

# Aplicació del *Material Point Method (MPM)* a la deformació d'objectes

Lliurable 2 GEP

Martín Garcia, Pol

Treball de final de grau d'Enginyeria informàtica

Especialitat de Computació  
Tardor 2019

Director: Susín Sánchez, Toni

# Índex

<b>1</b>	<b>Introducció</b>	<b>2</b>
1.1	Context . . . . .	2
1.1.1	Conceptes bàsics . . . . .	2
1.1.2	Formulació del problema . . . . .	5
1.1.3	Actors implicats . . . . .	5
1.2	Justificació . . . . .	6
1.3	Abast . . . . .	7
1.3.1	Objectius . . . . .	7
1.3.2	Requisits . . . . .	9
1.3.3	Obstacles i riscos . . . . .	9
1.4	Metodologia i rigor . . . . .	11
1.4.1	Metodologia . . . . .	11
1.4.2	Eines de seguiment . . . . .	11
1.5	Planificació temporal . . . . .	12
1.5.1	Descripció de tasques . . . . .	12
1.5.2	Taula resum . . . . .	17
1.5.3	Diagrama de Gantt . . . . .	18
1.5.4	Planificació d'alternatives . . . . .	19
<b>2</b>	<b>Referències</b>	<b>20</b>

# 1. Introducció

Aquest treball de fi de grau ha estat centrat en la simulació de fluids, una temàtica molt àmplia que tot i que conceptualment sembla trivial, és molt complexa. Molts detalls i decisions s'han de tenir en compte a l'hora d'implementar un simulador d'aquestes característiques, com la representació del fluid, si aquest és compressible, elàstic, interactiu amb col·lisions, etc.

Per a poder presentar el treball correctament, aquest document també serveix com a introducció a la simulació de fluids sense assumir coneixements previs en aquest aspecte, i des del punt de vista d'un enginyer informàtic.

I finalment també es busca mostrar com materials sòlids-elàstics es poden representar i simular com a fluids.

## 1.1 Context

La simulació de fluids, o dinàmiques de fluids computacionals (*CFD's*), és una disciplina que, mitjançant tècniques d'anàlisi numèrica, busca resoldre problemes que impliquen d'alguna manera un tipus de fluctuació o interacció amb fluids.

La base de qualsevol simulador de fluids són les equacions de Navier-Stokes, que descriuen la mecànica d'aquests; però necessitem molt més que les equacions per a poder implementar aquest programa, doncs també són necessaris coneixements de software i hardware per a una implementació eficient, tècniques de gràfics per computador per a una adequada visualització, i coneixements matemàtics d'àlgebra lineal que ja s'han vist durant l'educació oferida per la Facultat d'Informàtica de Barcelona, sobretot en la branca de computació. Tot i això, l'abast del projecte ens obliga a sortir de l'àmbit de coneixements d'un enginyer informàtic, i integrar múltiples conceptes matemàtics de diferents caires, que seran explicats degudament.

En el nostre cas les matemàtiques ens ajudaran a definir el comportament i les limitacions del nostre simulador, que a partir d'ara anomenarem també *solver*, per a poder configurar-lo amb coneixement a posteriori, i sempre ser conscients de l'estat del sistema que s'està processant.

### 1.1.1 Conceptes bàsics

Abans d'entrar en detalls, hi ha uns coneixements bàsics que s'han de tenir en compte degut a l'àmplia projecció dels simuladors de fluids.

## Tipus de solvers

Podem diferenciar els tipus de *solvers* en 3 subgrups, que varien intrínsecament en el mateix concepte de la representació del fluid i els algorismes emprats. Aquesta representació marcarà la manera d'interactuar i tractar el fluid tant amb si mateix, com amb cossos externs.

**Solvers en Graella** Podem representar el fluid en un instant de temps concret com a una magnitud en un punt d'una graella, identificant la quantitat de fluid en aquella posició en un moment concret; per altra banda, cada cel·la de la graella té alguna representació de direcció i magnitud de moviment del fluid, per a poder computar el desplaçament d'aquest.

A més a més, podem emmagatzemar altra informació a cada posició de la graella, o usar les dades ja guardades per a processar l'estat del sistema en un instant de temps posterior, tenint en compte que a cada cel·la només hi pot haver una quantitat màxima de fluid (volum màxim) perquè no sigui un fluid comprímbil, i així provocar una dissipació d'aquest i el consegüent moviment.

En conjunt, creem un espai acotat per les dimensions de la graella on el fluid es mou de manera quantitativa a través de les diferents cel·les, d'acord amb la mateixa informació d'aquestes.

Aquesta representació es coneix com a simulador Eulerià.

**Solvers de Partícules** Podem representar un petit volum de fluid com a una partícula en una posició determinada en un instant de temps, de manera que sempre representa la mateixa quantitat de fluid, i a més guardem tota la informació de moviment (força, velocitat, ...) a un nivell molt concret representat per la mateixa partícula.

És important entendre que les partícules no representen el fluid a escala de molècules o àtoms, sinó que cada partícula representa una porció contínua del material o fluid, o un subconjunt del domini físic a simular.

Aquesta representació permet, per exemple, barrejar dos fluids mantenint alhora les seves propietats separades, o simular interaccions entre partícules o sòlids externs amb un major nivell de detall. També cal dir, que aquest mètode dificulta la tasca de mantenir un volum constant del fluid, i sobretot necessita calcular interaccions entre totes les partícules, còmput que resulta molt costós.

Els simuladors de partícules són coneguts com a Lagrangians.

**Solvers Híbrids** Si mesclem els dos conceptes anteriors, obtenim un simulador que, podem dir, rep el millor dels dos mètodes.

Per una banda, es representa el fluid com partícules en cada instant de temps amb totes les seves propietats; i per altra banda el còmput d'interaccions entre aquestes el gestionem mitjançant una graella (la qual defineix l'espai del sistema) a on traspassem les característiques de les partícules en determinades zones, de manera que la computació de col·lisions i/o interaccions és molt més eficient i acotada.

## Propietats del fluid

Un fluid sempre ha de tenir representades, en alguna determinada estructura o directament en l'algorisme, un seguit de propietats que el defineixen i n'especifiquen el comportament. Cal dir, que només és necessari usar aquelles propietats que el nostre model té en compte.

**Volum** La quantitat de fluid que volem simular té un impacte directe en el nostre *solver*, i depenent de la representació del fluid aquest concepte afectarà de manera diferent; per exemple, en simuladors de graella, massa fluid pot saturar el model; i en simuladors de partícules hem de controlar la proximitat entre partícules perquè aquesta defineix el volum.

**Massa** Depenent del model a implementar, ens pot interessar la massa del fluid (representada, per exemple, en densitat) per a tractar la interacció amb forces externes (col·lisions o gravetat).

**Elasticitat** L'elasticitat és l'habilitat d'un cos o material de recuperar la seva forma original després d'una deformació, i alhora resistir-se a aquestes forces de deformació. Es descriu amb el model i la llei de Hook, que estipula que la força de retorn és proporcional a l'extensió de la deformació.

**Viscositat** Molt semblant a l'elasticitat, la viscositat d'un fluid és la resistència que té a deformar-se o, informalment, a escampar-se. Es mesura en pascals per segon ( $\frac{Pa}{s}$  o  $\frac{kg}{m \cdot s}$ ) i es representa amb  $\mu$  o  $\eta$ . És molt comú representar aquesta magnitud en relació amb la densitat  $\rho$  de manera  $\nu = \frac{\mu}{\rho}$  per a simplificar expressions.

**Plasticitat** Si l'elasticitat defineix la possible deformació d'un cos, la plasticitat marca el punt de no retorn d'una deformació, en què les forces que busquen el retorn de l'objecte a l'estat original són permanentment degradades, de manera que part de la deformació es torna permanent. La idea de modelar aquesta característica, i el pioner treball de Terzopoulos et al. [1] el 1988, ha fet que la simulació de materials topològicament canviant sigui una popular àrea d'estudi a gràfics per computador, on es poden tractar fets com la fractura (límit de la plasticitat), i el tallament d'objectes deformables. Totes aquestes interaccions s'anomenen inelàstiques.

**Viscoelasticitat** Propietat dels materials que són viscosos i elàstics alhora en el moment de deformar-se. La viscositat fa que, a mesura que el temps avança, el material sofreix una tensió que li provoca una certa deformació, de manera que tota deformació elàstica sempre comporta una pèrdua d'energia a causa de la viscositat, produint una deformació plàstica.

## Material Point Method

Un dels mètodes que ara ha ressorgit per a la simulació de fluids és el conegut *Material Point Method*, proposat per Sulsky [2], basat en partícules Lagrangianes i una graella Euleriana de rerefons. Aquest mètode és usat donat la seva alta eficiència de còmput, i la possibilitat de detallar la seva precisió amb la mida de la graella.

Aquesta tècnica intenta emmagatzemar les dades de les partícules interpolades en les cel·les més properes, du a terme els càlculs en la graella, i finalment es fa la interpolació de manera inversa i es produeix el desplaçament de la partícula.

Tot i això *MPM* té alguns problemes, ja que per culpa de la simplificació que es produeix a la graella no som capaços de representar velocitats discontinües, o col·lisions molt concretes amb objectes de menor resolució de la graella.

### 1.1.2 Formulació del problema

Podríem considerar que hi ha dos tipus de simulacions: les que tenen com a objectiu emular la realitat, i les que busquen obtenir una animació visualment realista.

Les simulacions que imiten la realitat tendeixen a requerir moltíssima precisió i algorismes poc optimitzables, les quals acostumen a necessitar ser processades en un supercomputador per obtenir resultats adequats en un temps raonable.

Per altra banda, la simulació amb objectius només visuals (com és el nostre cas) intenta ser ràpida, visualment realista i, sobretot, configurable per a poder triar quins resultats es volen aconseguir. Encara que no s'intenti emular la realitat al cent per cent, és important imitar-la al màxim acceptant generalitzacions i truncaments per a accelerar la velocitat de computació; depenent de la situació podríem simplificar el model el suficient per arribar a tenir una simulació en temps real.

A causa del fet que es busca trobar una fidel i ràpida aproximació a la realitat, no existeix una solució analítica (ni general ni específica) per a produir aquestes simulacions. També s'ha de tenir en compte quina és la versatilitat de la simulació, perquè el mateix algorisme implementat limitarà les propietats del fluid o sòlid. Finalment també hi ha la possibilitat, i el problema, de fer interaccionar el fluid a simular amb algun cos extern (tant en moviment com immòbil) i la transmissió de forces entre aquests.

Així doncs, aquest treball pretén estudiar com plantejar solucions per a implementar les qüestions anteriors, i alhora aplicar-les per a conèixer les limitacions i habilitats de cadascuna, en un determinat sistema; i també servir com a base introductòria a aquesta temàtica tan dispersa i tècnica a qualsevol lector.

### 1.1.3 Actors implicats

En aquesta secció es definiran els actors implicats, de manera directa i indirecta, en el projecte.

**Director** Figura essencial pel correcte desenvolupament del projecte, i màxim responsable en la guia i consell al desenvolupador del treball. La seva acció és clau per determinar errors en el projecte, tant de tipus proposicional com executius. En aquest cas el director és Toni Susín Sánchez, del departament de Matemàtiques Aplicades de la UPC, i cap del *Dynamic Simulation Lab* inclòs en el grup de recerca ViRVIG.

**Desenvolupador** La persona a càrrec de dur a terme la recerca, documentació i implementació del software requerit, així com de redactar la memòria del projecte sota la guia del director. També recau sobre aquest actor la gestió del projecte, i és en última instància la persona responsable de complir amb els terminis del treball. Aquest rol recau sobre Pol Martín Garcia, alumne de la FIB de la branca de computació.

**Entorn** El centre de treball ViRVIG (Visualització, Realitat Virtual i Interacció Gràfica) ha facilitat un entorn de treball per a poder usar i provar el projecte amb la potència de còmput necessària, i aprovisionat d'un clima de treball per al correcte desenvolupament del projecte.

**Usuaris** Aquest treball busca crear una dissertació i ampliació dels simuladors d'avui dia, i com a tal no tindrà usuaris directes. Qualsevol avanç en *solvers* de tota mena, però, té un impacte parcial en la investigació i implementació de noves tècniques d'animació; com per exemple en efectes especials de pel·lícules, com a *Big Hero 6*, *Zootopia*, o inclús *Frozen* de Disney [3] per a simular la neu.

**Beneficiaris** Perquè aquest projecte no crearà un producte, com s'ha dit abans, tampoc hi ha beneficiaris directes. No obstant això, altres estudiants d'enginyeria informàtica o matemàtiques poden trobar aquest projecte interessant com a base d'aprenentatge, i usar-lo com a punt de partida per a futurs projectes centrats en l'àmbit de simulació de fluids en *MPM*.

## 1.2 Justificació

Aquesta temàtica és una àmplia àrea d'estudi des de fa diverses dècades, que s'ha anat dividint en múltiples tècniques i cada branca d'estudi va aprofundint en paral·lel de la resta.

En aquest treball estudiarem i implementarem un *solver MPM*, que neix de la generalització de *Particle in Cell* (PIC) i *Fluid Implicit Particle Method* (FLIP) [2]. Hi ha hagut molt nombrosos i extensos treballs de FLIP en els darrers anys [4, 5], mentre que *MPM* ha sigut realment introduït i començat a estudiar recentment perquè, al contrari que FLIP, aquest pot tractar amb fluids i sòlids comprimibles.

D'aquesta manera, *MPM* s'està començant a estudiar i a usar per a simular diferents tipus d'interaccions innovadores. Com s'ha dit abans, ressalta la simulació de neu de Disney [3] o l'interacció entre o amb múltiples materials deformables[6], per posar alguns exemples. Es pot trobar més informació dels treballs en *MPM* a [7].

Just molt recentment ha sorgit una nova tècnica per a resoldre *MPM*, anomenada *MLS-MPM* (*Moving Least Squares Material Point Method*)[8] que, generalment, busca millors aproximacions a la graella gràcies a una *B-spline*<sup>1</sup> calibrada amb *MLS*.

D'aquesta manera, i amb base dels nous desenvolupaments de Yuanming Hu et. al. a [8] i usant com a base [9] dels mateixos autors, estudiarem aquesta tècnica i possibles diferents interpretacions del model. És a dir: adaptarem la solució de Yuanming a unes necessitats personals i investigar com reacciona el sistema a unes noves condicions.

## 1.3 Abast

En aquesta secció definirem quines són les fites del projecte, i la seva estructura progressiva.

### 1.3.1 Objectius

**Aprenentatge de nous conceptes** Donat que el treball requereix de coneixements allunyats dels adquirits en una enginyeria informàtica, cal una cerca intensiva d'informació i aprenentatge de mètodes numèrics, càlcul, i àlgebra lineal. Tots aquests conceptes són necessaris per entendre els diferents models d'objectes i simuladors a implementar.

En concret, existeixen alguns subtemes que mereixen una menció especial com a subobjectius:

- **Les equacions de Navier-Stokes** Aplicant la segona llei de Newton al moviment del fluid, amb algunes assumpcions de les forces internes (anomenades estrès), s'obtenen les equacions de Navier-Stokes; conegudes per la seva complexitat, que ja de per si obren una interessant àrea d'estudi de possibles implicacions i aplicacions.

És important entendre aquestes equacions, i saber interpretar-les, doncs descriuran el nostre sistema i són la base de qualsevol *solver*.

- **Càlcul vectorial** Fins a un cert nivell, és important i necessari conèixer diferenciació i integració de camps vectorials, doncs tractarem amb camps de partícules en aquest projecte.

---

<sup>1</sup>Funció corba; és detallarà en futurs apartats.



· **Tècniques de solvers** Finalment, amb els coneixements teòrics necessaris, s'ha d'aprendre el funcionament intern i els models que permeten simular els diferents tipus de *solvers* creats, en concret el de *MLS-MPM* [8].

**Implementació d'un solver bidimensional** Per a una primera versió, serà implementat un simulador de fluids de partícules bidimensional, seguint [9, 8] i programant la base de tot el projecte a una escala menor, per a fer proves i trobar errors. La idea és usar aquesta primera fase com a indicador, i prendre decisions d'acord amb els resultats obtinguts per a millorar el resultat final, i evitar errades.

L'objectiu d'aquesta etapa és el refinament de l'entorn de treball per a una posterior fase molt més complexa, i d'aquesta manera poder evitar nombrosos possibles errors, estudiar inconvenients que puguin aparèixer i, sobretot, com a guia d'aprenentatge progressiu de manera simplificada.

**Millora del solver a la tercera dimensió** El canvi de dimensió és un salt difícil, i essencial en aquest projecte, doncs tot simulador gràfic per a resultats realistes requereix aquesta tercera dimensió. Serà necessari redimensionar la interfície gràfica, les estructures de dades, i el mateix algorisme.

A partir d'aquesta fita, és quan s'inicien els tests i objectius secundaris.

· **Interacció entre diversos materials viscoelàstics diferents** Afegir suport per a poder mantenir en el simulador alhora diversos fluids de diferents característiques cadascun.

· **Sistema de físiques** Implementar un sistema d'interacció de cossos sòlids en moviment amb el *solver*, per aconseguir simular col·lisions amb un, o diversos, elements externs al simulador; com per exemple una roda amb sorra.

**Altres objectius** En la mesura del possible, s'intentarà dur a terme els següents objectius, que no són necessaris per al correcte desenvolupament del projecte, però el completen i ajuden a veure com tractar i solucionar altres problemes inherents en qualsevol sistema programat.

· **Eficiència** Sobretot en el *solver* tridimensional, l'eficiència i velocitat del programa és un problema, ja que aproximadament el còmput de cada fotograma serà major a un minut segons els articles de *MPM* [8]; per tant sempre que es pugui s'ha d'implementar el simulador sent conscients de l'arquitectura i de l'eficiència del codi.

· **Paral·lelisme** Aconseguir que el codi funcioni de manera paral·lela pot suposar un increment de la velocitat de computació molt important. Així doncs, qualsevol oportunitat de paral·lelitzar el codi s'haurà d'aprofitar.

### 1.3.2 Requisits

Podem separar els requisits del software a implementar en funcionals, i no funcionals. Els primers representen funcionalitats específiques que el nostre sistema haurà de complir, i el segon especifica criteris per jutjar els comportaments específics d'aquest.

#### Requisits funcionals

**Simulació d'un sistema personalitzable** Amb modificació del nombre, tipus, i posició de les partícules; canvi del medi (gravetat, mida de la graella, etc). Cal valorar si han de ser paràmetres modificables entre execucions, o constants en el codi.

**Exportació de la simulació** Exportació dels fotogrames produïts pel *solver*, o bé en arxius binaris per al seu posterior tractament, o bé en imatges *PNG* (*Portable Network Graphics*).

**Simulació d'objectes viscoelàstics** Importació de models de triangles, i simular-ne el comportament en unes condicions especificades com a materials elàstics o viscoelàstics.

#### Requisits no funcionals

**Velocitat de computació** La velocitat del còmput ha d'estar justificada i ser adient; ha de ser linealment dependent del nombre de partícules o a la mida de la graella.

**Estabilitat de la simulació** Les solucions del *solver* han de ser estables, i no descontrolar-se; en cas de ser impossible evitar la inestabilitat, aquest fet estarà documentat i estudiat, marcant els límits del simulador.

### 1.3.3 Obstacles i riscos

Per evitar imprevistos en el transcurs del projecte, analitzarem els obstacles que poden sorgir durant aquest procediment i, sobretot, com minimitzar el seu impacte.

#### Errors en el codi

És inevitable cometre petits errors en la implementació d'un software, que a vegades poden passar desapercebuts, i altres vegades poden trencar totalment el programa.

Donat que un simulador no pot ser testejat amb jocs de proves, serà necessari provar manualment cada nova característica afegida, i comprovar que totes les anteriors propietats segueixen funcionant adequadament. D'aquesta manera podrem minimitzar els riscos de perpetuar un error durant les diverses versions del software.

### Problemes o insuficiència de hardware

Els ordinadors poden tenir problemes, els discos poden fallar, la memòria RAM pot deixar de funcionar, els arxius es poden corrompre, etc. El qual suposa una pèrdua de temps important a curt termini per a arreglar el hardware espatllat i recuperar les dades perdudes.

Per evitar l'impacte d'aquest problema en la mesura de tot el possible, el projecte i la memòria d'aquest es guardaran per versions en un repositori a la xarxa, més concretament a *GitHub*. A més a més, la memòria es guardarà en paral·lel al *OneDrive* de Microsoft.

Per altra banda, durant les etapes més avançades del projecte el temps de simulació es veurà incrementat quantitativament, cosa que provocarà que per provar els canvis siguin necessàries hores.

Per a evitar perdre temps sense avançar adequadament, els tests (en forma d'execucions arbitràries del programa) es deixaran executant en una màquina diferent de la que es treballa, amb molta més potencia (com un dels ordinadors del ViRVIG) per obtenir els resultats tan aviat com sigui possible, sense entorpir el desenvolupament.

### Error de comprensió dels models matemàtics

A causa de la complexitat de les matemàtiques que hi ha darrere d'aquest projecte, i al fet que aquest és conceptualitzat des del punt de vista informàtic, és possible errar en l'enteniment de nombrosos conceptes tècnics en, per exemple, els models elàstics dels objectes, el procediment del simulador, o en simples formalismes d'expressivitat matemàtica.

### Problemes temporals

Les limitacions temporals són inherents a tot projecte de fi de grau, i aquest no n'és una excepció. Si hi ha una mala gestió del temps, o algun tipus d'imprevist, podem trobar-nos amb la impossibilitat de complir tots els requisits i objectius del treball a temps.

Per a aquest motiu és important escollir i seguir una metodologia, creant una planificació per a dur a terme el projecte. D'aquesta manera podem ser molt més conscients de l'avenç que s'està fent, i de ser necessari poder realitzar a temps modificacions a l'abast del mateix.

## 1.4 Metodologia i rigor

En qualsevol projecte de mínima magnitud, s'ha de triar i adherir-se a un sistema organitzatiu concret. D'aquesta manera podem tenir una visió molt més acurada del treball, i mantenir un ordre en aquest; a més de tenir sempre un procediment a seguir en cas de qualsevol inconvenient.

### 1.4.1 Metodologia

La metodologia emprada en el projecte per a la seva organització i concepció és la *Kanban* [10]. Aquesta és una metodologia àgil molt semblant a *Scrum*, però dissenyada per a mantenir un flux constant de resultats evitant discretitzar el projecte en *sprints*. Aquest mètode aporta molta flexibilitat i permet focalitzar el treball en només el que es considera essencial.

La principal idea darrere d'aquest mètode és mantenir sempre alguna tasca en procés, de manera que quan s'ha acabat quelcom, la següent tasca de més prioritat entra en acció. A més a més, s'ha d'evitar mantenir més d'una tasca en procés alhora, limitant la quantitat de *work in progress* al mínim, i centrar-se en el més important en aquell moment.

En cas de tenir qualsevol inconvenient, aquesta metodologia és molt responsiva als canvis i ens permet, ràpidament, corregir qualsevol aspecte mancant, alhora que es valora la seva prioritat amb el director del projecte.

### 1.4.2 Eines de seguiment

Per al correcte seguiment del projecte, el seu estat i les seves versions, s'usen diverses eines. Primerament és important tenir una visió global del que es porta fet, i el que queda per fer; per a això s'usa el mètode de *Bullet Journaling* per a ser conscients en tot moment de l'estat del projecte, a un nivell personal.

A escala de codi, un historial de versions, organitzat adequadament per funcionalitats, és emmagatzemat a *GitHub* juntament amb les diverses versions de la memòria del projecte. Amb aquesta eina podem tenir una visió ordenada del procés evolutiu del treball, veient les funcionalitats per separat.

Finalment, també es duren a terme reunions periòdiques amb el director on es revisaran, entre altres coses, els avanços fets des de l'última reunió, l'estat del projecte, i dubtes i qüestions sorgides. Donada la naturalesa del projecte, en aquestes reunions es duren a terme explicacions teòriques de conceptes no vistos a la FIB per al correcte desenvolupament del projecte.

En cas de sorgir problemes o imprevistos, es contactarà amb el director i, de ser necessari, es crearà un pla de contingència o una ruta a seguir per a solucionar qualsevol inconvenient.

## 1.5 Planificació temporal

Aquest projecte es realitzarà en el marc temporal d'un quadrimestre, i en les següents pàgines es fa una planificació en tasques i una estimació de la durada d'aquestes.

Ja que el desenvolupament del projecte no és una activitat a temps complert, s'estima un mínim de 14 hores de treball setmanal en el mateix; però s'espera una mitjana de 25 hores d'involucració a la setmana.

Concretament, el projecte s'inicia a mitjans de juliol del 2019, amb la seva concepció, i s'allargarà fins a inicis de gener de l'any 2020 per al torn de lectures del mateix mes.

### 1.5.1 Descripció de tasques

A continuació es descriuen les diferents tasques a dur a terme en el projecte. Les tasques engloben fins a 4 subtasques, com a molt, per a poder oferir el màxim de detall en el mínim espai de document possible, les quals seran descrites i especificades. També s'afegeix la seva estimació temporal i les dependències entre elles. La taula 1.1 mostra un resum d'aquesta secció a la pàgina 17.

#### Base del projecte i entorn de treball

La primera tasca del projecte consisteix en la creació de les eines més bàsiques i elementals, i la familiarització amb aquestes, que seran necessàries per a acompanyar la implementació del *solver*. També es codificarà una base adaptable per a interactuar amb el simulador en el futur.

Les eines a programar consisteixen en, usant la llibreria gràfica *OpenGL* [11], un visualitzador per pantalla de partícules per a veure en temps real els resultats; i seguidament també serà necessari implementar una eina per exportar els fotogrames en imatges, usant la llibreria *open source stb\_image* [12].

**Previsió temporal** 20 hores per al desenvolupament del visualitzador i la base del projecte, perquè han d'anar acoblades i requereixen certa planificació. 5 hores per a l'exportador d'imatges, ja que no és una eina complexa i no és innovadora, però es requereix prendre decisions sobre el format d'imatge a exportar.

**Dependències** Aquesta primera tasca no té cap dependència, més enllà del fet que abans de poder implementar quelcom, un s'ha de familiaritzar amb el concepte.

#### Simulador de partícules simple

Fase que té com a objectiu principal l'aprenentatge del funcionament dels simuladors *MLS-MPM*, implementant la seva versió més simple a sobre de la base creada a l'anterior etapa.

Aquest primer *solver* serà molt semblant al codi de *2D-MPM* en 88 línies de [8], però portat al terreny del projecte.

És important denotar que aquesta fase està dividida en dos: primerament s'ha de dur a terme un aprenentatge exhaustiu de la nova temàtica, i posteriorment l'anomenada implementació.

Es finalitzarà comprovant que la tasca ha estat realitzada correctament, mitjançant tests manuals, i es corregiran tots els errors trobats i esperats.

**Previsió temporal** A causa de la gran quantitat de continguts a estudiar, s'assignen 65 hores dedicades a l'aprenentatge, pràctica i assoliment de conceptes nous. Posteriorment, s'estimen 25 hores per a la implementació, i 10 hores extres per a dur a terme la validació i correcció d'errors.

**Dependències** D'aquesta tasca, a l'estar dividida en dos, només una part té dependències. L'aprenentatge es pot dur a terme de forma lliure; però la implementació requereix aquest aprenentatge previ, i que ja existeixi una base programada.

### Adició de model elàstic

De la mateixa manera que la fase anterior, aquesta està dividida en aprenentatge i implementació, però dels models elàstics a incorporar en el simulador. En concret s'estudiarà *CPIC* desenvolupat a [8], basat en [7], i la seva addició de *fixed corotated model* afegint l'estrès a *CPIC* [9, 7].

Finalment, s'haurà de validar el funcionament dels models elàstics.

**Previsió temporal** Perquè l'estudi de models elàstics és d'un nivell molt avançat, i d'una branca molt separada a la d'informàtica, s'estima unes 60 hores màximes per al seu aprenentatge; posteriorment pel seu disseny i implementació s'assignen 30 hores; més 8 hores de validació i correcció d'errors. Aquestes estimacions són degudes a la inestabilitat dels mètodes a implementar.

**Dependències** Igual que a l'anterior tasca, en aquesta només la implementació de models elàstics requereix del seu aprenentatge i de l'existència d'un simulador de fluids de partícules.

### Optimització i estudi 2D

Juntament amb el director, s'analitzarà el *solver* actual i s'estudiarà quines zones requereixen alguna millora, o poden admetre optimitzacions interessants. També es discutirà sobre com plantejar-se la migració del simulador de dues dimensions a una tercera.

Durant aquesta etapa també es veuran les oportunitats de paral·lelitzar el codi i millorar-ne la precisió del simulador, juntament amb la seva posterior implementació.

En aquesta fase també incorporem l'estudi del simulador, juntament amb el perfeccionament dels paràmetres del mateix, i els diferents comportaments depenent de l'entrada.

**Previsió temporal** S'estimen 30 hores de feina per a millorar i canviar fragments de codi necessaris per a una bona evolució. Per a implementar paral·lelisme s'assignen a part 15 hores a causa de la diferenciació de tipologia de problema que suposa.

**Dependències** Per a poder dur a terme l'estudi, és necessari que el simulador estigui complet i funcional.

**Recursos** En aquesta tasca, per a provar el paral·lelisme serà necessària una màquina amb diversos nuclis.

### Primera documentació

L'inici de la documentació, planificació, definició de l'abast i anàlisi detallat de l'enfocament del projecte es du a terme en aquesta tasca, a la tercera part del desenvolupament del projecte, per a puntualitzar el procediment a seguir per al seu correcte desenvolupament.

Tots aquests detalls es plasmaran en una memòria del projecte, efectuada en *LaTeX* [13], adjunta a la documentació i experimentació duta a terme.

És important puntualitzar el desenvolupament d'altres tasques en aquesta etapa com l'anàlisi de sostenibilitat, la planificació temporal, gestió econòmica i de riscos, i la metodologia del projecte.

**Previsió temporal** S'estimen unes hores invertides en aquesta etapa majors que les desitjables, perquè també s'hauran d'aprendre conceptes nous i redactar alhora. S'usaran 35 hores dedicades a l'escriptura de documentació, i 20 a l'estudi, planificació i concepció dels temes a tractar. S'afegeixen 10 hores per a realitzar correccions i replanificar el necessari.

**Dependències** La primera documentació no s'iniciarà fins que ja s'hagi tingut un tast en el desenvolupament del projecte, per a poder entendre'l bé i poder estimar el desenvolupament del mateix amb conseqüència.

### Millora a 3D

Aquesta etapa tracta íntegrament la translació del projecte a la nova dimensió, a discutir el procediment amb el director del projecte arribat el moment, doncs hi ha moltes maneres diferents d'aproximar aquest canvi depenent del desenvolupament del treball.

De totes formes, podem enumerar les característiques que s'hauran de canviar concretament:

- Estructures de dades: s'ha d'incorporar la nova coordenada a emmagatzemar per la posició de les partícules, i modificar la graella perquè sigui tridimensional.
- Model elàstic: La matemàtica darrere el model requereix canvis importants per a poder funcionar.
- Algorisme: Per a adaptar els canvis anteriors, alguns retocs a l'algorisme s'hauran de fer.
- Visualitzador: S'ha d'idear alguna forma de poder visualitzar una zona de l'espai 3D de la simulació.

Un cop els canvis han sigut implementats, és necessari dur a terme una validació intensiva i exhaustiva d'aquestes modificacions, doncs no són trivials i requereixen molta especialització.

**Previsió temporal** En aquesta etapa es dedicarà 30 hores a la implementació i gestió de canvis en el simulador; i posteriorment 30 hores més de validació de resultats, donat que es tracta d'una etapa molt delicada on la velocitat del sistema es reduirà, i la complexitat augmentarà.

**Dependències** La millora a la tercera dimensió requereix de l'existència d'un *solver* finalitzat en dues dimensions, optimitzat, i estudiat per a facilitar aquesta tasca.

**Recursos** En aquesta tasca, és necessari el PC del *ViRVIG* per a dur a terme la validació.

## Estudi 3D

Amb el simulador correctament implementat, podem passar a l'etapa de documentació, estudi, i perfecció dels paràmetres del simulador. Durant aquesta fase es documentaran totes les modificacions finals dutes a terme durant el canvi a la tercera dimensió, i perfeccionar la configuració del *solver* per a obtenir resultats adequats.

Si es valora necessari, amb l'estudi efectuat s'implementaran i es provaran aquelles millores que siguin senzilles i directes, ja que la complexitat del disseny de modificacions en una dimensió extra, i la posterior validació d'aquestes, seria molt costosa.

**Previsió temporal** S'assignen 10 hores de documentació de treball en la millora a 3D; 20 hores més d'estudi dels resultats i paràmetres del *solver*, i 30 més de valoració, implementació i validació de millores.

**Dependències** L'anàlisi del simulador tridimensional requereix de la correcta implementació d'aquest.



**Recursos** Per a la validació serà necessari el PC del *ViRVIG*, i aprofitar la seva potència.

## Físiques

Si la planificació no és esbiaixada de manera important, es podrà dur a terme aquesta etapa.

Diferenciant l'aprenentatge de la implementació, es busca arribar a codificar un sistema de col·lisions entre objectes externs al simulador i les partícules d'aquest, de manera que es puguin dur a terme animacions amb interaccions.

Serà necessari, primer de tot, implementar un sistema de càrrega de models simples tridimensionals, i posteriorment un mecanisme d'interacció i transmissió de forces unidireccional (del model al simulador), com els mostrats a [?].

Serà necessari validar, estudiar i documentar adequadament aquesta etapa, doncs portarà alguns dels resultats més interessants de tot el projecte.

**Previsió temporal** S'estima un temps d'estudi de 20 hores; i un temps d'implementació de 30 hores al complet. S'ha d'afegir unes 20 hores de validació de resultats, i unes altres 10 per a codificar eines externes de càrrega i prova de models.

**Dependències** Com que aquesta tasca està diferenciada en estudi i implementació, només la darrera requerirà l'aprenentatge de físiques i d'una codificació finalitzada d'un *solver* 3D.

## Documentació final

Finalitzant el projecte, s'ha de consolidar la documentació i la memòria d'acord amb el software creat, i tancat ja per a aquesta etapa. Aquí se sintetitzaran tots els resultats del programari, els estudis i descobertes interessants.

Serà important generar exemples per a poder mostrar les funcionalitats dels resultats del treball, i per justificar les conclusions.

Per últim s'hauria de tenir molt en compte la reunió de seguiment, que a ser possible es realitzarà ben entrada aquesta etapa, per a l'adequada finalització del projecte i la respectiva preparació de l'entrega final, la seva correcció, i la presentació final del treball de fi de grau.

**Previsió temporal** S'assignen 60 hores de treball en la memòria i documentació final, dividides en 18 hores per documentar el programa, 30 hores per a redactar el document de tot el que no ha estat ja escrit, i 12 hores més de correcció d'errors i revisió. Finalment també s'han de tenir en compte unes 20 hores de preparació de la defensa final.

**Dependències** En ser l'etapa final del projecte, dependrà de totes les fases anteriors.

### 1.5.2 Taula resum

A continuació s'adjunta la taula 1.1, que mostra l'estimació d'hores de les múltiples tasques del projecte, així com les dependències entre elles. En cas de cascada de dependències, només s'indica l'últim prerequisit.

Id	Nom	Hores	Dependències
1	Creació de la base	25	no
2.1	Aprenentatge <i>solver</i>	65	no
2.2	Implementació <i>solver</i>	25	1, 2.1
2.3	Validació <i>solver</i>	10	2.2
3.1	Aprenentatge models elàstics	60	no
3.2	Implementació models elàstics	30	2.3, 3.1
3.3	Validació models elàstics	8	3.2
4.1	Estudi 2D	30	3.3
4.2	Paral·lelisme	15	4.1
5.1	Documentació: planificació	20	2.3
5.2	Documentació: redacció	35	2.3, 5.1
5.3	Documentació: correcció	10	5.2
6.1	3D: migració	30	4.2
6.2	3D: validació	30	6.1
7.1	Estudi 3D	20	6.2
7.2	Documentació 3D	10	6.2
7.3	Estudi 3D: millores i validació	30	7.1
8.1	Aprenentatge físiques	20	no
8.2	Implementació físiques	30	7.3, 8.1
8.3	Validació físiques	20	8.2
8.4	Implementació eines	10	8.1
9.1	Documentació programa	18	8.3
9.2	Redacció memòria	30	9.1
9.3	Revisió i correcció	12	9.2
9.4	Preparació defensa final	20	9.3

Taula 1.1: Taula resum de tasques. Elaboració pròpia

### 1.5.3 Diagrama de Gantt



Figura 1.1: Diagrama de Gantt. Creat amb [14]. Elaboració pròpia.

#### 1.5.4 Planificació d'alternatives

En qualsevol projecte d'aquestes magnituds, tota planificació es pot esbiaixar a causa d'obstacles i inconvenients (tant directes com indirectes) que poden sorgir en qualsevol etapa, i ser de tipologies molt diverses. Tot això, en definitiva, sempre comportarà una pèrdua important de temps que cal tenir en compte.

Abans de res, és important notar que les duracions estimades de les tasques estan fetes a l'alça, i atès que usem una metodologia *Kanban* en acabar una tasca es començarà immediatament amb la següent. Aquest petit detall ens pot acabar proporcionant un marge de temps extra per a tractar qualsevol imprevist a sorgir.

Si succeís que una tasca essencial ens importunés, seria necessari endarrerir l'entrada en escena de les consecutives etapes de manera estratègica.

En qualsevol cas però, sempre s'haurà de discutir el pla d'acció amb el director del projecte durant les reunions continuades, i sobretot en cas d'aparèixer un problema.

De ser necessari, s'eliminaran etapes del projecte, amb l'objectiu de què aquest acabi satisfactòriament a temps.

Una altra opció és substituir una etapa d'implementació i estudi, per una basada en teoritzar el funcionament de quelcom d'acord amb els resultats obtinguts, sense requerir més recursos. Així podem reduir el temps d'una tasca de manera important, tot i que els resultats no seran tangibles.

## 2. Referències

- [1] D. Terzopoulos and K. Fleischer, “Modeling inelastic deformation: Viscoelasticity, plasticity, fracture,” *SIGGRAPH Comput. Graph.*, vol. 22, pp. 269–278, June 1988.
- [2] D. Sulsky, S. J. Zhou, and H. L. Schreyer, “Application of a particle-in-cell method to solid mechanics,” *Computer Physics Communications*, vol. 87, pp. 236–252, may 1995.
- [3] A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle, “A material point method for snow simulation,” *ACM Transactions on Graphics*, vol. 32, no. 4, 2013.
- [4] R. Bridson, *Fluid Simulation for Computer Graphics*. A K Peters/CRC Press, nov 2018.
- [5] Y. Zhu and R. Bridson, “Animating sand as a fluid,” *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 965–972, 2005.
- [6] J. Hegemann, C. Jiang, C. Schroeder, and J. M. Teran, “A level set method for ductile fracture,” in *Proceedings - SCA 2013: 12th ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pp. 193–202, 2013.
- [7] C. Jiang, C. Schroeder, J. Teran, A. Stomakhin, and A. Selle, “The material point method for simulating continuum materials,” *ACM SIGGRAPH 2016 Courses, SIGGRAPH 2016*, pp. 1–52, 2016.
- [8] Y. Hu, Y. Fang, Z. Ge, Z. Qu, Y. Zhu, A. Pradhana, and C. Jiang, “A moving least squares material point method with displacement discontinuity and two-way rigid body coupling,” *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, p. 150, 2018.
- [9] Y. Hu, Y. Fang, Z. Ge, Z. Qu, Y. Zhu, A. Pradhana, and C. Jiang, “High-performance material point method (MPM) solver for Taichi. (ACM Transactions on Graphics, SIGGRAPH 2018).”
- [10] D. J. Anderson, *Kanban: Successful Evolutionary Change For Your Technology Business*. 2010.
- [11] Silicon Graphics and Khronos Group, “OpenGL - The Industry Standard for High Performance Graphics,” 2013.
- [12] “stb: stb single-file public domain libraries for C/C++.”
- [13] S. Nelson, “LATEX: A document preparation system user’s guide and reference manual,” *Computers & Mathematics with Applications*, vol. 29, no. 11, p. 108, 1995.
- [14] Teamgantt, “Online Gantt Chart Software — TeamGantt,” 2019.