. Typical solutions for the polydisperse sedimentation model considered for batch settling in a column include stationary kinematic shocks separating layers of sediment of different composition, as we have seen throughout the experiments carried on this thesis. Analyzing this special structure of the solutions, it is easy to see that the solution is over-resolved in regions where the solution is smooth. Consequently, we can improve the computational cost of the scheme, while maintaining its high-order properties, if we use expensive resources only at a neighborhood of a singularity.

With this idea on mind, we are developing a hybrid scheme, not adaptive, that uses the characteristic information only on a neighborhood of a discontinuity, while uses a component-wise approach when we are located on a smooth region. Some more effort could be invested on the use of adaptive schemes to these models.

We are also analyzing the advantages of the use of a HLL flux-splitting and the different definition of the weights studied in this thesis on a characteristic-wise scheme.

In the case of the simulations involving shallow water flows, we are working on the parallelization of the code, needed because of the high computational cost of some problems, especially 2D realistic experiments, and its extension to deal with dry zones. We are also exploring the possibility of getting an adaptive scheme that preserves more stationary solutions if the underlying scheme does so.

Bibliography

[1] J. Anderson. A secular equation for the eigenvalues of a diagonal matrix perturbation. *Linear Algebra Appl.*, 246:49–70, 1996.

[2] J. D. Anderson. *Modern compressible flow*. McGraw-Hill, 1982.

[3] F. Aràndiga, A. Baeza, A. M. Belda, and P. Mulet. Analysis of WENO schemes for full and global accuracy. *SIAM J. Num. Anal.*, 42(2):893–915, 2011.

[4] F. Aràndiga and R. Donat. Nonlinear multiscale decompositions: the approach of A. Harten. *Numer. Algorithms*, 23:175–216, 2000.

[5] E. Audusse, F. Bouchut, M. O. Bristeau, R. Klein, and B. Perthame. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *SIAM J. Sci. Comp*, 25:2050–2065, 2004.

[6] A. Baeza. *Adaptive mesh refinement techniques for high order schemes for hyperbolic systems of conservation laws*. PhD thesis, Dept. Matemàtica Aplicada, Universitat de València, 2010.

[7] A. Baeza, R. Donat, and A. Martinez-Gavara. A numerical treatment of wet/dry zones in well-balanced hybrid schemes for shallow water flow. *Appl. Numer. Math.*, 62(4):264–277, 2012.

[8] A. Baeza, A. Martínez-Gavara, and P. Mulet. Adaptation based on interpolation errors for high order mesh refinement methods applied to conservation laws. *Appl. Numer. Math.*, 62(4):278–296, 2012.

[9] A. Baeza and P. Mulet. Adaptive mesh refinement techniques for high order shock capturing schemes for hyperbolic systems of conservation laws. Technical Report GrAN 04–02, Departament de Matemàtica Aplicada, Universitat de València, Spain, 2004.

[10] A. Baeza and P. Mulet. Adaptive mesh refinement techniques for high-order shock capturing schemes for multi-dimensional hydrodynamic simulations. *Internat. J. Numer. Methods Fluids*, 52(4):455–471, 2006.

[11] D. S. Balsara and C. W. Shu. Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy. *J. Comput. Phys.*, 160(2):405–452, 2000.

[12] G. K. Batchelor. *An introduction to fluid dynamics*. Cambridge University Press, 2000.

[13] M. J. Berger. *Adaptive mesh refinement for hyperbolic partial differential equations*. PhD thesis, Computer Science Dept., Stanford University, 1982.

[14] M. J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82:64–84, 1989.

[15] M. J. Berger, D. L. George, R. J. LeVeque, and K. Mandli. The geoclaw software for depth-averaged flows with adaptive refinement. *Advances in Water Resources*, 34(9):1195–1206, 2011.

[16] M. J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, 53:484–512, 1984.

[17] A. Bermúdez and M. E. Vázquez. Upwind methods for hyperbolic conservation laws with source terms. *Comput. & Fluids*, 23(8):1049–1071, 1994.

[18] S. Berres, R. Bürger, K. H. Karlsen, and E. M. Tory. Strongly degenerate parabolic-hyperbolic systems modeling polydisperse sedimentation with compression. *SIAM J. Appl. Math.*, 64(1):41–80, 2003.

[19] R. Borges, M. Carmona, B. Costa, and W. S. Don. An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws. *J. Comput. Phys.*, 227:3191–3211, 2008.

[20] F. Bouchut and T. Morales de Luna. A subsonic-well-balanced reconstruction scheme for shallow water flows. *SIAM J. Numer. Anal.*, 48(5):1733–1758, 2010.

[21] S. Bryson, Y. Epshteyn, A. Kurganov, and G. Petrova. Well-balanced positivity preserving central-upwind scheme on triangular grids for the Saint-Venant system. *ESAIM Math. Model. Numer. Anal.*, 45(3):423–446, 2011.

[22] R. Bürger, F. Concha, K.-K. Fjelde, and K. H. Karlsen. Numerical simulation of the settling of polydisperse suspensions of spheres. *Powder Technol.*, 113:30–54, 2000.

[23] R. Bürger, R. Donat, P. Mulet, and C. A. Vega. Hyperbolicity analisys of polydisperse sedimentation models via a secular equation for the flux jacobian. *SIAM J. Appl. Math.*, 70:2186–2213, 2010.

[24] R. Bürger, R. Donat, P. Mulet, and C. A. Vega. On the implementation of WENO schemes for a class of polydisperse sedimentation models. *J. Comput. Phys.*, 230:2322–2344, 2011.

[25] J. M. Burgers. A mathematical model illustrating the theory of turbulence. *Adv. Appl. Mech.*, 1:171–199, 1948.

[26] V. Caselles, R. Donat, and G. Haro. Flux-gradient and source-term balancing for certain high resolution shock-capturing schemes. *Comput. & Fluids*, 38(1):16–36, 2009.

[27] M. J. Castro, E. D. Fernández-Nieto, A. M. Ferreiro, J. A. García-Rodríguez, and C. Parés. High Order Extensions of Roe Schemes for Two-Dimensional Nonconservative Hyperbolic Systems. *J. Sci. Comput.*, 39:67–114, 2009.

[28] A. J. Chorin and J. E. Marsden. *A mathematical introduction to fluid mechanics*. Springer, New York, 3^{rd} edition, 2000.

[29] A. Cohen, S. M. Kaber, S. Müller, and M. Postel. Fully adaptive multiresolution finite volume schemes for conservation laws. *Math. Comp.*, 72(241):183–225 (electronic), 2003.

[30] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen Differenzengleichungen der mathematischen Physik. *Math. Ann.*, 100(1):32–74, 1928. English translation: ”On the partial difference equations of mathematical physics”, IBM Journal of Research and Development, 11:215–234, 1967.

[31] C. M. Dafermos. *Hyperbolic conservation laws in continuum physics*. Springer, 2000.

[32] M. J. Castro Díaz, J. A. López-García, and C. Parés. High order exactly well-balanced numerical methods for shallow water systems. *J. Comput. Phys.*, 246:242–264, 2013.

[33] R. Donat and A. Marquina. Capturing shock reflections: an improved flux formula. *J. Comput. Phys.*, 125(1):42–58, 1996.

[34] R. Donat and A. Martinez-Gavara. A hybrid second order scheme for scalar balance laws. *J. Sci. Comput.*, 48(1-3):52–69, 2011.

[35] R. Donat and P. Mulet. Characteristic-based schemes for multi-class Lighthill-Whitham-Richards traffic models. *J. Sci. Comput.*, 37(3):233–250, 2008.

[36] R. Donat and P. Mulet. A secular equation for the Jacobian matrix of certain multispecies kinematic flow models. *Numer. Methods Partial Differential Equations*, 26(1):159–175, 2010.

[37] A. Duran, Q. Liang, and F. Marche. On the well-balanced numerical discretization of shallow water equations on unstructured meshes. *J. Comput. Phys.*, 235:565–586, 2013.

[38] V. Elling. *A Lax-Wendroff type theorem for unstructured grids*. PhD thesis, Stanford University, 2004.

[39] H. Feng, F. Hu, and R. Wang. A new mapped weighted essentially non-oscillatory scheme. *J. Sci. Comput.*, 51:449–473, 2012.

[40] T. Gallouët, J. M. Hérard, and N. Seguin. Some approximate Godunov schemes to compute shallow water equations with topography. *Comput. Fluids.*, 32:479–513, 2003.

[41] Ll. Gascón and J. M. Corberán. Construction of second-order TVD schemes for nonhomogeneous hyperbolic conservation laws. *J. Comput. Phys.*, 172(1):261–297, 2001.

[42] D. L. George. Adaptive finite volume methods with well-balanced Riemann solvers for modeling floods in rugged terrain: Application to the Malpasset dam-break flood (France, 1959). *International Journal for Numerical Methods in Fluids*, 66(8):1000–1018, 2011.

[43] G.A. Gerolymos, D. Sénéchal, and I. Vallet. Very-high-order WENO schemes. *J. Comput. Phys.*, 228(23):8481–8524, 2009.

[44] 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.

[45] O. Gonzalez and A. M. Stuart. *A First Course in Continuum Mechanics*. Cambridge University Press, 2008.

[46] N. Goutal and F. Maurel. In: Proceedings of the 2nd workshop on dam-break wave simulation. *EDF-DER Report*, HE-43/97/016/B, 1997.

[47] J. M. Greenberg and A. Y. Leroux. A well-balanced scheme for the numerical processing of source terms in hyperbolic equations. *SIAM J. Numer. Anal.*, 33(1):1–16, 1996.

[48] J. M. Greenberg, A. Y. Leroux, R. Baraille, and A. Noussair. Analysis and approximation of conservation laws with source terms. *SIAM J. Numer. Anal.*, 34:1980–2007, 1997.

[49] H.P. Greenspan and M. Ungarish. On hindered settling of particles of different sizes. *Int. J. Multiphase Flow*, 8:587–604, 1982.

[50] A. Harten. Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. *Comm. Pure Appl. Math.*, 48:1305–1342, 1995.

[51] 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.

[52] A. Harten, P. D. Lax, and B. van Leer. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Rev.*, 25:35–61, 1983.

[53] A. Harten and S. Osher. Uniformly high order accurate essentially non-oscillatory schemes, I. *SIAM J. Numer. Anal.*, 24(2):279–309, 1987.

[54] A.K. Henrick, T.D. Aslam, and J.M Powers. Mapped weighted essentially non-oscillatory schemes: Achieving optimal order near critical points. *J. Comput. Phys.*, 207:542–567, 2005.

[55] C. Hirsch. *Numerical computation of internal and external flows (volume 1): fundamentals of numerical discretization.* John Wiley & Sons, Inc., New York, NY, USA, 1988.

[56] 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.

[57] M. E. Hubbard and N. Dodd. A 2D numerical model of wave run-up and overtopping. *Coastal Engineering*, 47(1):1–26, NOV 2002.

[58] H. Hugoniot. Sur la propagation du movement dans les coprs et spécialement dans les gaz parfaits. *J. Ecole Polytechnique*, 57:3–97, 1887.

[59] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *J. Comput. Phys.*, 126(1):202–28, 1996.

[60] D. Kroner, M. Rokyta, and M. Wierse. A Lax-Wendrof type theorem for upwind finite volume schemes in 2D. *East-West J. Numer. Math*, 4:279–292, 1996.

[61] A. Kurganov and E. Tadmor. New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations. *J. Comput. Phys.*, 160(1):241–282, 2000.

[62] P. Lamby, R. Müller, and Y. Stiriba. Solution of shallow water equations using fully-adaptive multiscale schemes. *Int. J. Numer. Meth. Fluids*, 49(4):417–437, 2005.

[63] L. D. Landau and E. M. Lifshitz. *Fluid mechanics.* Course of theoretical physics, vol. 6. Pergamon Press, Oxford, 2^{nd} edition, 1987.

[64] P. D. Lax. Weak solutions of nonlinear hyperbolic equations and their numerical computation. *Comm. Pure Appl. Math.*, 7:159–193, 1954.

[65] P. D. Lax. Hyperbolic systems of conservation laws, II. *Comm. Pure Appl. Math.*, 10:537–566, 1957.

[66] P. D. Lax. Shock waves and entropy. In E.A. Zarantonello, editor, *Contributions to nonlinear functional analysis*, pages 603–634. Academic Press, 1971.

[67] 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.

[68] P. D. Lax and R. D. Richtmyer. Stability of difference equations. *Comm. Pure Appl. Math.*, 9:267–293, 1956.

[69] P. D. Lax and B. Wendroff. Systems of conservation laws. *Comm. Pure Appl. Math.*, 13:217–237, 1960.

[70] R. J. LeVeque. *Numerical methods for conservation laws*. Birkhäuser Verlag, 1992.

[71] R. J. LeVeque. Balancing source terms and flux gradients in high-resolution Godunov methods: the quasi-steady wave-propagation algorithm. *J. Comput. Phys.*, 146:346–365, 1998.

[72] R. J. LeVeque. *Finite-volume methods for hyperbolic problems*. Cambridge University Press, 2004.

[73] R. J. LeVeque, D. L. George, and M .J. Berger. Tsunami modelling with adaptively refined finite volume methods. *Acta Numer.*, 20:211–289, 2011.

[74] D. Levy, G.Puppo, and G.Russo. Central weno schemes for hyperbolic systems of conservation laws. *M2AN, Mat. Mod. and Num. An.*, 33:547–571, 1999.

[75] D. Levy, G.Puppo, and G.Russo. A fourth-order central weno scheme for multidimensional hyperbolic systems of conservation laws. *SIAM J. Sci. Comput.*, 24:480–506, 2002.

[76] Q. Liang. A structured but non-uniform cartesian grid-based model for the shallow water equations. *Int. J. Numer. Meth. Fluids*, 66(5):537–554, 2011.

[77] T.-P. Liu. The entropy condition and the admissibility of shocks. *J. Math. Anal. Appl.*, 53:78–88, 1976.

[78] X-D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *J. Comput. Phys.*, 115:200–212, 1994.

[79] M. J. Lockett and K. S. Bassoon. Sedimentation of binary particle mixtures. *Powder Technol.*, 24:1–7, 1979.

[80] A. Martinez-Gavara and R. Donat. A hybrid second order scheme for shallow water flows. *J. Sci. Comput.*, 48(1-3):241–257, 2011.

[81] J. H. Masliyah. Hindered settling in a multiple-species particle system. *Chem. Eng. Sci.*, 34:1166–1168, 1979.

[82] S. Müller and Y. Stiriba. Fully adaptive multiscale schemes for conservation laws employing locally varying time stepping. *J. Sci. Comput.*, 30(3):493–531, 2007.

[83] H. Nessyahu and E. Tadmor. Nonoscillatory central differencing for hyperbolic conservation laws. *J. Comput. Phys.*, 87(2):408–463, 1990.

[84] S. Noelle, Y. Xing, and C.W. Shu. High order well-balanced finite volume weno schemes for shallow water equations with moving water.

[85] O. Oleinik. Discontinuous solutions of nonlinear differential equations. *Amer. Math. Soc. Transl. Ser. 2*, 26:95–172, 1957.

[86] K. G. Powell., P. L. Roe, and J. Quirk. Adaptive-mesh algorithms for computational fluid dynamics. In *Algorithmic trends in computational fluid dynamics (1991)*, ICASE/NASA LaRC Ser., pages 303–337. Springer, New York, 1993.

[87] J. J. Quirk. A parallel adaptive grid algorithm for computational shock hydrodynamics. *Applied Numerical Mathematics*, 20(4):427–453, APR 1996.

[88] W. J. M. Rankine. On the thermodynamic theory of waves of finite longitudinal disturbance. *Phil. Trans. Roy. Soc. London*, 160:277–288, 1870.

[89] M. Ricchiuto, R. Abgrall, and H. Deconinck. Application of conservative residual distribution schemes to the solution of the shallow water equations on unstructured meshes. *J. Comput. Phys.*, 222:287–331, 2007.

[90] J. F. Richardson and W. N. Zaki. The sedimentation of a suspension of uniform spheres under conditions of viscous flow. *Chemical Engineering Science*, 3(2):65–73, 1954.

[91] R. D. Richtmyer and K. W. Morton. *Difference methods for initial-value problems*, volume 4 of *Interscience Tracts in Pure and Applied Mathematics*. Wiley Interscience, New York, U.S.A., 2nd edition, 1967.

[92] P. T Shannon, E. Stroupe, and E. M. Tory. Batch, continuous thickening. *Ind. Eng. Chem. Fund.*, 2:203–211, 1963.

[93] C. Shen, J. Qiu, and A. Christlieb. Adaptive mesh refinement based on high order finite difference WENO scheme for multi-scale simulations. *J. Comput. Phys.*, 230(10):3780–3802, 2011.

[94] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.*, 77(2):439–471, 1988.

[95] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. *J. Comput. Phys.*, 83(1):32–78, 1989.

[96] G. A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J. Comput. Phys.*, 27:1–31, 1978.

[97] M. Sun and K. Takayama. An artificially upstream flux vector splitting scheme for the Euler equations. *J. Comput. Phys.*, 189(1):305–329, 2003.

[98] E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics*. Springer-Verlag, third edition, 2009.

[99] 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.

[100] R. F. Warming and R. W. Beam. Upwind second order difference schemes with applications in aerodynamic flows. *AIAA Journal*, 24:1241–1249, 1976.

[101] B. Wendroff. The Riemann problem for materials with nonconvex equation of state. *J. Math. Anal. Appl.*, 38:454–466, 1972.

[102] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.*, 54(1):115–173, 1984.

[103] Y. Xing and C. W. Shu. High order finite difference WENO schemes with the exact conservation property for the shallow water equations. *J. Comput. Phys.*, 208:206–227, 2005.

[104] N.K. Yamaleev and M.H. Carpenter. A systematic methodology for constructing high-order energy stable WENO schemes. *J. Comput. Phys.*, 228:4248–4272, 2009.

[105] N.K. Yamaleev and M.H. Carpenter. Third-order energy stable weno scheme. *J. Comput. Phys.*, 228:3025–3047, 2009.

[106] M. Zhang, C.-W. Shu, G. C. K. Wong, and S. C. Wong. A weighted essentially non-oscillatory numerical scheme for a multi-class Lighthill-Whitham-Richards traffic flow model. *J. Comput. Phys.*, 191(2):639–659, 2003.

[107] P. Zhang, R. X. Liu, S. C. Wong, and S. Q. Dai. Hyperbolicity and kinematic waves of a class of multi-population partial differential equations. *European J. Appl. Math.*, 17:171–200, 2006.