



**Universidad**  
Zaragoza

# Trabajo Fin de Grado

Estudio comparativo y análisis de  
sensibilidad de un modelo computacional  
de simulación hidráulica para flujos en  
superficie libre

Autor

Carlos Azara Egea

Directores

Mario Morales Hernández  
Asier Lacasta Soto

Ponente

Pilar García Navarro

Escuela de Ingeniería y Arquitectura (EINA)  
2014



# Agradecimientos

---

Una vez terminado este Trabajo fin de Grado tengo que agradecer muchas cosas a todas las personas que han hecho esto posible.

En primer lugar agradecer a la Universidad de Zaragoza, pero más concretamente al Área de Mecánica de Fluidos de la Escuela de Ingeniería y Arquitectura (EINA) la posibilidad de realizar este Trabajo con ellos. En especial a Pilar García Navarro, por su confianza y buena disposición desde el primer momento.

Destacar también dentro de este departamento a Mario Morales y Asier Lacasta, del Grupo de Hidráulica Computacional, sin quienes no habría sido posible este Trabajo. Gracias por la ayuda que me han brindado siempre que la he necesitado, por su colaboración y por su saber hacer, pero sobre todo por el trato recibido.

Mención especial merecen mi madre y mi hermano, por aguantarme siempre y darme libertad total para centrarme en la realización del Trabajo, sin nunca echarme en cara el poco tiempo disponible para ellos; así como a mi padre que seguro estaría orgulloso.

Para terminar, agradecer también a Carla el apoyo que me da y me ha dado siempre, y porque además fue la que me impulsó a preguntar si podría realizar el Trabajo fin de Grado con el Área de Mecánica de Fluidos. Pero sobre todo, agradecerle la paciencia que ha tenido y tiene conmigo, y el apoyo incondicional que ha sido durante estos años de carrera.



# Resumen

---

El objetivo del Trabajo fin de Grado que se presenta a continuación es el análisis de la calidad de los resultados de un modelo computacional de simulación en volúmenes finitos para estudios medioambientales utilizando distintas técnicas de discretización espacial (mallado) y diferentes implementaciones de programación. Los problemas que se estudian son de flujo transitorio en superficie libre sobre superficies irregulares. Se trata de abordar estudios de inundabilidad mediante propagación de ondas.

Para simular este tipo de fenómenos existen muchos modelos computacionales. La *U.K. Environment Agency* generó en 2009 un conjunto de situaciones tipo [9] que permiten la comparación objetiva entre paquetes de simulación hidráulica comerciales. En este trabajo, el modelo computacional posee un núcleo de cálculo desarrollado en el Área de Mecánica de Fluidos de la Escuela de Ingeniería y Arquitectura (EINA) de la Universidad de Zaragoza, concretamente por el Grupo de Hidráulica Computacional. Es este núcleo de cálculo el que se somete a análisis y sobre el que se realiza el estudio comparativo.

Durante el desarrollo del trabajo se han simulado los casos presentados en este documento, y a lo largo de la memoria se exponen los resultados obtenidos.



# Índice general

---

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Contexto del trabajo . . . . .	1
1.2	Estructura del trabajo . . . . .	2
<b>2</b>	<b>Método de cálculo</b>	<b>3</b>
2.1	Ecuaciones gobernantes . . . . .	3
2.2	Modelo en volúmenes finitos . . . . .	4
2.3	Región de estabilidad . . . . .	6
2.4	Condiciones de contorno y variables calculadas . . . . .	7
<b>3</b>	<b>Descripción del estudio comparativo</b>	<b>9</b>
3.1	Versiones del modelo computacional . . . . .	9
3.2	Tipos de mallas y CFL . . . . .	10
3.3	Variables de análisis . . . . .	11
3.3.1	Eficiencia del modelo computacional . . . . .	11
3.3.2	Precisión de los resultados . . . . .	12
<b>4</b>	<b>Resultados</b>	<b>13</b>
4.1	Caso 1 . . . . .	13
4.1.1	Descripción del caso . . . . .	13
4.1.2	Análisis de resultados . . . . .	15
4.2	Caso 2 . . . . .	17
4.2.1	Descripción del caso . . . . .	17
4.2.2	Análisis de resultados . . . . .	18
4.3	Caso 3 . . . . .	22
4.3.1	Descripción del caso . . . . .	22
4.3.2	Análisis de resultados . . . . .	23
4.4	Caso 4 . . . . .	28
4.4.1	Descripción del caso . . . . .	28
4.4.2	Análisis de resultados . . . . .	31
<b>5</b>	<b>Conclusiones</b>	<b>41</b>
	<b>Bibliografía</b>	<b>42</b>
	<b>Anexos</b>	<b>45</b>

<b>A</b>	<b>Tipos de malla</b>	<b>47</b>
A.1	Malla Triangular Delaunay (TD) . . . . .	47
A.2	Malla Triangular Equilátera (TE) . . . . .	48
A.3	Malla Rectangular (R) . . . . .	49
<b>B</b>	<b>Proceso de simulación</b>	<b>51</b>
B.1	Modos de implementación. CPU y GPU. . . . .	51
B.2	Ficheros y elementos del proceso de simulación . . . . .	53
<b>C</b>	<b>Programas informáticos auxiliares</b>	<b>59</b>
C.1	Representaciones gráficas: <i>Gnuplot</i> . . . . .	59
C.2	Visualización de datos: <i>Paraview</i> . . . . .	60
<b>D</b>	<b>Temporalización del Trabajo</b>	<b>63</b>



# Índice de tablas

---

4.1	Puntos a estudiar Caso 1 . . . . .	14
4.2	Tiempos de simulación Caso 1 . . . . .	16
4.3	Puntos a estudiar Caso 2 . . . . .	18
4.4	Tiempos de simulación Caso 2 . . . . .	21
4.5	Puntos a estudiar Caso 3 . . . . .	23
4.6	Tiempos de simulación Caso 3 . . . . .	28
4.7	Puntos a estudiar Caso 4 . . . . .	29
4.8	Tiempos de simulación Caso 4. Mallas no refinadas. . . . .	35
4.9	Tiempos de simulación Caso 4. Mallas refinadas . . . . .	39



# Índice de figuras

---

2.1	Representación uniforme a trozos (discreta) de las variables conservadas . . . . .	5
2.2	Parámetros de celda . . . . .	5
2.3	$\Delta X$ malla Rectangular (izquierda), malla Triangular Equilátera (centro) y malla Triangular Delaunay (derecha) . . . . .	7
4.1	Dominio Caso 1 . . . . .	13
4.2	Perfil Caso 1 . . . . .	14
4.3	Condición de contorno Caso 1 . . . . .	14
4.4	Mallas Caso 1. De arriba a abajo, TD, TE, R . . . . .	15
4.5	Vista 3D Caso 1 . . . . .	15
4.6	Evolución temporal de $h + z$ para el punto 1 (izquierda) y el punto 2 (derecha). Caso 1 . . . . .	16
4.7	Dominio Caso 2 . . . . .	17
4.8	Condición de contorno Caso 2 . . . . .	17
4.9	Vista 3D Caso 2 . . . . .	18
4.10	Mallas Caso 2. De izquierda a derecha, TD, TE, R . . . . .	19
4.11	Evolución temporal de $h + z$ de arriba-izquierda a abajo-derecha para los puntos 1, 2, 3 y 4. Caso 2 . . . . .	19
4.12	Evolución temporal de $h+z$ recorriendo las gráficas de arriba a abajo y de izquierda a derecha para los puntos 5, 6, 7, 8, 9, 10, 11 y 12. Caso 2 . . . . .	20
4.13	Evolución temporal de $h + z$ de arriba-izquierda a abajo-derecha para los puntos 13, 14, 15 y 16. Caso 2 . . . . .	21
4.14	Dominio Caso 3 . . . . .	22
4.15	Condición de contorno Caso 3 . . . . .	23
4.16	Vista 3D Caso 3 . . . . .	24
4.17	Mallas Caso 3. De izquierda a derecha, TD, TE, R . . . . .	24
4.18	Evolución temporal de $h + z$ (izquierda) y $modU$ (derecha) para el punto 1. Caso 3 . . . . .	24
4.19	Evolución temporal de $h + z$ (columna izquierda) y $modU$ (columna derecha) para los puntos 2, 3, 4 y 5. Caso 3 . . . . .	25
4.20	Evolución temporal de $h + z$ (izquierda) y $modU$ (derecha) para el punto 6. Caso 3 . . . . .	26
4.21	Evolución de $h + z$ (columna izquierda) y $modU$ (columna derecha) a lo largo de $x$ ( $y = 1.000$ ) en $t = 0,5 h$ (fila 1) y $t = 1 h$ (fila 2). Caso 3. . . . .	26
4.22	Evolución de $h + z$ (columna izquierda) y $modU$ (columna derecha) a lo largo de $x$ ( $y = 1.000 m$ ) en $t = 2 h$ (fila 1) , $t = 3 h$ (fila 2) y $t = 4 h$ (fila 3). Caso 3 . . . . .	27
4.23	Dominio Caso 4 . . . . .	29
4.24	Condición de contorno tipo lluvia Caso 4 . . . . .	30
4.25	Condición de contorno tipo fuente Caso 4 . . . . .	30

4.26	Vista en 3D del Caso 4 antes de comenzar la simulación (imagen superior) y para $t = 9 \text{ min}$ (imagen inferior) . . . . .	31
4.27	Mallas Caso 4. Cota $z$ (superior) y de izquierda a derecha mallas TD, TE y R (inferior) . . . . .	32
4.28	Evolución temporal de $h+z$ (columna izquierda) y $modU$ (columna derecha) para los puntos 1, 2, 3 y 4. Caso 4 . . . . .	33
4.29	Evolución temporal de $h+z$ (columna izquierda) y $modU$ (columna derecha) para los puntos 5, 6, 7 y 8. Caso 4 . . . . .	34
4.30	Evolución temporal de $h+z$ (izquierda) y $modU$ (derecha) para el punto 9. Caso 4 . . . . .	35
4.31	Mallas refinadas TD (izquierda) y TE (derecha) . . . . .	36
4.32	Evolución temporal de $h+z$ (columna izquierda) y $modU$ (columna derecha) para los puntos 1, 2, 3 y 4. Caso 4 malla refinada . . . . .	37
4.33	Evolución temporal de $h+z$ (columna izquierda) y $modU$ (columna derecha) para los puntos 5, 6, 7 y 8. Caso 4 malla refinada . . . . .	38
4.34	Evolución temporal de $h+z$ (izquierda) y $modU$ (derecha) para el punto 9. Caso 4 con malla refinada . . . . .	39
A.1	Diagrama de Voronoi (izquierda) y triangulación Delaunay (derecha) . . . . .	48
A.2	Malla Triangular Delaunay (TD) . . . . .	48
A.3	Malla Triangular Equilátera (TE) . . . . .	48
A.4	Malla Rectangular (R) . . . . .	49
B.1	Diagrama de flujo del proceso de simulación . . . . .	51
B.2	Proceso iterativo secuencial en 1 núcleo sobre el vector $A[ ]$ . . . . .	52
B.3	Proceso iterativo paralelo sobre el vector $A[ ]$ empleando 4 núcleos . . . . .	52
B.4	Proceso iterativo masivamente paralelo sobre el vector $A[ ]$ utilizando una arquitectura <i>Many-core</i> (GPU) . . . . .	53
C.1	Representaciones con <i>Gnuplot</i> . . . . .	60
C.2	Interfaz de <i>Paraview</i> . . . . .	60
D.1	Diagrama de Gantt del Trabajo fin de Grado . . . . .	64

# Capítulo 1

## Introducción

---

Desde los inicios, los ríos favorecen el desarrollo de asentamientos humanos permanentes en sus riberas al proporcionar terrenos adecuados para el cultivo en las llanuras próximas al cauce y agua de consumo necesaria para sus habitantes. No obstante, los ríos son susceptibles de sufrir variaciones de caudal de tal intensidad que pueden llegar a poner en peligro la vida que se desarrolla en sus alrededores. Las acciones de control sobre estos fenómenos se pueden clasificar en estructurales y no estructurales. Las primeras hacen referencia a aquellas actuaciones cuyo fin es reducir en lo posible el nivel de las inundaciones para que éstas no afecten a las actividades humanas. Entre otras destacan la modificación de la topografía o de la vegetación (variando su rugosidad), planificación del suelo, limpieza de cauces, construcción de presas, diques... Por otro lado, las medidas no estructurales son aquellas destinadas a adecuar el riesgo de inundación para que no afecte a los asentamientos humanos o al entorno, basándose en la naturaleza predictiva de este tipo de fenómenos.

La predicción de sucesos naturales y de sus consecuencias requiere de la modelización matemática. El modelo realizado debe concordar lo máximo posible con los resultados de sucesos pasados conocidos, de manera que las extrapolaciones que se realicen posean una base fiable. La gran ventaja de los modelos con base física radica en su capacidad predictiva y en la posibilidad de incorporar de manera relativamente rápida cambios en las condiciones.

Los fenómenos de inundación y propagación de ondas de agua debido a aumentos del nivel superficial o del caudal de un río se describen como flujos en superficie libre. Para la correcta representación de estos fenómenos es necesario conocer los valores que toman el calado, las velocidades y los caudales mediante la aplicación de unas determinadas ecuaciones bajo ciertas hipótesis. La simulación es una aproximación a la realidad, y será más fiable cuanto mejor sea el esquema numérico y la discretización de las ecuaciones que modelan este tipo de fenómenos.

### 1.1 Contexto del trabajo

El desarrollo de modelos computacionales para simulación medioambiental está cobrando importancia en la actualidad. El Grupo de Hidráulica Computacional ([ghc.unizar.es](http://ghc.unizar.es)), perteneciente al Área de Mecánica de Fluidos de la Escuela de Ingeniería y Arquitectura (EINA) de la Universidad de Zaragoza, ha desarrollado un modelo computacional para la simulación de flujos en superficie libre que responde a la necesidad de disponer de este tipo de modelos para la prevención en problemas de inundación. Este modelo se ha desarrollado bajo la aproximación del flujo mediante las ecuaciones de aguas poco profundas (*Shallow Water Equations*, de ahora en adelante SWE).

Para la utilización de este modelo es necesaria la correcta caracterización de ciertos parámetros, como el tipo de malla a emplear o el paso de tiempo a través del número de *Courant-Friedrichs-Lewy* (de ahora en adelante *CFL*). Este Trabajo Fin de Grado versa sobre la sensibilidad del modelo a estos parámetros, y de cómo influye la variación de los mismos en los resultados obtenidos, tanto en términos de precisión como de carga computacional.

Para poder llevar a cabo este estudio comparativo y análisis de sensibilidad del modelo se emplea un documento desarrollado por la *Environment Agency* (Agencia Medioambiental Británica) [9]. En este documento se recogen varios casos test que esta agencia propuso como básicos para la evaluación de modelos de simulación de flujos de superficie libre.

## 1.2 Estructura del trabajo

El Trabajo Fin de Grado que se presenta a continuación se estructura en dos documentos distintos, uno referido a la memoria y otro a los anexos, incluidos ambos en la misma encuadernación.

A lo largo de la memoria, en el capítulo 2 se explica la formulación matemática y el método de cálculo; en el capítulo 3 se describe el estudio comparativo llevado a cabo; se exponen los resultados obtenidos para los casos estudiados en el capítulo 4, y finalmente se detallan las conclusiones en el capítulo 5.

En los anexos se puede encontrar información importante para la comprensión completa del alcance del trabajo. En el apéndice A figura una explicación más detallada de los tipos de malla empleados. El apéndice B contiene la descripción del proceso de simulación con todos los ficheros que son necesarios así como una descripción de la simulación en GPU y CPU. Además, el apéndice C contiene una breve reseña a dos programas informáticos auxiliares que se han empleado para el tratamiento de datos. Por último, el apéndice D explica la temporalización del trabajo junto con un diagrama de Gantt del mismo.

# Capítulo 2

## Método de cálculo

---

El modelo computacional en el que se basa el estudio modela el flujo de agua mediante las ecuaciones bidimensionales de aguas poco profundas (*Shallow Water Equations* ó SWE), consiguiendo una herramienta de simulación para el manejo de problemas de inundación.

Los modelos hidráulicos unidimensionales (1D) son los más adecuados para la simulación de ríos o canales, siempre y cuando el flujo de agua fluya encauzado [5]. No obstante, para la modelización de fenómenos de inundación donde la representación de la llanura de inundación exige aproximaciones más sofisticadas, estas aproximaciones 1D no son válidas [5]. Por otro lado, el elevado coste de simulación de métodos tridimensionales no simplificados puede ser excesivo. Por lo tanto, una buena aproximación puede ser la utilización de modelos bidimensionales que modelan las ecuaciones de aguas poco profundas (SWE 2D) obtenidas de promediar en la vertical el problema 3D [8] .

Cuando se habla de las SWE, las aplicaciones reales siempre incluyen términos fuente para describir la variación del nivel de la superficie así como la fricción con la misma. La propagación de frentes de onda sobre topografía irregular, rugosa e inicialmente seca ha sido un tema de mucha actividad reciente en la literatura hidráulica [6]. De no estar correctamente discretizados, pueden conducir a problemas de inestabilidad. Recientemente, para poder incluir adecuadamente el efecto de estos términos en la formulación de la solución débil se han presentado soluciones basadas en las aproximaciones de Riemann [6]. De este modo, las soluciones correctas pueden ser programadas evitando la necesidad de imponer parámetros de ajuste dependientes del caso, los cuales se usan frecuentemente para evitar valores negativos de calado y otras inestabilidades numéricas que aparecen cuando se incluyen estos términos.

### 2.1 Ecuaciones gobernantes

Los flujos de aguas poco profundas 2D se pueden describir matemáticamente mediante las ecuaciones de conservación de masa y momento con valores promedios de calado o profundidad, con todas las suposiciones que eso conlleva [10]. Este sistema de ecuaciones diferenciales en derivadas parciales se formula de manera conservativa.

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{U})}{\partial y} = \mathbf{S}(\mathbf{U}, x, y) \quad (2.1)$$

Siendo  $\mathbf{U}$ :

$$\mathbf{U} = (h, q_x, q_y)^T \quad q_x = uh \quad q_y = vh \quad (2.2)$$

un vector con las variables conservadas calado,  $h$ , y  $q_x$  y  $q_y$  los caudales unitarios, con  $(u, v)$  las componentes promedio del vector velocidad a lo largo de las direcciones del plano  $x$  e  $y$  respectivamente. Los vectores del flujo en (2.1) son:

$$\mathbf{F} = \left( q_x, \frac{q_y^2}{h} + \frac{1}{2}gh^2, \frac{q_x q_y}{h} \right)^T \quad \mathbf{G} = \left( q_y, \frac{q_x q_y}{h}, \frac{q_y^2}{h} + \frac{1}{2}gh^2 \right)^T \quad (2.3)$$

El término  $g$  hace referencia a la aceleración de la gravedad, y la expresión  $\frac{1}{2}gh^2$  en los flujos se obtiene asumiendo una distribución de presión hidrostática en la columna de agua, consideración común en las SWE. El vector de términos fuente incorpora el efecto de la fuerza de presión de la superficie sólida y las fuerzas tangenciales generadas por el esfuerzo sobre la misma.

$$\mathbf{S} = (i, gh(S_{ox} - S_{fx}), gh(S_{oy} - S_{fy}))^T \quad (2.4)$$

En (2.4),  $i$  representa una tasa de aportación o extracción de agua del exterior, y las fuerzas comentadas se expresan en función de las pendientes de la superficie del nivel inferior,  $z$ , y de las pendientes de fricción. La contribución del esfuerzo superficial se modela mediante el coeficiente  $n$  de Manning, que evalúa la fricción entre el flujo de agua y la topografía.

$$S_{ox} = -\frac{\partial z}{\partial x} \quad S_{oy} = -\frac{\partial z}{\partial y} \quad (2.5)$$

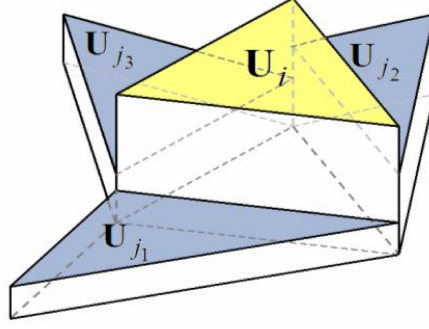
$$S_{fx} = \frac{n^2 u \sqrt{u^2 + v^2}}{h^{4/3}} \quad S_{fy} = \frac{n^2 v \sqrt{u^2 + v^2}}{h^{4/3}} \quad (2.6)$$

## 2.2 Modelo en volúmenes finitos

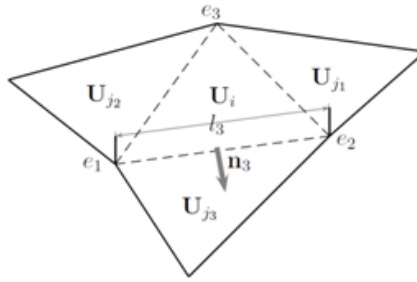
Para introducir el esquema de volúmenes finitos, la ecuación (2.1) se integra sobre un volumen o celda  $\Omega$  haciendo caso del teorema de Gauss, (2.7).

$$\frac{\partial}{\partial x} \int_{\Omega} \mathbf{U} d\Omega + \oint_{\partial\Omega} \mathbf{E} n dl = \int_{\Omega} \mathbf{S} d\Omega \quad (2.7)$$





**Figura 2.1:** Representación uniforme a trozos (discreta) de las variables conservadas



**Figura 2.2:** Parámetros de celda

donde  $\mathbf{E} = (\mathbf{F}, \mathbf{G})$  y  $\mathbf{n} = (n_x, n_y)$  es el vector unitario normal al exterior del volumen  $\Omega$ . Para obtener una solución numérica del sistema de (2.7) el dominio se divide en una serie de celdas computacionales,  $\Omega_i$ , usando una malla fija. Asumiendo una representación a trozos (discreta) de las variables conservadas (Figuras 2.1 y 2.2) y una formulación *upwind* unificada de los flujos y los términos fuente se llega a (2.8) [6].

$$A_i \frac{U_i^{n+1} - U_i^n}{\Delta t} + \sum_{k=1}^{NE} (\mathbf{E}\mathbf{n} - \bar{\mathbf{S}})_k^* l_k = 0 \quad (2.8)$$

La solución se puede definir utilizando una matriz Jacobiana aproximada  $\tilde{\mathbf{J}}_{n,k}$  del flujo normal no lineal  $\mathbf{E}\mathbf{n}^*$  tal que  $\delta(\mathbf{E}\mathbf{n}) = \tilde{\mathbf{J}}_{n,k} \delta \mathbf{U}$  y matrices aproximadas  $\tilde{\mathbf{P}} = (\tilde{\mathbf{e}}^1, \tilde{\mathbf{e}}^2, \tilde{\mathbf{e}}^3)$  y  $\tilde{\mathbf{P}}^{-1}$ , construidas usando los vectores propios del Jacobiano, de modo que lo diagonalizan, como se ve en (2.9).

$$\tilde{\mathbf{P}}^{-1} \tilde{\mathbf{J}}_{n,k} \tilde{\mathbf{P}} = \tilde{\mathbf{\Lambda}}_k \quad \tilde{\mathbf{J}}_{n,k} = \tilde{\mathbf{P}} \tilde{\mathbf{\Lambda}}_k \tilde{\mathbf{P}}^{-1} \quad (2.9)$$

Siendo  $\tilde{\mathbf{\Lambda}}_k$  una matriz diagonal con los valores propios  $\tilde{\mathbf{\Lambda}}_k^m$  en la diagonal principal.

$$\tilde{\mathbf{\Lambda}}_k = \begin{pmatrix} \tilde{\lambda}^1 & 0 & 0 \\ 0 & \tilde{\lambda}^2 & 0 \\ 0 & 0 & \tilde{\lambda}^3 \end{pmatrix} \quad (2.10)$$

Tanto la diferencia en el vector  $\mathbf{U}$  de variables conservadas como del vector de términos fuente son proyectadas sobre la base de los vectores propios de la matriz, (2.11).

$$\delta \mathbf{U}_k = \tilde{\mathbf{P}}_k \tilde{\mathbf{\Lambda}}_k \quad (\bar{\mathbf{S}})_k = \tilde{\mathbf{P}}_k B \quad (2.11)$$

Donde  $\mathbf{A}_k = (\alpha^1, \alpha^2, \alpha^3)_k^T$  y  $\mathbf{B} = (\beta^1, \beta^2, \beta^3)_k^T$ . Los detalles se pueden encontrar desarrollados en [6]. La linealización completa de todos los términos en combinación con la técnica *upwind* permite definir la función del flujo numérico  $(\mathbf{E}\mathbf{n} - \bar{\mathbf{S}})_k^*$ .

$$(\mathbf{E}\mathbf{n} - \bar{\mathbf{S}})_k^* = \mathbf{E}_i \mathbf{n}_k + \sum_{m=1}^3 \left( \tilde{\lambda}^- \Theta \alpha \tilde{\mathbf{e}} \right)_k^m \quad (2.12)$$

Con  $\tilde{\lambda}^- = \frac{1}{2} (\tilde{\lambda} - |\tilde{\lambda}|)$  y  $\Theta_k^m = \left( 1 - \frac{\beta}{\tilde{\lambda} \alpha} \right)$  que cuando se incluyen en (2.8) resulta un primer orden explícito del método de Godunov [7], dando lugar a (2.13).

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n + \sum_{k=1}^{NE} \left( \mathbf{E}_i \mathbf{n}_k + \sum_{m=1}^3 \left( \tilde{\lambda}^- \Theta \alpha \tilde{\mathbf{e}} \right)_k^m \right) \frac{l_k}{\mathbf{A}_i} \Delta t \quad (2.13)$$

Como la cantidad  $\mathbf{E}_i$  es uniforme para cada celda  $i$  y la propiedad geométrica expresada en (2.14) se da para todas las celdas,

$$\sum_{k=1}^{NE} n_k l_k = 0 \quad (2.14)$$

la ecuación (2.13) puede reescribirse como figura en (2.15).

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n + \sum_{k=1}^{NE} \left( \sum_{m=1}^3 \left( \tilde{\lambda}^- \Theta \alpha \tilde{\mathbf{e}} \right)_k^m \right) \frac{l_k}{\mathbf{A}_i} \Delta t \quad (2.15)$$

Por lo tanto, el método de volúmenes finitos puede ser descrito de manera compacta a través de la formulación denominada *wave-splitting*, siendo  $(\delta \mathbf{M}_{i,k}^-)^n = \sum_{m=1}^3 \left( \tilde{\lambda}^- \Theta \alpha \tilde{\mathbf{e}} \right)_k^m$ .

$$\mathbf{U}_i^k = \mathbf{U}_i^{n+1} + \sum_{k=1}^{NE} (\delta \mathbf{M}_{i,k}^-)^n \frac{l_k}{\mathbf{A}_i} \Delta t \quad (2.16)$$

El uso de la ecuación (2.16) resulta eficiente a la vez que asegura la conservación [6].

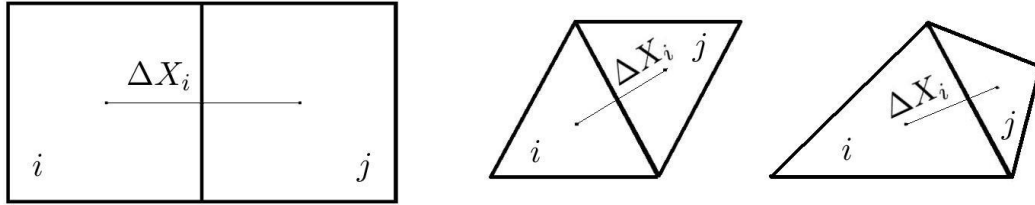
## 2.3 Región de estabilidad

La región de estabilidad numérica del esquema (2.16) para el caso homogéneo puede emplearse para calcular el tamaño del paso de tiempo. En el caso 2D, considerando mallas no estructuradas, la distancia relevante  $\Delta X_i$  para cada celda debe considerar el área de la celda y la longitud de los lados compartidos.

$$\Delta X_i = \frac{A_i}{\max_{k=1, NE} l_k} \quad (2.17)$$

$$\Delta t = \min \left\{ CFL \Delta t_k^{\tilde{\lambda}} \right\} \quad \Delta t_k^{\tilde{\lambda}} = \frac{\min(\Delta X_i, \Delta X_j)}{\max |\tilde{\lambda}_k^m|} \quad (2.18)$$

Por las características de este cálculo del paso de tiempo, el parámetro  $CFL$  variará según la manera de discretizar el dominio, ya sea mediante mallas Triangulares Delaunay (TD), mallas Triangulares Equiláteras (TE) o mallas Rectangulares (R), que son las estudiadas.



**Figura 2.3:**  $\Delta X$  malla Rectangular (izquierda), malla Triangular Equilátera (centro) y malla Triangular Delaunay (derecha)

En la Figura 2.3 se ilustra  $\Delta X_i$ , en la ecuación (2.17), para mallas R, TE y TD. En particular, con la condición aplicada de paso de tiempo (2.18), en mallas TE y R es necesario un valor de  $CFL$  de 0,5 para asegurar la estabilidad ya que el parámetro  $\Delta X_i$  está sobreestimado, mientras que en mallas TD, este valor puede aumentar hasta 1 [7].

## 2.4 Condiciones de contorno y variables calculadas

Por las características del sistema de ecuaciones, los resultados que aporta el modelo computacional para cada celda tras la resolución del sistema son el nivel de agua  $h$  que existe en cada celda; la altura del nivel superficial del agua, representada por el parámetro  $h + z$  y que informa del valor de la suma de la cota  $z$  y la altura de agua  $h$  sobre cada una de las celdas del dominio; la componente del vector velocidad del flujo en el eje  $x$ ,  $u$ ; la componente del vector velocidad en el eje  $y$ ,  $v$ ; y el módulo de la velocidad teniendo en cuenta el carácter bidimensional de ésta,  $modU$ . Con la información de estas variables se lleva a cabo el estudio comparativo y el análisis de sensibilidad, haciendo especial hincapié sobre la altura del nivel superficial  $h + z$  y sobre el módulo del vector velocidad,  $modU$ .

Para poder desarrollar el cálculo del sistema de ecuaciones es necesario definir unas condiciones iniciales y de contorno para comenzar a integrar la solución en cada una de las celdas que discretizan el dominio. Las condiciones de contorno se definen en las fronteras del dominio, distinguiéndose entre fronteras abiertas y cerradas. Se aplicarán condiciones de contorno de impenetrabilidad en cada una de las paredes de las fronteras cerradas del dominio. Dicho de otro modo, la velocidad normal a dichas superficies será 0, y el flujo no podrá atravesarlas.

En las fronteras abiertas hablaremos de dos tipos de condiciones de contorno, condición tipo nivel superficial o altura y condición tipo caudal de entrada. La condición tipo altura consiste en la asignación a un conjunto de celdas frontera del dominio un valor de  $h$  inicial. A partir de ese valor se resolverá el sistema para las celdas vecinas, definiendo todas las variables mencionadas mediante la resolución del sistema de ecuaciones. Lo mismo sucede con la condición tipo caudal cuando el dominio está mojado. No obstante, el tratamiento es algo distinto en el caso de que el dominio esté seco. En este último caso el caudal que se indica se integra en el tiempo, obteniendo un volumen dado que se distribuye sobre las celdas involucradas en la condición de contorno, definiendo sobre ellas de nuevo una altura  $h$  resultante del volumen que a cada una le corresponde.



# Capítulo 3

## Descripción del estudio comparativo

---

El objetivo de este Trabajo fin de Grado es fundamentalmente el de realizar un estudio comparativo y un análisis de sensibilidad del modelo computacional atendiendo a tres factores, el tipo de malla, el paso de tiempo a través del número  $CFL$  y el tipo de implementación del modelo, para poder sacar conclusiones sobre la manera más correcta de tratar un problema de simulación de flujos en superficie libre.

### 3.1 Versiones del modelo computacional

El modelo computacional que se analiza posee actualmente tres implementaciones, una versión secuencial en CPU en un núcleo (CPU 1), otra versión CPU paralelizada utilizando 4 núcleos (CPU 4) y una versión GPU en un núcleo.

Por las características del modelo y el tipo de problemas para los que está destinado a ser utilizado, el tiempo de cálculo es una variable fundamental. Si el dominio simulado es muy grande en extensión, como podría ser en situaciones reales de simulación de cuencas hidrográficas o se requiere una precisión muy grande en los resultados, puede ser necesario realizar un mallado muy fino del mismo, dando lugar a un número de celdas muy elevado dentro de la malla.

Teniendo en cuenta las restricciones en el tamaño de  $\Delta t$  impuestas por la estabilidad numérica, para simular estos dominios comentados podría ser necesario emplear horas o incluso días. Dentro del procesador de un ordenador pueden existir varios núcleos, que están íntimamente relacionados con la velocidad o el tiempo que tarda un programa en ejecutarse o en llevar a cabo las tareas para las que se ha diseñado. Es importante valorar dentro del estudio comparativo el tiempo que emplea cada una de las implementaciones del modelo en simular cada caso, así como calcular y valorar el *Speed-up* o relación existente entre los tiempos obtenidos en cada versión.

La versión CPU 1 es la más simple y puede ser utilizada por cualquier usuario en un ordenador. En esta versión solamente se involucra en el cálculo de las propiedades de todas las celdas de la malla a un núcleo del procesador. De este modo recaen sobre él todas las operaciones que realiza el programa. El tiempo de cálculo puede llegar a ser elevado y resultar problemático si el caso estudiado es muy exigente en número de celdas o tiempo real de simulación.

La implementación en CPU 4 está desarrollada para un procesador con al menos 4 núcleos. En este caso son 4 los núcleos que trabajan paralelamente para llevar a cabo la simulación, consiguiendo en teoría realizar el proceso de cálculo 4 veces más rápido. Este tipo de paralelización

en memoria compartida está implementada utilizando directivas OpenMP, que es una interfaz de programación de aplicaciones para este tipo de programación multiproceso. Permite añadir concurrencia a los programas escritos en C, C++ y Fortran sobre la base del modelo de ejecución y se compone de un conjunto de directivas de compilador, rutinas de biblioteca, y variables de entorno que influyen en el comportamiento en tiempo de ejecución. Es un modelo de programación portable y escalable que proporciona una interfaz simple y flexible para el desarrollo de aplicaciones paralelas para las plataformas, que van desde las computadoras de escritorio hasta las supercomputadoras. De este modo, gracias a esta manera de desarrollar el programa se consigue que el proceso de simulación sea sustancialmente más rápido [3].

La última de las versiones del programa ha sido implementada para simular en GPU (*Graphic Processing Units* o Unidades de procesamiento gráfico). Esta nueva tecnología está siendo cada vez más utilizada en la aceleración de software de cálculo intensivo requerido en distintos campos de la ingeniería, medicina o biología. Cabe destacar que el ordenador más rápido del mundo en la actualidad basa su capacidad de cómputo en este tipo de dispositivos [1]. Esto se traduce en un rendimiento de cálculo muy elevado, ya que se trasladan a una GPU las partes de la aplicación con mayor carga computacional, que son simuladas de forma muy eficiente en sus miles de núcleos. Estos núcleos son más pequeños que los de la CPU, pero están diseñados para llevar a cabo tareas de forma paralela, mientras que en una CPU se diseñan para llevarlas a cabo de manera secuencial. Por las características de este tipo de sistema de procesado de datos, la simulación se realiza más rápidamente que en otro tipo de sistemas basados únicamente en CPU [4].

## 3.2 Tipos de mallas y CFL

El tipo de malla utilizado para discretizar el dominio es otro aspecto fundamental dentro del estudio comparativo y del análisis de sensibilidad. Existen muchas maneras de crear las celdas sobre las que realizar los cálculos de los parámetros, pero en este estudio se tendrán en cuenta tres tipos: las mallas Triangulares Delaunay (TD), las mallas Triangulares Equiláteras (TE) y las mallas Rectangulares (R). Todos los casos se simularán utilizando los tres tipos de malla.

En las mallas TD los elementos que se emplean para discretizar el dominio son triángulos no equiláteros, de modo que se genera una malla no estructurada. La condición de Delaunay dice que una red de triángulos es una triangulación de Delaunay si la circunferencia circunscrita de cada triángulo de la red no contiene otros vértices aparte de los tres que definen el triángulo [2]. Esta definición original para espacios bidimensionales es posible ampliarla para espacios tridimensionales usando la esfera en vez de la circunferencia circunscrita. Esta condición asegura que los ángulos del interior de los triángulos son lo más grandes posibles, la longitud de los lados de los triángulos es mínima y la triangulación formada es única. Por la estructura de este tipo de malla es posible emplear un  $CFL = 0,9$  para conseguir estabilidad en el método de cálculo (2.18).

En las mallas TE los triángulos en los que se discretiza el dominio son equiláteros. Esto hace que la malla generada sea estructurada, lo que puede influir en la orientación del flujo a través de las caras del triángulo. Al utilizar este tipo de malla es preciso emplear un  $CFL = 0,5$ . A pesar de ello, en este estudio se analizan los resultados para este tipo de mallas con dos  $CFL$  distintos, uno de 0,45 con el fin de asegurar la estabilidad del método de cálculo, y otro de 0,9 para poder analizar las diferencias con los resultados aportados por mallas TD. Esta variación del  $CFL$  no solamente influye en la estabilidad del método de cálculo, sino también en el tiempo de simulación. Al disminuirlo aumenta el paso de tiempo, por lo que es previsible que los casos

simulados con malla TE con  $CFL = 0,9$  necesiten menos tiempo de simulación que los simulados con  $CFL = 0,45$ .

La diferencia fundamental que existe al emplear mallas R es que el dominio está discretizado por rectángulos, generándose mallas estructuradas. En este caso el  $CFL$  a emplear es 0,45, ya que con 0,9 es imposible alcanzar estabilidad en la solución. Este tipo de malla, por la estructura que lleva implícita, genera unas direcciones preferentes de flujo, de modo que puede dar lugar a errores en casos de simulación de flujos con componentes de velocidad o aceleración en más de una dirección.

### 3.3 Variables de análisis

#### 3.3.1 Eficiencia del modelo computacional

Para realizar la evaluación de la eficiencia la variable que se emplea es el tiempo de simulación, que sirve para comparar la rapidez con la que el modelo computacional calcula las propiedades de cada celda de la malla definida. Este tiempo de simulación dependerá tanto del tipo de malla con la que se discretiza el dominio (TD, TE, R) como del parámetro  $CFL$ , que está íntimamente relacionado con el paso de tiempo, y cuya influencia se observa en la comparación de los resultados obtenidos con mallas TE con  $CFL$  0,45 y 0,9.

Para cuantificar la aceleración aportada por las estrategias de implementación en CPU 4 y en GPU frente a la versión CPU 1 se define un parámetro denominado *Speed-up*,  $S_{up}$ .

$$S_{up} = \frac{t_{1core}}{t} \quad (3.1)$$

Donde  $t_{1core}$  hace referencia al tiempo o coste de simulación de la versión en CPU 1 y  $t$  al coste de simulación de la implementación empleada, de modo que el *Speed-up* de la implementación en CPU 1 es siempre 1, ya que relaciona su coste de simulación consigo misma.

Para el análisis de resultados se realizarán las simulaciones siguientes con las versiones en CPU 1 , CPU 4 y GPU:

- Malla TD y  $CFL = 0,9$
- Malla TE y  $CFL = 0,45$
- Malla TE y  $CFL = 0,9$
- Malla R y  $CFL = 0,45$

### 3.3.2 Precisión de los resultados

Para la validación de los modelos computacionales en términos de precisión, la *Environment Agency* [9] propone analizar la evolución temporal del parámetro nivel superficial,  $h + z$ , en varios puntos del dominio. Esto es lo que se hace en el estudio, colocando sondas de medida en los puntos en los que interesa representar gráficamente la evolución temporal de  $h + z$ . Estas sondas vuelcan en un fichero de salida los valores de todas las variables que calcula el modelo para cada celda. Las variaciones temporales de nivel superficial obtenidas tras la simulación de cada uno de los casos se representan superpuestas en una sola gráfica para cada punto especificado en el documento empleado como referencia [9].

Por otro lado, el documento en el que se basa el estudio propone el análisis del módulo de la velocidad, y no de las componentes por separado, por lo que será este parámetro el que se analice y compare, representado por  $modU$ . Las mismas sondas colocadas para el análisis del nivel superficial también contienen información referente al vector velocidad. El análisis de este valor se realiza comparando gráficamente las evoluciones temporales obtenidas en simulación.

En resumen, para poder corroborar la validez del método numérico en términos de precisión de cálculo se analizan dos variables, la altura del nivel superficial de agua  $h + z$  y el módulo de la velocidad  $modU$ .



# Capítulo 4

## Resultados

---

A continuación se van a presentar los distintos casos estudiados en este trabajo. Se incluye una descripción del caso, las condiciones de simulación, la topografía y los puntos del dominio estudiados. Estas descripciones se basan en las indicaciones dadas por la *Environment Agency* para la validación de modelos de simulación de inundación [9]. También se analizan los resultados obtenidos en cada caso así como la sensibilidad al tipo de malla, los tiempos de simulación y la precisión de los resultados mediante la evolución temporal de las variables estudiadas. Una vez expuestos los resultados se podrán sacar conclusiones sobre el estudio comparativo y el análisis de sensibilidad.

### 4.1 Caso 1

#### 4.1.1 Descripción del caso

El objetivo fundamental de este caso es evaluar cómo responde el programa de simulación a fenómenos básicos de inundación y movimiento de flujos de tipo unidireccional y la resolución de mojado y secado de zonas con diferente elevación [9].

El dominio estudiado en este caso es un rectángulo de  $700\text{ m}$  de longitud (eje  $x$ ) y  $100\text{ m}$  de anchura (eje  $y$ ). Dentro de este dominio interesa estudiar la altura del nivel superficial del flujo en dos puntos concretos que se reflejan en la Tabla 4.1. Esta información referida al dominio podemos verla en la Figura 4.1.

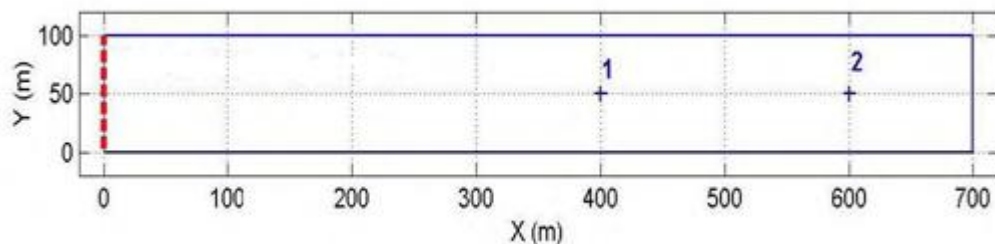
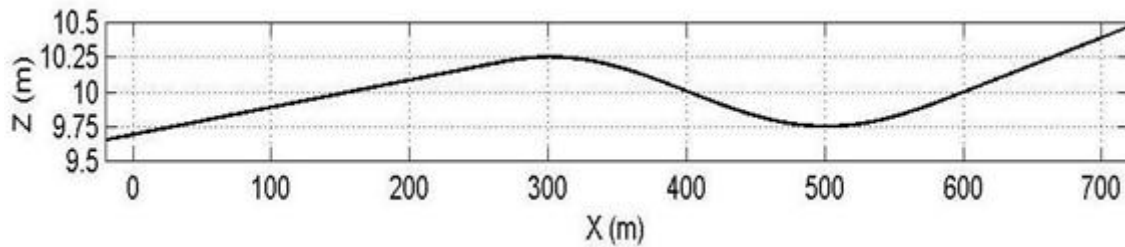


Figura 4.1: Dominio Caso 1

PUNTO	x(m)	y(m)
1	400	50
2	600	50

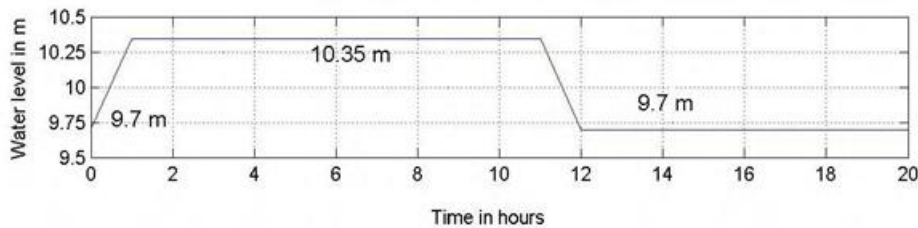
**Tabla 4.1:** Puntos a estudiar Caso 1

El perfil topográfico de este caso en dirección  $x$  comienza en la zona de coordenada  $x = 0\text{ m}$  con una cota  $z = 9,7\text{ m}$ , para llegar con una pendiente ascendente a cota  $z = 10,25\text{ m}$  en  $x = 300\text{ m}$ . Una vez alcanzada dicha cota se produce una depresión que alcanza su mínimo en  $z = 9,75\text{ m}$  para  $x = 500\text{ m}$ , aumentando luego hasta un valor  $z = 10,4\text{ m}$  en la frontera del dominio en  $x = 700\text{ m}$ . Este dominio se representa en la Figura 4.2.



**Figura 4.2:** Perfil Caso 1

En este caso se aplican condiciones de contorno de impenetrabilidad en todas las paredes del dominio excepto en  $x = 0\text{ m}$ . Sobre las celdas que se encuentran en esta línea se aplica una condición de contorno de nivel superficial de agua. Esta condición parte de una altura  $h + z = 9,7\text{ m}$  para elevar el nivel de agua en esa zona hasta una altura  $h + z = 10,35\text{ m}$ , manteniéndolo durante  $t = 10\text{ h}$ , y volviendo luego a su nivel original  $h + z = 9,7\text{ m}$ . Esta evolución temporal de los valores del nivel superficial de agua introducida como condición de contorno se ve en la Figura 4.3.



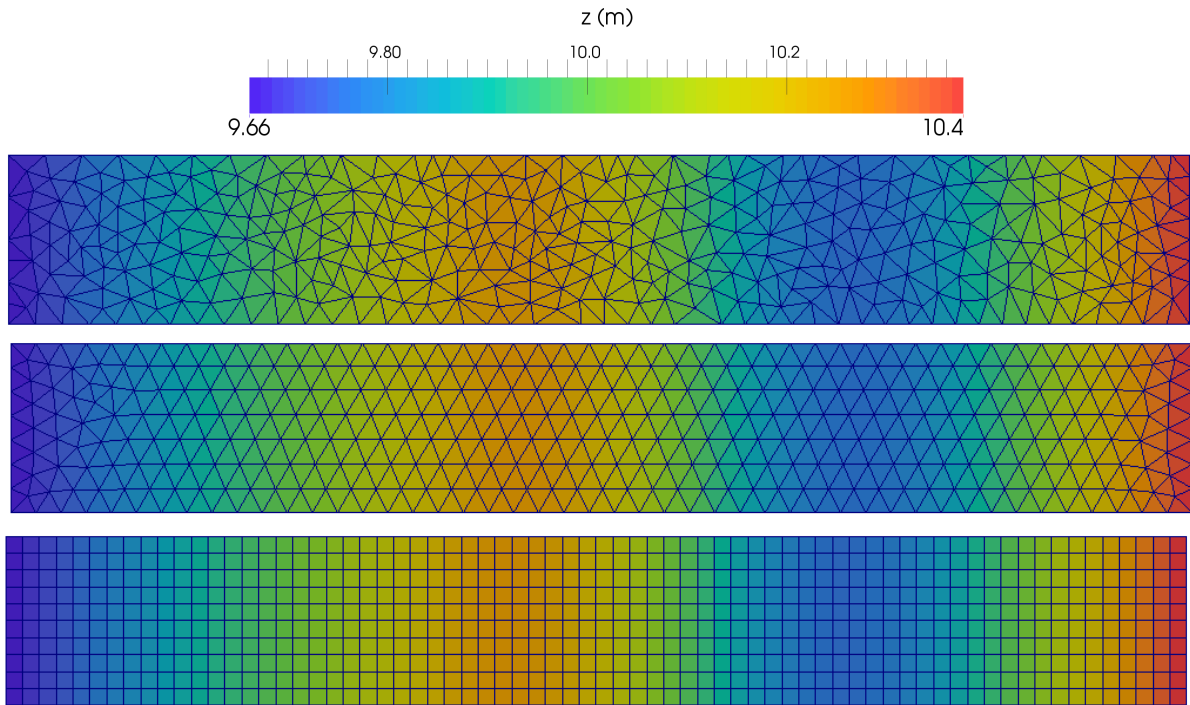
**Figura 4.3:** Condición de contorno Caso 1

Otros parámetros relativos a la descripción del caso y necesarios para poder llevar a cabo la simulación fijando unas condiciones comunes son:

- El coeficiente de Manning es uniforme;  $n = 0,03$ .
- La malla que se emplee ha de tener alrededor de 700 celdas.
- El tiempo de simulación es de  $t = 20\text{ h}$ .
- La forma de evaluar el modelo será realizando una representación gráfica de la altura del nivel superficial del agua frente al tiempo en cada uno de los dos puntos fijados.

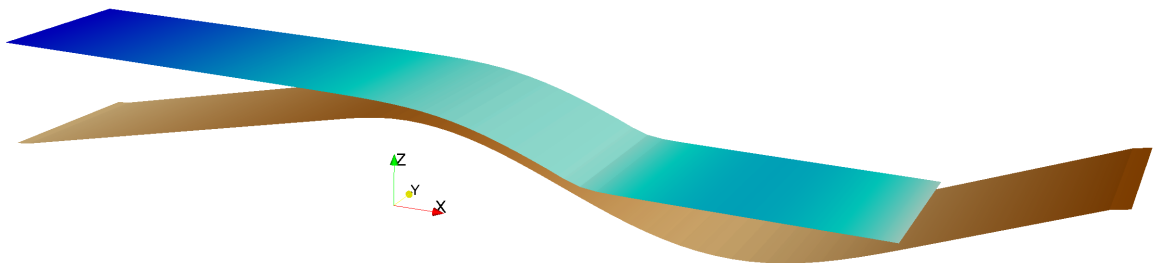
### 4.1.2 Análisis de resultados

Las mallas generadas según las discretizaciones Delaunay (TD), Equilátera (TE) y Rectangular (R) aparecen en la Figura 4.4.



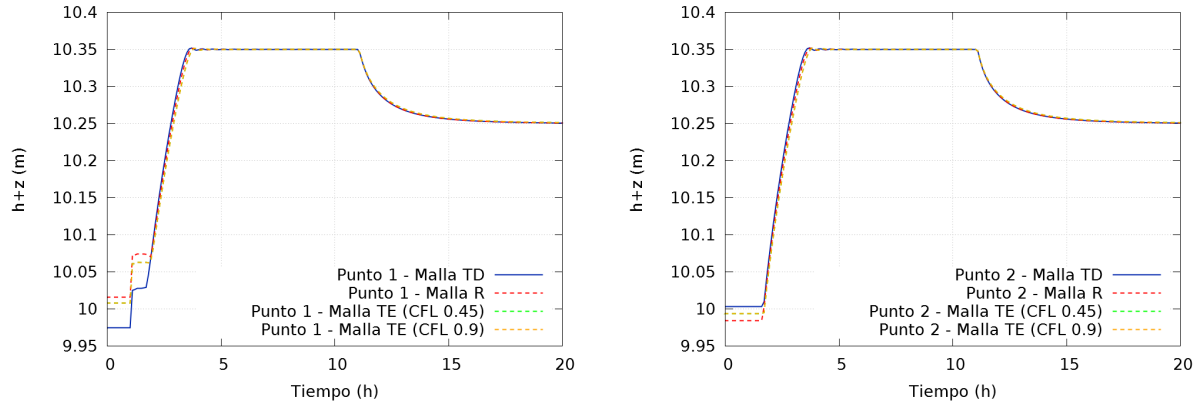
**Figura 4.4:** Mallas Caso 1. De arriba a abajo, TD, TE, R

En la Figura 4.5 se puede ver el dominio resultante de aplicar las condiciones de topografía ( $z$ ) en 3 dimensiones para la malla R. También se aprecia el el valor  $h + z$  para el instante  $t = 1,7 h$ .



**Figura 4.5:** Vista 3D Caso 1

Los resultados referidos al valor del nivel superficial del agua ( $h + z$ ) en los dos puntos especificados en la Tabla 4.1 pueden verse en la Figura 4.6



**Figura 4.6:** Evolución temporal de  $h + z$  para el punto 1 (izquierda) y el punto 2 (derecha). Caso 1

Se puede observar que apenas existe diferencia entre los resultados aportados para esta variable con los distintos tipos de mallas.

Los tiempos de simulación empleados para este caso se incluyen en la Tabla 4.2.

	Celdas Total	Tiempo (s)			Speed-up		
		CPU		GPU	CPU		GPU
		CPU 1	CPU 4		CPU 1	CPU 4	
TD	700	13,09	4,14	16,04	1	3,16	0,82
TE CFL 0,45	670	14,61	4,68	19,17	1	3,12	0,76
TE CFL 0,9	670	7,53	2,43	9,88	1	3,10	0,76
R	700	7,59	2,49	10,68	1	3,05	0,71

**Tabla 4.2:** Tiempos de simulación Caso 1

En términos de tiempo total, se comprueba que en este caso es más costoso simular empleando GPU que CPU, debido al bajo número de celdas que tiene el dominio. En lo que se refiere a la CPU, la versión CPU 4 es más rápida que la versión simple CPU 1.

Por otro lado, dentro de las mallas TE se ve que el tiempo de simulación con  $CFL = 0,45$  es mayor que el empleado con  $CFL = 0,9$ . Además, el tiempo de la malla R (simulada siempre con  $CFL = 0,45$ ) se aproxima más al de la malla TE con  $CFL = 0,9$ , mientras que el tiempo de simulación de la malla TD ( $CFL = 0,9$ ) es similar al de la malla TE con  $CFL = 0,45$ .

Sobre la tendencia del *Speed-up*, se puede ver que la simulación en CPU 4 es siempre alrededor de 3 veces más rápida que en la versión CPU 1, mientras que la simulación en GPU es entre 0,7 y 0,8 veces más lenta.

## 4.2 Caso 2

### 4.2.1 Descripción del caso

Este caso ha sido diseñado para evaluar la capacidad del programa de simular procesos extensos de inundación bidireccional, determinando su evolución temporal y los valores de altura superficial de agua (en este caso profundidades), teniendo en cuenta una topografía compleja [9].

El área modelada para el estudio es un cuadrado de 2.000  $m$  de lado. Este dominio contiene una matriz 4 x 4 de depresiones que se encuentran equidistantes entre ellas, y que tienen forma idéntica, con transiciones muy suaves entre unas y otras y obtenidas multiplicando sinusoides en dirección Norte-Sur y Oeste-Este. De este modo, los puntos cuyo estudio resultará relevante e interesante serán los 16 correspondientes a menor cota  $z$  de cada una de estas depresiones. Además, existe una ligera pendiente 1:1.500 en dirección Norte-Sur y otra 1:3.000 en dirección Este-Oeste, de modo que hay una caída de 2  $m$  de elevación desde la esquina noroeste hasta la esquina sureste. Este dominio explicado se ve en la Figura 4.7.

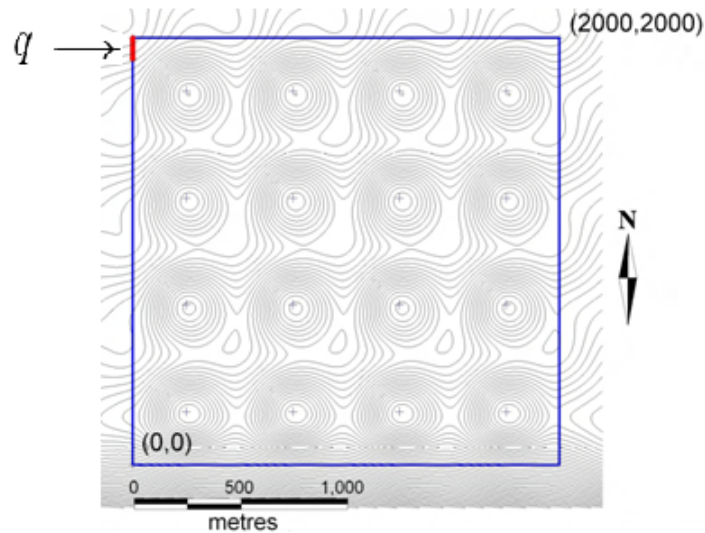


Figura 4.7: Dominio Caso 2

Por otro lado, en lo que a las condiciones de contorno se refiere, se establecen condiciones de impenetrabilidad en todas las fronteras del dominio, salvo en una línea (coloreada en rojo) donde se impone un caudal con dirección de Este-Oeste que se puede ver en la Figura 4.7. El hidrograma de la Figura 4.8 contiene la evolución temporal de este caudal aplicado, que alcanza un valor pico de 20  $m^3/s$  a los 10  $min$  y se prolonga hasta  $t = 85 min$ .

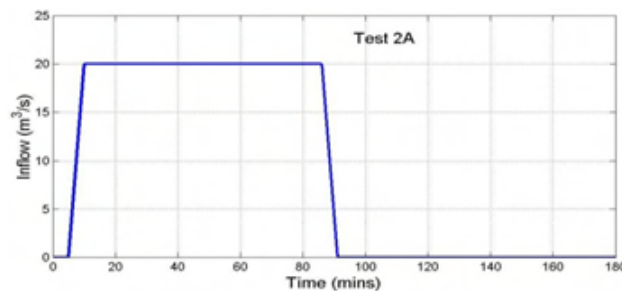


Figura 4.8: Condición de contorno Caso 2

Otros parámetros relativos a la descripción del caso y necesarios para poder llevar a cabo la simulación fijando unas condiciones comunes son:

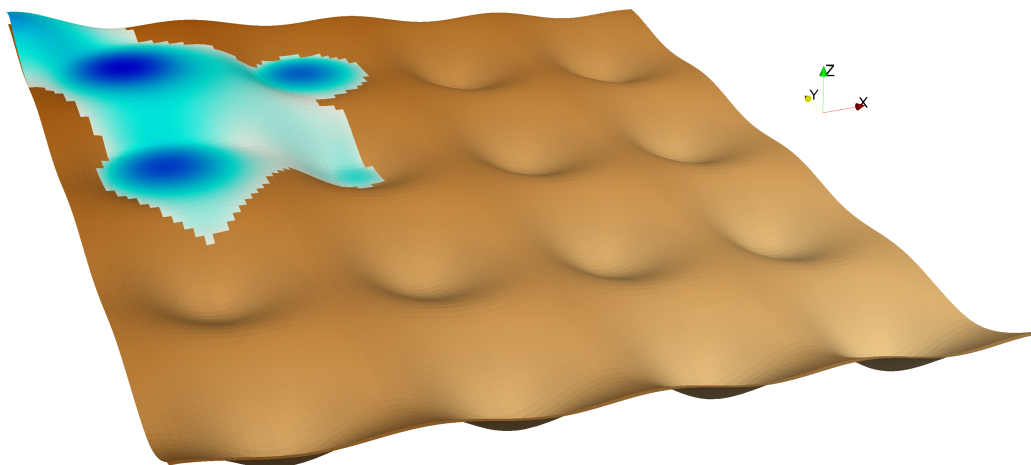
- El coeficiente de Manning es uniforme;  $n = 0,03$ .
- La malla que se emplee ha de tener alrededor de 10.000 celdas.
- El tiempo de simulación es de  $t = 48 h$ .
- La forma de evaluar el modelo será realizando una representación gráfica de la altura del nivel superficial del agua frente al tiempo en cada uno de los puntos fijados, que aparecen en la Tabla 4.3.

PUNTO	x(m)	y(m)	PUNTO	x(m)	y(m)
1	250	250	9	1.250	250
2	250	750	10	1.250	750
3	250	1.250	11	1.250	1.250
4	250	1.750	12	1.250	1.750
5	750	250	13	1.750	250
6	750	750	14	1.750	750
7	750	1.250	15	1.750	1.250
8	750	1.750	16	1.750	1.750

**Tabla 4.3:** Puntos a estudiar Caso 2

#### 4.2.2 Análisis de resultados

En la Figura 4.9 se puede observar una vista del dominio en 3 dimensiones generada con una malla R para el instante de simulación  $t = 1,44 h$  de manera que se puede observar el flujo de agua conforme va llenando las cavidades. Las mallas TD, TE y R generadas para este dominio aparecen en la Figura 4.10.



**Figura 4.9:** Vista 3D Caso 2

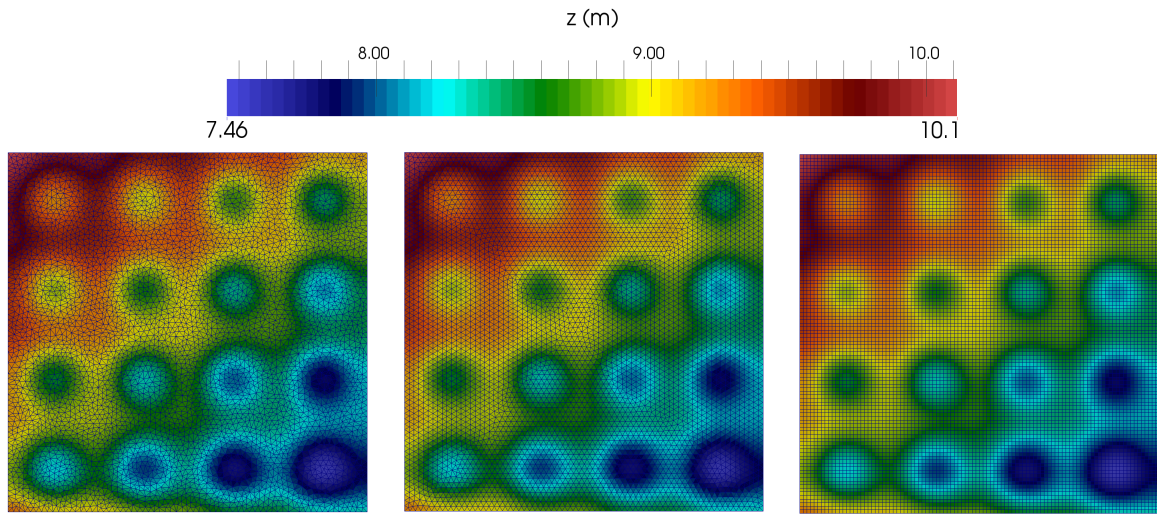


Figura 4.10: Mallas Caso 2. De izquierda a derecha, TD, TE, R

Los resultados requeridos en este caso son las representaciones de la evolución temporal del parámetro  $h + z$  en cada uno de los puntos de la Tabla 4.3, que coinciden con las 16 depresiones que existen en el dominio. Estas evoluciones temporales se muestran en las Figuras 4.11 , 4.12 y 4.13.

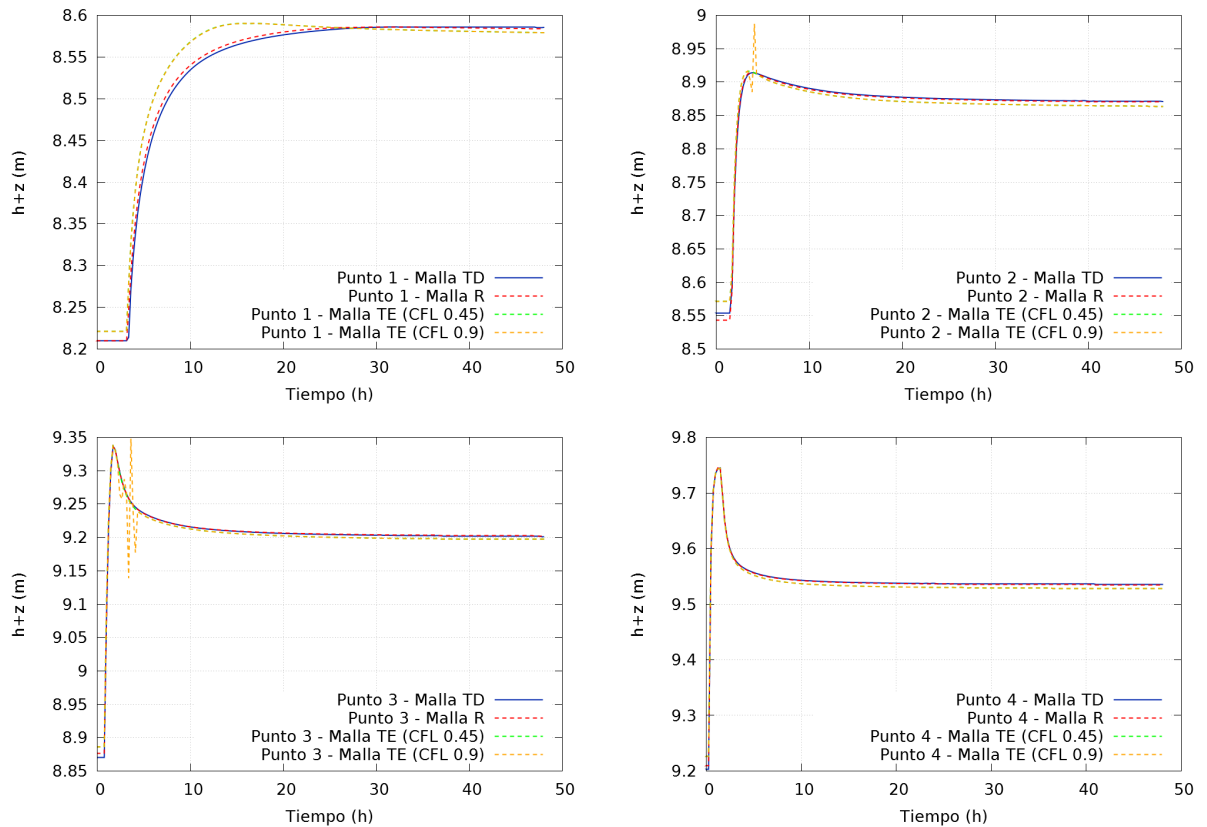
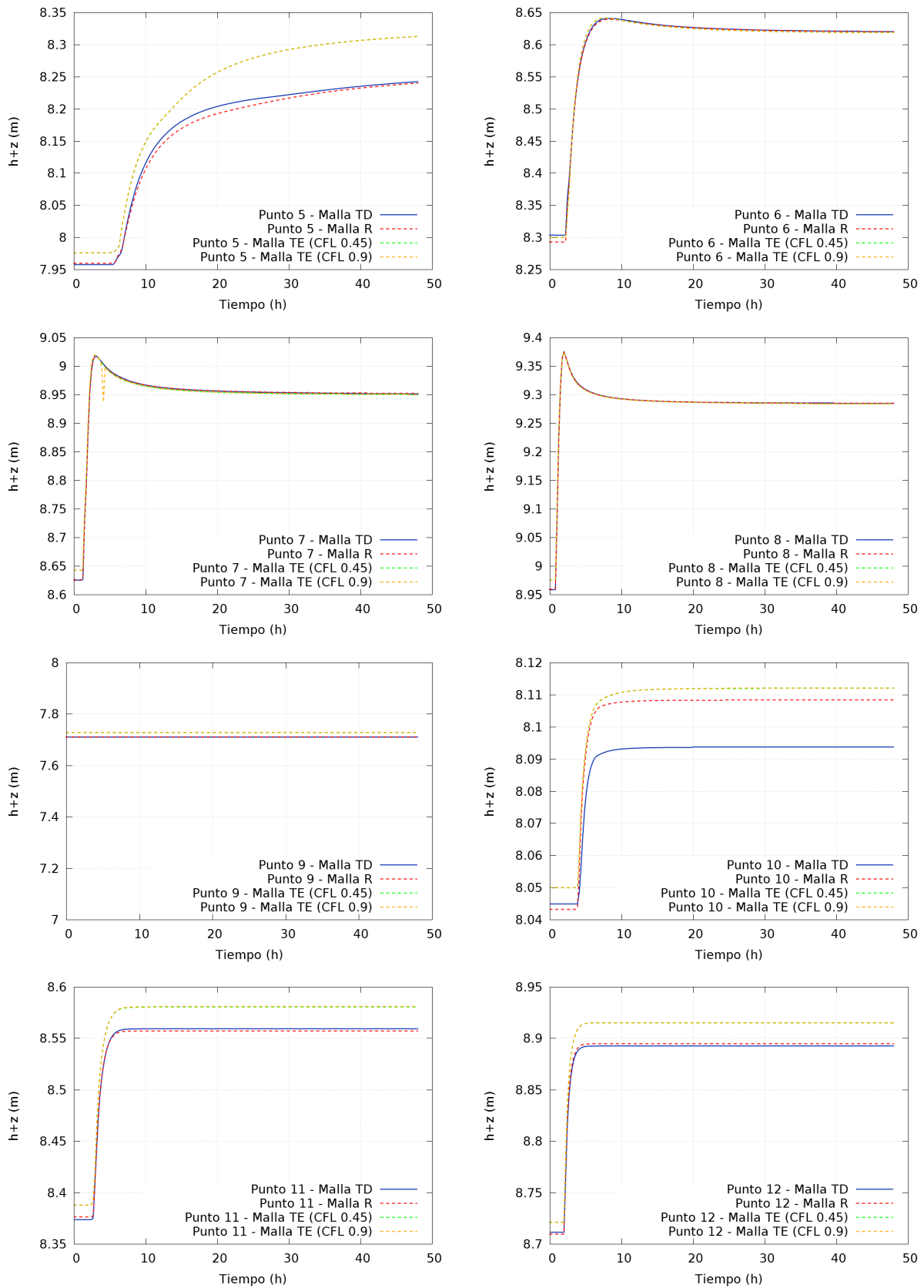
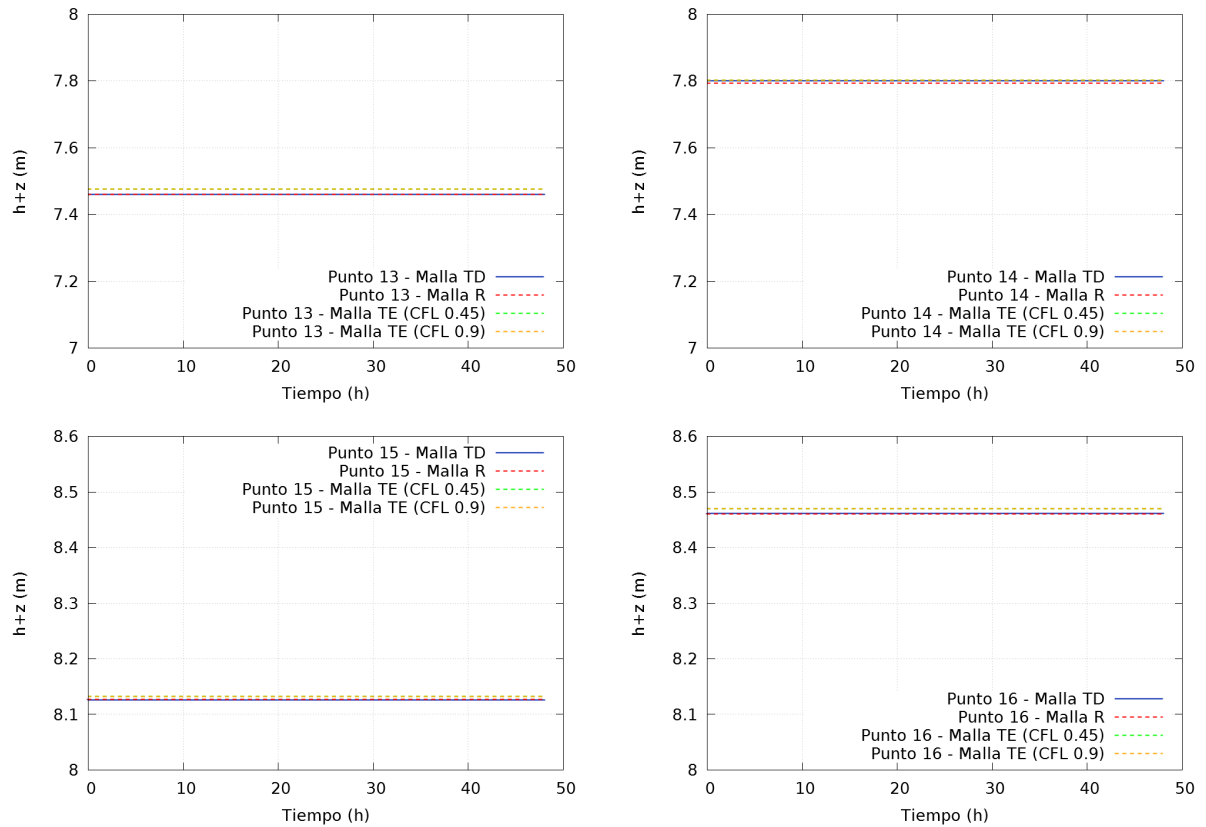


Figura 4.11: Evolución temporal de  $h + z$  de arriba-izquierda a abajo-derecha para los puntos 1, 2, 3 y 4. Caso 2



**Figura 4.12:** Evolución temporal de  $h+z$  recorriendo las gráficas de arriba a abajo y de izquierda a derecha para los puntos 5, 6, 7, 8, 9, 10, 11 y 12. Caso 2





**Figura 4.13:** Evolución temporal de  $h+z$  de arriba-izquierda a abajo-derecha para los puntos 13, 14, 15 y 16. Caso 2

Aunque las diferencias en valor absoluto son muy pequeñas (fijarse en las escalas que aparecen en el eje  $y$ ), aparecen desviaciones en cuanto a la estimación del valor alcanzado en permanente por el nivel superficial del agua. Esto puede observarse para los puntos 5, 10, 11 y 12 (Figura 4.12). En los resultados referidos a los puntos 2, 3 (Figura 4.11) y 7 (Figura 4.12), se observa que la malla TE con  $CFL = 0,9$  ofrece problemas de estabilidad en el cálculo del nivel superficial del agua.

Los tiempos de simulación empleados en este caso se incluyen en la Tabla 4.4.

	Celdas Total	Tiempo (s)			Speed-up		
		CPU		GPU	CPU		GPU
		CPU 1	CPU 4		CPU 1	CPU 4	
TD	9.989	86,99	27,99	19,03	1	3,11	4,57
TE CFL 0,45	10.054	161,56	54,54	33,07	1	2,96	4,89
TE CFL 0,9	10.054	80,98	27,21	16,76	1	2,98	4,83
R	10.000	65,61	20,45	15,30	1	3,21	4,29

**Tabla 4.4:** Tiempos de simulación Caso 2

Los tiempos empleados para la simulación en GPU en este caso son inferiores a los necesarios en CPU, ya que se empiezan a incluir más celdas en el proceso de cálculo. Se puede apreciar que para este caso la diferencia más significativa se encuentra entre la simulación en CPU 1 y CPU

4, y que la disminución del tiempo de simulación entre CPU 4 y GPU no es tan grande.

Analizando los resultados atendiendo al tipo de malla empleado, se puede comprobar que la malla R es la que menos tiempo emplea en la simulación, mientras que la malla TE con  $CFL = 0,45$  la que más necesita. Por otro lado, los tiempos de simulación de la malla TD y la malla TE de  $CFL = 0,9$  son muy similares aunque, como se ha visto, los resultados aportados por esta última sufren desviaciones puntuales considerables.

En cuanto a los distintos valores del *Speed-up*, se observa que la tendencia es que la versión CPU 4 sea alrededor de 3 veces más rápida que la versión CPU 1, y que además, para este caso (a diferencia del anterior) la versión en GPU es entre 4,3 y 4,9 veces más rápida que la versión CPU 1.

### 4.3 Caso 3

#### 4.3.1 Descripción del caso

El objetivo fundamental de este caso es evaluar la capacidad de simular correctamente el fenómeno de propagación de una onda de inundación bidireccional y simétrica, así como velocidades transitorias del flujo y su nivel superficial en distintas partes del dominio, especialmente en el frente de avance de la onda de inundación [9].

En este caso se parte de un dominio rectangular de  $1.000\text{ m}$  (eje  $x$ ) por  $2.000\text{ m}$  (eje  $y$ ). Además, la superficie de este dominio se considera horizontal. Es decir, no existen elevaciones ( $z = 0\text{ m}$ ) ni pendientes en el mismo. Este dominio se representa en la Figura 4.14, donde también se pueden ver los puntos en los que se tendrá que evaluar tanto el módulo de la velocidad del flujo,  $modU$ , como el parámetro  $h + z$ , cuyas coordenadas se muestran en la Tabla 4.5 así como los frentes de propagación esperados (con semicircunferencias) debido a la simetría del problema.

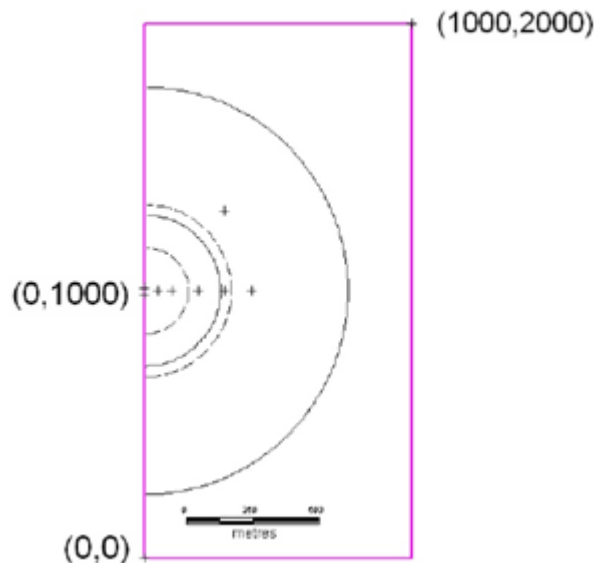
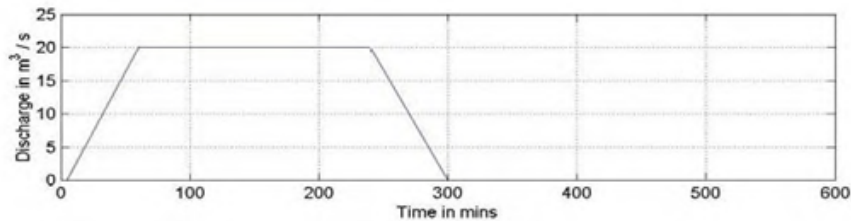


Figura 4.14: Dominio Caso 3

PUNTO	x(m)	y(m)
1	50	1.000
2	100	1.000
3	200	1.000
4	300	1.000
5	400	1.000
6	300	1.300

**Tabla 4.5:** Puntos a estudiar Caso 3

Las condiciones de contorno que se fijan en las fronteras del dominio son de impenetrabilidad salvo sobre una línea de 20 m de longitud situada en el centro de la pared del dominio con  $x = 0 m$ , como puede verse en la Figura 4.14. Esta condición de contorno es de tipo caudal en dirección  $x$ , y simula el fallo de un dique, ya sea por rotura o desbordamiento, con un pico de  $20 m^3/s$ , y siendo necesaria la simulación durante un periodo de  $t = 5 h$ . El hidrograma aparece representado en la Figura 4.15.



**Figura 4.15:** Condición de contorno Caso 3

De nuevo, existen unos parámetros fijos para realizar la simulación bajo unas condiciones dadas:

- El coeficiente de Manning es uniforme;  $n = 0,05$ .
- La malla que se emplee ha de tener alrededor de 80.000 celdas.
- El tiempo de simulación es de  $t = 5 h$ .

En este caso, la *Environment Agency* propone una evaluación diferente del modelo computacional. Por un lado será necesario representar gráficamente la evolución temporal tanto de  $h + z$  como de  $modU$  en cada uno de los puntos que figuran en la Tabla 4.5. Además, también deberán evaluarse estos dos parámetros a lo largo del eje de simetría del dominio en dirección  $x$  (coordenada  $y = 1.000 m$ ) para instantes de tiempo de  $t = 30 min$ ,  $t = 1 h$ ,  $t = 2 h$ ,  $t = 3 h$  y  $t = 4 h$ .

### 4.3.2 Análisis de resultados

La Figura 4.16 muestra una imagen 3D de la forma de propagación del frente de onda de inundación sobre el plano, generada con una malla R, y en este caso para  $t = 1 h$ .

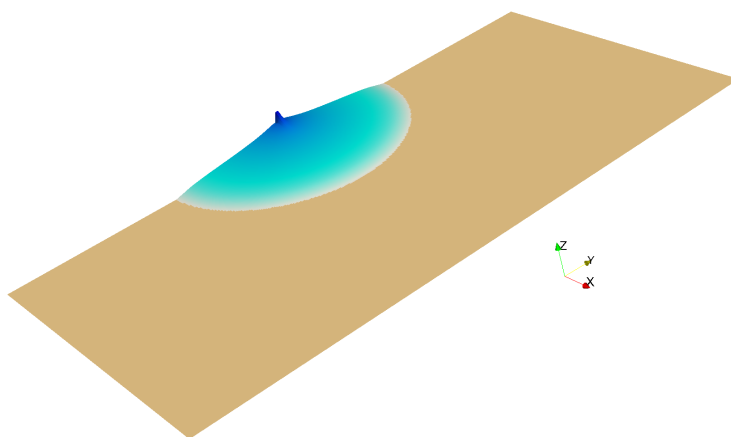


Figura 4.16: Vista 3D Caso 3

Se representa en la Figura 4.17 un detalle de la esquina superior izquierda del dominio con las tres mallas utilizadas. Este detalle tiene unas dimensiones en el dominio de  $100\text{ m} \times 100\text{ m}$ , lo que da una idea de la cantidad de celdas involucradas en el proceso de cálculo de este caso.

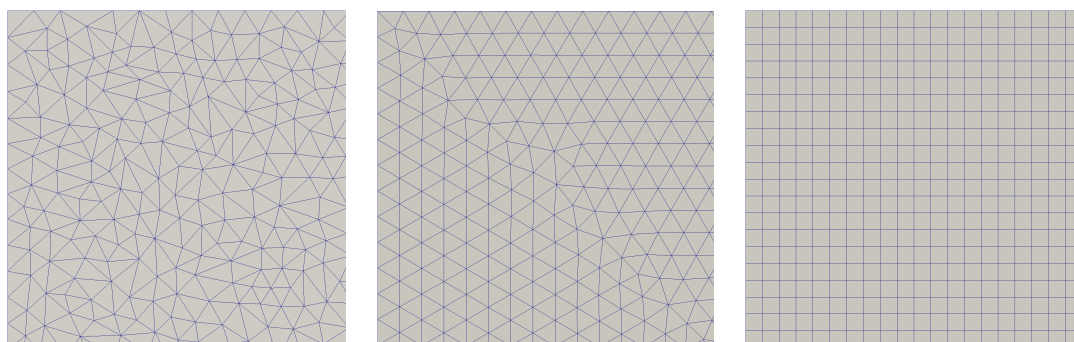


Figura 4.17: Mallas Caso 3. De izquierda a derecha, TD, TE, R

La evolución temporal de  $h + z$  y de  $modU$  en cada uno de los puntos se muestra en las Figuras 4.18, 4.19 y 4.20.

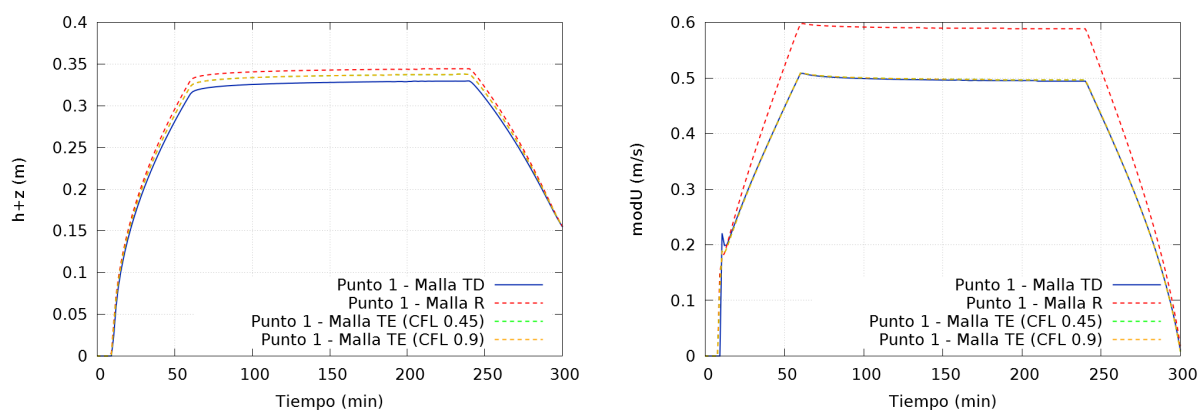


Figura 4.18: Evolución temporal de  $h + z$  (izquierda) y  $modU$  (derecha) para el punto 1. Caso 3

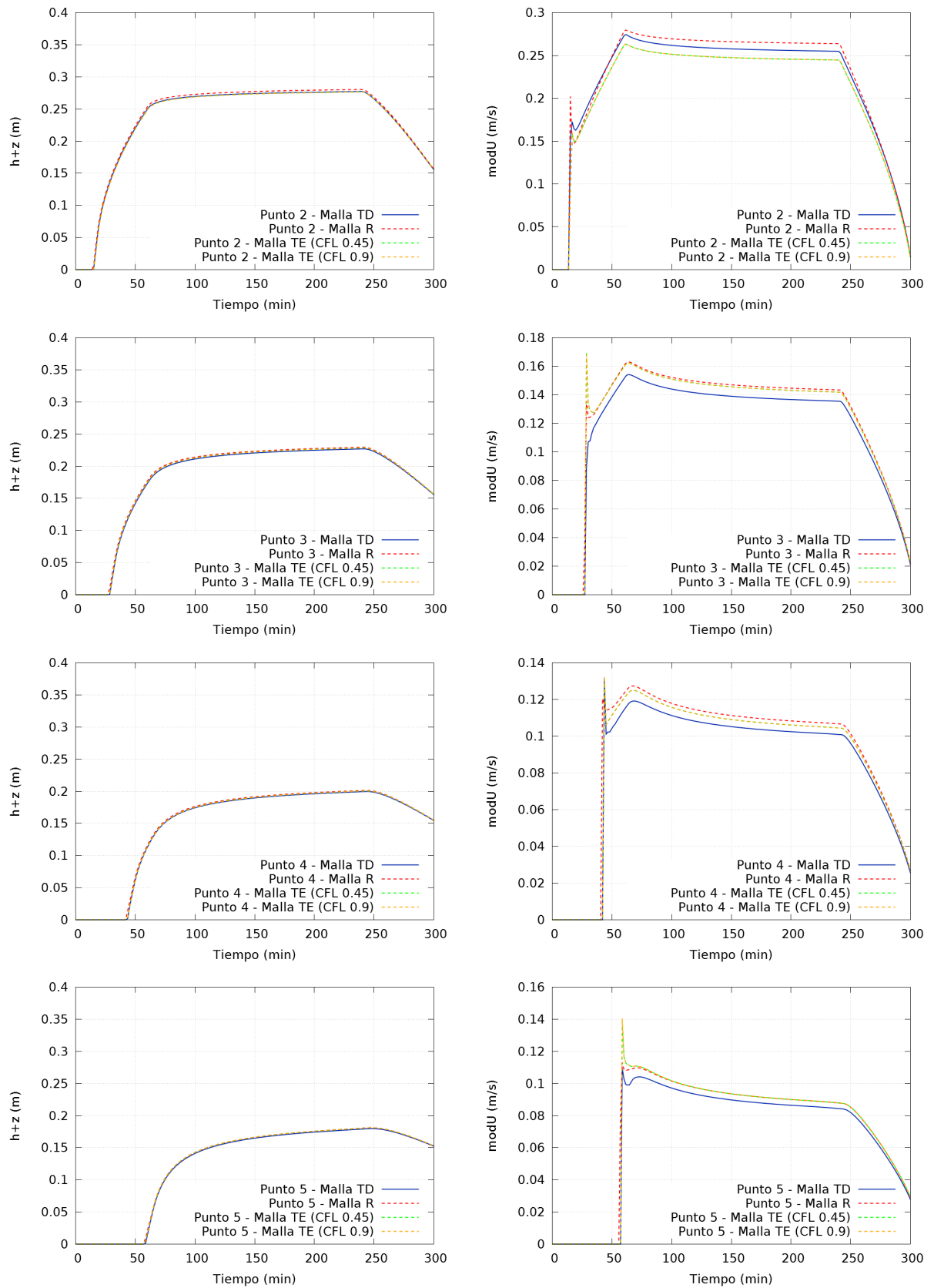


Figura 4.19: Evolución temporal de  $h+z$  (columna izquierda) y  $modU$  (columna derecha) para los puntos 2, 3, 4 y 5. Caso 3

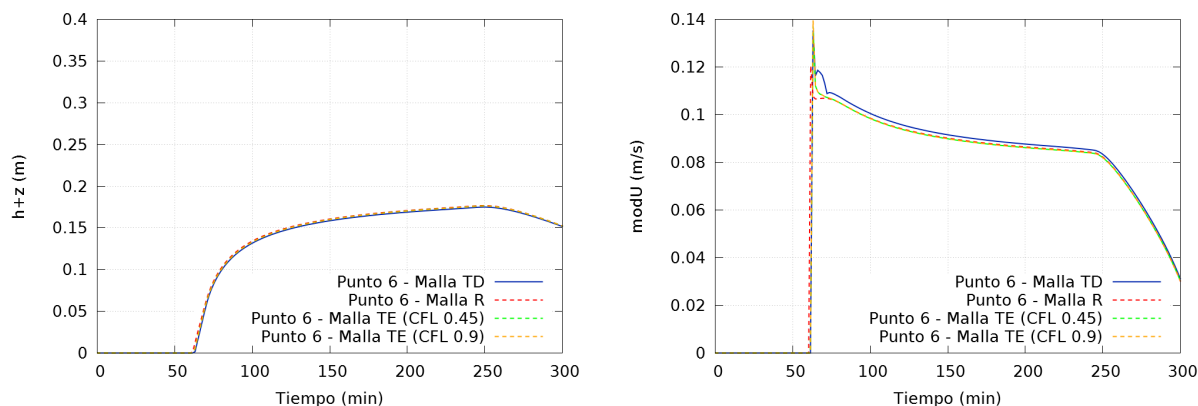


Figura 4.20: Evolución temporal de  $h + z$  (izquierda) y  $modU$  (derecha) para el punto 6. Caso 3

Es necesario representar también la situación tanto del nivel superficial como del módulo de la velocidad a lo largo de la línea que atraviesa transversalmente el dominio desde el punto  $(0, 1.000)$  hasta el punto  $(1.000, 1.000)$ . Se pueden ver estas representaciones en las Figuras 4.21 y 4.22.

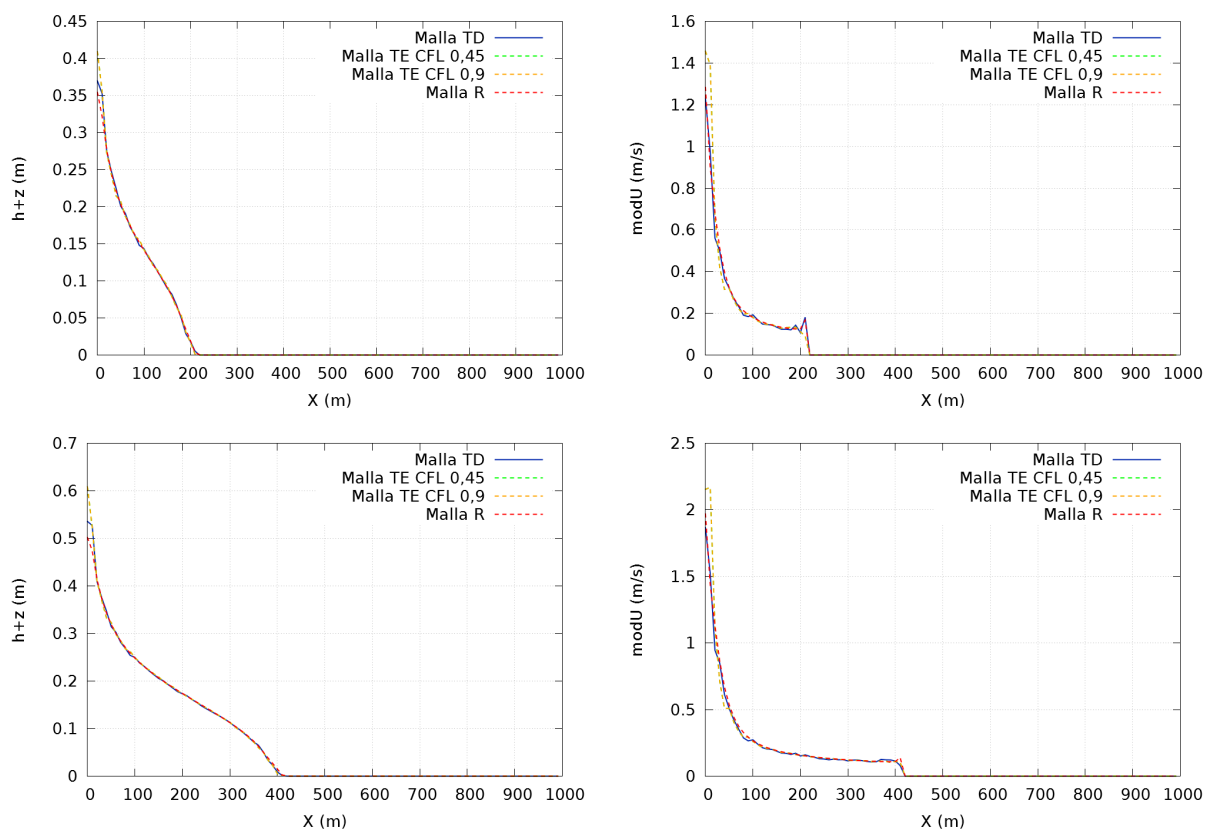
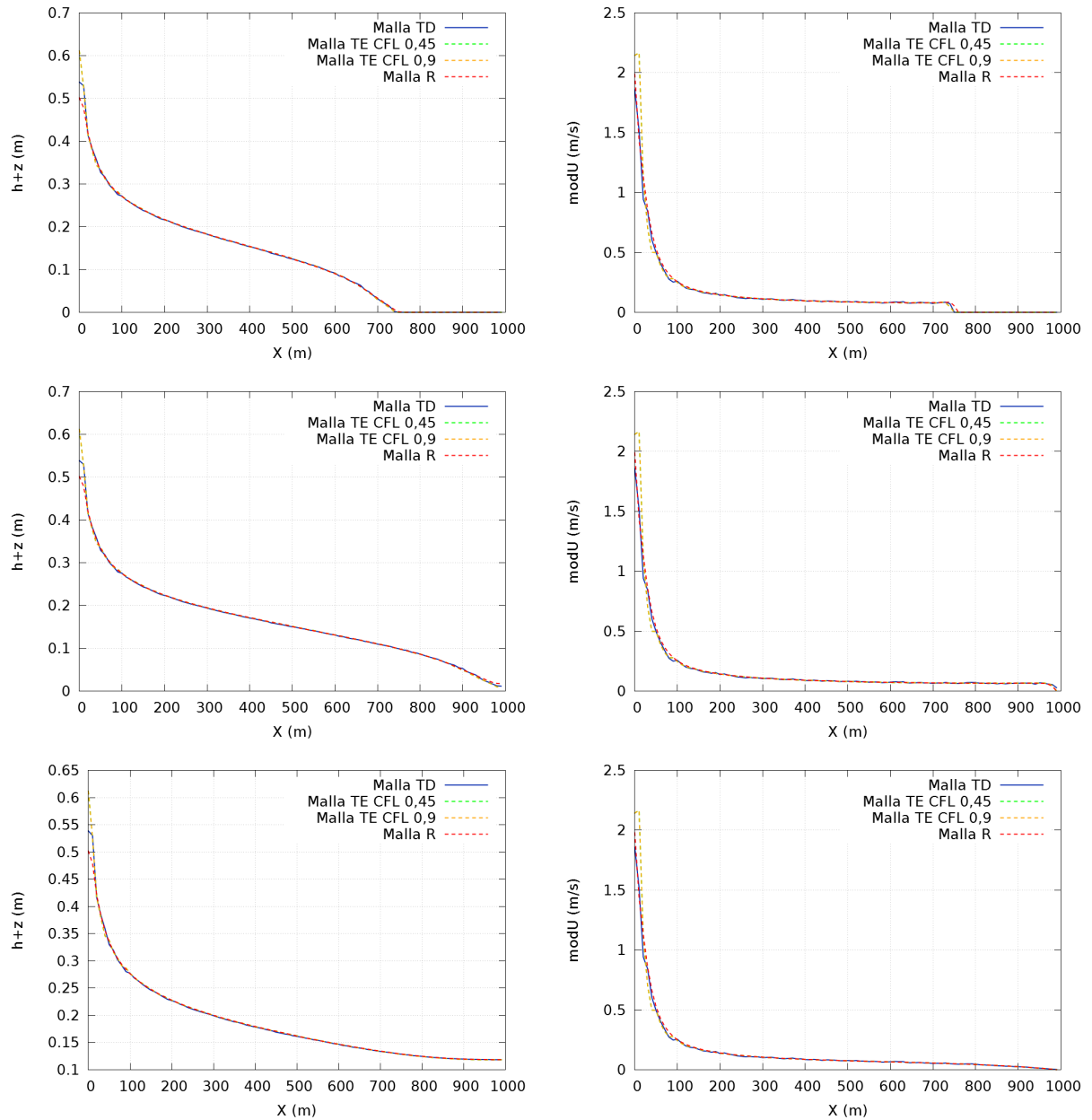


Figura 4.21: Evolución de  $h + z$  (columna izquierda) y  $modU$  (columna derecha) a lo largo de  $x$  ( $y = 1.000$ ) en  $t = 0,5 h$  (fila 1) y  $t = 1 h$  (fila 2). Caso 3.



**Figura 4.22:** Evolución de  $h+z$  (columna izquierda) y  $modU$  (columna derecha) a lo largo de  $x$  ( $y = 1.000\text{ m}$ ) en  $t = 2\text{ h}$  (fila 1),  $t = 3\text{ h}$  (fila 2) y  $t = 4\text{ h}$  (fila 3). Caso 3

En cuanto a las gráficas que representan la evolución temporal del nivel superficial ( $h+z$ ), merece la pena destacar que los resultados aportados por las simulaciones realizadas con los distintos tipos de mallas apenas difieren, encontrando las mayores diferencias en el punto 1 (Figura 4.18).

En cuanto a la evolución temporal del módulo de la velocidad ( $modU$ ), se puede comprobar que esta variable es mucho más sensible al tipo de malla empleado. Tanto la malla R como la malla TE, al ser estructuradas, introducen unas direcciones preferentes a través de las cuales se orienta el flujo, generando datos que se alejan del valor aportado por la malla TD.

Al evaluar la representación espacial de  $h + z$  y  $modU$  se puede ver que todas las mallas siguen un patrón muy similar independientemente del instante temporal escogido. Otro detalle importante es que en este caso la diferencia entre los resultados obtenidos con mallas TE para  $CFL = 0,9$  y  $CFL = 0,45$  son apenas apreciables, tanto en la representación de  $h + z$  como de  $modU$ .

Los tiempos de simulación se reflejan en la Tabla 4.6, donde también se aprecia que el número de celdas del dominio es mucho mayor que en los dos casos anteriores.

	Celdas Total	Tiempo (s)			Speed-up		
		CPU		GPU	CPU		GPU
		CPU 1	CPU 4		CPU 1	CPU 4	
TD	79.936	1.411,16	527,99	48,48	1	2,67	29,11
TE CFL 0,45	79.822	1.537,93	690,80	46,92	1	2,23	32,78
TE CFL 0,9	79.822	754,96	312,02	23,50	1	2,42	32,12
R	80.000	1.014,44	411,22	35,69	1	2,47	28,42

**Tabla 4.6:** Tiempos de simulación Caso 3

En tiempos totales invertidos para el cálculo se observa que el coste de simulación es mayor en la versión CPU 1 que en el caso paralelizado CPU 4. No obstante, donde se ve que la disminución del tiempo necesario es mucho más acusada es en el caso de la simulación en GPU. Esta relación de tiempos se traduce en valores de *Speed-up* muy elevados para la comparación entre CPU 1 y GPU, alcanzando cotas de aproximadamente 30. En cuanto al *Speed-up* entre CPU 1 y CPU 4, los valores se sitúan cercanos a 2,5, siendo menores que en los otros dos casos estudiados, donde eran cercanos o ligeramente superiores a 3.

Sobre el tipo de malla empleada, la malla TE con un  $CFL = 0,9$  es la que menos coste de simulación tiene, seguida por la R, la malla TD y por último la TE con  $CFL = 0,45$ .

## 4.4 Caso 4

### 4.4.1 Descripción del caso

Este caso intenta poner a prueba el modelo de simulación frente a condiciones diferentes que la mera aplicación de caudales o niveles superficiales en un contorno del dominio. Concretamente trata de evaluar la respuesta frente a una entrada tipo fuente (comienza a manar agua de un punto localizado en una zona interior del dominio) y a una entrada de tipo lluvia, que se aplica, mientras dura, a todas las celdas existentes [9].

Este dominio responde a un rectángulo de 960 m (eje  $x$ ) por 400 m (eje  $y$ ). Representa un área de la ciudad de Glasgow, existiendo dos zonas distintas (pavimento y calles y resto de zonas) con diferente coeficiente  $n$  de Manning, pero sin estar incluidos los edificios que pudieran existir para el proceso de simulación. Este dominio se representa en la Figura 4.23, donde también se pueden ver los puntos en los que se realiza el análisis. Teniendo en cuenta que la esquina inferior izquierda de este dominio no se corresponde con el origen, sino que se encuentra en una posición  $x = 263.977$  m;  $y = 664.409$  m, cobra sentido la Tabla 4.7, donde se muestran las coordenadas de dichos puntos de análisis.



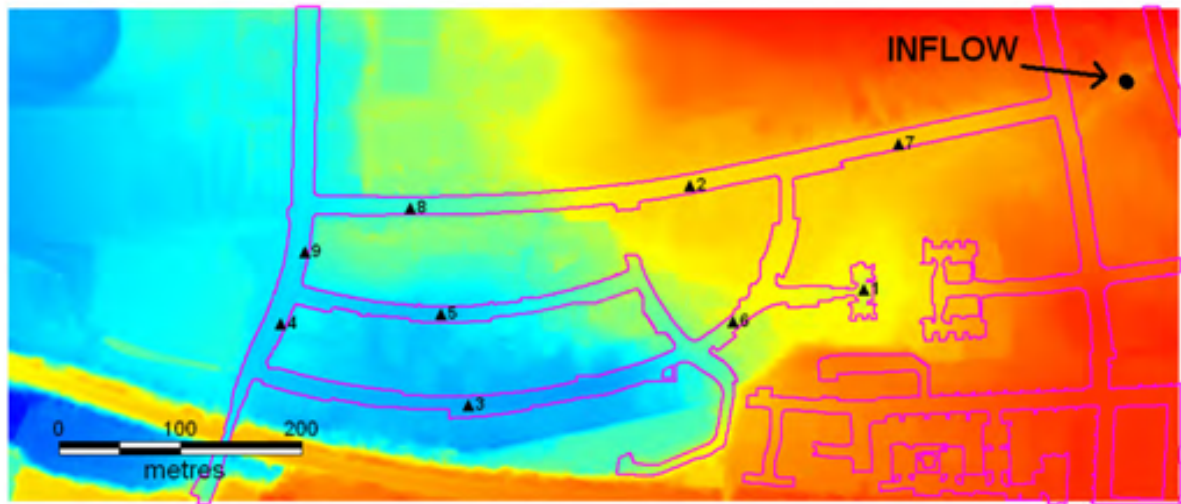


Figura 4.23: Dominio Caso 4

PUNTO	x(m)	y(m)
1	264.682	664.581
2	264.538	664.665
3	264.356	664.487
4	264.202	664.553
5	264.334	664.561
6	264.574	664.555
7	264.710	664.699
8	264.308	664.647
9	264.222	664.611

Tabla 4.7: Puntos a estudiar Caso 4

En este caso se imponen condiciones de impenetrabilidad en todas fronteras del dominio, y a diferencia de en el resto de casos, las aportaciones de agua no se definen como condiciones de contorno, sino en el interior del mismo en forma de términos fuente de masa. Estos términos así definidas son dos:

- **Condición tipo lluvia.** Un evento *lluvia* se distribuye de manera uniforme por todo el dominio estudiado. Su hidrograma aparece en la Figura 4.24.
- **Condición tipo fuente.** Un evento *fuentes* se incluye en el interior del dominio, en el punto que aparece en la Figura 4.23, de coordenadas  $x = 264.896 \text{ m}$ ,  $y = 664.747 \text{ m}$ . Este tipo de entrada, cuyo hidrograma se puede ver en la Figura 4.25 simula la entrada de un caudal determinado de agua en un punto del dominio, equivalente a suponer que de una alcantarilla en la ciudad comienza a manar agua.

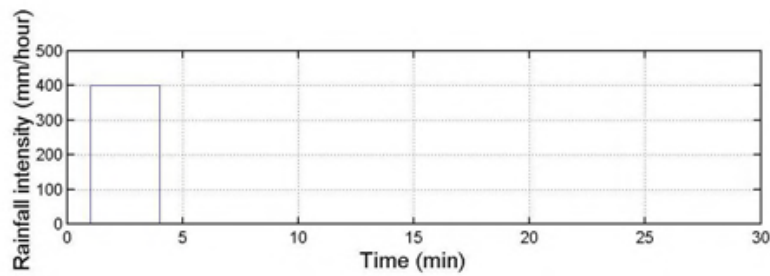


Figura 4.24: Condición de contorno tipo lluvia Caso 4

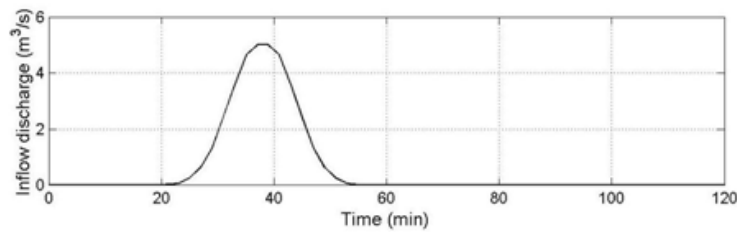


Figura 4.25: Condición de contorno tipo fuente Caso 4

Teniendo en cuenta el significado del coeficiente de Manning, que evalúa la fricción existente entre el flujo de agua y la superficie por la que discurre, resulta lógico pensar que una variación del terreno conlleva un cambio en este coeficiente. Por lo tanto se considera un Manning distinto para las zonas con pavimento ( $n = 0,02$ ) y para el resto del área del dominio ( $n = 0,05$ ).

Se especifica que la malla ha de poseer aproximadamente 96.000 celdas. Comparando las dimensiones del dominio con el número de celdas, este caso tiene la mayor relación de *celdas/superficie* de todas las situaciones estudiadas. Además, el tiempo durante el que tendrá que realizarse la simulación será de  $t = 5 h$ , ya que así se indica en el artículo utilizado como referencia [9].

Los resultados que tendrán que extraerse de la simulación serán:

- Representación del nivel superficial del agua ( $h + z$ ) en cada uno de los puntos considerados frente al tiempo.
- Representación del módulo de la velocidad ( $modU$ ) en cada uno de los puntos estudiados frente al tiempo.

En la Figura 4.23 se puede comprobar que todos los puntos del dominio en los que se insta a calcular el nivel superficial del agua y el módulo de la velocidad se encuentran en la zona pavimentada. Esto tiene sentido teniendo en cuenta las características de este caso, ya que dentro de una ciudad puede resultar más interesante saber con certeza qué es lo que ocurre en sus calles, calzadas o carreteras que en otros puntos. Por este motivo se plantea la posibilidad de estudiar con más detalle las zonas pavimentadas.

El método de mallado con mallas triangulares, ya sean TE o TD, permite definir distintos tamaños de celda en diferentes zonas del dominio. Si en una zona del dominio se definen celdas de menor tamaño existen entonces más puntos sobre los que el modelo computacional realizará el cálculo de las variables, por lo que la resolución aumenta y el error cometido en la predicción

puede disminuir. Esto hace que se plantee la posibilidad de emplear mallas con estas características para evaluar también este caso. Este hecho de variar el tamaño de la celda en ciertas zonas del dominio se conoce como refinamiento.

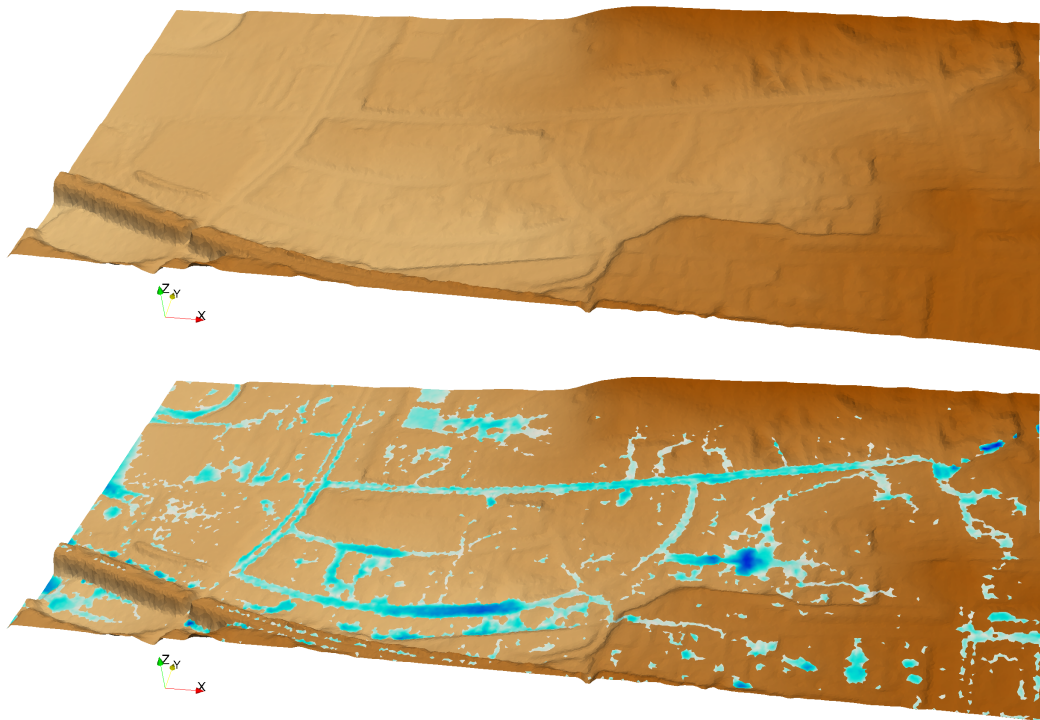
A la hora de refinar la malla se ha de tener en cuenta la limitación del número de celdas (96.000), ya que si se disminuye el tamaño de las mismas para un área determinada del dominio es necesario emplear un mayor número en la discretización. Por ello, si se refina en ciertas zonas, también se tiene que aumentar el tamaño de las celdas en otras que no sean de interés, aumentando la resolución en unos puntos pero perdiéndola en otros.

A la hora de analizar y probar los resultados también se presentan las gráficas de la evolución temporal del nivel superficial del agua y del módulo de la velocidad para las mallas refinadas TE y TD. En este análisis el resto de parámetros del caso no varían, y el tiempo de simulación, número de Manning, número de celdas aproximado y localización de las entradas son los mismos que en la malla sin refinar.

#### 4.4.2 Análisis de resultados

En el análisis de este caso se diferencian los resultados obtenidos para la malla no refinada y refinada.

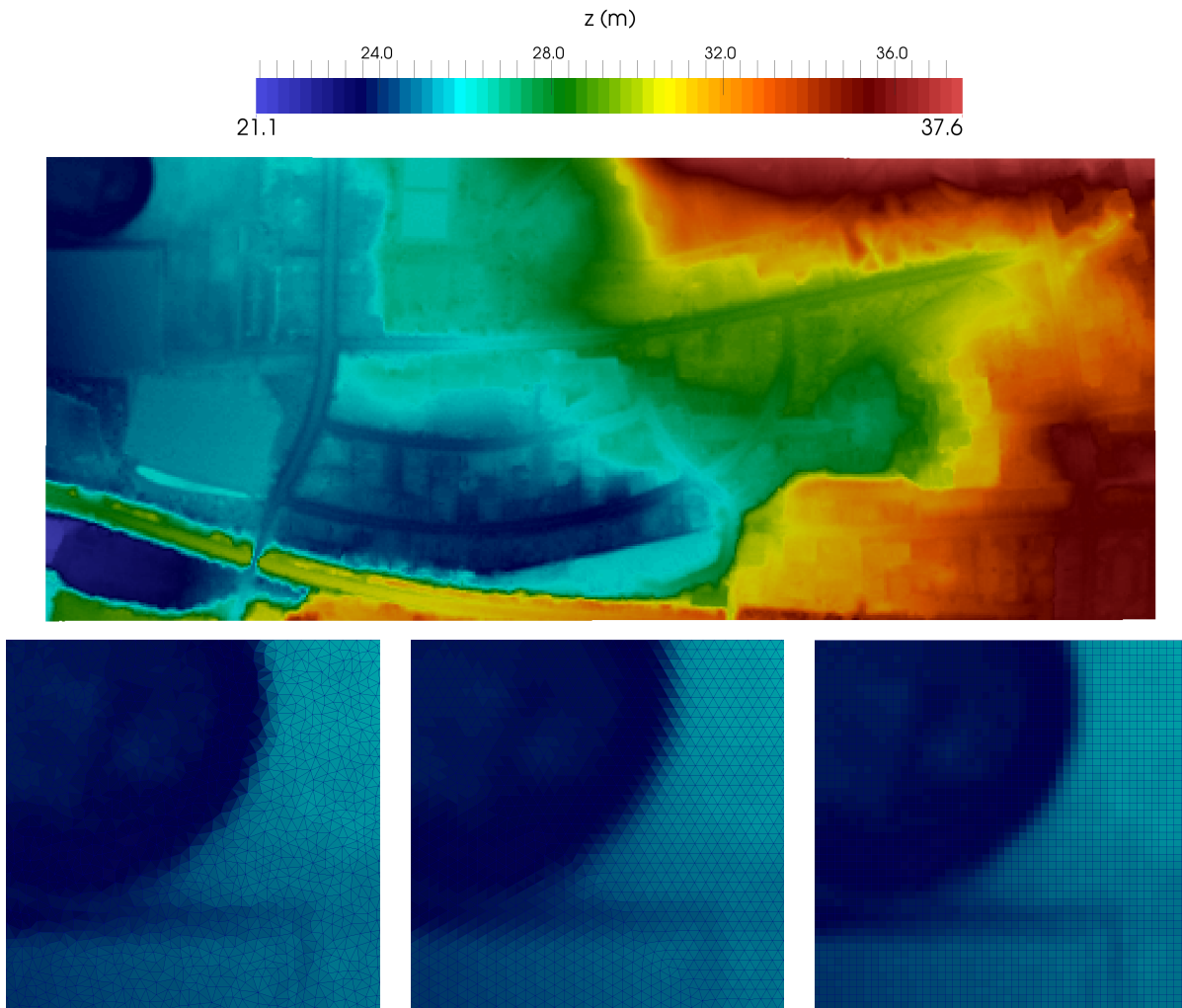
En la Figura 4.26 se muestra una vista en 3 dimensiones del dominio, mostrando la topografía del mismo, así como la situación del flujo de agua en las distintas zonas del terreno para un instante de simulación  $t = 9 \text{ min}$ .



**Figura 4.26:** Vista en 3D del Caso 4 antes de comenzar la simulación (imagen superior) y para  $t = 9 \text{ min}$  (imagen inferior)

### Mallas no refinadas

En el caso de la malla no refinada, la discretización obtenida para las diferentes tipologías de celda empleadas se puede ver en la Figura 4.27 , donde aparece el dominio a estudiar con la cota  $z$  de cada una de sus celdas según la escala de colores reflejada (parte superior), además de un detalle (zona cercana a la esquina superior izquierda) de las mallas TD, TE y R generadas.



**Figura 4.27:** Mallas Caso 4. Cota  $z$  (superior) y de izquierda a derecha mallas TD, TE y R (inferior)

El detalle mostrado tiene unas dimensiones de  $50\text{ m} \times 50\text{ m}$ , lo que da una idea de la densidad de celdas en este dominio.

Los resultados necesarios para evaluar este caso obtenidos con mallas no refinadas se muestran en las Figuras 4.28 , 4.29 y 4.30. Estos resultados son la evolución temporal del nivel superficial ( $h + z$ ) y del módulo de la velocidad ( $modU$ ) en los puntos que aparecen en la Tabla 4.7.

4.4. CASO 4

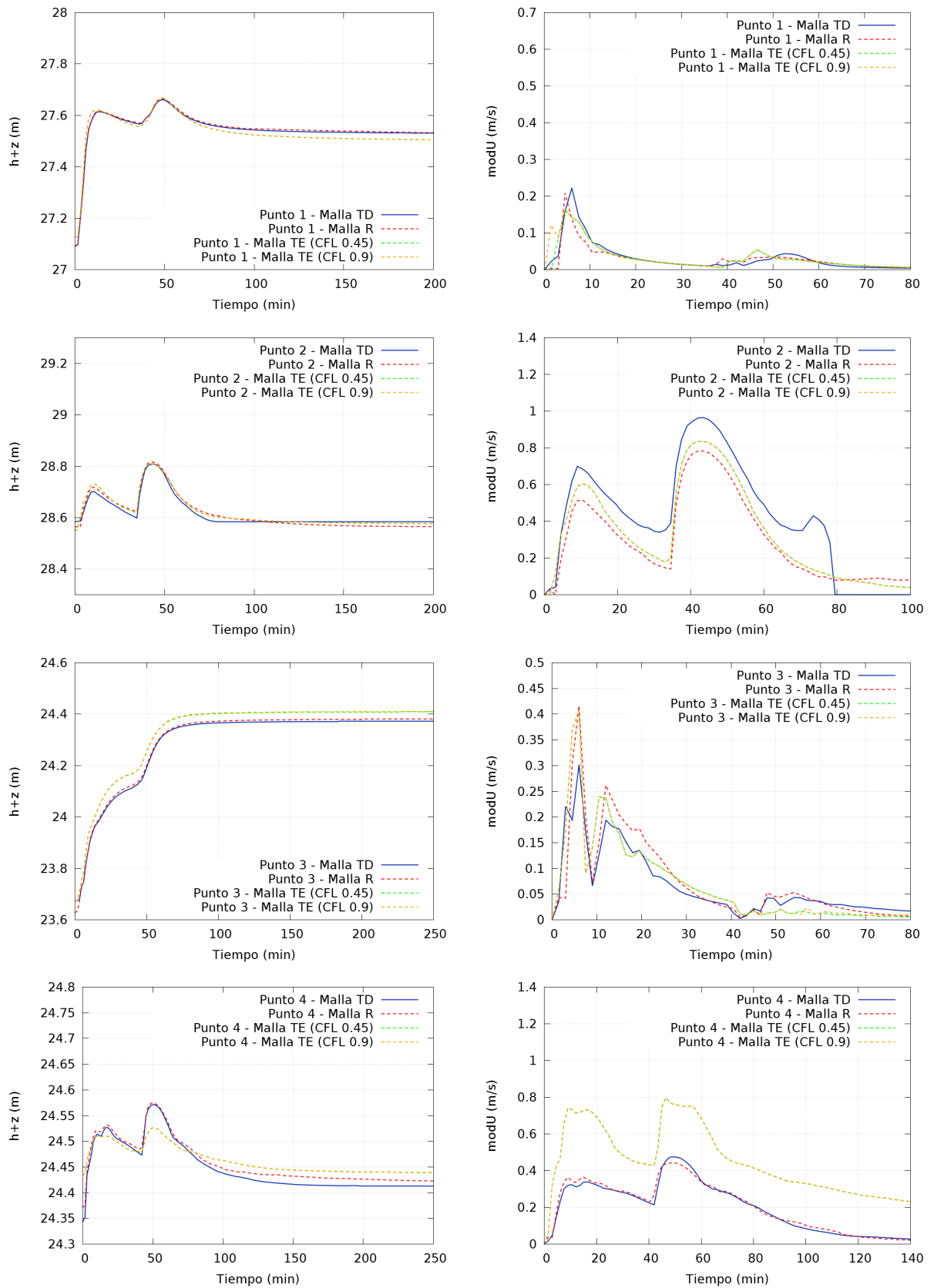


Figura 4.28: Evolución temporal de  $h+z$  (columna izquierda) y  $modU$  (columna derecha) para los puntos 1, 2, 3 y 4. Caso 4

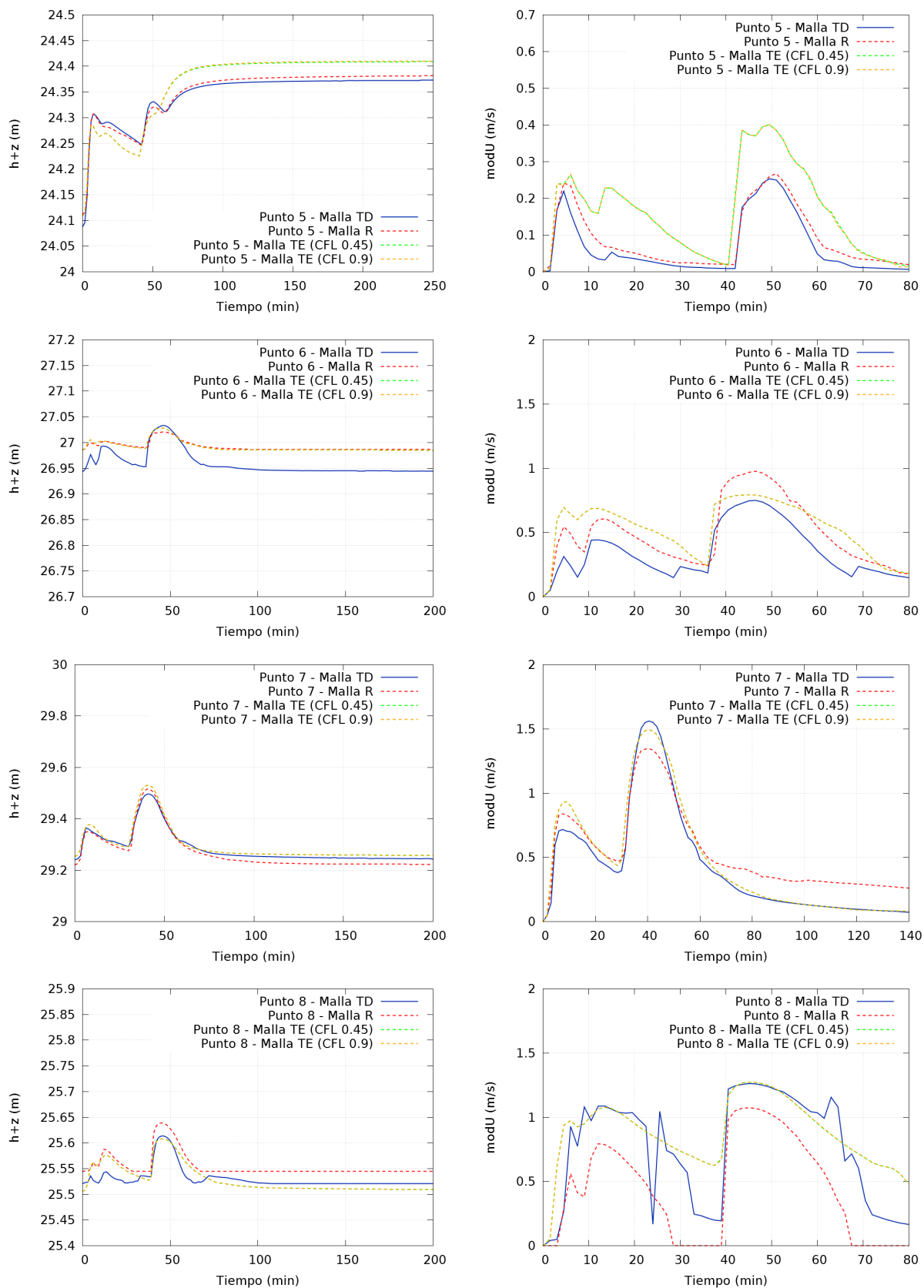
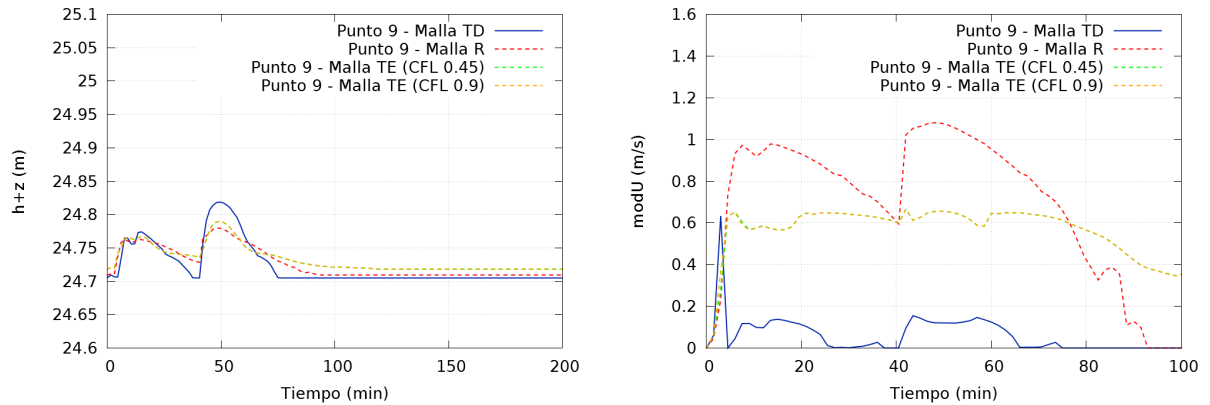


Figura 4.29: Evolución temporal de  $h+z$  (columna izquierda) y  $modU$  (columna derecha) para los puntos 5, 6, 7 y 8. Caso 4



**Figura 4.30:** Evolución temporal de  $h + z$  (izquierda) y  $modU$  (derecha) para el punto 9. Caso 4

Se puede ver que en este caso la variación entre los resultados obtenidos por los distintos tipos de mallas es mucho mayor que en el resto. Se aprecia que ahora las diferencias están presentes incluso en los valores de  $h + z$ , especialmente en los puntos 5 y 6 (Figura 4.29). En el resto de puntos evaluados el nivel superficial ofrece resultados similares entre unas mallas y otras.

Las representaciones de la evolución temporal del módulo de la velocidad son muy diferentes entre las mallas empleadas. Especialmente interesante es que no existe apenas diferencia entre las mallas TE de  $CFL = 0,45$  y las de  $CFL = 0,9$ . Entre las mallas R, TD y TE, las desviaciones son considerables, especialmente en los puntos 8 (Figura 4.29) y 9 (Figura 4.30), que ni siquiera siguen un patrón común. Los resultados de estos dos puntos 8 y 9 obtenidos con la malla TD tienen además la peculiaridad de presentar cambios muy bruscos de valor, y experimentar picos muy acusados, por lo que se podría dudar sobre la veracidad de los mismos.

Los tiempos de simulación empleados aparecen en la Tabla 4.8.

	Celdas Total	Tiempo (s)			Speed-up		
		CPU		GPU	CPU		GPU
		CPU 1	CPU 4		CPU 1	CPU 4	
TD	96.016	4.899,81	2.873,90	84,74	1	1,70	57,82
TE CFL 0,45	96.334	4.564,71	2.641,23	72,35	1	1,73	63,09
TE CFL 0,9	96.334	2.792,81	1.526,96	39,94	1	1,83	69,93
R	96.000	3.414,95	2.081,52	57,87	1	1,64	59,01

**Tabla 4.8:** Tiempos de simulación Caso 4. Mallas no refinadas.

Los tiempos globales de simulación de este caso son mucho mayores que los de cualquiera de los anteriores, y para simular un periodo real de  $t = 5 h$  se sitúan en valores cercanos y superiores a  $t = 1 h$  en la versión CPU 1. Los tiempos en la versión paralelizada CPU 4, a pesar de ser algo menores, se encuentran en valores cercanos o superiores a los  $t = 30 min$ . La implementación del programa en GPU realiza todo el cálculo en aproximadamente  $t = 1 min$ , que supone una velocidad de simulación mucho mayor que las otras versiones.

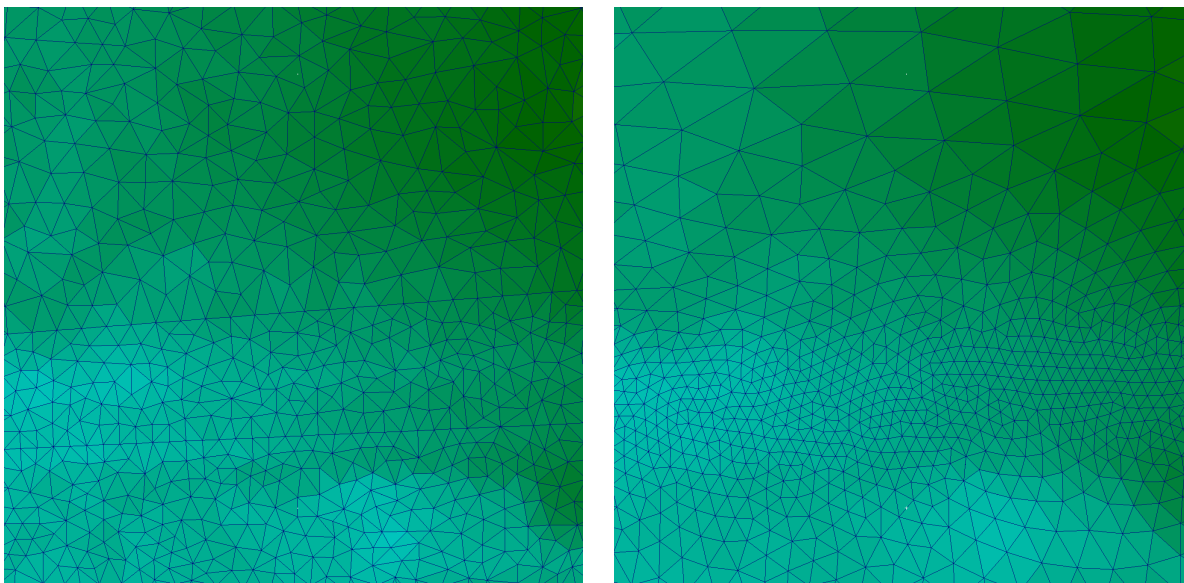
Estos tiempos tan elevados de simulación pueden ser debidos a las condiciones de este caso, sobre todo a la condición tipo lluvia, ya que obliga al modelo computacional a realizar cálculos

en todas las celdas del dominio en cada instante de tiempo.

Los valores de *Speed-up* son también distintos a los de los anteriores. La relación entre la versión de CPU 1 y CPU 4 ya no es cercana a 3, sino que apenas se reduce el tiempo de cálculo en un factor 2. Puede deberse de nuevo a la condición tipo lluvia introducida. Por otro lado, el *Speed-up* de la versión GPU alcanza valores entre 57 y 70, lo que supone una aceleración muy grande del proceso de cálculo.

### Mallas refinadas

Una vez analizados los resultados con mallas no refinadas se pasa al análisis de las TD y TE refinadas. Se ha de tener en cuenta que las condiciones del caso son las mismas y que el dominio y su representación en 3 dimensiones no varían. La única diferencia radica en el refinamiento de las mallas, que podemos ver en la Figura 4.31, donde se representa un detalle de  $50\text{ m} \times 50\text{ m}$  con el fin de observar el cambio en el tamaño de celda.



**Figura 4.31:** Mallas refinadas TD (izquierda) y TE (derecha)

Las celdas de la parte superior del detalle tienen mayor tamaño que las de la parte central. En el detalle de la malla TE es interesante el hecho de que, al forzar el refinamiento, algunos triángulos dejan de ser equiláteros y la malla en esa zona deja de ser tan estructurada, aunque mantiene una mayor estructura que la TD.

Las evoluciones temporales de  $h + z$  y de  $modU$  calculadas con el modelo computacional empleando mallas TD y TE refinadas se muestran en las Figuras 4.32 , 4.33 y 4.34.



4.4. CASO 4

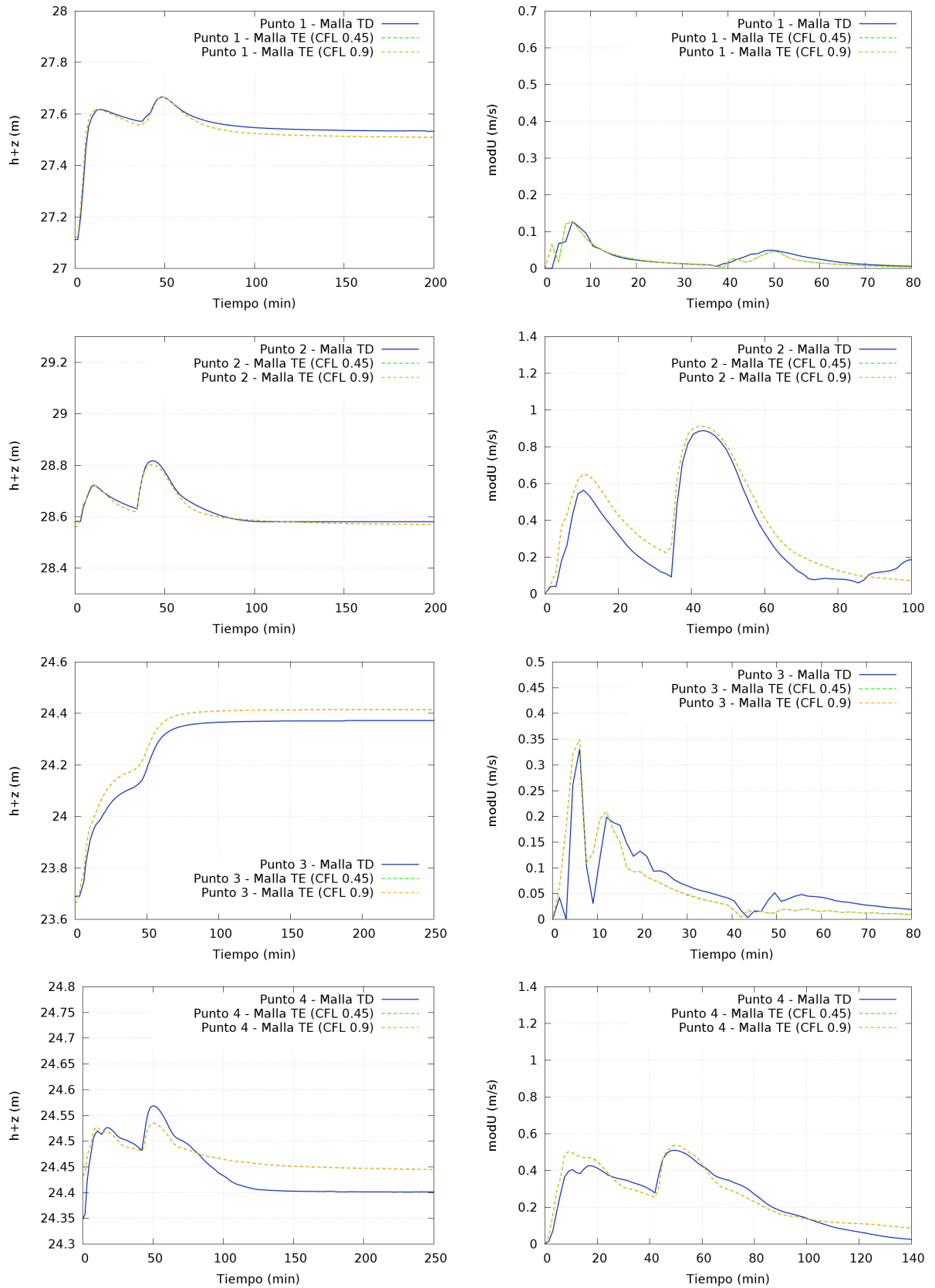


Figura 4.32: Evolución temporal de  $h+z$  (columna izquierda) y  $modU$  (columna derecha) para los puntos 1, 2, 3 y 4. Caso 4 malla refinada

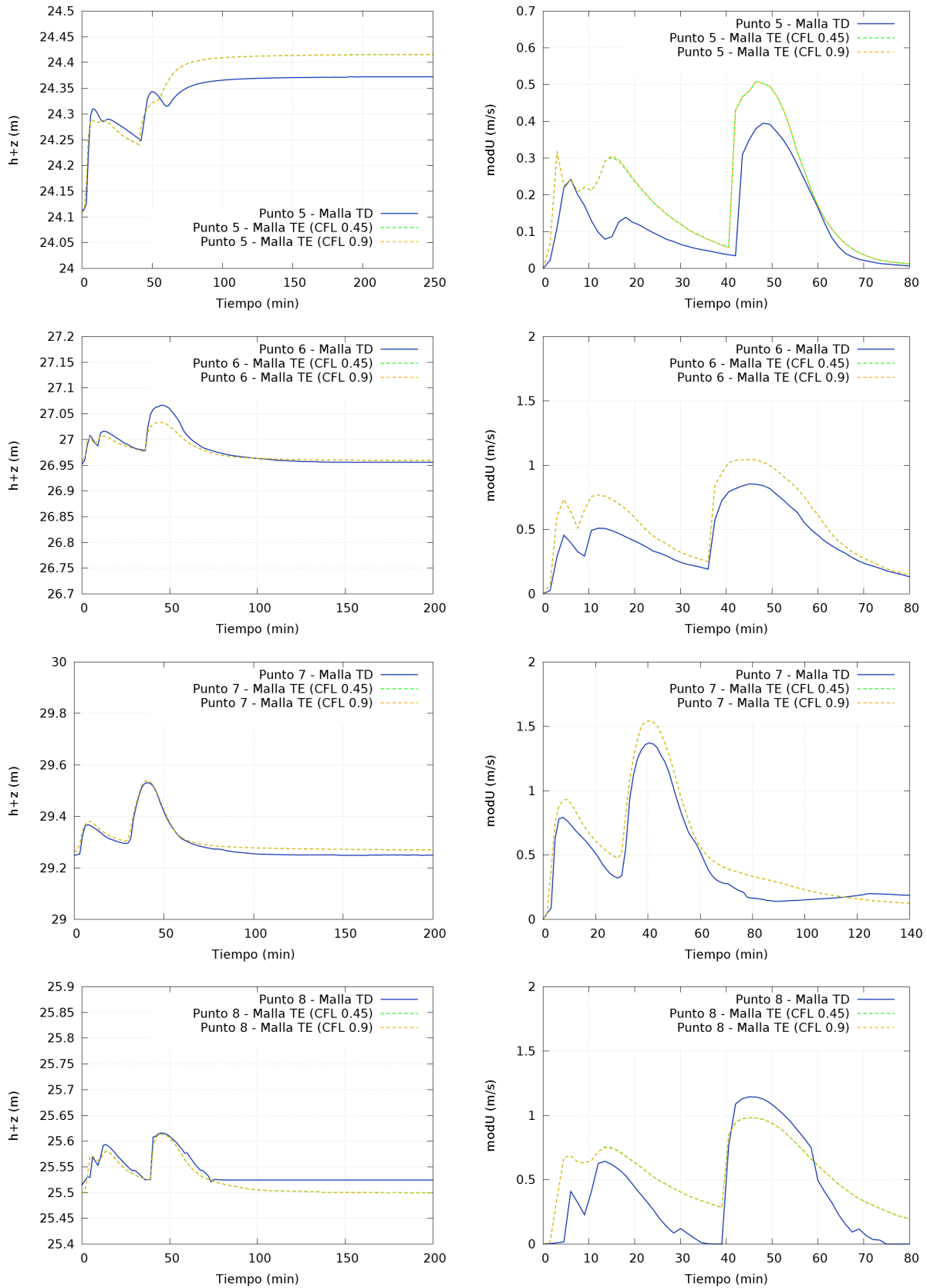
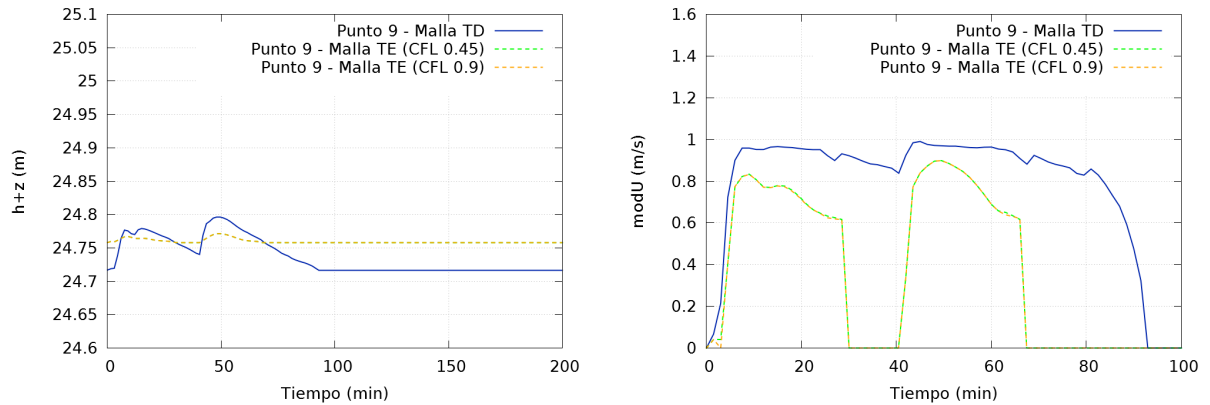


Figura 4.33: Evolución temporal de  $h+z$  (columna izquierda) y  $modU$  (columna derecha) para los puntos 5, 6, 7 y 8. Caso 4 malla refinada



**Figura 4.34:** Evolución temporal de  $h + z$  (izquierda) y  $modU$  (derecha) para el punto 9. Caso 4 con malla refinada

Las tendencias que siguen las evoluciones temporales, tanto del nivel superficial como del módulo de la velocidad, son ahora mucho más parecidas que en el caso no refinado. Esto se debe a que al refinar, la malla TE pierde parte de esa estructura característica, y se asemeja más a una composición tipo TD en las zonas refinadas. Como todos los puntos estudiados se encuentran en áreas refinadas, a priori los resultados obtenidos para este caso son más precisos que los representados sin refinamiento.

De nuevo la diferencia entre  $CFL = 0,45$  y  $CFL = 0,9$  en las mallas TE no se aprecia ni en el valor de  $h + z$  ni en el de  $modU$ . Las diferencias entre malla TD y malla TE aparecen fundamentalmente en la representación del  $modU$ , especialmente en los puntos 8 (Figura 4.33) y 9 (Figura 4.34). Se observa también que en estos dos puntos, en los resultados con malla TD han desaparecido esas variaciones bruscas y picos de módulo de la velocidad.

Los tiempos de simulación que se han obtenido aparecen en la Tabla 4.9 .

		Tiempo (s)			Speed-up		
		CPU		GPU	CPU		GPU
	Celdas Total	CPU 1	CPU 4		CPU 1	CPU 4	
TD	95.784	7.305,14	3.958,88	119,63	1	1,85	61,06
TE CFL 0,45	94.434	11.850,97	7.129,28	192,98	1	1,66	61,41
TE CFL 0,9	94.434	6.026,08	2.877,76	97,22	1	2,09	61,98

**Tabla 4.9:** Tiempos de simulación Caso 4. Mallas refinadas

Estos tiempos son mucho mayores que los resultantes de la simulación del caso sin refinar. Aunque al refinar el número de celdas no varía apreciablemente, sí que lo hace el tamaño de las mismas. Esta disminución de tamaño de algunas celdas provoca que el paso de tiempo sea menor que en el caso no refinado produciendo este aumento tan significativo de los tiempos de cálculo. El  $CFL$  juega de nuevo un papel fundamental en el coste de simulación, siendo aproximadamente inversamente proporcional al tiempo de cálculo. Sobre los *Speed-up* registrados, de nuevo los valores entre la versión CPU 1 y la CPU 4 no llegan a 3, sino que apenas consiguen un aumento en la velocidad de simulación en un factor 2. Por otro lado, la aceleración experimentada con la implementación en GPU sigue siendo muy grande, aproximándose a valores de 61-62.



# Capítulo 5

## Conclusiones

---

A continuación se exponen las conclusiones resultantes de la realización del Trabajo fin de Grado *Estudio comparativo y análisis de sensibilidad de un modelo computacional de simulación hidráulica para flujos en superficie libre*.

En primer lugar, de todas las gráficas presentadas se puede extraer la conclusión de que el tipo de malla empleado es un parámetro mucho más crítico a la hora de valorar el módulo de la velocidad ( $modU$ ) que el nivel superficial del agua ( $h + z$ ).

A la hora de valorar el  $modU$ , ni las mallas TE ni las mallas R aportan resultados precisos, especialmente en el último de los casos estudiados, que hace referencia a una posible situación real. En cambio, si el objetivo de una simulación es únicamente obtener resultados sobre el valor  $h + z$ , tanto las mallas TE como las R pueden ser válidas.

Otro detalle importante es la influencia del  $CFL$ . A lo largo de la memoria se ha observado que en la mayoría de los casos, los resultados obtenidos con  $CFL = 0,45$  y los obtenidos con  $CFL = 0,9$  apenas difieren. La excepción a esto se encuentra en algunas gráficas del caso 2. En ellas se puede observar que el nivel superficial obtenido con  $CFL = 0,9$  presenta una serie de picos que muestran la inestabilidad que puede llegar a aparecer si el  $CFL$  no es acorde al tipo de malla empleada. Se recomienda que en caso de emplear mallas TE se utilice un valor de  $CFL = 0,45$ , mientras que si la malla empleada es TD, éste puede ser 0,9, ya que no ha presentado ningún problema de inestabilidad a lo largo de todo el estudio. En caso de simulaciones con malla R, el  $CFL$  a emplear será 0,45, ya que al ser mallas tan marcadamente estructuradas la solución podría ser muy inestable si se aumenta este parámetro.

Sobre los tiempos de simulación, la primera de las conclusiones es que a mayor número de celdas involucradas en el proceso de cálculo, más tiempo tendrá que ejecutarse el programa para obtener la solución. Por otro lado, se ha observado también el papel del  $CFL$  en este coste de simulación, que aumenta de manera inversamente proporcional a este parámetro. Esto lo podemos ver en todas las tablas referidas a tiempos de simulación, donde los valores con  $CFL = 0,45$  son aproximadamente el doble que con  $CFL = 0,9$ .

Otra característica a destacar dentro de los tiempos de simulación es que en la práctica totalidad de los casos, el coste de simulación según el tipo de malla, de mayor a menor es, en primer lugar la malla TE con  $CFL = 0,45$ , luego la malla TD, la malla R y la malla TE con  $CFL = 0,9$ . Esto es importante de cara a valorar qué malla se puede emplear según el tipo de estudio. Si el objetivo es valorar únicamente el nivel superficial del agua, tanto la malla R como

la malla TD y la TE de  $CFL = 0,45$  ofrecen resultados precisos. Por ello, siempre y cuando el flujo sea prácticamente unidireccional, atendiendo al coste de simulación será preferible emplear la malla R. Hay que tener en cuenta que en situaciones aproximadamente reales (como el caso 4) esta malla R ofrece problemas de estabilidad, por lo que es más preciso emplear la malla TD o la TE con  $CFL = 0,45$  (teniendo en cuenta su elevado tiempo de simulación). Si el objetivo es evaluar el módulo de la velocidad, la malla TD es la única que ofrece resultados precisos en la mayoría de los casos, mientras que la R y las TE, al ser mallas estructuradas y definir direcciones prioritarias del flujo no son útiles en estos estudios.

Si se atiende al refinamiento, al aumentar la resolución (densidad de celdas) en alguna zona del dominio el tiempo de simulación aumenta enormemente. Si las necesidades de un caso son tales que pueda merecer la pena el refinamiento, entonces se puede llevar a cabo. Estos casos son aquéllos en los que sea necesario obtener resultados solo en ciertos puntos del dominio, bien definidos y situados en zonas concretas; o bien aquellos que, por las características de la zona discretizada, solo sea necesario el estudio en detalle de una pequeña parte de la misma.

Sobre la necesidad de desarrollar versiones del programa mediante distintos métodos de implementación, a la vista de los resultados obtenidos de *Speed-up*, se puede concluir que cuanto más se acerca un caso a condiciones reales y mayor es la exigencia en cuanto a coste de simulación, más importancia cobra la implementación en GPU, ya que de no ser así muchos casos reales no podrían simularse porque su tiempo de simulación sería del orden del propio periodo que se desea estudiar.

Para terminar, y después de todo lo expuesto, las mallas TD se consolidan como la mejor opción en cuanto a relación *coste de simulación/precisión de los resultados*; y la versión del programa en GPU se hace necesaria para simulaciones de casos con condiciones reales.

# Bibliografía

---

- [1] TOP500 Supercomputer Site.
- [2] Boris Delaunay. Sur la sphère vide. A la mémoire de Georges Voronoï. *Bulletin de l'Académie des Sciences de l'URSS*, (6):793–800, 1934.
- [3] A. Lacasta, P. García-Navarro, J. Burguete, and J. Murillo. Preprocess static subdomain decomposition in practical cases of 2d unsteady hydraulic simulation. *Computers & Fluids*, 80(0):225 – 232, 2013. Selected contributions of the 23rd International Conference on Parallel Fluid Dynamics ParCFD2011.
- [4] A. Lacasta, M. Morales-Hernández, J. Murillo, and P. García-Navarro. An optimized gpu implementation of a 2d free surface simulation model on unstructured meshes. *Advances in Engineering Software*, -(-):Under Review, 2014.
- [5] M. Morales-Hernández, P. García-Navarro, J. Burguete, and P. Brufau. A conservative strategy to couple 1d and 2d models for shallow water flow simulation. *Computers & Fluids*, 81(0):26 – 44, 2013.
- [6] J. Murillo and P. García-Navarro. Weak solutions for partial differential equations with source terms: Application to the shallow water equations. *Journal of Computational Physics*, 229(11):4327–4368, 2010.
- [7] J. Murillo, P. García-Navarro, and J. Burguete. Time step restrictions for well-balanced shallow water solutions in non-zero velocity steady states. *International Journal for Numerical Methods in Fluids*, 60(12):1351–1377, 2009.
- [8] E.F. Toro. *Shock-capturing methods for free-surface shallow flows*. John Wiley, 2001.
- [9] UK Environment Agency. *Benchmarking of 2D Hydraulic Modelling Packages*. Environment Agency Report. UK Environment Agency, 2010.
- [10] C.B. Vreugdenhil. *Numerical Methods for Shallow-Water Flow*. NATO Asi Series. Series C, Mathematical and Physical Science. Springer, 1994.





# ANEXOS

Apéndice A: Tipos de malla

Apéndice B: Proceso de simulación

Apéndice C: Programas informáticos auxiliares

Apéndice D: Temporalización del Trabajo



# Apéndice A

## Tipos de malla

---

La malla es uno de los elementos fundamentales dentro del proceso de modelado y simulación de un dominio, y además uno de los parámetros que se varían para realizar el estudio comparativo y el análisis de sensibilidad.

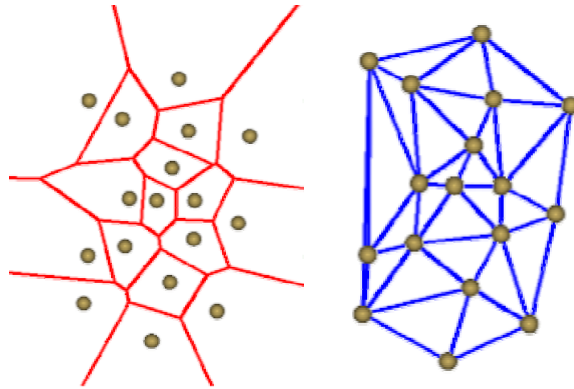
Para poder estudiar y predecir de manera computacional lo que ocurre en un dominio es necesario discretizarlo. La superficie o zona a estudiar es un medio continuo, y no puede ser tratado como tal por un modelo computacional debido a las limitaciones de los métodos numéricos. Por eso es necesario dividir este continuo en unidades de un tamaño suficientemente reducido como para poder recoger con fidelidad las propiedades del medio. Estas unidades reciben el nombre de celdas, y el conjunto de todas las celdas que reproducen el dominio es lo que se llama malla. Además, el proceso mediante el cual se divide un dominio en celdas se puede denominar *mallado*.

La forma de las celdas que se emplean en el *mallado* es también fundamental dentro de la malla. Se han estudiado tres tipos de malla atendiendo a la forma de la celda: Mallas triangulares Delaunay (TD), Mallas triangulares Equiláteras (TE) y Mallas Rectangulares (R).

### A.1 Malla Triangular Delaunay (TD)

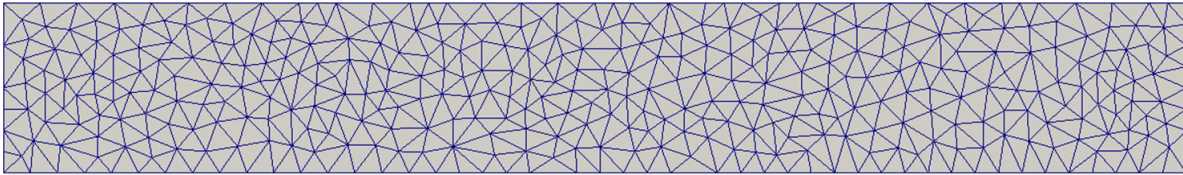
Este tipo de malla marca que el dominio ha de ser discretizado por elementos de planta triangular. De este modo se genera una malla no estructurada en la que cada elemento, analizado en 2D, responde a un triángulo no equilátero, y en la que las celdas no tienen por qué tener todas el mismo tamaño.

Las mallas TD se basan en los diagramas de Voronoi. El Diagrama de Voronoi de un conjunto de puntos aislados o nodos,  $V(N)$ , es una partición del espacio completo en regiones, donde a cada nodo  $p_q$  se le asigna la región cuyos puntos no son más cercanos a ningún otro nodo (Figura A.1). Un punto de la frontera de esta región equidista de dos o más nodos, es decir pertenece a alguna mediatriz. Dado que las mediatrices son planas o rectas, las fronteras solo pueden ser polígonos, cuyos vértices equidistan de cuatro o más puntos. De entre todas las posibles triangulaciones de un conjunto  $N$  de nodos, la triangulación de Delaunay (Figura A.1) es aquella en la que las celdas generadas mediante la triangulación deben cumplir la condición Delaunay, que dice que la circunferencia circunscrita a cada triángulo ha de ser vacía. Es decir, que dicha circunferencia no puede contener ningún vértice que no corresponda a los del propio triángulo que la forma.



**Figura A.1:** Diagrama de Voronoi (izquierda) y triangulación Delaunay (derecha)

Gracias a esta manera de triangular un conjunto de puntos se puede discretizar un dominio generando así una malla de tipo Delaunay, compuesta por triángulos no equiláteros de diferente tamaño, que es única y no posee ninguna estructura definida, por lo que no genera direcciones ni orientaciones preferentes de flujo. Un ejemplo de esta malla se puede ver en la Figura A.2 que corresponde a la malla TD generada para el Caso 1 de los presentados en la memoria.

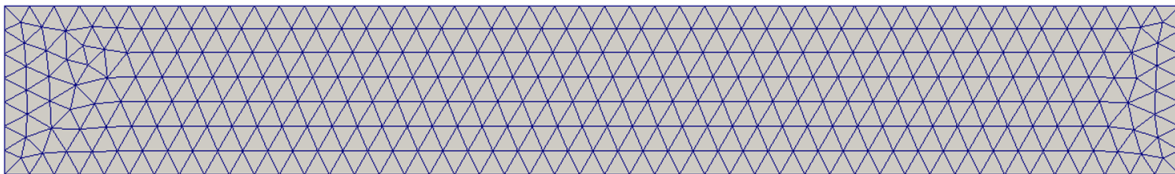


**Figura A.2:** Malla Triangular Delaunay (TD)

## A.2 Malla Triangular Equilátera (TE)

La malla TE es un tipo de malla estructurada en la que se intenta discretizar el dominio mediante triángulos lo más equiláteros posibles. Existen muchas formas de generar este tipo de mallas. Las empleadas para el modelo computacional estudiado han sido creadas definiendo un valor característico de área de la planta del triángulo a un programa generador de malla, y a partir de dicho valor se triangula el dominio consiguiendo un número de elementos aproximadamente similar a los que requiere cada caso.

La única similitud de este tipo de malla con la TD es que el elemento que se escoge para generar la malla del dominio es el triángulo. No obstante, al ser estos triángulos prácticamente equiláteros se generan en la malla estructuras características, como puede verse en la Figura A.3, que responde a la malla TE generada para el Caso 1 expuesto en la memoria.

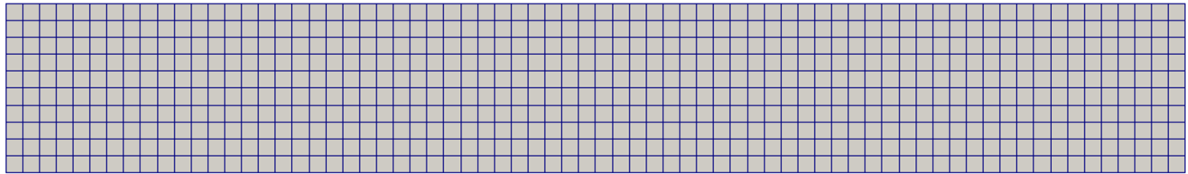


**Figura A.3:** Malla Triangular Equilátera (TE)

Al poseer la malla TE esta estructura característica pueden existir problemas ya que el flujo puede llegar a orientarse según las caras de estos triángulos equiláteros, generando direcciones preferentes de flujo y pudiendo llegar a proporcionar valores no reales de variables de carácter vectorial como la velocidad, donde la dirección y el sentido sí tienen importancia.

### A.3 Malla Rectangular (R)

En este caso la diferencia principal con los otros dos tipos de mallas es el elemento seleccionado para discretizar el dominio. En este tipo de mallas en lugar de emplear triángulos se usan rectángulos, como podemos ver en la Figura A.4, correspondiente al Caso 1 de los analizados en el estudio.



**Figura A.4:** Malla Rectangular (R)

Como se puede ver en la Figura A.4 la malla posee una estructura muy definida, por lo que la resolución de las ecuaciones gobernantes del sistema se realizará con unas orientaciones del flujo mucho más marcadas que en el caso de las mallas TD y TE. Este tipo de malla puede responder muy bien ante simulaciones de problemas cuasi unidireccionales debido a la disposición de las celdas en el dominio, mientras que en casos con flujos bidireccionales se pueden dar problemas de precisión.

De cara a la generación de la malla, la malla R tiene la ventaja de que es más sencillo llevar a cabo la discretización del dominio, ya que al ser los elementos de mallado rectángulos, ante geometrías simples con frontera y contornos planos basta definir el número de celdas deseadas para obtener la malla.



# Apéndice B

## Proceso de simulación

---

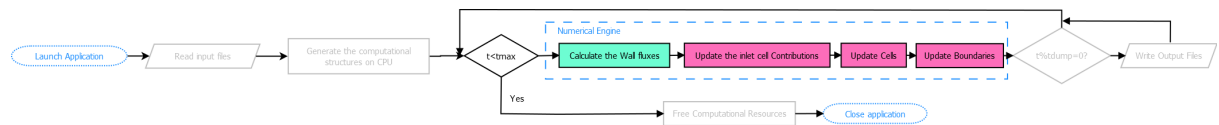
Una vez se obtiene la malla del dominio que se quiere simular se pasa al propio proceso de simulación. Para llevar a cabo dicho proceso es necesario que al modelo computacional se le definan una serie de ficheros a partir de los cuales pueda realizar el cálculo de las propiedades del flujo en cada una de las celdas mediante la resolución del método numérico con el dominio discretizado.

También es importante dentro del modelo computacional diferenciar entre los tres tipos de arquitectura en los que está implementado, en CPU (1 core y 4 cores) y en GPU.

A continuación se explican las diferencias entre la forma en que se procesan los datos en una arquitectura tipo CPU y otra tipo GPU, así como los ficheros que se han de definir al modelo computacional para que pueda llevar a cabo la simulación.

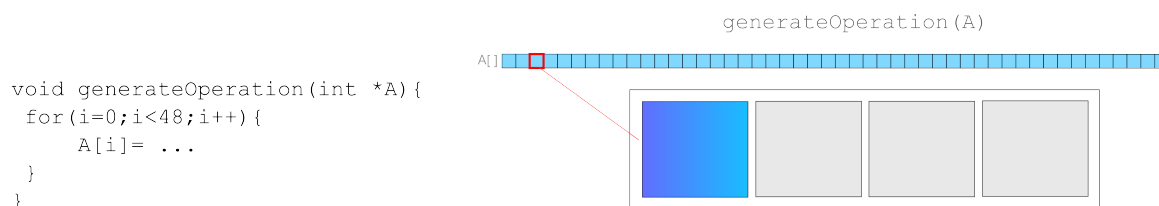
### B.1 Modos de implementación. CPU y GPU.

La implementación del modelo computacional sigue un proceso iterativo donde, mientras no finalice el tiempo total de simulación, el método numérico se aplica sobre todo el dominio. La manera en la que este proceso se lleva a cabo se describe en la Figura B.1, donde las operaciones principales se muestran en colores verde y rosa. El cálculo de los flujos se realiza siguiendo los contornos (en verde en la Figura B.1) y el proceso de actualización mediante bucles para cada celda (en rosa en la Figura B.1). Ambos procesos son iterativos y abarcan desde 1 hasta  $n_{contornos}$  y  $n_{celdas}$  respectivamente.



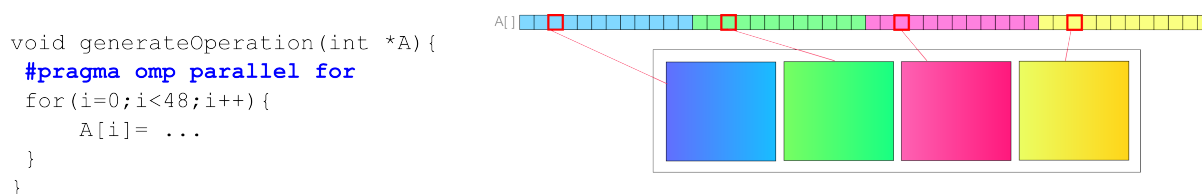
**Figura B.1:** Diagrama de flujo del proceso de simulación

Para conseguir que el proceso sea más rápido, algunas partes pueden ser paralelizadas o incluso adaptadas para llevar a cabo las operaciones en GPU (*Graphics Processing Units*). Actualmente, la mayoría de los procesadores se diseñan con capacidad de multi-proceso. A pesar de esta capacidad, no es directo llevar a cabo operaciones utilizando cada núcleo por separado. En la Figura B.2 se muestra un proceso iterativo típico empleando un núcleo de un procesador multi-núcleo.



**Figura B.2:** Proceso iterativo secuencial en 1 núcleo sobre el vector A [ ]

Cuando es posible, el proceso iterativo debe ser dividido en diferentes subprocesos, consiguiendo así procesar diferentes elementos al mismo tiempo en diferentes unidades de procesamiento. Esto es posible cuando el bucle contiene código secuencial (un único bloque simple sin saltos) y cuando no existe dependencia de datos entre iteraciones (no se necesita el elemento  $i - 1$  para el elemento  $i$ ). Si el flujo de datos no satisface estas condiciones es necesario replantear la manera de procesar los elementos para poder hacerlos paralelizables. OpenMP es un modelo muy útil para desarrollar paralelizaciones de memoria compartida, que incorpora una serie de directivas en muchos lenguajes de programación como C o Fortran que hacen muy fácil implementar una solución paralela en un tiempo razonable. La principal desventaja de este tipo de paralelización es que los procesadores *Multi-core* (propios de CPU) no están creciendo tanto como las más recientes arquitecturas *Many-core* (típicos de GPU), donde una elevada cantidad de unidades de procesamiento especializadas permiten realizar muchas más operaciones que en una unidad *Multi-core*. Un ejemplo de un bucle paralelo utilizando OpenMP se muestra en la Figura B.3.



**Figura B.3:** Proceso iterativo paralelo sobre el vector A [ ] empleando 4 núcleos

Los procesadores de tipo *Many-core* proporcionan una arquitectura de Hardware que permite realizar de manera paralela un elevado número de operaciones independientes. La principal diferencia entre los hilos de ejecución de los procesadores *Multi-core* y los *Many-core* es que la carga de procesamiento es mucho mayor en estos primeros. En un procesador *Multi-core* cada uno de los núcleos es capaz de realizar la totalidad de las operaciones requeridas para el funcionamiento y procesamiento de datos, conllevando esto una gran carga para los hilos de ejecución. Por otro lado, en los procesadores *Many-core* cada núcleo es capaz solo de realizar tareas simples lógico-aritméticas. Esto hace que los hilos de ejecución de este tipo de procesadores tengan menor carga, pero limita la utilidad de la GPU a operaciones de cálculo y desarrollo de métodos numéricos. El objetivo con el empleo de este tipo de procesamiento es el de obtener una solución numérica gracias a la implementación en GPU y trasladar dicha solución a una CPU para que el procesador represente los resultados de la manera requerida. Además, a diferencia de los hilos de ejecución de los procesadores *Multi-core*, donde cada hilo procesa un grupo de elementos (Figura B.3), en los procesadores *Many-core* cada hilo de ejecución procesa únicamente un elemento. En la Figura B.4 se muestra un ejemplo esquemático de cálculo en GPU.



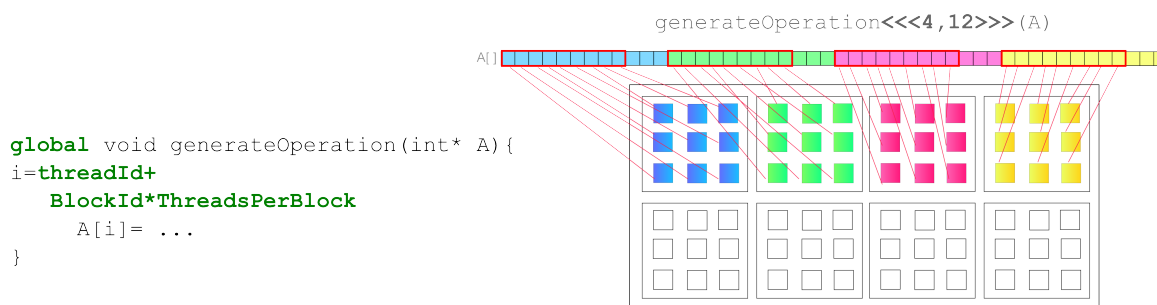


Figura B.4: Proceso iterativo masivamente paralelo sobre el vector A [ ] utilizando una arquitectura *Many-core* (GPU)

## B.2 Ficheros y elementos del proceso de simulación

### Condiciones de simulación

Para poder llevar a cabo la simulación hay que crear un fichero con formato *.cond* de nombre *case.cond* que será necesario para especificarle al modelo parámetros relevantes para la simulación como:

- **Tiempo inicial de la simulación:** el modelo puede partir su simulación desde cualquier instante temporal. Lo usual es introducir en este parámetro el valor 0, aunque se puede incluir cualquier otro valor numérico (en segundos) si se quiere simular a partir de otro estado ya conocido, o si la franja temporal total estuviera dividida en intervalos de tiempo menores.
- **Tiempo final de la simulación:** para que el modelo sepa el momento en el que tiene que finalizar la simulación es necesario indicar este parámetro. También tiene que ser introducido en segundos y su valor debe ser mayor que el tiempo inicial de la simulación.
- **CFL:** es un parámetro adimensional muy importante en el proceso de simulación. La variación del mismo influye en la estabilidad del cálculo y en el paso de tiempo.
- **Volcado de pantalla:** este parámetro se indica en número de pasos de tiempo. Para poder conocer el estado de la simulación existe la opción de que en la pantalla de simulación se vuelque información del proceso cada cierto número de pasos de tiempo.
- **Volcado de datos:** de nuevo aparece otro parámetro temporal y por lo tanto expresado en segundos. En este caso está relacionado con la frecuencia con la que se desea que el programa capture los valores de las variables del flujo de cada celda a lo largo de toda la simulación. De este modo, si se quieren simular 1.000 segundos de tiempo real y se le indica al programa que el volcado de datos sea cada 10 segundos, entonces se obtienen 100 ficheros con formato *.vtk* que contienen los valores de los parámetros que calcula el programa en cada celda en los instantes de tiempo 1, 10, 20... hasta los 1.000 segundos de tiempo real. Estos ficheros permiten visualizar esta simulación.
- **Volcado de medidas:** el modelo ofrece la posibilidad de incluir sondas de medida en el proceso de simulación. Este parámetro temporal expresado en segundos indica la frecuencia con que se desea que se recoja información en esas sondas, que son unos ficheros con formato *.out* que recogen los valores de las variables calculadas por el programa (en columnas) en intervalos de tiempo (en filas) separados entre sí tantos segundos como marca dicho parámetro, que se denomina en el fichero *volcado\_query*.

- **Coefficiente n de Manning:** este parámetro hace referencia a propiedades de fricción del terreno en el que se desarrolla la simulación. Cada terreno tiene un coeficiente de Manning propio y dentro del dominio de simulación puede variar. Si se da este caso es necesario indicar en el fichero de condiciones el coeficiente de Manning de cada celda.
- **Calado mínimo:** este parámetro, expresado en metros, indica cual es el nivel de elevación superficial de agua mínimo a partir del cual se considera que el flujo permanece estático. De nuevo puede variar dentro del dominio si lo hace el número de Manning.

Un fichero de este tipo, *case.cond* podría tener el formato siguiente:

TIEMPO_INICIAL	0.0
TIEMPO_FINAL	36000.0
CFL	0.9
volcado_pantalla	1000
VOLCADO_datosA	360.0
VOLCADO_QUERY	360.0
N_MANNING	0.03
CALADO_MINIMO	0.001

## Entradas

Una vez definidas las condiciones de simulación es necesario especificar al modelo cuáles y cómo son las entradas del sistema. Esto se consigue gracias a un fichero *case.inlet* en el que se incluyen los siguientes parámetros:

- **Número de entradas:** se le tiene que indicar al modelo cuantas entradas existen en los contornos del dominio.
- **Tipo de entradas:** las entradas de flujo al dominio pueden ser de varios tipos. Dentro de este fichero solo se pueden recoger dos de ellos, entradas de tipo nivel superficial de agua en metros ( $h$ ), que se indican con un 1 en este parámetro, y entradas de tipo caudal ( $m^3/s$ ) que se indican con un 2.
- **Coordenadas de las zonas de entrada de flujo:** para indicarle al modelo por qué zona del dominio tiene lugar la entrada de flujo es necesario indicar dos puntos dentro de este fichero. El modelo emplea estos dos puntos para trazar una diagonal, y sobre dicha diagonal construye el rectángulo correspondiente. Este rectángulo incluye los puntos centrales de algunas celdas. A dichas celdas se les asignará el valor de la condición de nivel superficial de agua o de caudal correspondiente a la función de entrada en cada instante temporal. Para indicar estos puntos se le tendrán que señalar al programa las coordenadas  $x$  e  $y$  de los mismos en metros.
- **Dirección de entrada:** gracias a este parámetro se le indica la dirección y el sentido del flujo en los contornos de entrada a nuestro dominio. Para indicarlo son necesarios dos valores numéricos que indican el módulo de los vectores en las direcciones  $x$  e  $y$  respectivamente. De este modo, si colocamos (1 0) estamos haciendo referencia a un flujo que tiene dirección  $x$  y sentido positivo.
- **Número de Froude:** este parámetro adimensional hace referencia a las características del flujo a la entrada. Durante todos los problemas y casos resueltos se emplea el valor de 0,9 ya que el flujo es siempre subcrítico en la entrada.

- **Número de puntos de la función:** estos son los últimos parámetros que se indican dentro de este fichero. Las entradas han de ser especificadas al modelo como una función temporal. Para ello hay que indicar el número de puntos que tiene la función y posteriormente incluir dichos puntos por filas en pares de coordenadas  $x$  (tiempo),  $y$  (dimensiones de la entrada, es decir, caudal o altura).

Un ejemplo de este fichero podría ser el siguiente:

```
NUMERO_INLETS          1
TIPO_DE_ENTRADA        2
X_LIMITE_PUNTO2        -5.0  -1.0
X_LIMITE_PUNTO1        5.0   101.0
DIRECCION_ENTRADA      1.0   0.0
FROUDE_ENTRADA         0.9
NUMERO_PUNTOS_FUNCION  6
0.0      9.7
300.0    9.7
3600     10.35
39600    10.35
43200    9.7
72000.0  9.7
```

## Salidas

Otro fichero que hay que facilitarle al modelo es el fichero de salidas, *case.outlet*. En este fichero se tienen que especificar las salidas de flujo que existen en el dominio. No obstante, en ningún caso de los estudiados es necesario dicho fichero, motivo por el que no se incide más sobre él, ya que en todos los casos simulados tiene la siguiente forma:

```
N_OUTLETS  0
```

## Elementos de medida

Además de los ficheros necesarios para el correcto funcionamiento del programa existen otros que pueden aportar información importante de cara a la visualización y obtención de datos. Por norma general, el programa aporta un número determinado de ficheros con formato *vtk* que depende del parámetro de volcado de datos. Estos ficheros pueden ser interpretados y mostrados mediante varios programas de visualización. Estos programas además de ofrecer representaciones del proceso de simulación y del desarrollo temporal del mismo y de las variables calculadas también permiten realizar gráficos de las propiedades de cada celda. No obstante, esta manera de graficar puede resultar laboriosa, y puede simplificarse y optimizarse gracias a una función del modelo computacional. Esta función consiste en aportarle al modelo un fichero de nombre *case.probes*, que puede ser de mucha utilidad si queremos recopilar el valor de las variables del flujo de uno o varios puntos concretos a lo largo del tiempo. En este fichero se incluyen los siguientes datos:

- **Número de sondas o elementos de medida:** se incluye en la primera línea del fichero un número precedido por el texto *NPROBES* que indica el número de sondas o puestos de medida que se incluyen en el dominio.

- **Coordenadas de las sondas:** cada uno de estos elementos de medida hay que situarlo espacialmente en el dominio. Para ello es necesario incluir en el fichero tantas líneas como sondas, indicando en cada línea las coordenadas  $x$ ,  $y$  en metros de la sonda dentro del dominio.

Este fichero podría tener el formato siguiente:

```
NPROBES  2
400      50
600      50
```

Una vez se incluye el fichero, tras la simulación el programa genera  $M$  ficheros de nombre *probei.out* donde  $i$  varía desde 1 hasta  $M$ , siendo  $M$  el número de sondas incluidas. Cada uno de estos ficheros incluye tantas filas como instantes de tiempo en los que se han recogido, dependiendo este número del parámetro `VOLCADO_QUERY`, que nos indica cada cuantos segundos queremos recopilar datos. Estos ficheros de salida tienen además 6 columnas. En la primera de ellas figura el instante de tiempo en el que nos encontramos al tomar medidas (en segundos), y en las otras 5 las variables que calcula el modelo computacional, apareciendo en las columnas 2, 3, 4, 5 y 6 los valores de  $h + z$ ,  $h$ ,  $u$ ,  $v$ , y  $modU$  respectivamente.

### Otros ficheros

Por último merece la pena explicar otros dos ficheros que se han utilizado para la definición de las condiciones iniciales tipo *lluvia* y tipo *fuentes*. Estos ficheros son *case.rain* y *case.sources*:

- **Fichero para incluir entradas tipo lluvia:** se le denomina *case.rain* y se utiliza para especificarle al modelo una entrada en forma de lluvia. La lluvia, como el resto de entradas es necesario indicarla en forma de función, por lo que en el fichero tendremos que incluir de nuevo el número de puntos de la función y las coordenadas de los mismos,  $x$  (s) e  $y$  (mm/s). Este fichero tiene el siguiente formato:

```
RAIN      1
NPOINTS   7
0.0       0.0
59.994    0.0
60.0      0.11111111
240.0     0.11111111
240.006   0.0
3600.0    0.0
18000     0.0
```

- **Fichero para incluir entradas tipo fuentes:** para incluir este tipo de entradas se tiene que crear un fichero de nombre *case.sources*. Este tipo de fichero simula la entrada de un caudal de agua a través de un punto (celda) concreto del dominio que no tiene por qué estar en el contorno. Este caudal se transforma para la simulación en una altura de nivel de agua que se le introduce a la celda correspondiente. Se indica el número de entradas de este tipo después del texto `N_SOURCES` con el valor correspondiente. Además se tienen que localizar en el dominio dichas fuentes con las coordenadas de las mismas en metros ( $x$ ,  $y$ ). De nuevo hay que especificar la función de entrada, por lo que tenemos que indicar

el número de puntos de la misma así como los valores  $x$  (s) e  $y$  ( $m^3/s$ ). Este fichero tiene el formato siguiente:

N_SOURCES	1
COORD	264896 664747
N_POINTS	23
0.00000000	0.00000000
1199.99999880	0.00000000
1379.99999880	0.06250000
1500.00000120	0.25000000
1620.00000000	0.62500000
2940.00000120	0.62500000
3060.00000000	0.25000000
3179.99999880	0.06250000
3300.00000120	0.00000000
18000.00000000	0.00000000



# Apéndice C

## Programas informáticos auxiliares

---

Para poder llevar a cabo el estudio comparativo y el análisis de sensibilidad, además del propio modelo computacional estudiado han sido necesarios otros dos programas informáticos que merece la pena resaltar, uno empleado para poder visualizar los resultados de las simulaciones y otro para realizar las gráficas requeridas.

### C.1 Representaciones gráficas: *Gnuplot*

Para poder realizar el estudio comparativo es necesario representar gráficamente la evolución temporal tanto del nivel superficial del agua ( $h + z$ ) como del módulo de la velocidad ( $modU$ ) en varios puntos dentro de cada uno de los casos estudiados.

Estas representaciones gráficas se pueden realizar con el mismo programa empleado para la visualización de los resultados. No obstante el proceso es relativamente lento en el momento en el que se tienen que presentar gráficos superpuestos de casos de simulaciones distintas. Por este motivo ha sido necesario emplear un programa informático adicional de cara a optimizar la realización de las gráficas necesarias.

El programa empleado es *Gnuplot*. Es un programa muy flexible para generar gráficas de funciones y datos y que es compatible con los sistemas operativos más populares como Linux, UNIX, Windows o Mac OS X. *Gnuplot* puede producir sus resultados directamente en pantalla, así como en multitud de formatos de imagen como PNG, EPS, SVG, JPEG, etc. Se puede usar interactivamente o en modo por lotes (batch) usando scripts.

En el estudio, el programa se ha utilizado empleando y desarrollando un script para poder realizar las gráficas mediante un bucle iterativo y así generar todas las representaciones necesarias para cada caso con una sola orden a partir de los datos contenidos en los ficheros de medida *case.probes*. Esto ha permitido agilizar el proceso de obtención de gráficas, y ha posibilitado presentar varias superpuestas sobre la misma imagen de un modo visual e intuitivo.

Además de para gráficas sencillas en 2 dimensiones, este programa puede emplearse para fines mucho más complejos como representación de superficies y volúmenes, funciones matemáticas, series de datos o incluso imágenes animadas. Aunque estas opciones no han sido utilizadas, en la Figura C.1 se pueden ver algunos ejemplos de representaciones con *Gnuplot*.

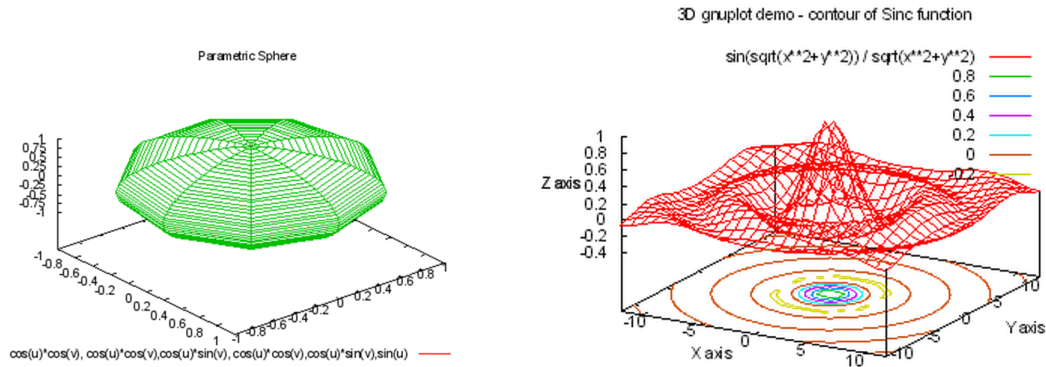


Figura C.1: Representaciones con *Gnuplot*

## C.2 Visualización de datos: *Paraview*

Para poder interpretar los ficheros de salida que se obtienen gracias al modelo computacional ha sido necesario emplear un programa auxiliar denominado *Paraview*, que es una aplicación multiplataforma y de código abierto diseñada para visualizar series de datos de tamaños muy diversos, desde muy pequeñas a muy grandes. Tiene una interfaz muy sencilla de utilizar, flexible e intuitiva, que permite moverse fácilmente por la pantalla y seleccionar las distintas funciones del programa, como puede verse en la Figura C.2, donde se muestra con *Paraview* la malla generada para el caso 1 del estudio, con la escala de colores correspondiente a la cota  $z$  de la topografía del dominio .

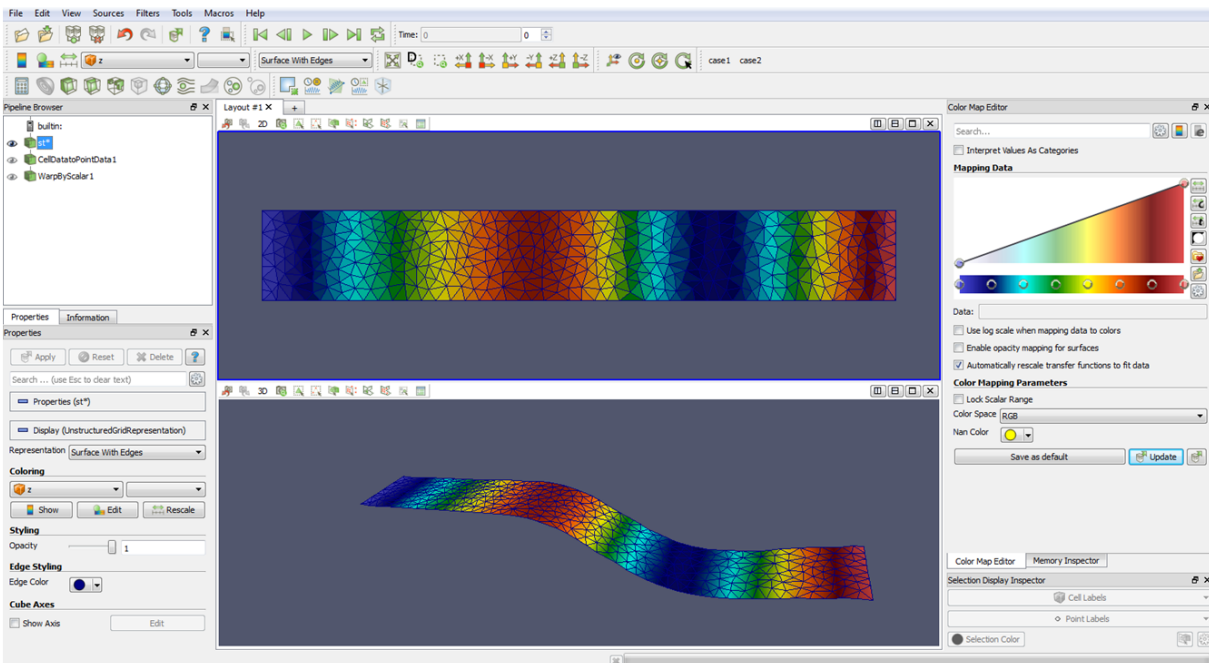


Figura C.2: Interfaz de *Paraview*

Para el estudio ha sido de gran utilidad ya que este programa permite abrir los ficheros de formato *.vtk* generados por el modelo computacional y así se pueden mostrar por pantalla



las variables del flujo que éste calcula, propiedades o animaciones de la simulación. Entre otras funcionalidades podemos destacar:

- Identificación de los valores de las variables mediante escalas de colores.
- Visualización temporal de la variación de las variables con escalas de colores.
- Representación de magnitudes vectoriales mediante iconos que intuitivamente muestran módulo, dirección y sentido.
- Visualización en 3 dimensiones mediante la proyección de diversas propiedades de la malla.

Estas opciones pueden visualizarse también en la Figura C.2, donde se muestra un ejemplo de lo que se puede ver por pantalla mientras se trabaja con este programa.



# Apéndice D

## Temporalización del Trabajo

---

Este anexo hace referencia a la estructura temporal que se he seguido durante el desarrollo del Trabajo fin de Grado *Estudio comparativo y análisis de sensibilidad de un modelo computacional de simulación hidráulica para flujos en superficie libre*.

Esta traza temporal comienza con el inicio del segundo cuatrimestre del último curso del Grado en Ingeniería de Tecnologías industriales y llevándose a cabo al mismo tiempo que tres asignaturas de dicho curso, tal y como se recoge dentro del plan de estudios del grado.

Durante la realización del trabajo se han podido distinguir varias fases diferentes. En primer lugar, desde el día 10 hasta el día 20 de febrero tuvo lugar la fase de introducción al Trabajo fin de Grado. Dentro de esta fase se incluye la comprensión del método numérico que el modelo computacional desarrolla para el cálculo de las propiedades del flujo.

Una vez realizada la introducción, la segunda fase del trabajo consistió en la familiarización tanto con el modelo computacional como con los programas informáticos auxiliares que se han empleado. La familiarización con el modelo computacional se llevó a cabo simulando un caso test similar a los que se tendrían que evaluar en la memoria, y fue también costosa porque el modelo se ejecuta en remoto y con una interfaz de comandos en linux.

Una vez comprendido el uso del modelo computacional se llegó a la tercera fase, que ha englobado la mayor parte de las horas dedicadas al Trabajo, y que hace referencia a la simulación de los 4 casos estudiados. El coste temporal de esta fase resultó ser elevado al tener que compatibilizar el desarrollo del Trabajo con asignaturas del Grado de Tecnologías Industriales. A la vez que se llevaban a cabo estas simulaciones también se realizaban las gráficas de  $h+z$  y  $modU$  que se requerían para el análisis de los resultados según el documento de la *Environment Agency* empleado como referencia [9].

A día 8 de junio en mitad del periodo de exámenes y con la memoria todavía por redactar tomé la decisión de posponer la entrega del trabajo hasta el mes de septiembre, teniendo así más tiempo para redactar la memoria y preparar la defensa. Esta nueva fase de redacción de la memoria comenzó el día 25 de junio, ya terminado el periodo de exámenes, con una introducción a *Latex*, el programa que he empleado para desarrollar el documento. Una vez comprendido se pasó a redactar la memoria, y al finalizarla se llevaron a cabo las correcciones pertinentes y la maquetación.

Finalmente tuvo lugar la última de las fases del Trabajo, la preparación de la defensa. Esta

se ha realizado posteriormente al depósito de la memoria, por lo que la duración es estimada. No obstante se ha basado en la experiencia del personal del Grupo de Hidráulica Computacional en cuanto a preparación de este tipo de presentaciones se refiere.

Toda esta estructura temporal llevada cabo a lo largo de estos meses se representa en el diagrama de Gantt de la Figura D.1.

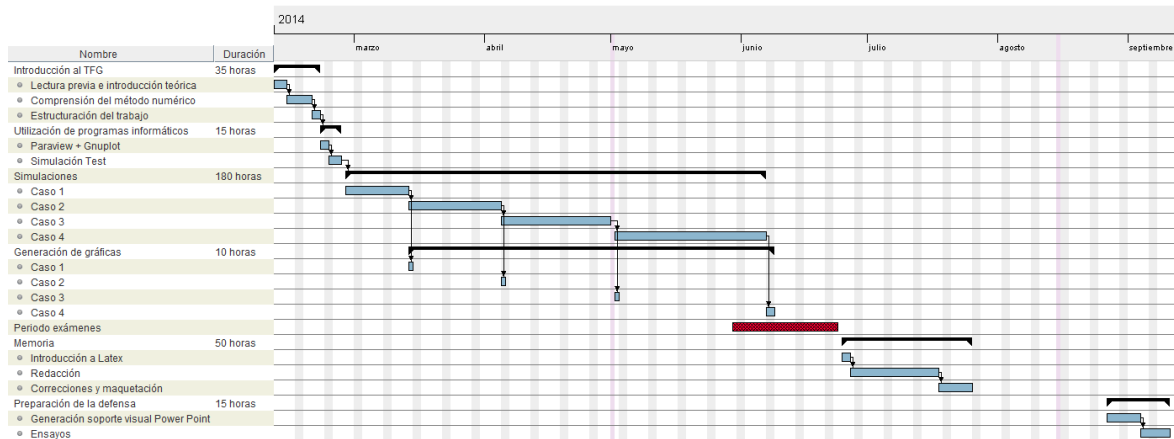


Figura D.1: Diagrama de Gantt del Trabajo fin de Grado