International Conference
ENGINEERING MECHANICS 2010
Svratka, Czech Republic, May 10 – 13, 2010

DEVELOPMENT OF EDT GENERATOR

D. Krybus, B. Patzák [1]

Summary: *The paper deals with the development of a mesh generating algorithm. First, reasons for the use of Lagrangian formulation in modeling of fluid-structure interaction are given. The need of a suitable meshing algorithm is discussed. Bowyer/Watson algorithm for constructing Delaunay triangulation was chosen and implemented in the finite element package OOFEM. An octree data sorting is introduced in order to decrease the algorithm time consumption.*

1. Úvod

V posledních desetiletích lze pozorovat zvýšený rozvoj numerických metod v oblasti modelování interakce kapalin s konstrukcí. Tyto metody se uplatňují v řešení úloh zahrnujících např. hydrodynamiku plovoucích předmětů, konstrukce vystavené proudění vody nebo proudění s volnou hladinou či simulaci slévárenských procesů.

Klasický přístup k modelování proudění představuje použití Eulerovy formulace řídících rovnic, kdy pohyb proudící tekutiny je vztažen k pevné síti. Naproti tomu, Lagrangeovský přístup vychází z provázanosti sítě s konkrétními částicemi modelované domény. Tzv. *arbitrary Lagrange-Eulerova (ALE)* formulace kombinuje tyto dva základní přístupy, aby těžila z jejich výhod a omezovala některá z negativ. Jak Eulerův, tak ALE přístup jsou pro řešení úloh pohybu pevných těles v kapalinách, jejich vzájemné interakce, či úloh s volnou hladinou velmi rozšířené.

Oběma formulacím je společné, že vztah mezi pohybem sítě a částic je v momentových rovnicích popsán konvektivní rychlostí, což vyžaduje numerickou stabilizaci. Další obtíže pak v úlohách interakce představují zajištění podmínky nestlačitelnosti kapaliny, modelování a sledování volné hladiny, přenos informací mezi doménami pevné a kapalné části na kontaktním rozhraní či popis velkých pohybů.

Důsledkem použití Lagrangeovského přístupu je vymizení značně problematického konvektivního členu, což řešení úloh proudění a interakce z numerického hlediska velmi usnadňuje. Jedná se o moderní metody, které zažívají nebývalý rozvoj.

2. Lagrangovská formulace proudění

Jednou z metod vycházejících z Lagrangeovské formulace je *particle finite element method — PFEM* (Oñate at al., 2004) vyvíjená na Katalánské technické univerzitě v Barceloně v

[1] Ing. David Krybus, doc. Dr. Ing. Bořek Patzák, Department of Mechanics, Faculty of Civil Engineering, Czech Technical University in Prague, Thákurova 7, 166 29 Prague 6, tel. +420 224 355 417, email: david.krybus@fsv.cvut.cz

centru CIMNE. Metoda PFEM řeší úlohu představovanou množinou částic, jejichž pohyb se uskutečňuje na základě jejich vlastní tíhy a výslednice vnějších či vnitřních sil, které na ně působí. Z hlediska metody konečných prvků pak tyto částice představují uzly sítě. Částice reprezentuje hmotu a přenáší informaci o objemu ve svém okolí. Veškeré fyzikální a matematické vlastnosti jsou vztaženy přímo k samotné částici a nikoliv ke konečnému prvku. V každém kroku se z těchto částic vytváří konečněprvková síť, na níž jsou pak standardním způsobem metody konečných prvků řešeny charakteristické rovnice. PFEM se postupně vyvinula jako výsledek práce kolektivu autorů na řešení úloh interakce prostřednictvím Lagrangeovských konečných prvků a bezsíťových metod.

Zásadní výhoda Lagrangeovské formulace spočívá v absenci konvektivních členů, které v jiných formulacích řešení komplikují a vyžadují jejich stabilizaci. Na druhou stranu sebou tento přístup nese potřebu věnovat značnou pozornost pohybu uzlů sítě. Jelikož již z charakteristiky tekutiny lze předpokládat rozsáhlé pohyby částic, je nutné, aby síť byla v každém kroku řešení obnovována. Autoři vsadili na tzv. rozšířené Delauneyovské mozaikování (*extended Delauney tesselation*), představující rychlý a účinný algoritmus sloužící ke generování hledané sítě. Jeho výsledkem je kombinace simplexových a mnohostěnových prvků, které vyžadují použití zvláštních tvarových funkcí např. z bezsíťové metody konečných prvků (Idelsohn at al., 2003).

Použití Lagrangeovského přístupu se jeví jako velmi vhodné, je-li třeba detailně sledovat volnou hladinu nebo rozhraní mezi dvěma materiály. Snadno lze tak modelovat velké pohyby hladiny, lomící se vlny nebo rozstřikující se kapaliny. Předpokladem je však algoritmus identifikující hranici domény množiny daných uzlů, která v úloze není explicitně definována. Hranici tvoří v úlohách proudění jak volná hladina, tak jednotlivé částice pohybující se ven z hlavní domény. Pro účely identifikace se osvědčila technika *Alpha Shape* (Edelsbrunner and Mücke, 1994).

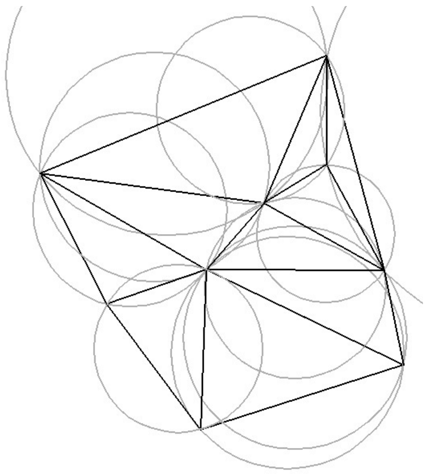
3. Delaunayova triangulace

Jednou z metod často používaných ke generování sítí je takzvaná Delaunayovská triangulace. Ta byla definována Borisem Delaunayem v roce 1934 pro obecnou množinu bodů P jako taková triangulace, že každá z opsaných kružnic k jednotlivým trojúhelníkům neobsahuje žádný z bodů P, kromě tří tvořících vrcholy trojúhelníku. Delaunayovská triangulace maximalizuje minimální úhel ze všech úhlů v trojúhelnících, což umožňuje získat „kvalitní" síť zaručující stabilní výpočet.

Tuto definici lze rozšířit i do trojrozměrného prostoru. Výsledkem triangulace jsou potom čtyřstěny, jimž opsané koule opět neobsahují jiné body než definiční body na kulové ploše. Teoreticky je možné i rozšíření do jiných než eukleidovských prostorů, což ovšem nezaručuje existenci nebo jedinečnost triangulace.

Duální k Delaunayovské triangulaci je Voronoiův diagram někdy také označovaný jako Voronoiovské mozaikování. Jeho výsledkem je rozdělení metrického prostoru na základě množiny bodů takové, že jakákoliv část je ke svému definičnímu bodu blíže než k jakémukoliv jinému. Pro případ dvou bodů v rovině je tato dělena přímkou kolmou a půlící jejich spojnici. Máme-li tři body neležící na přímce, je průsečíkem hranic oddělujících části příslušejících bodům právě střed opsané kružnice trojúhelníku s vrcholy v těchto bodech.

Existuje množství algoritmů sloužících k výpočtu Delaunayovské triangulace (Shewchuck, 1999). *Inkrementální* algoritmy pracují s udržováním stávající Delaunayovy triangulace, do které jsou postupně vkládány jednotlivé body. Po vložení bodu je síť upravena tak, aby nadále



Obr.1 Delaunayovská síť

splňovala základní vlastnost. *Flipping* algoritmy jsou založeny na záměně, tzv. „flippovaní" stran trojúhelníků, které nejsou Delaunayovské. Obecně jsou tyto metody založeny na dvou krocích. Nejprve je vygenerována obecná trojúhelníková síť a ta je následně optimalizována záměnou stran tak, aby výsledná síť splňovala Delaunayovskou podmínku. Algoritmy *„rozděl a panuj"* nejprve rekurzivně rozdělí úlohu na elementární a po jejich triangulaci zase zpět sloučí do celkové triangulace. *Projekční* algoritmy převádí úlohu hledání Delaunayovské triangulace na úlohu výpočtu konvexní obálky (*convex hull*). To spočívá v zobrazení bodů na o jednu dimenzi vyšší paraboloid umístěný v počátku. Výpočtem konvexní obálky a její zpětnou projekcí do původní dimenze získáme strany triangulace.

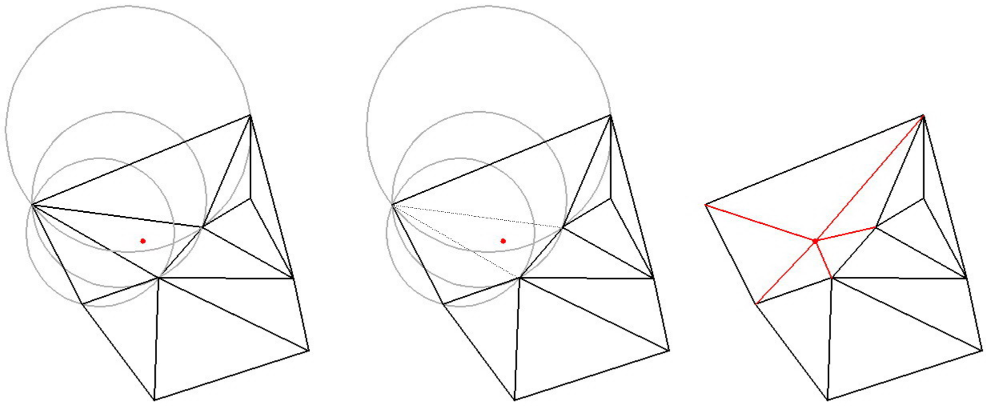
4. Bowyer/Watsonův algoritmus

Mezi jedny z nejpoužívanějších metod patří Bowyer/Watsonův inkrementální algoritmus založený na postupném přidávání jednotlivých bodů do triangulace. Tento algoritmus odvodili nezávisle na sobě Adrian Bowyer (1981) a David F. Watson (1981) a prezentovali jej v jednom čísle *The Computer Journalu*.

Tab.1 Schéma algoritmu

1. vytvoř *bounding-box*, zahrnující všechny body *p*
2. dokud nejsou vloženy všechny body *p*
   (a) vlož bod *p* do triangulace
   (b) najdi všechny opsané kružnice s příslušnými trojúhelníky obsahující bod *p*
   (c) smaž tyto trojúhelníky → vkládací polygon
   (d) vytvoř triangulaci vkládacího polygonu spojením jeho vrcholů s bodem *p*
3. odstraň *bounding-box*

Na počátku bývá zpravidla vytvořen tzv. *bounding-box*, který zahrnuje všechny body. Do této úvodní triangulace jsou pak body jeden po druhém postupně vkládány a je vždy vyžadováno splnění Delaunayovské vlastnosti. To spočívá v hledání trojúhelníků, do jejichž opsané kružnice vložený bod spadá. Tyto jsou pak smazány a strany jim sousedních trojúhelníků vytvoří tzv. vkládací polygon. Spojením vrcholů polygonu a nově vloženého bodu vznikne nová triangulace splňující Delaunayovskou vlastnost. Po přidání všech bodů do triangulace jsou pak bounding-box a všechny strany směřující od jeho vrcholů do triangulace vymazány.



Obr.2 Vkládání bodů

Výhodou tohoto algoritmu je snadná rozšiřitelnost z roviny do prostoru. Na místo trojúhelníkům opsaných kružnic jsou hledány dotčené koule opsané čtyřstěnům. Nedelaunayovské čtyřstěny jsou pak smazány, čímž vznikne „dutý" vkládací mnohostěn obsahující pouze vkládaný bod. Výsledkem jeho spojení s vrcholy vkládacího mnohostěnu je opět triangulace splňující předpoklady.

Nevýhoda tohoto algoritmu spočívá v možnosti vzniku defektní sítě vlivem zaokrouhlovací chyby. Při hledání trojúhelníků, které mají být mazány, může touto cestou dojít k opomenutí některého z dotčených a tím pádem není vkládací polygon prázdný. Aby bylo možno dosáhnout robustní implementace, je potřeba použít zvláštní algoritmus k přesnému určení vkládacího polygonu jako je například depth-first hledání.

Další z nevýhod, kterou je nutno zmínit, souvisí s „kvalitou" vzniklých trojúhelníků jakožto prvků pro výpočet metodou konečných prvků. Obecně lze říct, že Delaunayovská síť je kvalitní, ale existenci nekvalitních prvků nelze vyloučit. Jedná se především o trojúhelníky na hranici oblasti, případně trojúhelníky, jejichž středy opsaných kružnic se nacházejí v nepatrné vzdálenosti a jejichž poloměry se sobě také blíží. Ve většině generátorů sítí se takovéto prvky „napravují" vložením dalších bodů do těžiště nebo jejich vymazáním a vložením bodu do těžiště jejich opsané kružnice. Eventuálně je možné použít kombinaci s metodou postupné fronty. Alternativou, pro kterou se rozhodli autoři metody PFEM je použití rozšířeného Delaunayovského mozaikování, které zavádí nesimplexové prvky.

5. Octree

Z hlediska rychlosti triangulace velkého počtu bodů se ukazuje jako zásadní prohledávání celé databáze trojúhelníků, což vede ke značnému nárůstu času potřebného k vytvoření sítě. Aby bylo možné algoritmus urychlit, bylo přistoupeno k roztřídění dat na základě tzv. oktálového stromu (*octree*).

Octree (v rovině *quadtree*) představuje stromovou strukturu používanou k třídění dat na základě jejich umístění v prostoru. Nejprve jsou data (např. částice) zahrnuta do kořenové buňky (*root cell*). Je-li překročen limit pro počet dat v buňce, rozdělí se tato na osm potomků (*child*) - v každém rozměru na polovinu - a data jsou do nich převedena. Přerozdělování se děje rekurzivně, dokud není dosaženo maximálního počtu dat pro jednotlivou buňku. Strukturování umožňuje rychlý přístup k datům zvolené lokace bez nutnosti procházet kompletní databázi.

V této konkrétní aplikaci se používá octree k roztřídění jednotlivých dočasných trojúhelníků během fáze tvorby triangulace. Na počátku je vytvořena z částic kořenová buňka, která svým způsobem odpovídá bounding-boxu. Prvky jsou pak do datové struktury zařazovány na základě příslušnosti svých opsaných kružnic do jednotlivých buněk. Během sekvenčního vkládání bodů do triangulace lze snadno a rychle z jejich pozice určit příslušnou buňku octree a dotčené opsané kružnice hledat pouze v ní.

6. Implementace

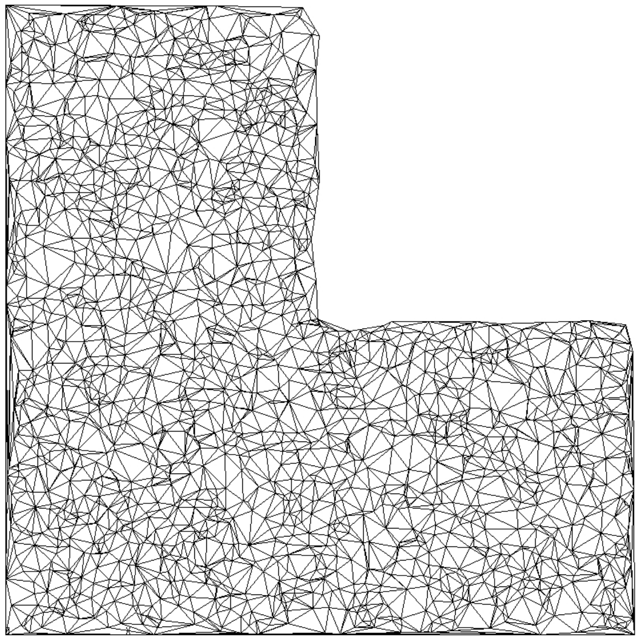
Lagrangeovská formulace proudění a následná implementace interakce fází vycházející z particle finite element method představují budoucí rozšíření programu pro výpočet metodou konečných prvků OOFEM (2010) vyvíjeného na Katedře mechaniky Fakulty stavební ČVUT v Praze. OOFEM je volně šiřitelný program pro řešení multifyzikálních problémů metodou konečných prvků s objektově orientovanou architekturou pro řešení úloh mechaniky pevné fáze, transportních jevů a mechaniky tekutin.

Prvním a zároveň nutným krokem pro zahrnutí metody PFEM je implementace interního generátoru sítě. Pro účely jeho testování byl vytvořen prototypový program v jazyce C++ využívající knihovny *oofemlib* a *Elixir* (Krysl, 1997). Kód v sobě zahrnuje nově implementované třídy částice a prvku. První z nich obsahuje polohu a identifikátor částice, druhá pak již informace o částicích tvořících prvek a o opsanou kružnici. Algoritmus byl aplikován na jednoduché úlohy, jak lze vidět na obrázku 3, a prokázal svoji funkčnost a spolehlivost.

Na druhou stranu byla pozorována vysoká časová náročnost pro rozsáhlé úlohy, která byla způsobena prohledáváním celé databáze bodů. Za účelem zefektivnění celého algoritmu bylo přistoupeno k jeho kompletní integraci do prostředí OOFEM, aby mohly být využity všechny existující nástroje. Zároveň bylo do kódu zahrnuto třídění dat vycházející z octree, popsaného v předcházející kapitole. Momentálně je algoritmus již plně implementován, probíhá jeho testování a porovnávání rychlosti s původní verzi prohledávající celou databázi.

7. Závěr

V článku byl představen algoritmus pro tvorbu sítě konečných prvků. Ten se jeví jako zásadní v kontextu řešení úloh proudění tekutin založených na Lagrangeovské formulaci řídících rovnic. Jako vhodná metoda byla zvolena Delaunayova triangulace používající inkrementální Bowyer/Watsonův algoritmus. Pro účely testování byl vytvořen prototypový kód a následně byla ověřena jeho funkčnost. Jeho aplikace na vybrané úlohy však ukázala nedostatečnou rychlost. Vlastní optimalizací kódu byla upřednostněna plná integrace algoritmu do prostředí OOFEM a zahrnutí lokálního strukturování dat využívající octree.



Obr.3 Triangulace náhodných bodů v doméně tvaru L

8. Poděkování

Tato práce vznikla za podpory grantu Grantové agentury České republiky č. 103/09/H078 a interního grantu studentské grantové soutěže Českého vysokého učení technického v Praze.

9. Reference

Bowyer, A. 1981: Computing Dirichlet tessellations. *The Computer Journal* Vol. 24, 162-166

Edelsbrunner, H., Mücke E.P. 1994: Three-dimensional alpha shapes. *ACM Transaction on Graphics* 13, 43-72

Idelsohn, S.R., Calvo, N., Oñate, E. 2003: Polyhedrization of an arbitrary 3D point set. *Computer methods in applied mechanics and engineering* Vol. 192, 2649-2667

Krysl, P. 1997: Petr Krysl's Page *http://hogwarts.ucsd.edu/ pkrysl/software.html*

Oñate, E., Idelsohn, S.R., Del Pin, F., Aubry, R. 2004: The particle finite element method. An overview. *International Journal of Computational Method* Vol. 1, 267-307

Patzák, B. & Bittnar, Z. 2001: Design of object oriented finite element code. *Advances in Engineering Software* 32(10-11), 759-767.

Patzák, B. 2010: OOFEM project homepage *http://www.oofem.org*

Shewchuck, J.R. 1999: *Lecture notes on Delaunay mesh generation*

Watson, D.F. 1981: Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. *The Computer Journal* Vol. 24, 167-172