



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STAVEBNÍ

FACULTY OF CIVIL ENGINEERING

ÚSTAV STAVEBNÍ MECHANIKY

INSTITUTE OF STRUCTURAL MECHANICS

TROJROZMĚRNÉ PRUŽINOVÉ SÍTĚ A JEJICH APLIKACE

THREE-DIMENSIONAL SPRING NETWORKS AND THEIR APPLICATIONS

ZKRÁCENÁ VERZE DIZERTAČNÍ PRÁCE

SHORT VERSION OF DOCTORAL THESIS

AUTOR PRÁCE

AUTHOR

Ing. Michal Štafa

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Petr Frantík, Ph.D

BRNO 2017

KLÍČOVÁ SLOVA

Fyzikální diskretizace, FyDiK, pružinová síť, lattice model, tetrahedronizace, CUDA, nelineární modelování

KEYWORDS

Physical discretization, FyDiK, Spring network, Lattice model, Tetrahedralization, CUDA, Non-linear modelling

Dizertační práce je uložena:

Vysoké učení technické v Brně, Fakulta stavební
Ústav stavební mechaniky
Veveří 95, 602 00 Brno

© Michal Štafa, 2017

ISBN 80-214-

ISSN 1213-4198

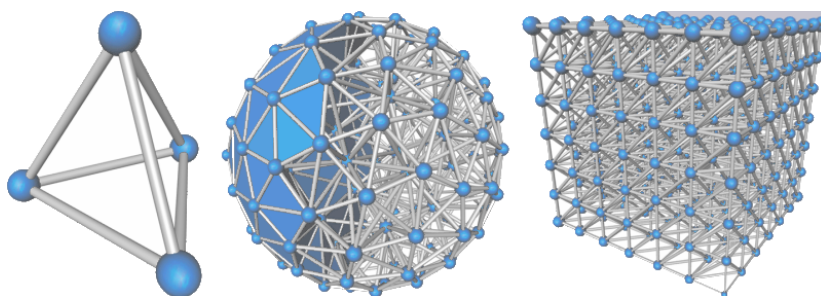
OBSAH

| | | |
|----------|--|-----------|
| 1 | Úvod | 5 |
| 1.1 | FyDiK | 5 |
| 1.2 | Modelované konstrukce | 6 |
| 2 | Teorie | 7 |
| 2.1 | Diskretizace | 7 |
| 2.2 | Geometrie modelu | 7 |
| 2.2.1 | Generování bodů | 7 |
| 2.2.2 | Triangulace | 8 |
| 2.2.3 | Voroného diagramy | 10 |
| 2.3 | FyDiK | 11 |
| 2.4 | Programová platforma | 14 |
| 2.5 | Aplikace FyDiK na platformě CUDA | 15 |
| 3 | Vyhodnocení | 17 |
| 3.1 | Beton | 17 |
| 3.2 | Ocel | 22 |
| 3.3 | Použití CUDA platformy | 23 |
| 4 | Závěr | 25 |
| | Literatura | 27 |
| | Životopis | 29 |
| | Abstrakt | 30 |

1 ÚVOD

Přestože dostupnost výpočetního výkonu umožňuje pro lineární úlohy zpřesňovat modely a snižovat jejich výpočetní čas, silně nelineární problémy mechaniky jako plastická deformace nebo rozvoj trhlin v heterogenních materiálech se již několik dekad velmi obtížně modelují klasickými metodami mechaniky kontinua [2].

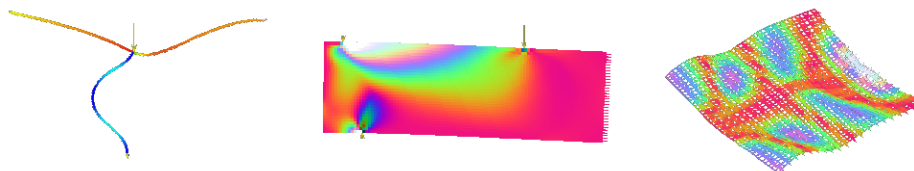
Jedním ze způsobů jak tento problém řešit je použití metody *fyzikální diskretizace*. Ta rozděluje modelované kontinuum na jednotlivé částice, mezi nimiž sleduje jejich vzájemnou interakci. Zpravidla jsou částice nositelé hmotnosti a jejich vzájemné vazby jsou zajištěny pevně danou, či variabilní sítí vazeb - pružin definující mechanické vlastnosti materiálu, viz obr. 1.1.



Obr. 1.1: Příklad fyzikální diskretizace [19]

1.1 FyDiK

Na VUT v Brně je již více než 10 let vyvíjen vedoucím této práce, doc. Ing. Petrem Frantíkem, Ph.D. fyzikálně diskretizační model *FyDiK* [8]. Ten je postaven na bázi dynamických systémů, umožňující zmíněné nelineární chování modelované konstrukce vystihnout i v čase. Tento model má navíc velmi dobré předpoklady, aby mohl být paralelizován.



Obr. 1.2: Příklady použití modelu FyDiK

1.2 Modelované konstrukce

Beton

Pro validaci modelu byla použita měření na betonových vzorcích vytvořených Ústavem stavebního zkušebnictví na VUT – FAST v roce 2008 [11]. Byly namíchány čtyři směsi s referenční recepturou pro beton C20/25. Betonovaná tělesa měla rozměry $100 \times 100 \times 400$ mm, z nichž byly po 28 dnech nařezány vzorky přibližně tvaru krychle o délce strany 100 mm, viz obr. 1.3. Vzorky byly označeny kódy R3 – R6 s rozdílným podílem jednotlivých frakcí kameniva a zatěžovány celoplošně (CC) nebo středovou příložkou o rozměrech 8×8 mm (1C).



Obr. 1.3: Betonové vzorky, foto: Ondřej Vodák [20]

Ocel

Motivací pro modelování ocelového nosníku, se stal výsledek experimentu na Ústavu kovových a dřevěných konstrukcí Fakulty stavební na VUT v Brně, viz obr. 1.4. Tento prvek byl zatěžován ve středu horní pásnice s celkovou deformací asi 40 mm, přičemž bylo dosaženo zřetelného vybočení a plastizace stojiny.

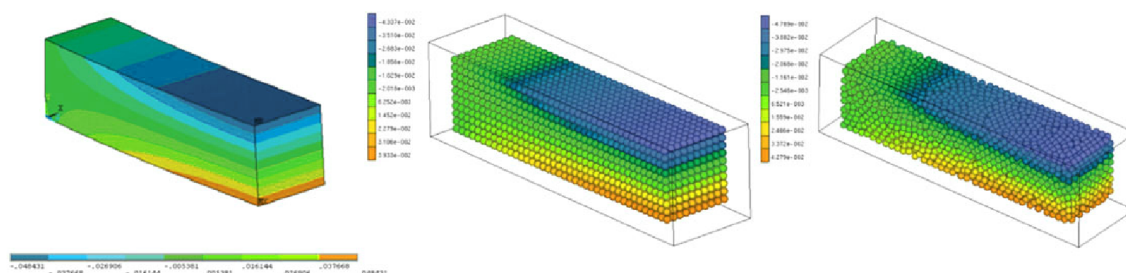


Obr. 1.4: Deformovaný I profil, vystavený na Ústavu kovových a dřevěných konstrukcí Fakulty stavební na VUT v Brně

2 TEORIE

2.1 Diskretizace

Vytváření fyzikálně diskretizovaného modelu obvykle předchází použití jednoduchého, zpravidla spojitého, náhradního modelu reálné konstrukce. Ten je následně rozdělen – diskretizován na jednotlivé elementy, za použití matematického aparátu sledujícího interakci jednotlivých částí. Výhodou tohoto přístupu je jasná fyzikální představa o dění v modelu, stejně tak jako možnost jej reálně vyrobit a ověřit. Jistou nevýhodou je určitá neshoda jak s mikroskopickou tak i makroskopickou strukturou modelovaného objektu a v případě modelů s velkým počtem částic značná výpočetní náročnost [8].



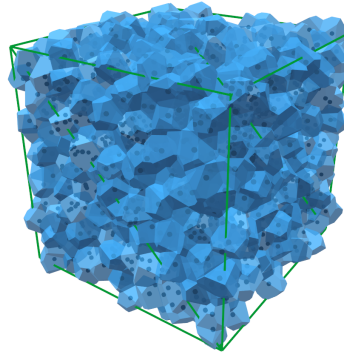
Obr. 2.1: Příklad matematické diskretizace na metodě konečných prvků (vlevo) a fyzikální diskretizace – lattice model s pravidelnou (střed), resp. nepravidelnou strukturou (vpravo) [26]

2.2 Geometrie modelu

Samotný proces diskretizace nejprve vytváří v modelovaném objemu množinu bodů, coby nositelů hmotnosti. Dalším krokem je jejich vzájemné propojení. Ačkoliv může být sestaveno různými způsoby, jako jeden z nejvhodnějších se jeví využití *Deloneho tetrahedronizace*, viz dále. Takto vytvořeným elementům sítě pružin a bodů je následně nutné přiřadit fyzikální vlastnosti. Pro jejich kvantifikaci lze s výhodou použít *Voroného mozaiku*, jež si lze představit jako množinu buněk zcela vyplňující diskretizovaný prostor s jádry odpovídajícími generovaným bodům. Voroného mozaika je duální strukturou k Deloneho tetrahedronizaci a bude rovněž dále více přiblížena.

2.2.1 Generování bodů

Existuje několik přístupů jak efektivně body v cíleném prostoru generovat a zároveň splnit uvedené podmínky. Zřejmě výpočetně nejjednodušší přístup je vygenerování pravidelné mřížky, například ortogonální a následná modifikace polohy každého bodu

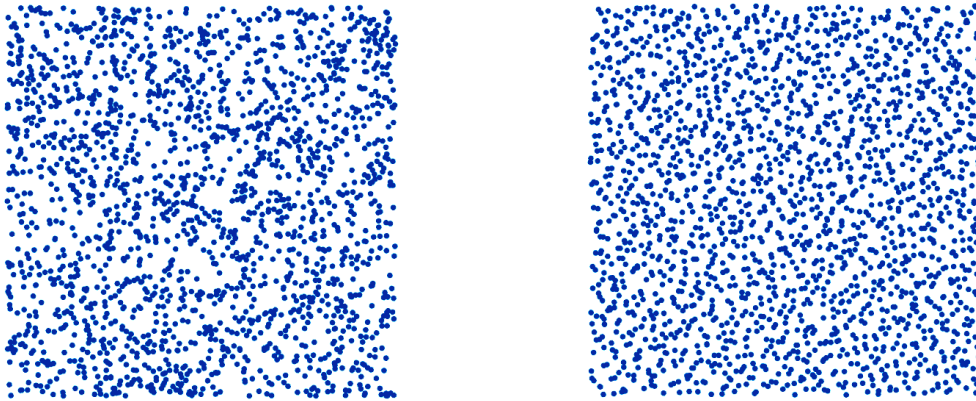


Obr. 2.2: Voroného diagram – rozdělení krychle na buňky podle vložených bodů

náhodným posunutím v předem definovaném rozsahu, viz [7]. Nevýhodou tohoto přístupu zůstává stále značná podobnost s pravidelnou mřížkou, případně může docházet k lokálnímu naředění, či nahuštění mřížky pokud je náhodné rozptýlení příliš velké.

Nízce diskrepanční algoritmy

Jinou možností, jak vhodně body rozmístit, je využití *nízce diskrepančních* sekvenčních algoritmů. Jedná se o skupinu algoritmů, která je schopna generovat takovou posloupnost, která rovnoměrně pokryje stanovený prostor, ale přitom není lineárně závislá. Příkladem je *Sobolova* nebo *Haltonova* posloupnost [9]. Do aplikace byla implementována Haltonova sekvence (obr. 2.3) s modifikacemi pro vyloučení možných lineárních korelací.



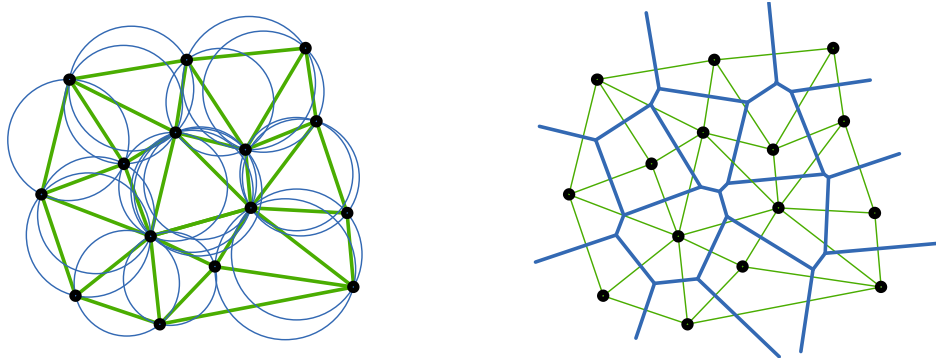
Obr. 2.3: Body vkládány generátorem náhodných čísel (vlevo) a Haltonovou sekvencí (vpravo)

2.2.2 Triangulace

Před samotným propojením vzniklých bodů je nezbytné správně určit dvojice, mezi kterými budou jednotlivé pružiny vytvořeny. Aby takto vytvořená síť vykazovala co

nejvěrnější chování, je potřebné vzájemně propojit body ležící ve svém nejbližším okolí s co možná nejpravidelnější sítí, viz [4]. Tuto podmínku můžeme splnit za pomoci *triangulace*, resp. ve 3D za pomoci *tetrahedronizace*.

Triangulací rozumíme rozdělení rovinné oblasti na soustavu trojúhelníků vytvořené ze spojnic bodů. V případě tetrahedronizace rozdělujeme prostorovou oblast na soustavu čtyřstěnů – tetraedrů.¹ Zvláštním případem je tzv. Deloneho tetrahedronizace, která splňuje podmínku, že žádný další bod neleží v kouli opsané tetraedru, tedy spojnice vrcholů takových čtyřstěnů spojují právě nejbližší body [3]. Příklad takového řešení v 2D je uveden na obr. 2.4.



Obr. 2.4: Deloneho triangulace množiny bodů a její duální struktura – Voroného mozaika

Tento algoritmus tedy rozděljuje prostor tak, aby jednotlivé čtyřstěny odpovídaly co nejvíce pravidelnému čtyřstěnu a je hojně využíván v inženýrských úlohách [1], [23].

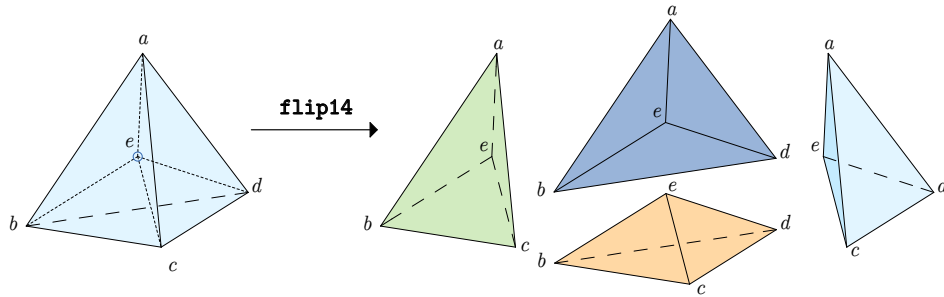
Triangulace metodou inkrementálního vkládání

Výchozím stavem pro takovou konstrukci je fiktivní čtyřstěn – tetraedr mnohonásobně svými rozměry překračující obálku všech vkládaných bodů sítě, který se po dokončení triangulace odstraní spolu se všemi dalšími čtyřstěny majícími jako vrchol některý z fiktivních bodů.

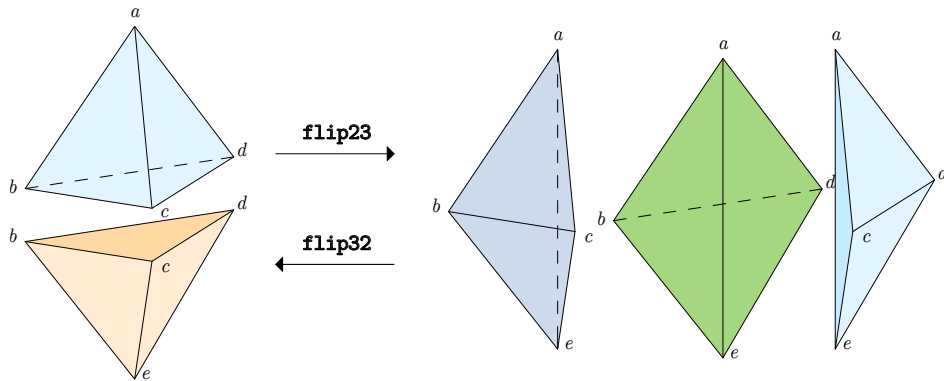
Po vložení bodu do prostoru existujícího čtyřstěnu se z jeho vrcholů vedením spojnic k vloženému bodu vytvoří jiné čtyři čtyřstěny (obr. 2.5) a původní tetraedr se zahodí. Tento postup je zpravidla označován jako `flip14`. Novým čtyřstěnům se předají informace o sousedících čtyřstěnech a sousedech mateřského tetraedru, které jsou rovněž informovány o změně jejich sousedů. Následně se zaregistrují požadavky na ověření Deloneho podmínky pro všechny nově vytvořené dvojice čtyřstěnů s přilehlými mateřskými sousedy.

Pokud posuzovaná dvojice sousedících čtyřstěnů nesplňuje Deloneho podmínku, dojde k transformacím typu `flip23`, `flip32` (obr. 2.6) nebo `flip44` a po kterékoliv

¹Ačkoliv by jako český výraz pro 3D triangulaci byla vhodnější spíše “tetrahedronizace“, v literatuře se dle anglického “tetrahedron“ objevuje právě výraz tetrahedronizace.



Obr. 2.5: Transformace vložení bodu flip14



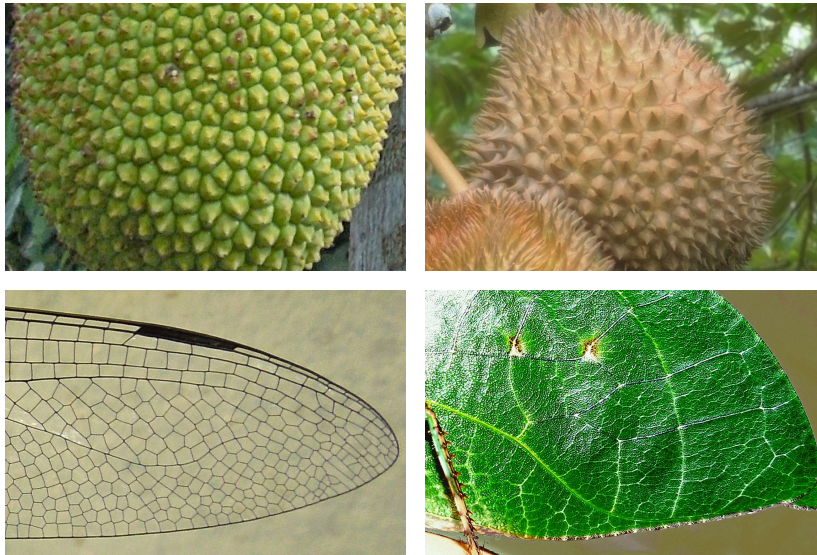
Obr. 2.6: Transformace flip23 a flip32

z transformací dochází dále k informování okolí a novým testům. Rozhodnutí o tom, jakou transformaci provést, je závislé na poloze testovaných bodů a k nim přilehlých čtyřstěnů. Nejprve se proto ověří, zda čtyři body sousedících čtyřstěnů neleží v rovině. Pokud ano, provede se transformace **flip44**, která transformuje čtyřstěny se čtyřmi body ležícími v rovině rozdělením podle sdílené roviny s vytvořením jiných čtyř čtyřstěnů. Prakticky se jedná o analogický případ známý ve 2D jako **flip22** [12], [13].

2.2.3 Voroného diagramy

Z vytvořené triangulace lze snadno odvodit geometrii pružinové sítě, jelikož je tvořena hranami vytvořených čtyřstěnů. Triangulace sama ale neposkytuje informace nezbytné k tomu, aby mohla být vzniklým pružinám přiřazena adekvátní plocha průřezu a spojeným bodům odpovídající hmotnost. Právě tyto informace lze ale efektivně získat využitím již zmíněného *Voroného diagramu*, též nazývaného jako *Voroného dekompozice* či *Dirichletova tesalace*.

Voroného diagram je tzv. duální strukturou k triangulaci, či tetrahedronizaci ve 3D, viz obr. 2.4. Voroného diagram vytváří kolem bodů diagramu buňky, jejichž stěny leží v přesné polovině vzdáleností sousedících bodů. Praktickým příkladem Voroného diagramu může být včelí plástev, vznik prasklin při vysychání půdy, některé typy žilnatin listů, rozložení buněk v organismu a další přírodní struktury, viz obr. 2.7.



Obr. 2.7: Příklady Voroného buněk: plod chlebovníku (vlevo nahoře) a durianu (vpravo nahoře), křídlo vážky (vlevo dole) a kobylky s žilnatinou listu (vpravo dole)

Jelikož stěny Voroného buňky leží v polovině vzdáleností sousedních bodů, pak průsečíky těchto stěn, resp. vrcholy buněk budou identické se středy opsaných kulových ploch okolních čtyřstěnů. Vrcholy mnohoúhelníku tvořícího stěnu Voroného buňky získáme, použijeme-li středy koulí právě těch čtyřstěnů, jenž mají jako vrcholy oba body spojované příslušnou pružinou, která odpovídá i normále hledané stěny.

2.3 FyDiK

Jakmile je výše uvedenými způsoby zajištěno geometrické uspořádání modelu, lze přistoupit k definování fyzikálních vlastností a k samotnému sestavení modelu FyDiK, sestávajícího z hmotných bodů a translačních pružin.

Hmotný bod

Do hmotných bodů (obr. 2.8) je soustředěna veškerá hmotnost a kinetická energie E_k systému. Pro bod s indexem i , hmotností m_i a rychlostí \vec{v}_i lze kinetickou energii vyjádřit vztahem:

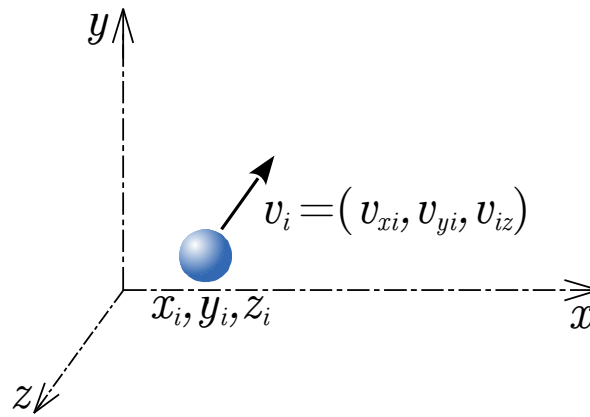
$$E_{ki} = \frac{1}{2} m_i v_i^2, \quad (2.1)$$

rozepsáno pomocí složek vektoru rychlosti v_{xi} , v_{yi} , v_{zi} pak do tvaru:

$$E_{ki} = \frac{1}{2} m_i (v_{xi}^2 + v_{yi}^2 + v_{zi}^2). \quad (2.2)$$

Celková kinetická energie modelu je pak:

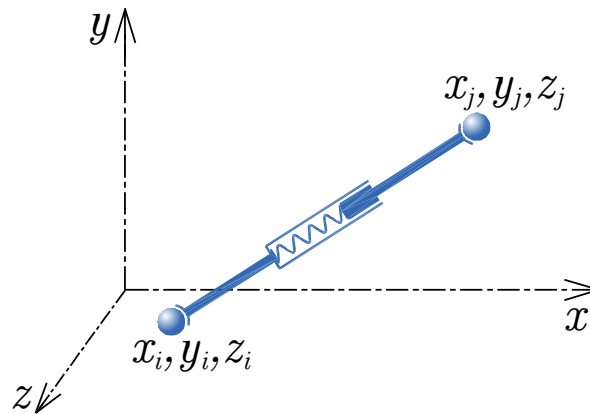
$$E_k = \sum E_{ki}. \quad (2.3)$$



Obr. 2.8: Hmotný bod

Translační pružina

Veškerá potenciální energie E_p modelu je koncentrována do translačních pružin (obr. 2.9). Ty spojují vždy dva hmotné body a zajišťují přenos normálových sil modelu.



Obr. 2.9: Translační pružina

Pro lineární pružinu spojující body s indexem i a j , tuhostí k_{ij} ² a délkou l_{ij} lze normálovou sílu F_{ij} vyjádřit jako:

$$F_{ij} = k_{ij} \Delta l_{ij}, \quad (2.4)$$

kde Δl_{ij} je prodloužení, či zkrácení pružiny oproti její výchozí délce l_{0ij} :

$$\Delta l_{ij} = l_{ij} - l_{0ij}, \quad (2.5)$$

Potenciální energii pružiny pak lze vyjádřit vztahem:

$$E_{pij} = \frac{1}{2} k_{ij} \Delta l_{ij}^2, \quad (2.6)$$

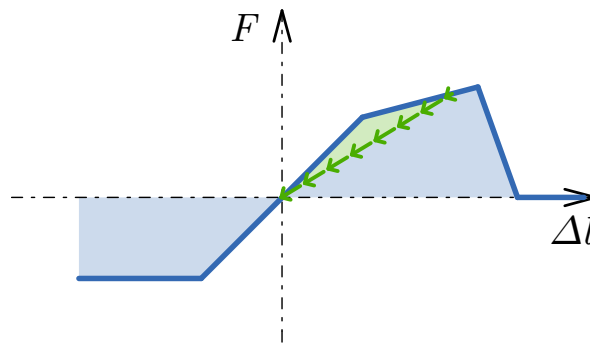
²Pro zjednodušení předpokládáme lineární chování materiálu a tedy konstantní tuhost pružiny.

nebo pomocí složek souřadnic jednotlivých bodů jako:

$$E_{pij} = \frac{1}{2} k_{ij} \left(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} - l_{0ij} \right)^2. \quad (2.7)$$

Pracovní funkce translačních pružin

Funkce použitá v modelu má polygonální tvar, což umožňuje její snadné definování i implementaci. Pro příklad mějme pracovní funkci znázorněnou na obr. 2.10. V případě malého protažení, či zkrácení se pružina chová ideálně elasticky. Při větším tlaku ideálně plasticky. V případě rostoucího tahu, následuje plastická část se zpevněním až do náhlého přetržení. Další protažení již nevyvozuje v pružině žádnou sílu. V případě, že se již natažená pružina navrací k výchozí délce, výsledná síla je dopočítaná podle tuhosti získané proložením dosaženého extrému s výchozím stavem (znázorněno zpětnými šipkami).



Obr. 2.10: Příklad pracovní funkce translační pružiny

Řešení systému

S vyjádřením kinetické a potenciální energie systému pomocí zobecněných souřadnic a rychlostí pak lze sestavit tzv. *Lagrangián* L :

$$L = E_k - E_p, \quad (2.8)$$

a získat pohybové rovnice systému užitím vztahu:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) = \frac{\partial L}{\partial q_i}, \quad i = 1, 2, \dots, n, \quad (2.9)$$

kde q_i je zobecněná souřadnice, $\dot{q}_i = dq_i/dt$ její zobecněná rychlost a t je čas. Tyto proměnné jednoznačně popisují stav systému v čase.

Za předpokladu, že známe počáteční stav systému $\mathbf{x}(t_0) = \mathbf{x}_0$ je řešením diferenciálních rovnic vektor funkcí:

$$\mathbf{x}(t) = (x_1(t), x_2(t), \dots, x_n(t)), \quad (2.10)$$

udávající hodnoty stavových proměnných \mathbf{x} v čase t .

Řešení takového systému je možné nalézt s využitím řady numerických metod, viz [18], přičemž v tomto případě se jako nejvhodnější³ jeví použití *Symplektické Eulerovy metody*, jež určuje polohový vektor \vec{x}_i bodu i s časovým krokem h v čase $t+h$ vztahem:

$$\vec{x}_i(t+h) = \vec{x}_i(t) + h\vec{v}_i(t+h), \quad (2.11)$$

přičemž za předpokladu, že v daném časovém kroku zůstává zrychlení daného bodu \vec{a}_i konstantní, můžeme určit rychlost \vec{v}_i v čase $t+h$ jako:

$$\vec{v}_i(t+h) = \vec{v}_i(t) + h\vec{a}_i(t). \quad (2.12)$$

Zrychlení můžeme snadno odvodit z druhého *Newtonova zákona*:

$$\vec{a}_i = \frac{\vec{F}_i}{m_i}. \quad (2.13)$$

Hmotnost jednotlivých bodů m_i uvažujeme v čase konstantní, mění se tedy pouze vektor síly \vec{F}_i působící v daném bodě. Ten je výslednicí normálových sil přilehlých translačních pružin, tedy:

$$\vec{F}_i = \sum \vec{F}_{ij}, \quad j = 1, 2, \dots, n_i, \quad (2.14)$$

kde n_i je počet připojených translačních pružin k danému bodu. V případě zohlednění lineárního a kvadratického tlumení c_1 resp. c_2 pak:

$$\vec{F}_i = \sum \vec{F}_{ij} - (c_1 v_i + c_2 v_i |v_i|), \quad j = 1, 2, \dots, n_i. \quad (2.15)$$

Tímto způsobem je možné s využitím relativně jednoduchého matematického aparátu systém přehledně řešit.

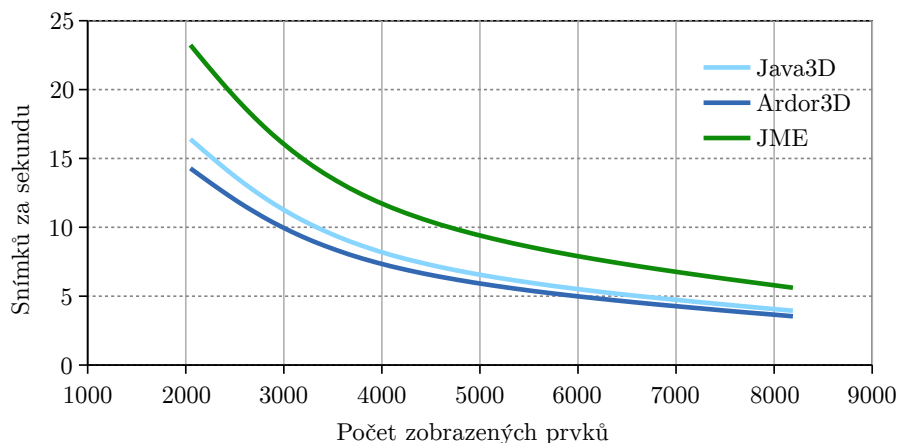
Podrobnější informace k modelu *FyDiK* a metodám řešení lze nalézt především v habilitační práci autora modelu [8], případně v publikacích, jež se věnují obdobným modelům [6] a [5].

2.4 Programová platforma

Efektivní práce s modelem vyžaduje naprogramování aplikace, jež bude uplatňovat výše uvedené principy. Pro vytvoření takového programu byl zvolen programovací jazyk *Java* společnosti *Oracle*. Pro výběr nejvhodnější vývojového nástroje byla zohledněna řada kritérií. Z výběru byly vyřazeny komerční projekty, projekty již ukončené, projekty bez uživatelské podpory a s malou uživatelskou základnou, čímž došlo ke zúžení pouze na tři nástroje a to *Java3D*, *Ardor3D*⁴ a *jMonkeyEngine (JME)*.

³Vhodnost metody závisí především na její numerické stabilitě a výpočetní náročnosti.

⁴Vývoj prostředí *Ardor3D* byl oficiálně ukončen v roce 2014 a v současné době je udržován pod názvem *JogAmp's Ardor3D Continuation*



Obr. 2.11: Porovnání výkonu jednotlivých vývojových nástrojů v testu dynamické geometrie [17]

Ze všech zmíněných nástrojů byl zvolen 3D framework jMonkeyEngine 3.0 a to díky své široké komunitě, velmi dobré dokumentaci s řadou praktických příkladů, výkonu a v neposlední řadě licenční politice a IDE.

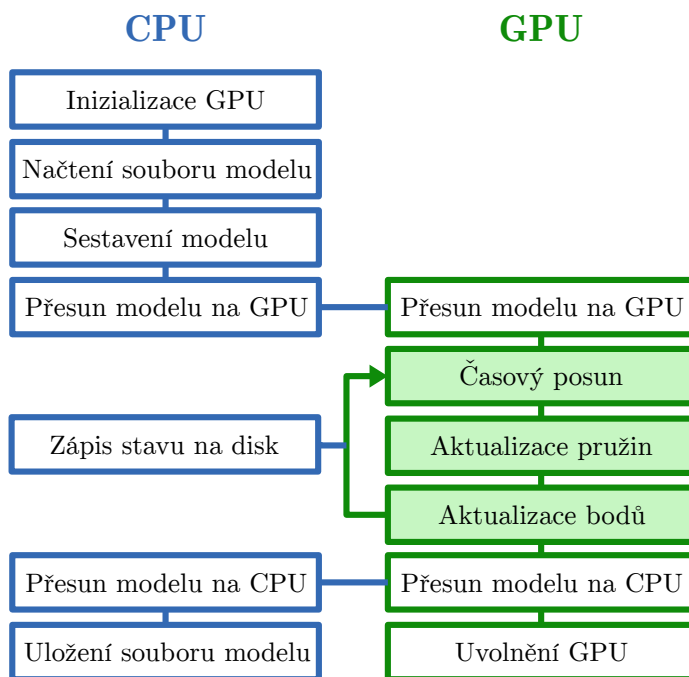
2.5 Aplikace FyDiK na platformě CUDA

Z hlediska výpočetního potenciálu hrají v současnosti GPU procesory určitě velmi důležitou roli, ne-li roli hlavní. Navíc, z hlediska širokého spektra možností jejich využití se jeví, že tento trend bude ještě nějakou dobu pokračovat. Vzhledem k struktuře modelu FyDiK umožňující jeho paralelizaci, by jistě byla škoda v tomto případě potenciálu GPGPU nevyužít.

Zdá se, že největší slabinou paralelizace je řešení sériově závislých úloh, tedy úloh, které není možné rozdělit mezi více procesorů - řešičů. V případě modelu FyDiK to jsou například jednotlivé časové kroky. Ty jsou tvořeny třemi, rovněž závislými, celky:

- **Časový posun** – aktualizace stavových proměnných (rychlosti a polohy bodu v závislosti na jejich zrychlení, resp. rychlosti).
- **Aktualizace pružin** – aktualizace délky pružin a normálových sil působících na přilehlé body.
- **Aktualizace bodů** – aktualizace zrychlení v závislosti na přilehlých pružinách.

Výpočet jednotlivých uvedených celků ale závislý není. Tedy např. aktualizace jednotlivých bodů v daném časovém kroku je nezávislá na ostatních aktualizovaných bodech, což umožňuje tyto celky paralelizovat. V případě modelu FyDiK obsahujícího 10^5 bodů to pak znamená, 10^5 současně spuštěných vláken, které jsou zpracovávány paralelně podle instrukcí v kódu a patřičných možností hardware. Tedy použijeme-li kartu Tesla K20X architektury Kepler, pak může probíhat aktualizace 2688 bodů současně.



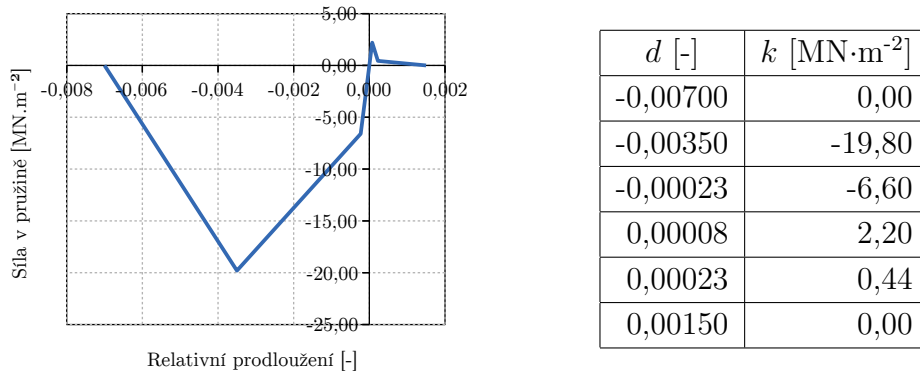
Obr. 2.12: Zjednodušené schéma aplikace FyDiK na platformě CUDA s rozdělením instrukcí vykonávanými na CPU a GPU

Analogicky totéž platí pro ostatní celky, jenž jsou ve schématu aplikace na obr. 2.12 podbarveny zeleně a v prostředí GPU jsou samostatnými kernely.

3 VYHODNOCENÍ

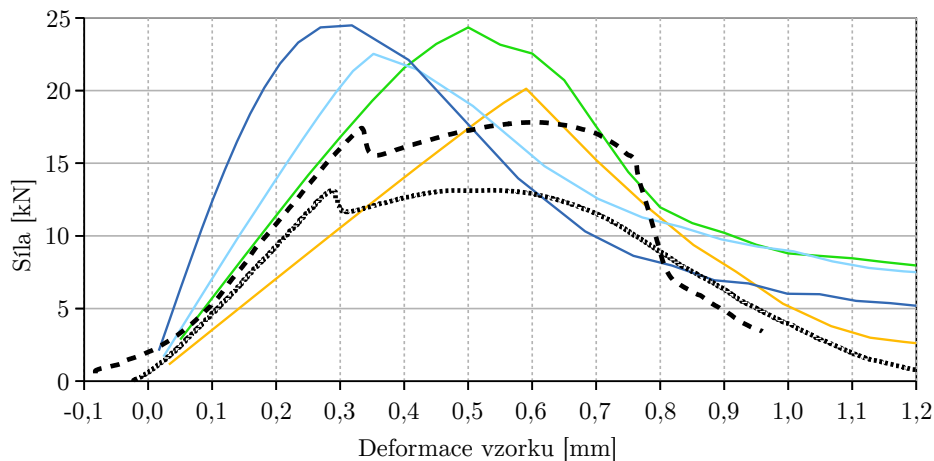
3.1 Beton

Pro hmotné body betonu byla použita nejprve objemová hmotnost $\rho = 2300 \text{ kg/m}^3$ a lineární tlumení o hodnotě $c = 0,003 \text{ Ns/m}$. Počáteční funkce translačních pružin byla odvozena s přihlédnutím k hodnotám uvedeným v ČSN EN 206-1, viz obr. 3.1.¹



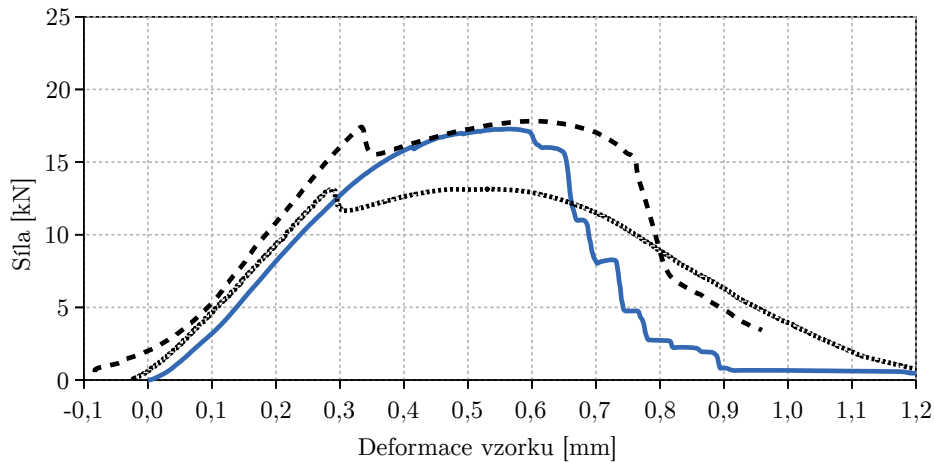
Obr. 3.1: Počáteční pracovní funkce translačních pružin

Se zohledněním vlivu kontaktu a odladěnými parametry modelu pro beton odpovídají výsledky simulací velmi dobře naměřeným hodnotám, a to zvláště v kontextu výsledků stejného vzorku získaných prostřednictvím softwaru *ATENA 3D*, srovnej obr. 3.2 a 3.3.



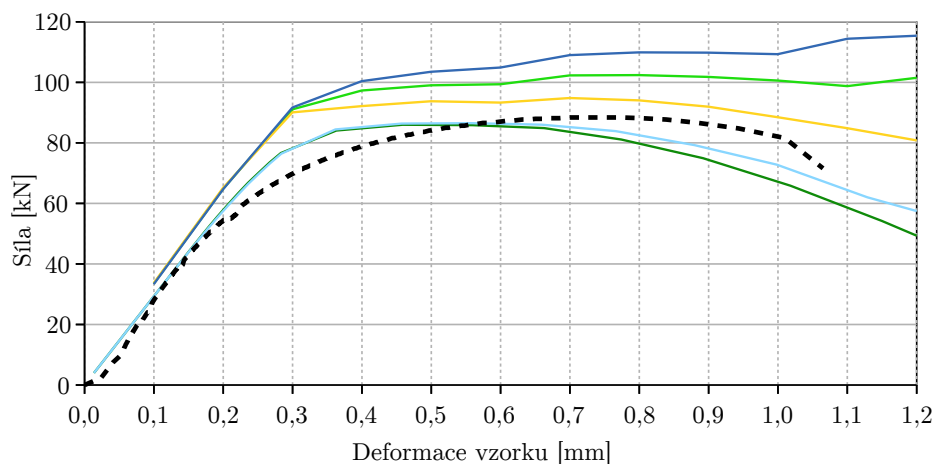
Obr. 3.2: Diagramy síla–posun zatěžování vzorku R3-1C (přerušovaně) a simulací prostřednictvím softwaru *ATENA* s různými parametry (barevně odlišeno) [15]

¹Je potřeba vzít v úvahu, že mezi pracovním diagramem modelovaného materiálu a pracovní funkcí translačních pružin neexistuje zatím zcela prozkoumaná souvislost, a tudíž se skutečně jedná pouze o inicializační hodnoty následujícího iteračního procesu.



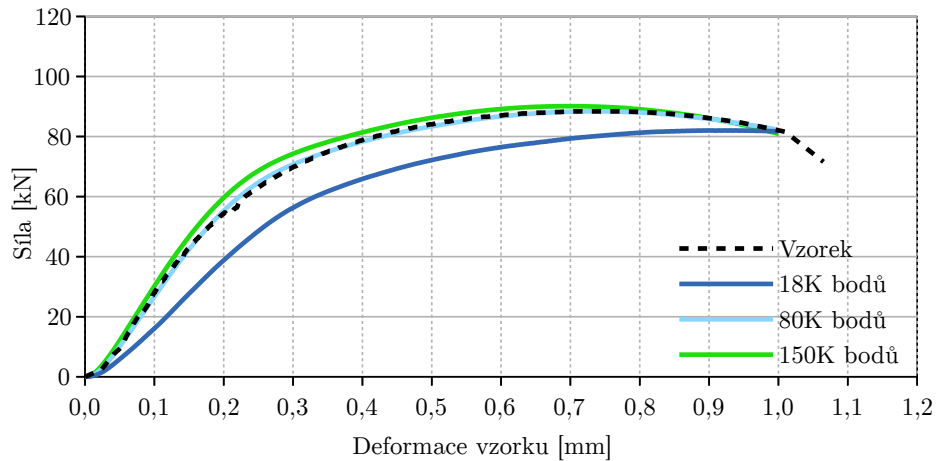
Obr. 3.3: Diagramy síla–posun zatěžování vzorku R3-1C (přerušovaně) a FyDiK simulace se zohledněním kontaktu (modře)

Uvedený model posloužil pro všechny následující simulace s pouhou modifikací parametrů pracovních funkcí pružin. Ačkoliv je tímto způsobem možné velmi věrně vystihnout naměřený průběh diagramů síla–posun, je potřeba mít na paměti, že se jedná o diskretizovaný model určitého zrnitostního stupně a i samotná diskretizace podléhá pseudonáhodnému generování bodů. Tedy i při použití stejného materiálu a způsobu zatěžování se mohou dosažené výsledky odlišovat. Do jaké míry se takové výsledky liší, znázorňuje obr. 3.5 (srovnej s obr. 3.4), kde byl případ R3-CC za použití stejného materiálu modelován pomocí asi $1,8 \cdot 10^4$, $8 \cdot 10^4$ a $1,5 \cdot 10^5$ bodů, resp. s limitní vzdáleností 3,0 mm, 2,0 mm a 1,5 mm.



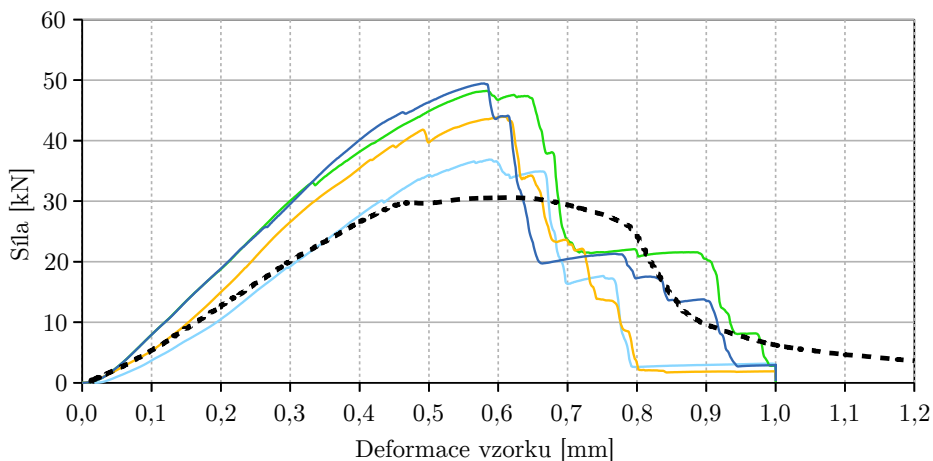
Obr. 3.4: Diagram síla–posun zatěžování vzorku R3-CC (přerušovaně) a simulace prostřednictvím softwaru ATENA s různými parametry (barevně odlišeno) [15]

Obdobně se projeví i vliv různého rozložení bodů v důsledku různého počátku generování pseudonáhodných bodů. Takovýto rozdíl je obzvláště patrný v místech, kde



Obr. 3.5: Diagramy síla–posun zatěžování vzorku R3-CC (přerušovaně) a FyDiK simulace se zohledněním kontaktu a různým stupněm zrnitosti - počtem bodů

je model relativně řídký, tedy kde vzdálenost jednotlivých bodů není výrazně menší, nežli okolní geometrie. Takovýmto příkladem je typ zatěžování 1C, kde příložka 8×8 mm zatěžuje body se vzájemnou průměrnou vzdáleností dosahující téměř 3 mm. Rozdíl zachycují výsledky v porovnání se zatěžováním R4-1C na obr. 3.6.

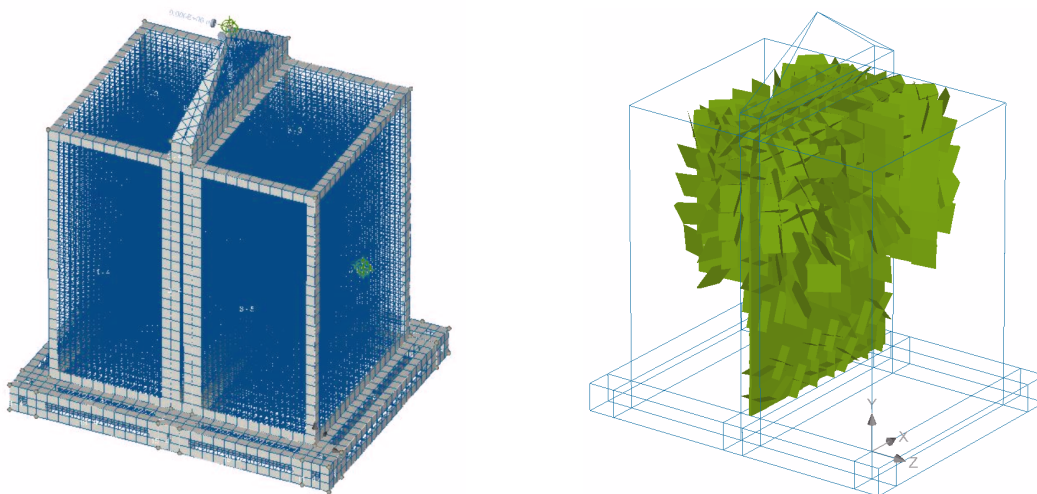


Obr. 3.6: Diagram síla–posun vzorku R4-1C pro různé realizace výpočetní sítě modelu³

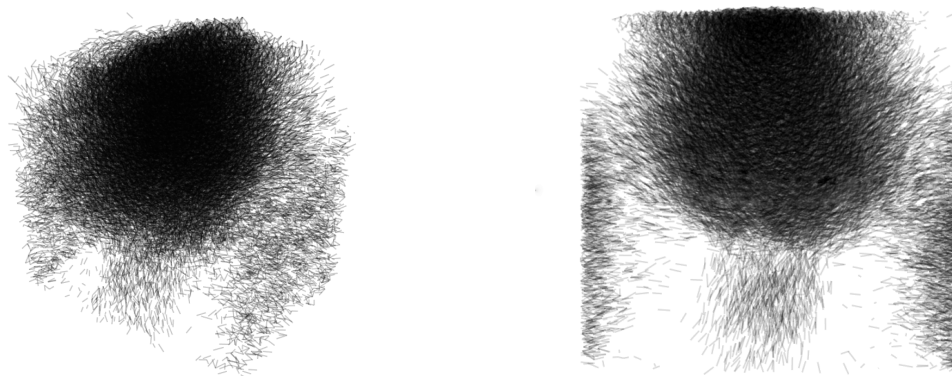
Výstupy dosažené modelem FyDiK byly dále porovnány z hlediska rozvoje trhlin s výsledky modelu ATENA 3D získané Jakubem Sobkem [16]. Pro toto porovnání byl použit vzorek R3-1C, který je z hlediska zatěžování i rozvoje trhlin na rozdíl od vzorku R3-CC podstatně složitější.

Obrázky 3.8 a 3.9 znázorňují rozvoj trhlin, resp. dosažení mezního tahového napětí v pracovní funkci pružin modelu FyDiK. Ačkoliv model ATENA 3D (obr. 3.7) používá

³Pro různé počáteční hodnoty generátoru pseudonáhodných čísel, jež jednoznačně určují sekvenci generovaných bodů.



Obr. 3.7: Model v prostředí ATENA 3D [20] (vlevo) a vznik trhlin vzorku R3-1C [16] (vpravo)



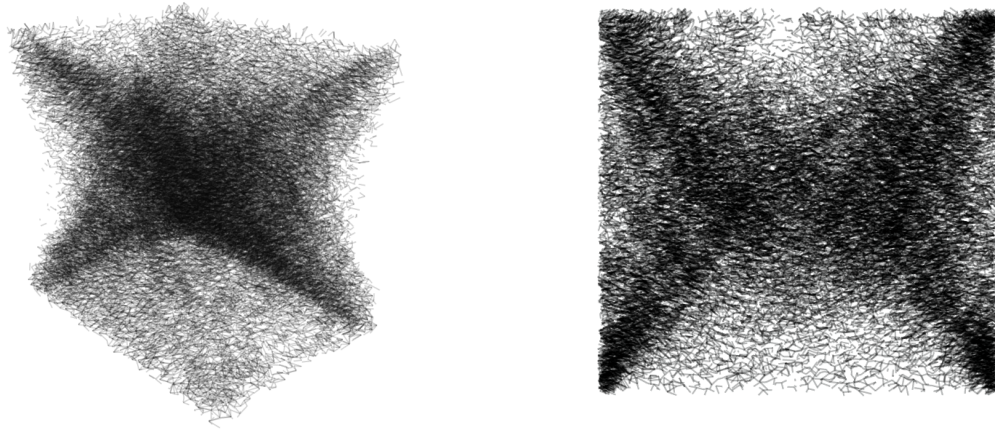
Obr. 3.8: Vizualizace vzniku trhlin prostřednictvím modelu FyDiK na vzorku R3-1C (vlevo celkový pohled v perspektivě) (vpravo řez středem modelu)

zřetelně hrubější výpočetní síť než zrnitost druhého modelu, oba přístupy vykazují pod zatěžovací příložkou oblast porušení prakticky totožných rozměrů.

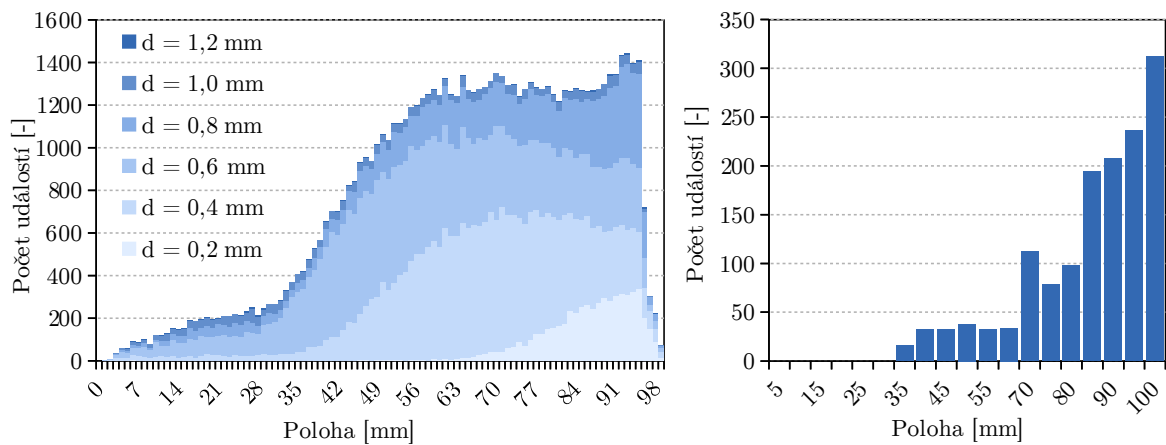
Za povšimnutí zřejmě stojí i výsledek simulace celoplošného zatěžování R3-CC. Ten na rozdíl od předchozího případu vykazuje v porušování zřetelnou symetrii v prostrovém tvaru písmene „X“, viz 3.9.

Modely vzorku R3-1C byly dále porovnány prostřednictvím zaznamenaných událostí v jednotlivých souřadných osách, tedy dle vzniku porušení ve smyslu výše uvedeného případu. Model byl v dané ose rozdělen po pevně daných vzdálenostech a v každé takto vytvořené oblasti sledována četnost událostí – porušení.

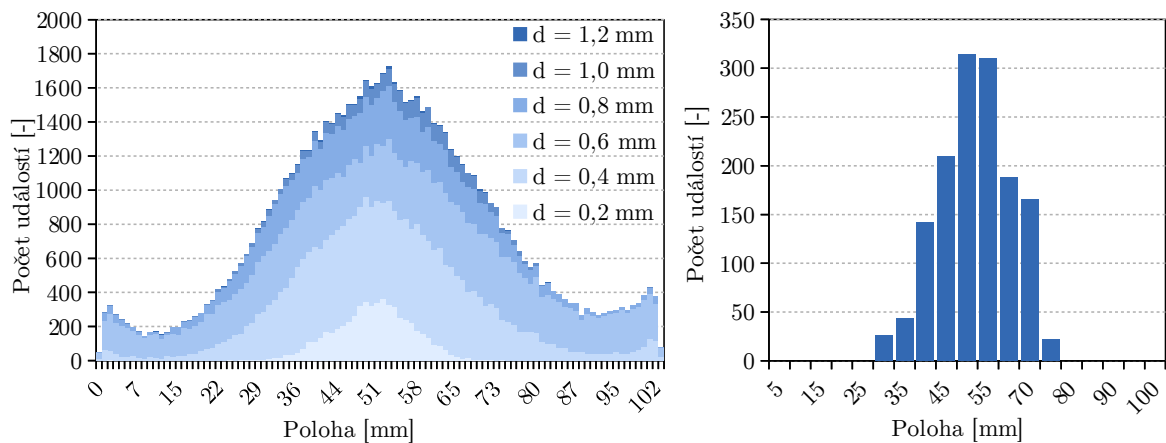
Zatímco výsledky modelu ATENA 3D zachycují konečný počet událostí v čase maximálního zatížení, model FyDiK v jednotlivých časech odpovídajících vynucené deformaci vzorku. Z histogramů uvedených na obr. 3.10 a 3.11 je opět patrná poměrně dobrá



Obr. 3.9: Vizualizace vzniku trhlin prostřednictvím modelu FyDiK na vzorku R3-CC (vlevo celkový pohled v perspektivě) (vpravo řez středem modelu)



Obr. 3.10: Zaznamenané události (vznik trhlin) na vzorku R3-1C ve směru osy Y: prostřednictvím modelu FyDiK (vlevo), ATENA 3D (vpravo) [16]



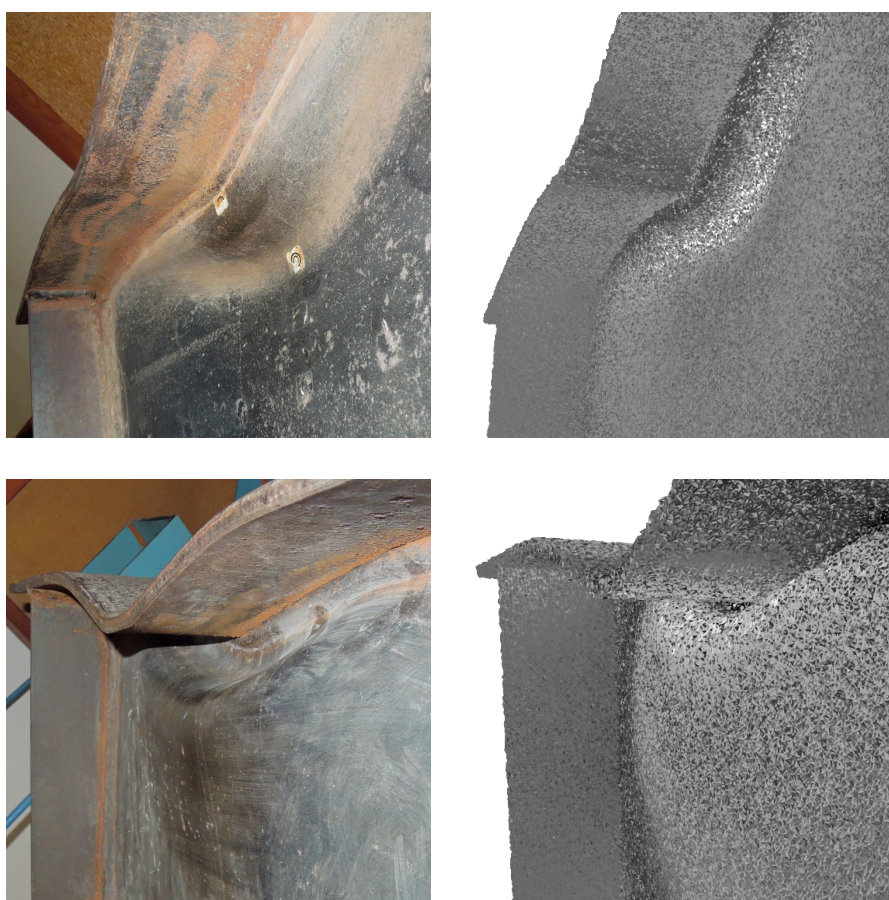
Obr. 3.11: Zaznamenané události (vznik trhlin) na vzorku R3-1C ve směru osy Z: prostřednictvím modelu FyDiK (vlevo), ATENA 3D (vpravo) [16]

shoda obou modelů.⁴ Toto zobrazení navíc v případě modelu FyDiK dává přehledně znát jakým způsobem docházelo k propagaci porušení v jednotlivých fázích zatěžování.

3.2 Ocel

Základní materiálové charakteristiky byly převzaty z běžných hodnot stavební oceli S235. Objemová hmotnost materiálu $\rho = 7850 \text{ kg/m}^3$, modul pružnosti $E = 210 \text{ GPa}$, mez kluzu $f_y = 235 \text{ MPa}$, mez pevnosti v tahu $f_u = 360 \text{ MPa}$ a minimální tažnost 20 %.

Jelikož ze zmíněného experimentu svařovaného I profilu nebyla dostupná data zatěžování, výsledky modelu FyDiK lze proto obtížně porovnat.

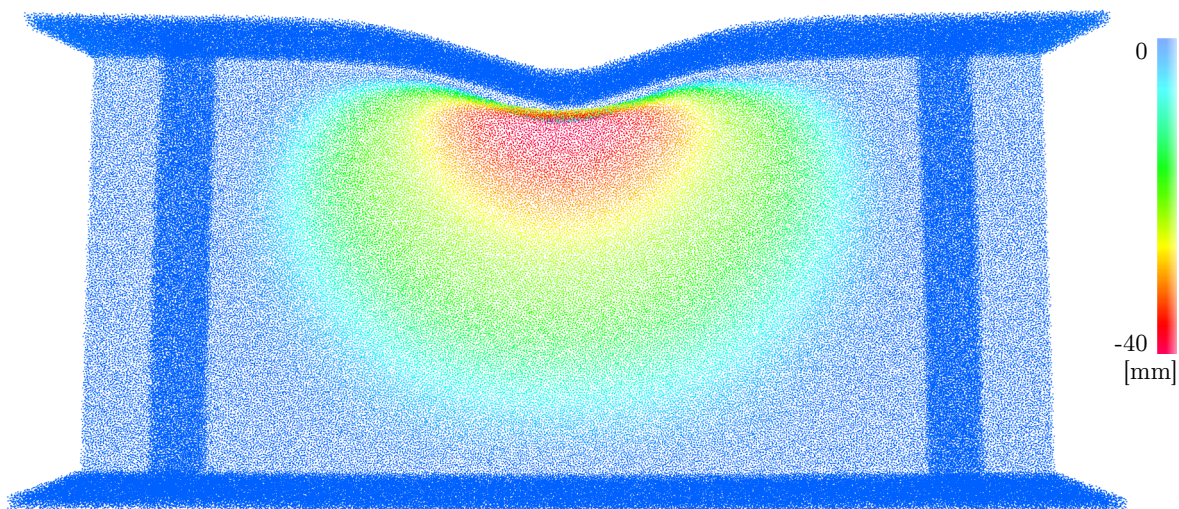


Obr. 3.12: Vizuální porovnání detailů deformace z experimentu (vlevo) a simulace (vpravo)

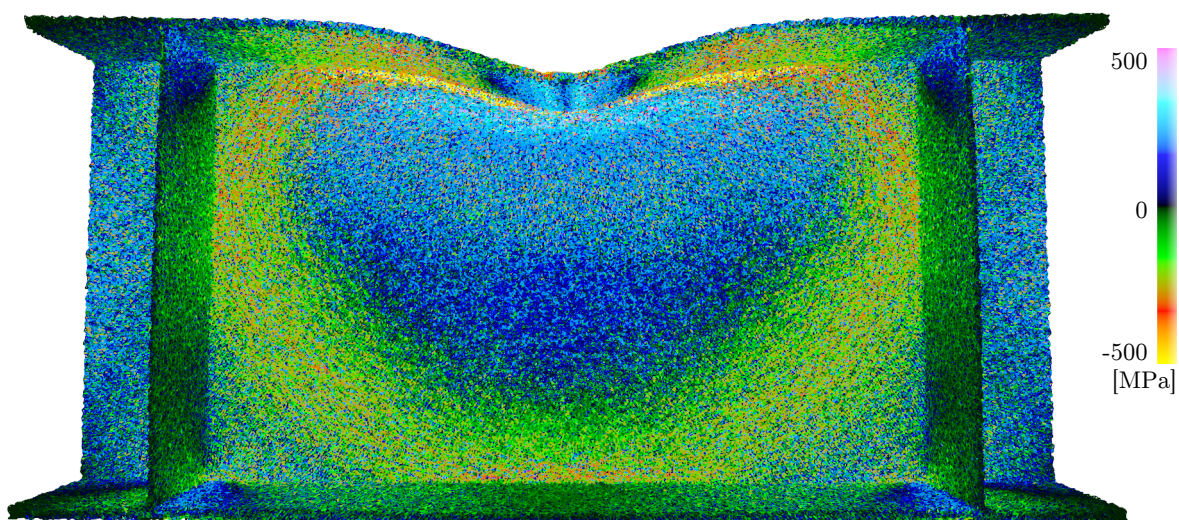
Přesto alespoň vizuální porovnání detailu deformace pod místem zatěžování na obr. 3.12 vykazuje poměrně dobrou shodu modelu s experimentem. Výrazněji se pouze

⁴Výrazná odlišnost v absolutním počtu událostí mezi modely ATENA 3D a FyDiK pouze podtrhuje rozdílnost rozlišení modelu, tedy odlišnost velikostí použitých prvků.

odlišuje místo maxima, které je u experimentu posunuto níže. To ovšem může být přisouzeno svarům, které tak lokálně zvětšují tuhost stojiny a v modelu nejsou zohledněny. Naměřená hodnota maximálního vybočení v ose kolmé ke stojině je v obou případech prakticky totožná $40,0 \text{ mm} \pm 2,0 \text{ mm}$.⁵



Obr. 3.13: Celková konečná deformace nosníku v ose kolmé na stojinu



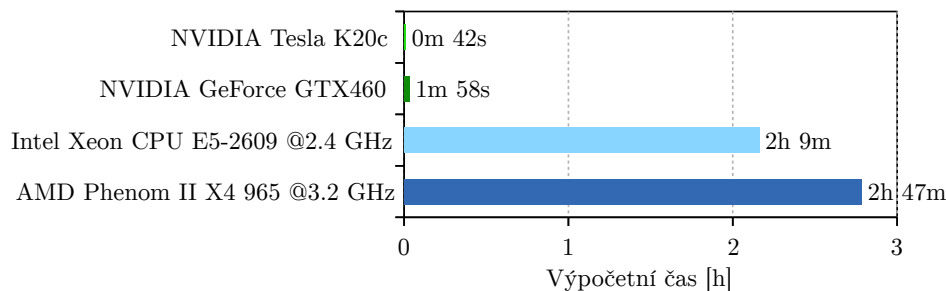
Obr. 3.14: Konečný průběh napětí v jednotlivých pružinách modelu nosníku

3.3 Použití CUDA platformy

S využitím všech optimalizačních technik, uvedených v celém znění práce, byl model počítán nejen na platformě CUDA, ale pro porovnání také na běžně dostupných CPU,

⁵Vzhledem k celkové deformaci stojiny bylo celkové vybočení obtížně změřitelné.

viz obr. 3.15. Ve srovnání s uvedeným procesorem Intel Xeon je výpočet na architektuře Fermi, kartě GeForce GTX460 rychlejší $66\times$ na architektuře Kepler, kartě Tesla K20c $184\times$. Ve srovnání s procesorem AMD Phenom je v tomto případě platforma CUDA $85\times$ resp. $238\times$ rychlejší. Přesto výpočet betonového prvku s $8\cdot 10^4$ body trvá na kartě Nvidia Tesla K20 asi 30 min a výpočty ocelových profilů na stejnojmenné kartě zabraly 4 až 15 hodin.



Obr. 3.15: Spotřebovaný čas pro výpočet 25 000 kroků - porovnání platforem a procesorů

Pro porovnání rychlosti výpočtu s výsledky v literatuře lze výkon dosažený na kartě GeForce GTX460 uvést hodnotou $22,58\cdot 10^6$ částic/s. Při přepočtu na normativní kartu architektury Fermi Tesla C2050 bude výkon odpovídat $25,65\cdot 10^6$ částic/s. Obdobné řešení na stejné architektuře lze nalézt v článku [22], kde při použití karty Tesla C2050 byl dosažen výkon $22,36\cdot 10^6$ částic/s. Zjednodušeně lze tedy říci, že představené řešení dosahuje srovnatelné rychlosti. Z jiných článků nelze s odpovídající přesností určit rychlost výpočtu, ovšem např. [24] provádí výpočet také na architektuře Fermi s pokročilejší kartou GeForce GTX580⁶ a přitom ve srovnání s CPU⁷ řešením dosahuje zrychlení pouze asi $23\times$. Samotný faktor zrychlení kromě toho, že je závislý na mnoha činitelích, bývá v literatuře také různě vyjádřen. A to buď ve srovnání vůči CPU, jako ve výše uvedeném případě, nebo vůči jedno-vláknovému výpočtu na stejném hardware, viz [25] a [21].

Zřejmě je vhodné rovněž poznamenat, že pokud je pro výpočet použita běžná GPU grafická karta, jakou je například GeForce GTX460, jsou její pořizovací náklady srovnatelné, nebo dokonce nižší než cena samostatného CPU. Tento použitý hardware přitom zdaleka nereprezentuje vrchol dnes dostupného výkonu a zároveň, jak ukazuje srovnání architektury Fermi a Kepler, další vývoj poskytuje i nové možnosti optimalizace kódu. Díky tomu se zdá, že potenciál grafických karet pro GPGPU, resp. pro aplikaci FyDiK není ještě zdaleka vyčerpán a stále existují další metody jak zvýšit efektivitu a rychlost tohoto již tak působivého řešení.

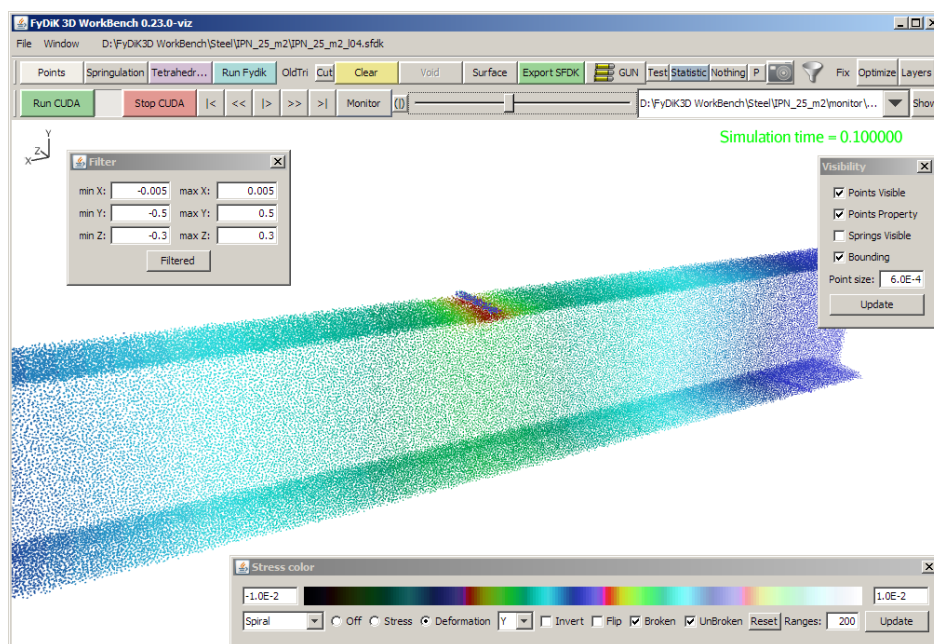
⁶Nárůst výkonu oproti GeForce GTX460 je více než 70 %.

⁷V tomto případě byl zřejmě použit procesor *Intel Core i7-960 @3.20 GHz*, který výkonnostně přibližně odpovídá procesoru *Intel Xeon E5-2609 @2.40 GHz*.

4 ZÁVĚR

K dosažení uvedených výsledků práce byla vytvořena aplikace, jež umožňuje uživateli komplexní práci s představeným modelem FyDiK. K jejímu vytvoření bylo nezbytné naprogramovat nejen zmíněný systém generování bodů ve vymezeném prostoru, Deloneho tetrahedronizaci, převod na Voronoiovu mozaiku a následné sestavení a export modelu FyDiK, ale také uživatelské prostředí poskytující celkovou správu modelu a jeho zobrazení, to vše s důrazem na co nejnižší výpočetní náročnost.

V současné podobě tak aplikace umožňuje práci s modely obsahujícími více než 10^6 bodů a 10^7 pružin, což pro většinu případů zcela postačuje. Při použití běžného hardware¹, je model s cca $3 \cdot 10^6$ pružinami překreslován $30\times$ za sekundu.



Obr. 4.1: Okno vytvořené aplikace pro práci s modelem FyDiK

K samotnému výpočtu modelu bylo použito masivní paralelizace na platformě CUDA. Ta ve srovnání s běžným CPU řešením poskytuje více než stonásobné zrychlení, a to také díky provedené optimalizaci zohledňující samotnou strukturu použitého hardware.

S využitím uvedených nástrojů prokázalo použití modelu FyDiK na uvedených příkladech schopnost výstižně simulovat prvky jak z kvazikřehkých, tak pružno-plastických materiálů. Třebaže použití tohoto modelu vyžaduje v porovnání s obvyklými MKP metodami odlišný přístup, svojí výstižností a transparentností je schopen je v některých ohledech dokonce i předčit. Nicméně stále existují některé ne zcela probádané vlastnosti modelu, kterým by v budoucnu bylo dobré věnovat další pozornost.

¹CPU AMD Phenom II X4 a GPU NVIDIA GeForce GTX460

Zůstává například otázkou, do jaké míry je důležité v případě modelování kompozitního (nehomogenního) materiálu usilovat o co nejrovnoměrnější síť – rozložení generovaných bodů. Ukázalo se sice, že model je na hustotu a rovnoměrnost sítě v místech, kdy se velikost změny geometrie blíží zrnitosti modelu citlivý. Nicméně generátor rozšířený o schopnosti tato úskalí zohledňovat by mohl napomoci nejenom v řešení těchto kritických oblastí, ale i optimalizaci modelu jako celku. Možným přínosem by mohlo být rovněž i zavedení statistického sledování vytvářené sítě, aby uživatel již během vkládání bodů měl k dispozici přehled o její dosažené kvalitě.

S otázkou homogenity, ale zřejmě podstatně více souvisí samotné přiřazení materiálových funkcí jednotlivým pružinám. Ty jsou pouhým násobkem definované materiálové funkce v závislosti na pružinové geometrii, čímž je z hlediska materiálu na globální úrovni opět dosahována jeho maximální stejnorodost.² Eventuálním řešením by byla modifikace přiřazovaných parametrů pružinových sítí podle vhodných pravděpodobnostních rozložení. To by mohlo kromě lepšího vystižení nehomogenního materiálu také umožnit i zjednodušení sestupné větve tahové funkce pružin a tudíž i urychlení výpočtu.

V případě modelování ocelových prvků se zdá být největším úskalím globální nastavení pružinových funkcí. Zejména pak jejich parametrů v blízkosti a po dosažení meze kluzu. Možným nástrojem jak tuto oblast vystihnout, je použití *Ramsberg-Osgoodovy* rovnice [14], případně její modifikace [10] pro aproximaci pracovní funkce materiálu. Tyto metody by mohly zároveň napomoci k porozumění tlakovému chování a bylo by tak možné i patřičně nastavit tlakovou část pružinové funkce.

I přesto, že model mohl být vyzkoušen při simulování odlišných typů materiálu a způsobů zatěžování, rozsah použitých experimentů je stále poměrně úzký. Modelování jiných prvků z dalších materiálů tak stále představuje výzvu, jež může sloužit k odhalení zatím neprozkoumaných předností, a možná i úskalí, modelu FyDiK.

²To pak zřejmě souvisí i se zvýšenou elasticitou modelu.

LITERATURA

- [1] Aurenhammer, F.: Voronoi diagrams – a survey of a fundamental geometric data structure. *ACM COMPUTING SURVEYS*, ročník 23, č. 3, 1991: s. 345–405.
- [2] Bazant, Z. P.; Tabbara, M. R.; Kazemi, M. T.; aj.: Random particle model for fracture of aggregate or fiber composites. *Journal of engineering mechanics*, ročník 116, č. 8, 1990: s. 1686–1705.
- [3] Beni, L. H.: *Development of a 3D kinetic data structure adapted for a 3D spatial dynamic field simulation*. Dizertační práce, Université Laval, Québec, 2009.
- [4] Bolander, J.; Moriizumi, K.; Kunieda, M.; aj.: Rigid-Body-Spring Network modeling of cement-based composites. *Fracture Mechanics of Concrete Structures*, 2001: s. 773–780.
- [5] Eberly, D. H.: *Game physics*. CRC Press, 2010.
- [6] Echtler, F.: *Efficient Realization of Mass-Spring Systems on Graphics Hardware*. Diplomová práce, Technische Universität München, 2004.
- [7] Eliáš, J.: *Discrete simulation of fracture processes of disordered materials*. Dizertační práce, Vysoké učení technické v Brně. Fakulta stavební. Ústav stavební mechaniky., 2009.
- [8] Frantík, P.: *Fyzikální diskretizace a její aplikace na řešení nelineárních úloh*. Habilitační práce, Vysoké učení technické v Brně, 2016.
- [9] Halton, J. H.: On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, ročník 2, č. 1, dec 1960: s. 84–90, doi:10.1007/bf01386213.
- [10] Kamaya, M.: Ramberg–Osgood type stress–strain curve estimation using yield and ultimate strengths for failure assessments. *International Journal of Pressure Vessels and Piping*, ročník 137, jan 2016: s. 1–12, doi:10.1016/j.ijpvp.2015.04.001.
- [11] Kucharczyková, B.: Protokol - Betonáž směsí obyčejného betonu. Technická zpráva, Ústav stavebního zkušebnictví, Vysoké učení technické v Brně, Fakulta Stavební, 2008.
- [12] Ledoux, H.: Computing the 3D Voronoi Diagram Robustly: An Easy Explanation. In *4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD 2007)*, IEEE Computer Society.
- [13] Ledoux, H.: *Modelling Three-dimensional Fields in Geoscience with the Voronoi Diagram and its Dual*. Dizertační práce, University of Glamorgan, 2006.

-
- [14] Ramberg, W.; Osgood, W. R.: Description of stress-strain curves by three parameters. 1943.
- [15] Sobek, J.; Štafa, M.; Veselý, V.: Numerical simulations towards verification of acoustic emission events localization in splitting tests of cementitious composites. In *32th Spanish Conference on Fracture and Structural Integrity.*, Zamora, Spain, J. Toribio, B. Gonzáles, J-C. Matos, 2015, s. 160–165, iISSN: 0213- 3725.
- [16] Sobek, J.; Štafa, M.; Veselý, V.: Fracture events localization by numerical simulations of cementitious composites. In *14th International Conference on Fracture and Damage Mechanics, FDM 2015. Key Engineering Materials (print).*, Trans Tech Publications Ltd, 2016, iISBN: 978-3-03835-541- 0. ISSN: 1013-9826.
- [17] tensei @ Java-Gaming.org: Benchmarks for 3D Scene Graph APIs.
- [18] Teschner, M.; Heidelberger, B.; Muller, M.; aj.: A versatile and robust model for geometrically complex deformable solids. In *Computer Graphics International, 2004. Proceedings*, IEEE, 2004, s. 312–319.
- [19] The CGAL Project: *CGAL User and Reference Manual*. CGAL Editorial Board, 4.8.1 vydání, 2016.
- [20] Vodák, O.: *Analysis of Fracture of Quasi-Brittle Materials Using Numerical Modelling and Acoustic/Electromagnetic Emission*. Diplomová práce, Vysoké učení technické v Brně, 2013.
- [21] Wang, L.; Li, S.; Zhang, G.; aj.: A GPU-based parallel procedure for nonlinear analysis of complex structures using a coupled FEM/DEM approach. *Mathematical Problems in Engineering*, ročník 2013, 2013.
- [22] Washizawa, T.; Nakahara, Y.: Parallel Computing of Discrete Element Method on GPU. *arXiv preprint arXiv:1301.1714*, 2013.
- [23] Z, P. C.; Z, C. M.; Scopigno, R.: DeWall: A Fast Divide & Conquer Delaunay Triangulation Algorithm in E d. Consiglio Nazionale delle Ricerc he - ITALY, 1997.
- [24] Zhao, G.; Khalili, N.: Graphics processing unit based parallelization of the distinct lattice spring model. *Computers and Geotechnics*, ročník 42, 2012: s. 109–117.
- [25] Zhao, G.-F.: *High Performance Computing and the Discrete Element Model: Opportunity and Challenge*. Elsevier, 2015.
- [26] Zhao, G.-F.; Fang, J.; Zhao, J.: A 3D distinct lattice spring model for elasticity and dynamic failure. *International Journal for Numerical and Analytical Methods in Geomechanics*, ročník 35, č. 8, 2011: s. 859–885.



Michal Štafa

Oldřichovice 343
739 61 Třinec

Telefon: +420 777 837 479

E-mail: stafa@centrum.cz

Stav: svobodný

Narozen: 10.12.1983

Vzdělání

| | | |
|-------------|-------------|--|
| 2003 – 2008 | Škola: | Vysoké učení technické v Brně - Fakulta stavební |
| | Zaměření: | Pozemní stavby - Konstrukce a statika staveb |
| | Ukončení: | Titul Bc. |
| 2008 – 2010 | Škola: | Vysoké učení technické v Brně - Fakulta stavební |
| | Zaměření: | Pozemní stavby - Konstrukce a statika staveb |
| | Ukončení: | Titul Ing. |
| od 2010 | Škola: | Vysoké učení technické v Brně - Fakulta stavební |
| | Zaměření: | Konstrukce a dopravní stavby |
| | Typ studia: | Doktorské |

Praxe

| | |
|------------------------------------|---|
| od IV.2015 | Projekce Štafová Judita |
| | Inženýr – statik |
| VII.08 – IX.2013 (2 měsíce/rok) | Vysoké učení technické v Brně - Fakulta stavební |
| | Technický pracovník |
| | Programování numerických simulací lomových zkoušek. |

Pobyty

| | |
|-----------|--|
| XI. 2011 | Universiti Malaysia Sabah – Malajsie |
| (7 dní) | Prezentace software vyvíjeného na ústavu STM |
| | Garant: Janice Lynn Ayog |
| VIII.2010 | Bauhaus-Universität Weimar – Německo |
| (15 dní) | Letní škola se zaměřením na validaci modelů a simulace |
| | Garant: Lars Abrahamczyk |

Projekty a ocenění

| | |
|-----------------|--|
| I.11 – XII.2011 | Projekt Fondu rozvoje vysokých škol G1 2817/2011 |
| | Nástroje pro inovaci výukových metod k řešení staticky neurčitých konstrukcí silovou metodou |
| VII.2008 | Uznání děkana Fakulty stavební za vzorně vypracovanou bakalářskou práci a její obhajobu |

Znalosti a dovednosti

| | |
|---------------|---|
| Cizí jazyky: | Angličtina (Pokročilý – C1), Němčina (Středně pokročilý – B1) |
| Programování: | JAVA SE (Pokročilý), C++/CUDA (Základy) |

ABSTRAKT

Předkládaná práce poukazuje na pozoruhodný potenciál fyzikálně diskretizačního lattice modelu FyDiK pro prostorové modelování nelineárních problémů stavební mechaniky. K dosažení cílů práce byla naprogramována aplikace, implementující model FyDiK společně s 3D grafickým uživatelským prostředím zajišťující sestavení modelu pružinových sítí. Ta byla následně využita k modelování vzniku trhlin a lomu v betonových vzorcích a k modelování plastizace stojiny ocelového I profilu, přičemž výpočty byly realizovány prostřednictvím masivní paralelizace na platformě CUDA.

V úvodní části jsou nejprve představeny základní principy, z nichž práce vychází. Následně jsou podrobněji rozebrány jednotlivé prvky modelu a přiblížena problematika paralelizace prostřednictvím grafických karet. Vytvoření potřebného software a ladění modelu na uvedených materiálech se věnuje další část, která je následována vyhodnocením dosažených výsledků ve srovnání s jinými programy pro modelování. Závěrem jsou shrnuty dosažené výsledky a přestřeny další možnosti vývoje představeného způsobu modelování.