

# SVILUPPO DI UNA PIATTAFORMA INTEGRATA PER LA SIMULAZIONE DI SCENARI D'INCENDIO

N. De Santis<sup>1</sup>, F. Micchetti<sup>2</sup>, V. Primavera<sup>2</sup>, R. Lala<sup>3</sup>, P. Cinnella<sup>1</sup>

**1** Università del Salento, Dip. di Ingegneria dell'Innovazione, Via per Monteroni, 73100 Lecce, Italia

**2** EnginSoft s.p.a., Via Marconi 207, Mesagne, 72023 Brindisi, Italia

**3** Comando Provinciale Vigili del Fuoco di Lecce, Viale Grassi 86, 73100 Lecce, Italia

## SOMMARIO

In base al DM del 09/05/07 recante “Direttive per l’attuazione dell’approccio ingegneristico alla sicurezza antincendio” viene recepita la necessità di affiancare alla metodologia prescrittiva quella prestazionale in materia di prevenzione incendi. Tale metodologia prevede una valutazione del livello di sicurezza tramite un’accurata analisi del rischio di incendio, basata sulla simulazione del tipo di incendio che può ragionevolmente svilupparsi in quella determinata attività e delle conseguenze che da esso possono derivare, definendo, per ogni singolo caso, i fattori di rischio e gli obiettivi da raggiungere per ottenere una sicurezza equivalente a quella prescritta dalle cosiddette norme verticali. Oggi il quadro normativo trova un importante strumento di approfondimento in una nuova disciplina che si è affermata negli ultimi decenni, chiamata FSE (Fire Safety Engineering).

Il presente lavoro descrive le fasi di sviluppo di una piattaforma software per lo studio dell’evoluzione di un incendio nel tempo e nello spazio per via numerica. Il progetto si basa su un algoritmo di calcolo che risolve un modello fluidodinamico mediante tecniche CFD (Computational Fluid Dynamics). Il codice di simulazione incendi in questione è il ‘Fire Dynamics Simulator’ (FDS), elaborato dal NIST (National Institute of Standards and Technology). Tale modello risolve numericamente le equazioni di Navier-Stokes applicate ad un incendio, con particolare attenzione alla propagazione di fumo e calore.

L’attività è finalizzata alla realizzazione di un codice generatore di scenari, strumento la cui funzione è quella di permettere all’utente della piattaforma software di creare diverse situazioni di incendio al variare delle condizioni al contorno e delle variabili di processo. La soluzione tecnologica sviluppata, per la realizzazione dell’obiettivo prefissato, consiste in una piattaforma informatica che, integrandosi con il codice fluidodinamico FDS, è in grado di assolvere alla funzione di pre-processore per lo stesso, e che quindi fornisce la possibilità, tramite una interfaccia grafica, di riprodurre sotto forma di CAD 3D la geometria, di impostare le proprietà termo-fisiche della struttura edilizia presa in esame, e di elaborare i dati forniti, generando automaticamente il dati di input richiesti dal solutore FDS. L’attività è stata condotta approcciando il lavoro secondo le *best-practice* solitamente utilizzate in campo informatico.

Allo stato attuale di avanzamento del progetto sono disponibili i seguenti elementi:

- 1) Algoritmo di processamento automatico della geometria e delle condizioni al contorno.
- 2) Versione preliminare dell’interfaccia grafica

Questi due punti verranno descritti nel dettaglio nei paragrafi che seguono. Si vedrà infine come i software sviluppati rappresentino uno strumento di grande valore nel settore della progettazione antincendio, in vista della semplificazione che essi apportano all’utilizzo del codice sopraccitato FDS.

## 1.0 INTRODUZIONE

Il presente studio mira ad affrontare la prevenzione incendi in chiave ingegneristica tramite un modello progettuale, da realizzare tramite l’utilizzo di codici di calcolo numerico.

Tali codici, tramite la soluzione delle equazioni di termofluidodinamica correlate al fenomeno incendio, implementano modelli in grado di simulare e prevedere lo sviluppo degli incendi in funzione della geometria degli ambienti interessati e delle caratteristiche costruttive di questi.

I modelli di simulazione permettono, inoltre, di analizzare i diversi possibili scenari conseguenti ad un incendio, variandone le condizioni al contorno.

Lo studio dei fenomeni riguardanti i diversi scenari d’incendio risale all’incirca ai primi anni di utilizzo dei calcolatori. Alla base dello studio dei diversi scenari vi sono le equazioni di governo della fluidodinamica, del trasferimento del calore e le leggi che regolano il processo di combustione. La difficoltà principale che si incontra nello studio di tali fenomeni è, essenzialmente, la grande quantità sia degli scenari di incendio, dovuta alla natura accidentale del fenomeno, sia delle variabili che condizionano il tipo di scenario. Si pensi,

per esempio, alla possibilità di flussi multi fase, alla presenza di moti turbolenti, al trasporto radiativo di calore, allo scambio termico combinato (conduzione, convezione ed irraggiamento).

Allo stato attuale, esistono ben pochi codici di calcolo utilizzabili per risolvere problemi di questo tipo; uno di questi è l’FDS (Fire Dynamics Simulator), elaborato dal NIST (National Institute of Standards and Technology) [1],[2]. FDS è un simulatore CFD (Computational fluid dynamics). Esso risolve una forma delle equazioni di Navier-Stokes per flussi a basse velocità dominati dagli effetti termici, avendo come obiettivo il calcolo accurato del trasporto del calore e dei fumi all'interno dell'ambiente in cui ha luogo l'incendio. Le equazioni di conservazione della massa, della quantità di moto e dell'energia sono approssimate mediante un metodo alle differenze finite, mentre la radiazione termica è valutata con la tecnica dei volumi finiti sulla stessa griglia utilizzata per la risoluzione delle equazioni di flusso. Il simulatore utilizza inoltre un approccio di tipo Lagrangiano per simulare il moto del fumo e l’iniezione di acqua allo stato liquido dagli sprinklers.

Data la natura delle metodologie numeriche utilizzate, tale simulatore permette di prendere in considerazione solo geometrie di tipo parallelepipedo, discretizzate mediante griglie di calcolo cartesiane. Tuttavia nell’architettura moderna, così come in molti edifici storici, ricorrono spesso geometrie non parallelepipede. Nasce pertanto la necessità di un simulatore che permetta di prendere in considerazione ambienti con geometria complessa per l’analisi antincendio.

Attualmente l’unico modo per trattare geometrie non parallelepipede con l’FDS consiste nello scrivere manualmente, all’interno del file contenente i dati di input, le coordinate di una serie di blocchi solidi che descrivono la geometria non parallelepipeda. Il risultato è un’approssimazione “a gradini” delle superfici curve, tanto più accurata quanto maggiore è il numero di blocchi utilizzati. Questo approccio, allo stato attuale di sviluppo dell’FDS, comporta un intervento manuale da parte dell’utente che diventa tanto più oneroso, e al limite addirittura ingestibile, quanto maggiore è la complessità della geometria considerata e il livello di accuratezza voluto (proporzionale, dunque, al numero di blocchi parallelepipedi necessari ad approssimare la geometria).

Nel presente lavoro verranno presentate le caratteristiche del programma di pre-processing che rende completamente automatica la procedura di generazione dei blocchi parallelepipedi e dei relativi dati di input per l’FDS a partire dalla geometria (più o meno complessa) fornita dall’utente in formato STL (Sthelolitography), sviluppato dall’Università del Salento e già presentato al convegno nazionale sulla 'Sicurezza nei sistemi complessi', tenutosi al Politecnico di Bari dal 16 al 18 Ottobre 2007 . Verrà inoltre presentata la versione preliminare dell’interfaccia grafica grazie alla quale è possibile sia importare file di input dell’FDS che semplificare la procedura di scrittura degli stessi.

## 2.0 ALGORITMO DI PRE-PROCESSING DELLA GEOMETRIA

L’obiettivo principale del programma di pre-processing della geometria da analizzare è di approssimarla, una volta che l’utente ne abbia fornito la descrizione in formato *STL*<sup>1</sup>, mediante una serie di blocchi solidi parallelepipedi (ostruzioni). La descrizione della geometria risulterà tanto più accurata quanto maggiore sarà il numero di blocchi usati e dunque quanto più fitta risulterà la griglia di calcolo utilizzata; naturalmente occorrerà stabilire un compromesso tra accuratezza della soluzione e incremento del tempo computazionale necessario.

Il programma di pre-processing, detto “*Geometry Solver*”, compie le seguenti operazioni:

- lettura della geometria in formato STL, precedentemente generata mediante un programma CAD 3D;
- generazione della griglia computazionale;
- classificazione delle celle della griglia in base alla loro posizione rispetto alla struttura;
- scrittura dei dati di input per FDS nel file di input.

Innanzitutto è necessario disegnare la struttura presa in considerazione, con un programma di grafica 3D, che fornisca il disegno in formato STL, e che permetta di scegliere il numero di triangoli con cui si approssima la superficie.

Il geometry solver legge i dati geometrici relativi alla struttura e crea una griglia cartesiana che ingloba il modello. A tal fine, l’algoritmo individua il più piccolo parallelepipedo contenente la struttura da analizzare e lo discretizza mediante una griglia cartesiana. Le linee della griglia, in generale, non seguono i contorni

---

<sup>1</sup> La rappresentazione in formato *STL* di una superficie non è altro che una collezione di triangoli non connessi, di dimensioni inversamente proporzionali alla curvatura locale della superficie di partenza.

della struttura, ma possono intersecarli liberamente. In tale situazione, alcune delle celle della griglia risultano completamente interne allo spessore della struttura, altre esterne e altre di confine.

Per classificare le celle di calcolo in base alla loro posizione rispetto alla struttura si è adottato un algoritmo di *ray-tracing* [6], come discusso ad esempio in [7]. Tale tecnica è molto utilizzata nei programmi di grafica 3D per distinguere, ad esempio, le parti in ombra di una figura da quelle in vista.

Tramite la tecnica di *ray-tracing* si definisce la posizione di ciascun nodo di griglia rispetto al modello. Più precisamente, le celle sono classificate come interne, esterne o di confine. Successivamente le celle interne e di confine sono trattate come ostruzioni solide. A tal fine, il Geometry Solver prepara automaticamente il file di input *CHID.data* per l'FDS, scrivendo le coordinate dei blocchi solidi che definiscono la geometria della struttura considerata, nonché le coordinate delle superfici che identificano le aperture sulle superfici perimetrali ed, infine, le aperture sulle pareti interne.

Dopo aver constatato la validità del programma Geometry Solver [8], l'obiettivo attuale consiste nel verificare quali sono le capacità e potenzialità dello stesso.

Nell'articolo presentato al convegno nazionale sulla 'Sicurezza nei sistemi complessi', tenutosi al Politecnico di Bari dal 16 al 18 Ottobre 2007, è stato analizzato come esempio applicativo dell'implementazione dell'FDS con il programma che sviluppa la geometria, uno scenario d'incendio caratterizzato da un'autorimessa di dimensioni 8x15x7 metri lungo le tre direzioni del sistema di riferimento, costituita da sei ambienti con volte a crociera, al cui interno si è fatto bruciare un blocco solido parallelepipedo.

Successivamente si è puntata l'attenzione sulla procedura di lettura da parte del Geometry Solver di strutture con all'interno altri oggetti, introdotti come file stl, con in più la presenza di finestre e porte.

In particolare, si vuole presentare lo scenario d'incendio caratterizzato da una struttura cilindrica con cupola, con all'interno una serie di elementi solidi ed avente un'apertura sulle superfici curvilinee della struttura principale, che può essere creata o eliminata in istanti di tempo prefissati dall'utente. Si noti inoltre che la struttura può all'occorrenza essere suddivisa in più parti costituite da materiali diversi, e quindi con diverse caratteristiche termo-fisiche.

## 2.1 Struttura cilindrica con cupola

La struttura presa in considerazione è costituita da un cilindro con diametro di base pari a 2 metri, ricoperto superiormente da una cupola. L'altezza totale della struttura risulta di 2.10 metri. Una sua rappresentazione grafica è mostrata in Fig. 1, che presenta una diapositiva della geometria disegnata col programma cad3D Rhinoceros. Il programma di visualizzazione grafica è invece lo Smokeview. La griglia di calcolo ha un numero di nodi pari a 90x90x90 lungo i tre assi di riferimento. All'interno della struttura considerata sono stati disegnati e salvati in formato Stl diversi elementi solidi, quali un tavolo, un tappeto, due panche ed una finestra circolare. Il file che definisce le caratteristiche di questi elementi viene letto in maniera automatica dal Geometry Solver.

In particolare, la struttura principale è caratterizzata da superficie inerte; il tavolo è l'elemento che prende fuoco ed è caratterizzato dalla superficie 'WOOD', introdotta nel file di input dell'FDS ed avente un HRRPUA (*Heat Release Rate Per Unit Area*) pari a 500 KW/m<sup>2</sup>; la finestra circolare è introdotta all'istante di tempo 2 secondi.

Implementando il Geometry Solver, ciò che si ottiene è la lettura dei vari elementi; il tempo d'iterazione impiegato per la simulazione è di 1'30".

Successivamente viene fatto girare l'FDS dando in input il file ottenuto in uscita dal programma di pre-processing. L'FDS ha impiegato 5h d'iterazione per la definizione delle grandezze fluidodinamiche d'interesse, per un tempo di durata dell'incendio pari a 6".

La macchina hardware utilizzata ha le seguenti caratteristiche:

processore Dual core, cache size pari a 1024KB, cpu 2200MHz, ram di 4GB.

I risultati che mostrano la corretta lettura da parte dell'FDS della geometria possono essere visualizzati nelle figure presentate di seguito. Vengono mostrati quattro immagini di Smokeview, in cui è rappresentato l'andamento della temperatura in piani a y costante e a differenti istanti di tempo (Fig. 2 e 3). Dalle due foto di Fig.3 si nota come il flusso passi al di fuori della struttura dopo il tempo di 2", in cui si ha l'apertura della finestra (definita 'rosone' nel file di input del Geometry Solver). La Fig.4 mostra la geometria modificata dalla creazione della cavità che simula la finestra circolare.

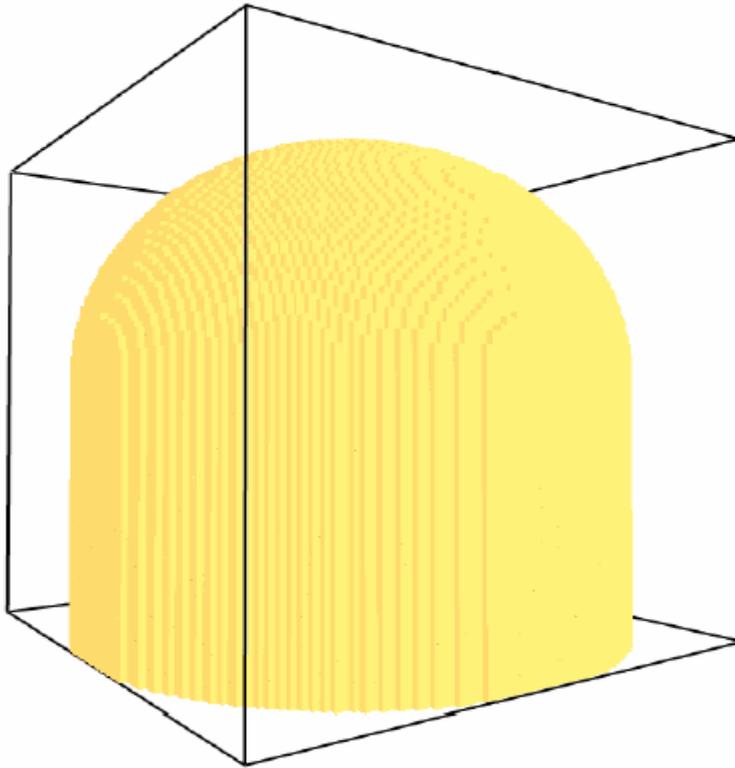


Figura 1: Struttura cilindrica visualizzata tramite Smokeview.

## 2.2 Potenzialità del pre-processor

L'ultima versione del codice di pre-processing è stata migliorata rispetto alla precedente per quanto concerne l'introduzione della possibilità da parte del codice di leggere dal file di input parametri caratteristici relativi sia alla struttura principale che agli elementi solidi e aperture. Questo perché, nel momento in cui si vogliono caratterizzare i vari elementi all'interno dello scenario d'incendio, è necessario associare a ciascun stl caratteristiche relative alla superficie e/o ai tempi di rimozione e di creazione delle porte e finestre presenti. Inoltre si è riusciti ad implementare un metodo più efficace e veloce di lettura della struttura principale.

Infatti il codice di pre-processing, nella versione precedente era caratterizzato da un particolare procedimento di scrittura del file di input dell'FDS, per cui i blocchi solidi che costituiscono tutte le pareti perimetrali della struttura considerata sono ottenuti come insiemi di più parallelepipedi solidi, ciascuno avente dimensioni pari a quella delle singole celle di calcolo della griglia computazionale, costruita attorno alla scenario d'incendio. Questa procedura accelera la simulazione, visto che alla lettura della geometria segue la scrittura automatica nel file di input, senza la necessità d'effettuare altri controlli sulle coordinate dei blocchi solidi.

L'ottimizzazione che si è voluta introdurre riguarda la diminuzione del numero dei blocchi che vengono scritti nel file data per rappresentare la struttura, vista la dimensione considerevole del file di output generato dal Geometry Solver nella sua prima versione. Il contributo apportato consiste nel controllo sui valori delle coordinate dei blocchi solidi, grazie al quale è possibile accorpere più parallelepipedi vicini in un blocco unico, con la conseguente riduzione dei valori delle coordinate da scrivere all'interno del file data di input all'FDS e delle sue dimensioni in termini di bytes.

Nel caso di studio presentato di seguito si evidenzierà il miglioramento apportato al codice di calcolo dal punto di vista della diminuzione dello sforzo computazionale che ne deriva e dalla maggiore facilità di caricamento da parte dell'interfaccia user-friendly del file data, di cui si parlerà nel capitolo 3.

Per evidenziare l'ottimizzazione realizzata si è preso in considerazione il caso di una galleria di dimensioni 10x10x7 metri rispettivamente lungo i tre assi di riferimento x, y e z; al suo interno è poi definita una banchigia. Sia la struttura principale che il blocco secondario, vengono forniti come

file stl. La lettura di questi file effettuata tramite il Geometry Solver avviene analizzando le coordinate dei blocchi solidi che individuano le strutture lungo l'asse x (che rappresenta l'asse lungo il quale si sviluppa la lunghezza del corpo). Viene effettuato un controllo sequenziale di tali blocchi per diminuirne il numero nel momento in cui possono essere accorpati, perché aventi stesso valore della coordinata y, come indicato nella Fig.5:

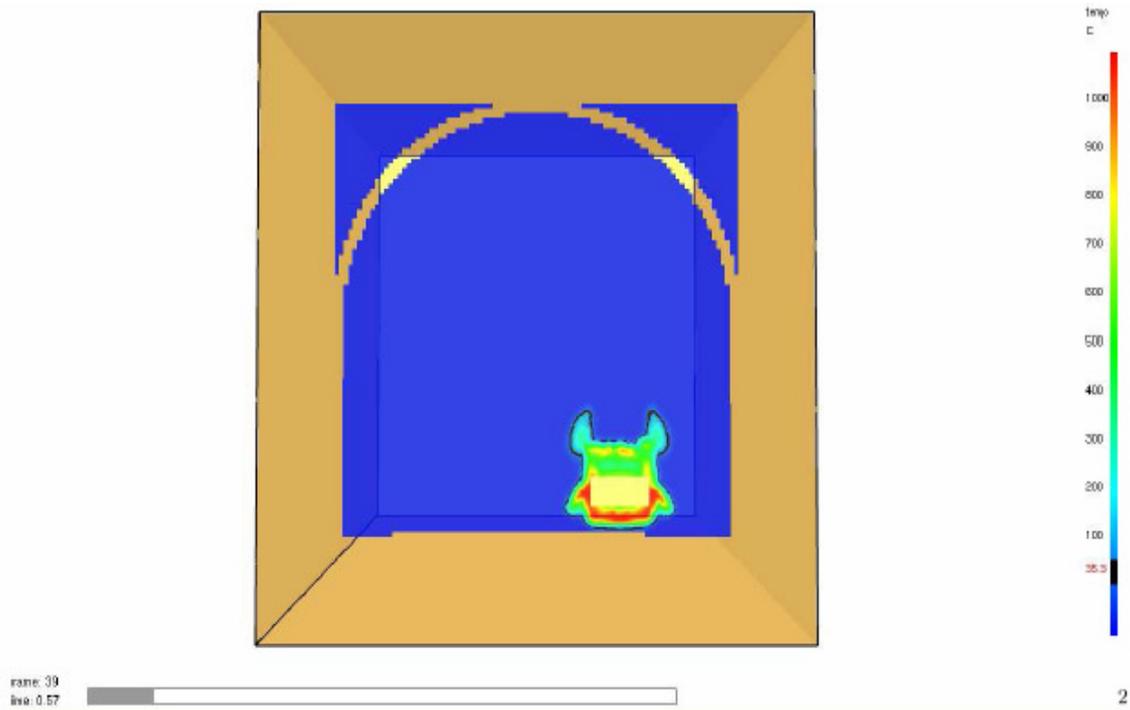


Figura 2: Andamento della temperatura a  $y=\text{cost}$  e a  $t=0.57$  (sez. trasversale della panca).

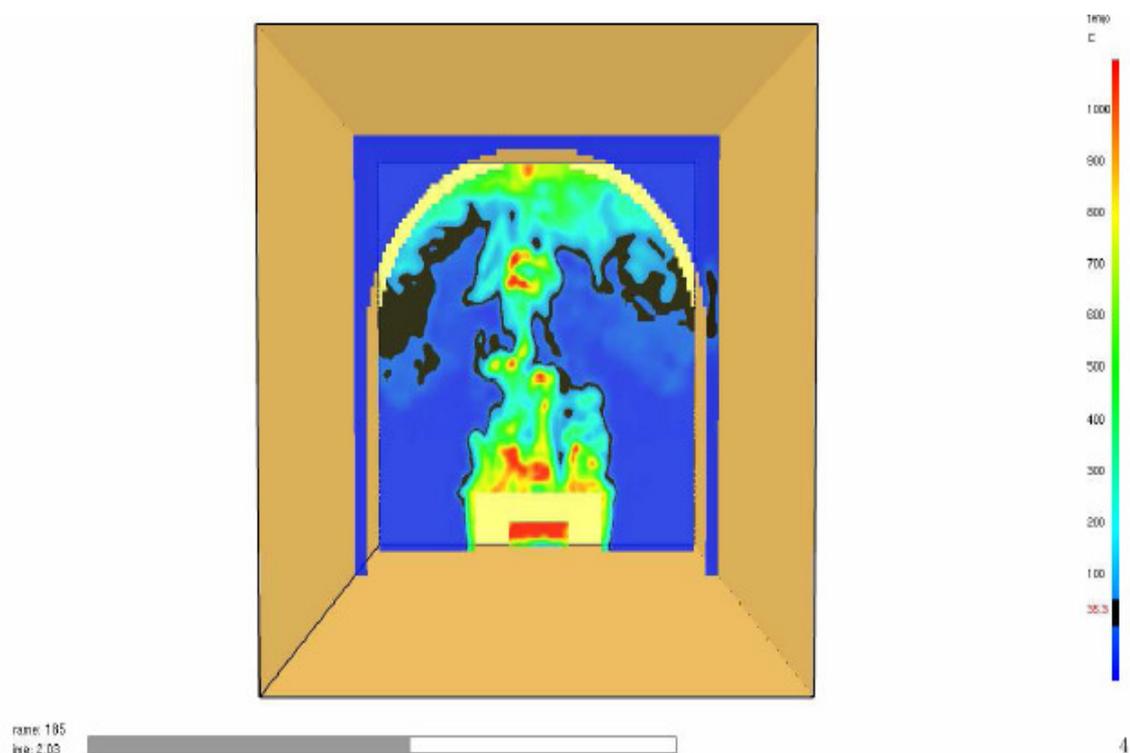


Figura 3: Andamento della temperatura a  $y=\text{cost}$  e a  $t=2.03$  (sez. longitudinale del tavolo).

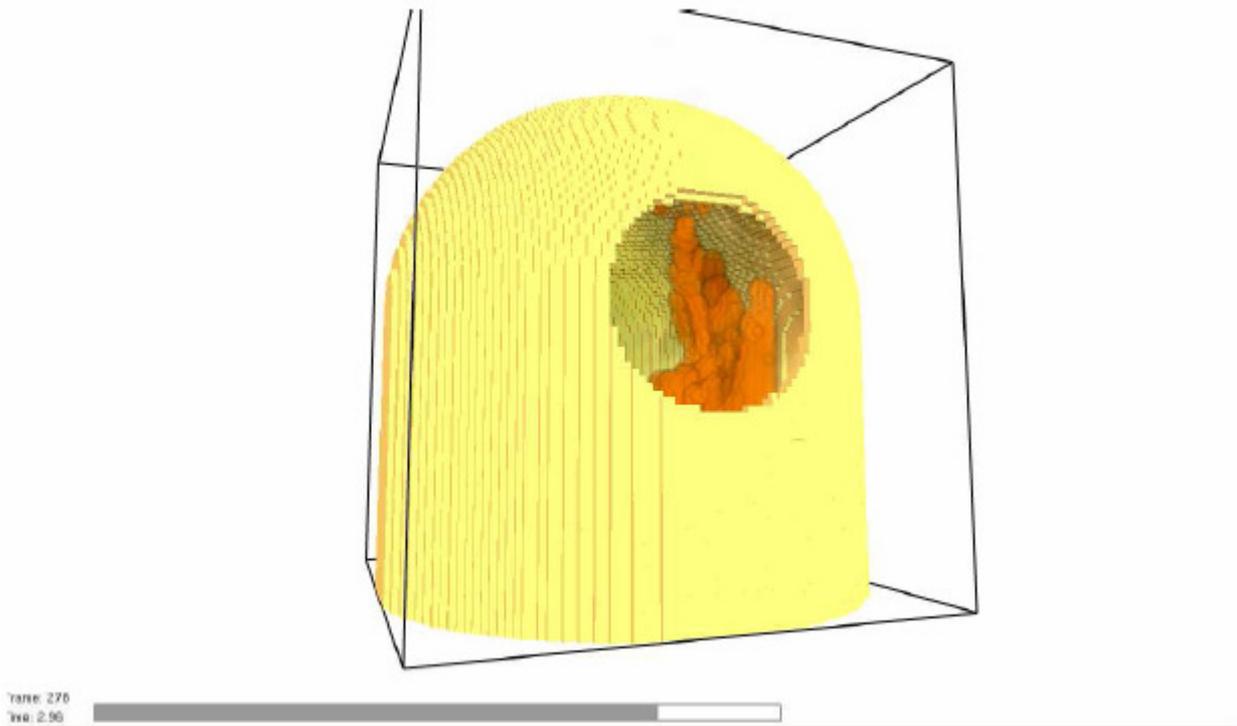


Figura 4: Visualizzazione della finestra circolare.

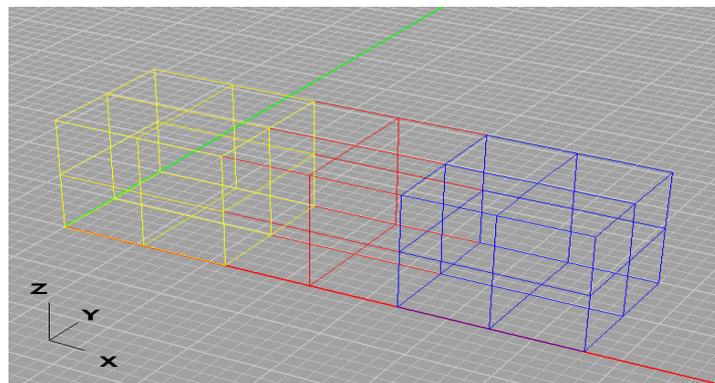


Figura 5: La sequenza sei tre blocchi solidi è definita come unico parallelepipedo.

L'ottimizzazione consiste nell'unire i tre blocchetti in uno unico, in modo da far leggere all'FDS, in tempi inferiori rispetto alla prima versione del codice, le coordinate della struttura data. Il risultato che si è ottenuto è la corretta lettura e semplificazione del modello da parte dell'ultima versione del codice; sono inoltre rispettati anche tempi di simulazione molto brevi in accordo da quanto stabilito dall'algoritmo del ray-tracing, ed in particolare essi risultano pari a quelli che si ottenevano con la versione precedente (con un hardware avente le seguenti caratteristiche: processore Dual core AMD Opteron, cache size pari a 1024KB, cpu 2200MHz, ram di 4GB). Tutte le caratteristiche della simulazione, confrontate con quelle che si ottenevano con la prima versione del programma, sono riportate di seguito:

### GeoSolv\_01

- Simulazione : 18”
- File data : 99408 obst (7900 kb)

### GeoSolv\_02

- Simulazione : 16”
- File data : 1556 obst (113 kb)

Il file data che si ottiene in output, viene fatto leggere dall’FDS, che realizza il run della simulazione in un tempo pari a 3’38” per 10” di evoluzione dell’incendio (l’hardware utilizzato ha le caratteristiche elencate in precedenza).

La lettura corretta della geometria analizzata viene evidenziata dalla Fig.6, che rappresenta una slide ottenuta con il visualizzatore Smokeview.

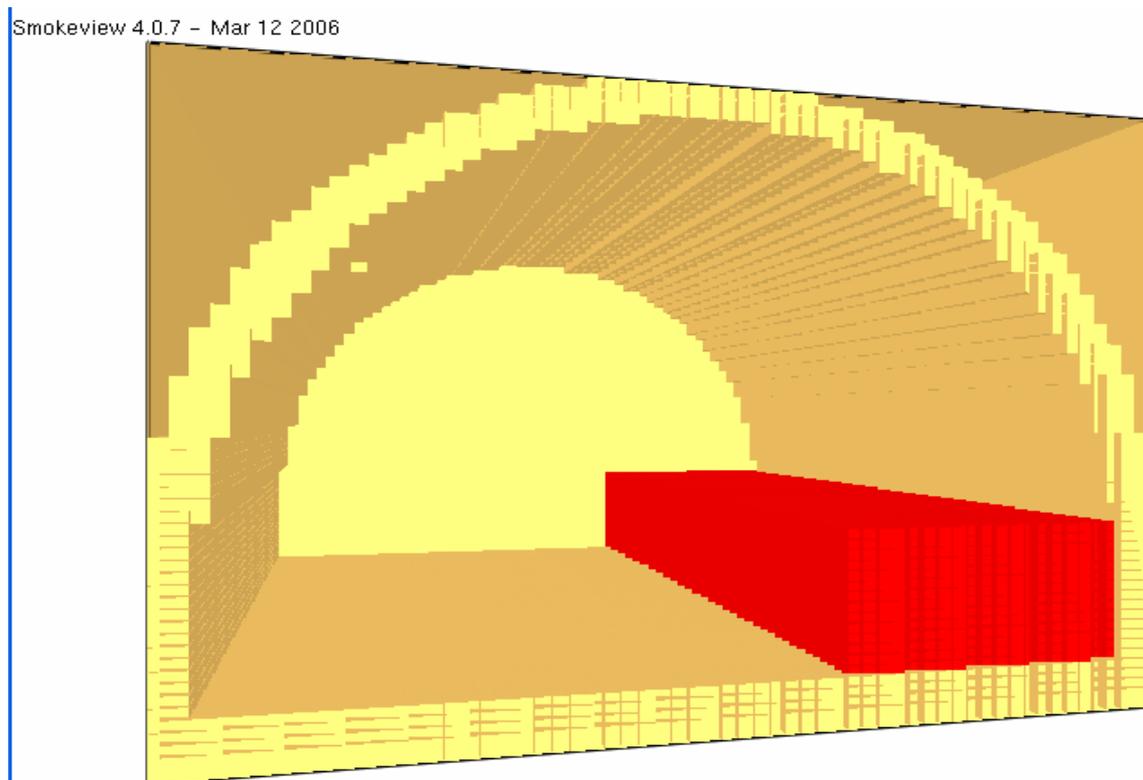


Figura 5: Rappresentazione della geometria mediante il programma Smokeview

### **3.0 SVILUPPO DELL’INTERFACCIA GRAFICA**

Parallelamente allo studio della lettura automatica di geometrie complesse, è stata sviluppata un’interfaccia utente che si propone di essere uno strumento completo di pre-processing per il software FDS. Tale interfaccia offre la possibilità di creare od importare una geometria e/o parti di essa e di definire tutte le condizioni al contorno per la generazione di un caso completo di studio FDS. Scritta in linguaggio Java e completamente strutturata secondo logiche di progettazione ‘orientate oggetto’, l’interfaccia offre strumenti di sviluppo di facile utilizzo, oltre che controlli tali da ridurre possibili errori da parte dell’utente finale. Il software fornisce, inoltre, un sistema di controllo tramite command files, implementato per offrire la possibilità di compiere una serie di azioni pre-impostate sul caso di studio, come ad esempio lo spostamento di gruppi di entità geometriche, oltre che fornire un potente mezzo di controllo sullo stesso in caso di lanci batch. Tale funzione permette di studiare lo scenario impostato partendo da un “Design of Experiment” (DOE) per esplorare una serie di configurazioni e guidare nelle scelte ingegneristiche, oltre che predisporre il software ad eventuali studi di ottimizzazione, permettendo la modifica dinamica del caso oggetto di studio.

Tutte le componenti dell'interfaccia grafica sono state scritte seguendo la tecnica a struttura MVC (Model-View-Controller) per semplificarne il funzionamento e per renderle molto flessibili in vista di eventuali futuri aggiornamenti.

La struttura MVC è una scelta di design semplice ed efficiente, che si basa sulla suddivisione dei “compiti” di un componente grafico. Nel presente lavoro si parlerà di come il componente grafico viene suddiviso in una vista, in dati gestiti tramite un modello ed in un oggetto che permette la comunicazione fra questi due. L'obiettivo è quello di offrire una piattaforma che a 360 gradi possa guidare l'utente nella progettazione “Fire Safe Safety”.

### 3.1 Caratteristiche tecniche dell'interfaccia grafica

La soluzione tecnologica sviluppata consiste in un codice che fornisce la possibilità, tramite una comoda interfaccia grafica, di riprodurre in modo sufficientemente accurato la geometria e le proprietà della struttura edilizia presa in esame, generando automaticamente il file di input per il solutore fluido-dinamico dedicato allo studio degli incendi FDS.

Lo sviluppo del software può essere schematizzato in 5 fasi distinte:

1. LOR (List Of Requirement)
2. Analisi del Software
3. Scrittura Codice
4. Debug
5. Ottimizzazione Codice

Nella prima fase del progetto è stata definita la LOR (List Of Requirement), vale a dire le proprietà e le caratteristiche che devono contraddistinguere la scrittura del generatore di scenari.

In un secondo passo sono state individuate, secondo una logica “top-down”, le componenti software e le loro rispettive funzioni. Tali componenti sono elencate di seguito, mettendo in evidenza la funzione che esse assolvono:

1) motore vero e proprio del codice:

- **Model Features:** è il pacchetto “base”, contenente le specifiche di ogni gruppo di parametri definiti da FDS; gestisce la creazione di questi gruppi (o features) ed i vincoli ingegneristici interni ad ognuno di essi.
- **Model Container:** gestisce, tramite strutture dati ad hoc (strutture ad array, a pila, a coda e ad albero binario bilanciato), i gruppi del **Model Features**; ne consente, inoltre, l'inserimento, la cancellazione, la modifica e la ricerca, su memoria ram e su disco, gestendo le funzioni di swapping (passaggio di dati da memoria random a memoria fissa in funzione della percentuale di utilizzo associato ai diversi gruppi).
- **Model Manager;** gestore “finale” del motore del software, ed assolve quindi ad una funzione di tramite tra l'interfaccia utente (GUI) ed i dati di base. Nello stesso tempo, controlla i vincoli ingegneristici esistenti fra i gruppi.

2) interfaccia grafica:

**GUI;** Si è deciso di comporre l'interfaccia grafica con un albero di visualizzazione e controllo, una barra di menu, una barra di stato, un pannello di visualizzazione tridimensionale e delle finestre di pop up per la gestione delle features.

Tutte le componenti dell'interfaccia grafica sono state scritte seguendo il design a struttura MVC (Model-View-Controller) per semplificarne il funzionamento e per renderle molto flessibili in vista di eventuali futuri aggiornamenti.

La struttura complessiva del codice, è rappresentata nel diagramma in Fig. 6.

#### 3.1 La tecnica a struttura MVC

La struttura MVC è una scelta di design tanto semplice, quanto “potente”, che si basa sulla suddivisione dei “compiti” di un componente grafico.

Il componente grafico viene suddiviso in una vista (o View), in dei dati gestiti tramite un modello (o Model) ed in un oggetto che permette la comunicazione fra questi due (Controller).

Ogni parte del componente non “conosce” il funzionamento delle altre e, di conseguenza, ognuna di esse può essere modificata senza che le altre debbano essere “toccate”.

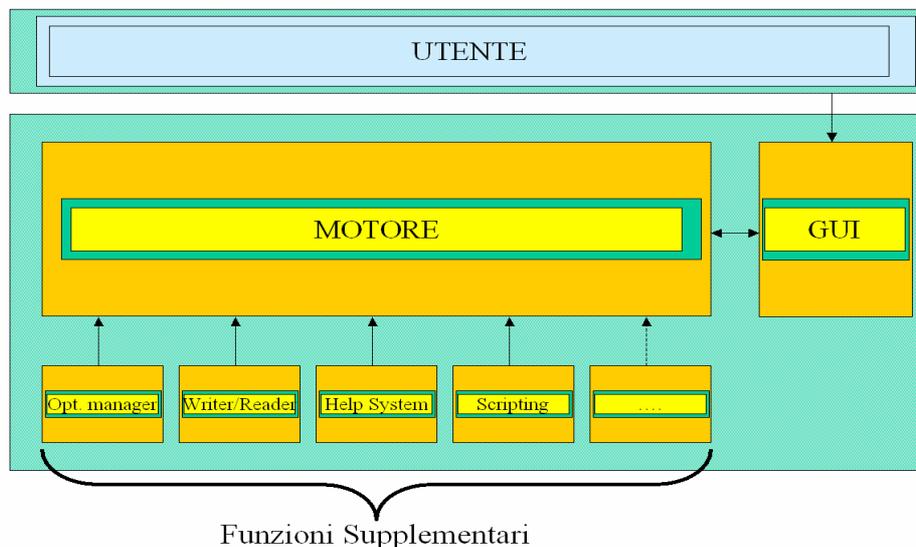


Figura 6: Struttura complessiva dell'interfaccia grafica.

### 3.2 Potenzialità dell'interfaccia grafica

Tramite l'interfaccia grafica è possibile ridurre notevolmente la difficoltà legate alla scrittura del file di input dell'FDS. Infatti il programma permette sia di importare i file di input dell'FDS, sia di definire scenari d'incendio inserendo direttamente le coordinate delle varie strutture solide presenti e tutti i parametri ad esse associate (che ne determinano le caratteristiche termo-fisiche), come stabilito dall'FDS. Nelle Fig. 7 e 8 si evidenzia come viene importato correttamente uno scenario d'incendio e come, con semplici comandi attivati con il mouse, è possibile selezionare i vari elementi che lo compongono per eventuali modifiche o caratterizzazioni (ad es., per assegnare materiali o altre grandezze alle varie superfici).

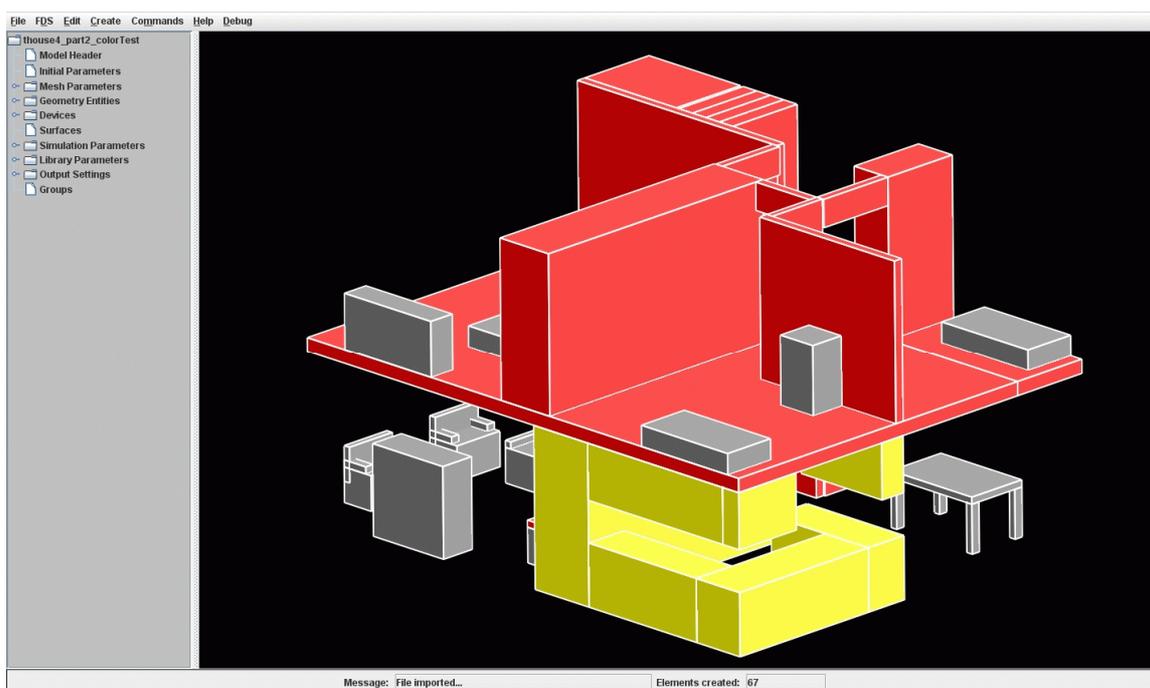


Figura 7: Esempio di definizione di scenari d'incendio tramite l'interfaccia grafica.

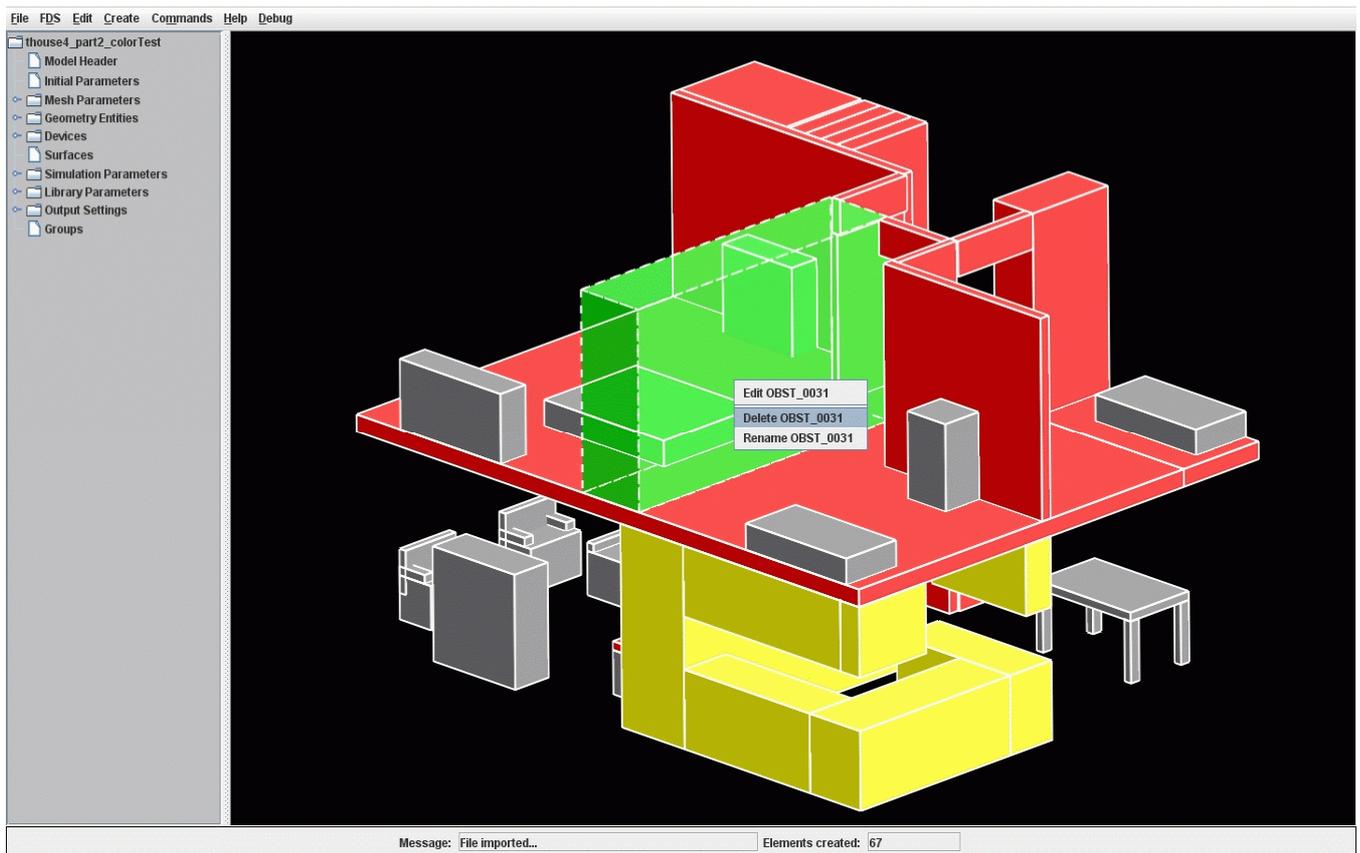


Figura 8: Caratterizzazione delle parti solide tramite l'interfaccia grafica.

#### 4.0 CONCLUSIONI E SVILUPPI FUTURI

La piattaforma sviluppata e qui presentata è un prodotto nuovo ed innovativo. Grazie al suo utilizzo, i progettisti operanti nel settore della Fire Engineering avranno accesso ad un strumento efficace e flessibile che permetterà loro di ridurre i tempi e i costi di progettazione e migliorerà l'efficienza degli impianti di sicurezza antincendio. Avendo infatti a disposizione risultati attendibili provenienti dall'analisi fluido dinamica, strumenti di gestione delle informazioni, di confronto normativo e di previsione di scenari, al variare di una serie di parametri di ingresso, sarà sicuramente più agevole ed affidabile la scelta progettuale di contromisure che accrescano la sicurezza delle strutture e dei relativi impianti, nonché la verifica della sicurezza ai fini antincendio di opere preesistenti.

Sinteticamente, la suddetta piattaforma software può rappresentare uno strumento che, utilizzato dagli organi di controllo supporta la formulazione di pareri tecnici e accertamenti normativi, mentre utilizzato dai progettisti che si occupano di Fire Engineering agevola la valutazione delle condizioni di rischio, delle prestazioni dei sistemi e dei coefficienti di confidenza, migliorando complessivamente il livello di sicurezza generale delle misure antincendio e sviluppando delle nuove metodologie che possono essere adottate da tutto il settore di riferimento.

## Bibliografia

- [1] Kevin McGrattan, Glenn Forney, Fire Research Division Building and Fire Laboratory in cooperation with Building and Transport: “Fire Dynamics Simulator (Version 4)”, User’s Guide, NIST Special Publication 1019, Finland, March 2006
- [2] Kevin McGrattan, , Fire Research Division Building and Fire Laboratory in cooperation with Building and Transport : “Fire Dynamics Simulator (Version 4) ”, Technical Reference Guide, NIST Special Publication 1018, Finland, March 2006.
- [3] Mohd-Yusof J.: “Combined immersed boundaries/b-slines methods for simulation of flows in complex geometries”, CTR Annual Research Briefs, NASA Ames/Stanford Univ., pages 317-327, 1997.
- [4] Orlandi P., Mohd-Yusof J., Verzicco R.: “Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations”, Journal of computational Physics, 161:35-60, 2000.
- [5] [www.fire.nist.gov/fds](http://www.fire.nist.gov/fds)
- [6] O’Rourke, J.: Computational Geometry in C. Cambridge University Press (1998)
- [7] Iaccarino, G., Verzicco, R.: “Immersed boundary technique for turbulent flow simulations”. App. Mech. Rev. ASME 56, 331–347 (2003)
- [8] D.Bello, N.De Santis, P.Cinnella, R.Lala.: “Sviluppo di un codice fluidodinamico per lo studio dei fenomeni di incendio in ambienti con geometria complessa” (2007)
- [9] Gamma E., Helm R., Johnson R. e Vlissides J.: “Design Patterns: Elements of Reusable Object-Oriented Software”, Addison-Wesley Professional Computing Series, (1995)
- [10] John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman: “Introduction to Automata Theory, Languages, and Computation”, Pearson, (2001)
- [11] Donald E. Knuth: “Art of Computer Programming, Volume 3: Sorting and Searching”, Addison-Wesley, (1998)
- [12] Mark de Berg, Marc Van Kreveld, Mark Overmars, Otfried Schwarzkopf: “Computational Geometry: Algorithms and Applications”, Springer, (2000)

