



**FAKULTA  
INFORMAČNÍCH  
TECHNOLOGIÍ  
ČVUT V PRAZE**

## Zadání bakalářské práce

**Název:** Simulování vodního povrchu  
**Student:** Hong Son Ngo  
**Vedoucí:** Ing. Petr Pauš, Ph.D.  
**Studijní program:** Informatika  
**Obor / specializace:** Webové a softwarové inženýrství, zaměření Počítačová grafika  
**Katedra:** Katedra softwarového inženýrství  
**Platnost zadání:** do konce letního semestru 2022/2023

### Pokyny pro vypracování

Voda a vodní hladina se vyskytuje ve spoustě grafických aplikací a hrách a jejich realistická real-time simulace je žádoucí. Cílem této práce je vytvořit simulátor vodního povrchu a jeho případných světelných efektů na tělesa pod vodou.

1. Analyzujte možnosti real-time simulace vodní hladiny.
2. Analyzujte vhodné nástroje pro její simulaci (např. OpenGL, Unity, atd.).
3. Na základě analýzy vyberte vhodný nástroj pro simulaci a navrhňte prototyp simulace.
4. Ve zvoleném nástroji implementujte.
5. Vytvořte testovací scénu, která simulaci demonstruje.

---

*Elektronicky schválil/a Ing. Radek Richtř, Ph.D. dne 16. prosince 2021 v Praze.*





**FAKULTA  
INFORMAČNÍCH  
TECHNologiÍ  
ČVUT V PRAZE**

Bakalářská práce

## **Simulování vodního povrchu**

*Hong Son Ngo*

Katedra softwarového inženýrství  
Vedoucí práce: Ing. Petr Pauš, Ph.D.

19. března 2022



---

## Poděkování

Doplňte, máte-li komu a za co děkovat. V opačném případě úplně odstraňte tento příkaz.



---

# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principu při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu) licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 19. března 2022

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2022 Hong Son Ngo. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Ngo, Hong Son. *Simulování vodního povrchu*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2022.



---

## Abstrakt

V několika větách shrňte obsah a přínos této práce v češtině. Po přečtení abstraktu by se čtenář měl mít čtenář dost informací pro rozhodnutí, zda chce Vaši práci číst.

**Klíčová slova** Nahradte seznamem klíčových slov v češtině oddělených čárkou.

---

## Abstract

Sem doplňte ekvivalent abstraktu Vaší práce v angličtině.

**Keywords** Nahradte seznamem klíčových slov v angličtině oddělených čárkou.



---

# Obsah

<b>Úvod</b>	<b>1</b>
<b>1 Cíl práce</b>	<b>3</b>
<b>2 Základ teorie mechaniky tekutin</b>	<b>5</b>
2.1 Navierovy–Stokesovy rovnice . . . . .	5
2.1.1 Vysvětlení . . . . .	5
2.2 Využití . . . . .	7
2.3 Popisy tekutiny . . . . .	7
2.3.1 Lagrangeův popis . . . . .	7
2.3.2 Eulerův popis . . . . .	7
<b>3 Vlastnosti vodního povrchu</b>	<b>9</b>
3.1 Dynamické vlastnosti . . . . .	9
3.1.1 Procedurální metody . . . . .	10
3.1.1.1 Simulace podle sinusoid . . . . .	11
3.1.1.2 Simulace podle Gerstnerovy vlny . . . . .	12
3.1.1.3 Simulace podle Fourierovy transformace . . . . .	14
3.1.2 Částicové metody . . . . .	15
3.1.2.1 Simulace podle Smoothed Particle Hydrodynamics . . . . .	16
<b>Závěr</b>	<b>21</b>
<b>Bibliografie</b>	<b>23</b>
<b>A Seznam použitých zkratek</b>	<b>27</b>
<b>B Obsah přiloženého CD</b>	<b>29</b>



---

## Seznam obrázků

3.1	Ukázka Guendelmanovy off-line simulace Eulerovy tekutiny [3]	10
3.2	Simulace vlnění pomocí jedné sinusoidy	11
3.3	Simulace vlnění pomocí sčítání sinusoid	11
3.4	Porovnání mezi vlnami funkcí 3.1 a 3.3	12
3.5	Gerstnerova vlna s fázovou rychlostí $c$ , vlnovou délkou $\lambda$ a rozdílem hřebenu a údolí $H$ [8]	12
3.6	Porovnání hodnot $s$ ve dvourozměrném prostoru	13
3.7	Gerstnerovy vlny v simulaci vodního povrchu podle tutoriálu od Jaspera Flicka [10]	14
3.8	Ukázka simulace podle FFT od Asylum Darth [15]	15
3.9	Ilustrace vážení <i>smoothing kernel</i> [21]	17
3.10	Demonstrace rozdílů v hustotě tekutiny [23]	18
3.11	Interpolace pole hustoty podle SPH [23]	20
3.12	Müllerova simulace vody založená na SPH [20]	20



---

# Úvod

Kvalita herních titulů za poslední let ohromně vzrostla. Jejich úspěch lze připsat nejen zajímavému příběhovému obsahu, ale také i jejich vizuálnímu zpracování. Právě v grafickém provedení můžeme vidět největší pokrok. Díky hardwarovým zlepšením v grafických kartách se obraz her čím dál více blíží fotorealistickým výsledkům a s příchodem zařízení pro virtuální a rozšířenou realitu je o realistický obraz ještě větší zájem.

Co rozlišuje scény napříč historií her, jsou speciální efekty. Na základě kvality jejich provedení je hráč hlouběji vnořen do virtuálního světa a následně i do jeho příběhu. Jedním z takových efektů jsou např. přírodní jevy jako pohyb plamene, vlnění hladiny vody, proudění větru...

Fyzikálně korektní chování přírodních jevů však zůstává do dnešního dne mezi komplexnějšími problémy. Mezi těmi výpočetně nejobtížnějšími je považováno proudění tekutin, do kterých patří jak plynné, tak i kapalná skupenství. Pro vizualizaci se většina metod zabývá zejména kapalinami kvůli její povaze (jsou lidským okem viditelné), ale v některých případech lze postupy pro simulování proudění kapalin využít i pro zobrazení plynů jako např. kouře nebo plamene.

Kvůli vysoce dynamickému chování tekutin se však v současnosti nejrealističtějších výsledků dosáhne hlavně off-line metodami. Real-time aplikace využívají stejných principů, ale s podstatnými kompromisy jak paměťovými, tak i výpočetními. Vzhledem k tomu, že některé aplikace nevyužijí přesné a hlavně výpočetně drahé simulace, tak napodobují jen výsledné efekty, které se nijak neopírají o fyzikální zákony. Ačkoli off-line metody zobrazují věruhodně chování kapalin, resp. plynů, nejsou nijak zastoupeny v hrách, neboť je interakce s uživatelem a dynamické prostředí scén nedovolí využít.

Obsah této bakalářské práce se zabývá real-time simulací vodního povrchu a jeho případnými světelnými efekty jak na hladině, tak i na tělese pod ní, jejíž aplikace by bylo možné využít k fotorealistickému zobrazení virtuálních scén v hrách.





---

## Cíl práce

Cílem teoretické části této bakalářské práce je prozkoumat možnosti real-time simulace vodního povrchu a případných vlastností, které interagují se světelnými paprskami osvětelující vodní plochu. Na základě analýzy současných metod prozkoumat vhodné nástroje pro její zobrazení v grafických aplikacích jako jsou počítačové hry.

Následně v praktické části je cílem vybrat vhodné simulační metody pro real-time grafické aplikace, navrhnout prototyp simulátoru a ve zvoleném nástroji ho implementovat a nakonec vytvořit testovací scénu, která simulaci demonstruje.



# Základ teorie mechaniky tekutin

V této kapitole je shrnutí teorie mechaniky tekutin, která je nezbytnou součástí pro simulování fyzikálně korektního chování vody. Teorie je zde vyložena takovým způsobem, aby byl čtenář s ní seznámem a co nejrychleji pochopil principy chování tekutin, neobsahuje žádné rigorózní vysvětlení problematiky.

## 2.1 Navierovy–Stokesovy rovnice

Proud tekutin se v reálném světě řídí podle Navierových–Stokesových rovnic (NSE), soustavou nelineárních diferenciálních rovnic:

$$\nabla \cdot \mathbf{v} = 0 \quad (2.1)$$

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} + \frac{1}{\rho} \nabla p = \mathbf{g} + \nu \nabla^2 \mathbf{v}, \quad (2.2)$$

kde vektor  $\mathbf{v} = (u, v, w)$  označuje rychlost tekutiny,  $\rho$  označuje hustotu tekutiny,  $p$  tlak, kterým působí tekutina na své okolí,  $\mathbf{g} = (x, y, z)$  je gravitační zrychlení,  $\nu$  je značení kynematické viskozity tekutiny. Symboly  $\nabla$ ,  $\nabla \cdot$ ,  $\nabla^2$  označují diferenciální operátory nabra, divergence a Laplacův operátor.

### 2.1.1 Vysvětlení

Před vysvětlení jednotlivých rovnic je dobré zmínit, že většina teorie mechaniky tekutin je založena na odvětví matematiky vektorové analýzy a pracují s vektorovými poli, resp. skalárními poli. NSE např. pracují s vektorovým poli, resp. se skalárními poli, které jednotlivým bodům prostoru přiřazují vektor určující rychlost proudu tekutiny, resp. hodnotu tlaku.

Na první pohled vypadají rovnice těžce uchopitelné, ale myšlenka za nimi je velmi jednoduchá. První rovnice 2.1 popisuje zákon o zachování hmotnosti, tím je myšleno, že není možné, aby hmota tekutiny na některém místě z ničeho vznikla nebo zanikla.

Druhá rovnice 2.2 popisuje zákon o zachování hybnosti a je ve své podstatě Newtonův druhý zákon.

$$\mathbf{f} = m\mathbf{a} \quad (2.3)$$

Zrychlení  $\mathbf{a}$  lze přepsat jako derivace rychlosti podle času.

$$\mathbf{f} = m \frac{D\mathbf{v}}{Dt}, \quad (2.4)$$

kde  $\frac{D\mathbf{v}}{Dt}$  značí tzv. materiálovou derivací. Pro odvození celé rovnice je třeba rozvést sílu  $\mathbf{f}$ , které na tekutinu působí [1]. Jedna z nich je samozřejmě gravitace, jejíž hodnota je vyčíslena jako  $m\mathbf{g}$ .

Další síly vytváří tekutina sama na sobě. První z nich je síla způsobena rozdílem tlaků v tekutině. Tekutina se v oblastech s vyšším tlakem přesouvá do oblastí s nižším tlakem. Její hodnotu můžeme zapsat jako  $-V\nabla p$ <sup>1</sup>.

Další síla působící na tekutinu je ovlivněna její viskozitou. Viskozita působí při každém pohybu částic tekutiny a snaží se vyrovnat rychlost částice rychlostí svých sousedních částic. Její hodnotu můžeme vyjádřit jako  $V\mu\nabla^2\mathbf{v}$ <sup>2</sup>, kde  $\mu$  označuje koeficient dynamické viskozity.

$$m\mathbf{g} - V\nabla p + V\mu\nabla^2\mathbf{v} = m \frac{D\mathbf{v}}{Dt} \quad (2.5)$$

Rovnice 2.3, 2.4 a 2.5 předpokládají, že tekutinu lze rozložit na konečně mnoho malých částí, tímto způsobem je do vyčíslení představena výpočetní chyba. Řešením toho problému je celou rovnici 2.5 vydělit  $V$ , aby zachytila pohyb nekonečně mnoho nekonečně malých částic tekutin. Dobré je připomenout, že hustota  $\rho$  lze vyjádřit jako  $\frac{m}{V}$ .

$$\rho\mathbf{g} - \nabla p + \mu\nabla^2\mathbf{v} = \rho \frac{D\mathbf{v}}{Dt} \quad (2.6)$$

Následně po vydělení hustoty  $\rho$ , vyjádření derivace  $\frac{D\mathbf{v}}{Dt}$  podle pravidla pro složené funkce a prohození sčítanců získáme druhou Navierovu–Stokesovu rovnici. Kynematická viskozita  $\nu$  je rovna  $\frac{\mu}{\rho}$ .

$$\frac{\partial\mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla\mathbf{v} + \frac{1}{\rho}\nabla p = \mathbf{g} + \nu\nabla^2\mathbf{v} \quad (2.7)$$

---

<sup>1</sup>Operace  $\nabla p$  vyjadřuje vektorové pole, ve kterém vektory směřují, z jednotlivých bodů na své sousední body, aby jeho hodnota ve skalárním poli nejrychleji vzrostla.

<sup>2</sup>Operace  $\nabla^2\mathbf{v}$  vyjadřuje míru odchýlení rychlosti částice okolo svého okolí.

## 2.2 Využití

Výhodou NSE je, že je lze aplikovat na téměř jakékoliv tekutiny. V praxi se využívají pro modelování počasí, podnebí pro předpověď počasí, proudění v oceánech, výpočtu aerodynamických vlastností vozidel. Přestože využití rovnic je nespočetné, mají zásadní problém, neboť do dnes nevíme, zdali mají pro náhodný vstup nějaké řešení. Pro zjednodušení výpočtu existují několik aproximací, které je umožňují aplikovat se zanedbatelnými chybami v modelovacích systémech.

## 2.3 Popisy tekutiny

V teorii mechaniky tekutin existují dva různé pohledy popisu tekutin. Na základě těchto popisů jsou následně založeny algoritmy pro simulaci kapalin, resp. plynů, které jich využívají pro diskretizaci NSE.

### 2.3.1 Lagrangeův popis

Lagrangeův popis je jeden z popisů kontinua, který si většina nejspíše vybaví. Na tekutinu nahlíží jako na systém částic. Na jednotlivé body tekutiny nahlíží jako částice, které mohou být na základě potřeby různě velké, např. jako molekule nebo části tekutiny o nějakém objemu. Ke každé z nich Lagrangeův popis přiřazuje její pozici  $\mathbf{x}$  a rychlost  $\mathbf{v}$  a sleduje je, jak se tyto hodnoty v čase mění.

### 2.3.2 Eulerův popis

Eulerův popis kontinua je na první pohled neintuitivní. Eulerův postup diskretizuje prostor tekutiny na pevně dané oblasti, ve kterých měří vlastnosti tekutiny, např. tlak, rychlost proudu, teplotu, jak se v čase mění.

Přestože tento postup vypadá omezující a složitý kvůli sledování veličin tekutin jen v pevně daných bodech, je v metodách pro simulování tekutin preferovaným popisem kontinua. Hlavní výhodou Eulerova pohledu je jednoduchost výpočtu prostorových derivací jako  $\nabla p$  nebo  $\nabla \mathbf{v}$ , které se lépe aproximují v pevné Eulerově mřížce oproti shlukům pohybujících se částic [1].



## Vlastnosti vodního povrchu

Tato kapitola obsahuje popis vizuálních vlastností vodní hladiny, které v reálném světě lze pozorovat, a k jejím hlavním rysům jsou vypsány algoritmy, které je simulují. Z pohledu počítačové grafiky lze vlastnosti vody rozdělit do dvou kategorií:

**dynamické vlastnosti** popisující pohyb vodní hladiny,

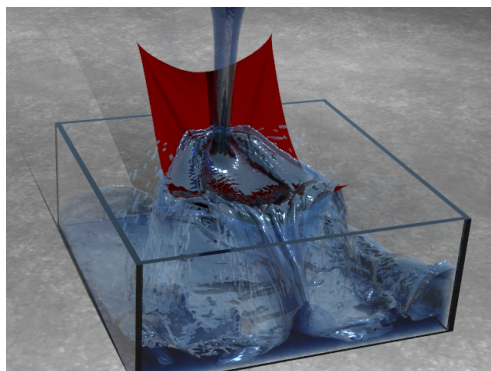
**světelné vlastnosti** popisující interakce povrchu vody se světelnými prsky.

### 3.1 Dynamické vlastnosti

Dynamické vlastnosti zachycují, jak se vodní hladina pohybuje a jak reaguje na dynamické prostředí. V současnosti nejrealističtější výsledků je dosaženo simulací Eulerovy vody (nahlíží na vodu Eulerovým popisem), jejíž prostor je rozdělen do alespoň  $512^3$  buněk. Takto řídké rozdělení obsahuje přes 100 milion neznámých pro vyřešení a s užitím globálních zobrazovacích metod jako ray-tracing pro realistickou vizualci odrazů, refrakcí a kaustik je výpočetně nemožné simulaci provést v reálném čase [2].

Proto real-time simulace vody vhodné pro aplikace jako hry musí nutně splňovat tyto podmínky [2]:

- být výpočetně rychlé – zlomek 15 ms, který je třeba pro vykreslení jednoho snímku,
- být paměťově nenáročný,
- být stabilní – korektně reagovat i na nefyzikálně pohybující se objekty.



Obrázek 3.1: Ukázka Guendelmanovy off-line simulace Eulerovy tekutiny [3]

Jeden přístup, jak se nejvíce přiblížit off-line simulaci a stále dodržet podmínky pro real-time simulaci ve hrách, je zachovat stejný algoritmus jako při off-line simulaci, ale zmenšit rozlišení prostoru simulace. Tento postup ale spíše ubírá na realitě vodního povrchu, neboť se voda nefyzikálně shlukuje a detaily z obrazu jsou vynechány.

Další z možností, jak splnit výše zmíněné podmínky, je omezit míru interakce s vnějším prostředím, která tvoří největší výpočetní překážku. Na základě významnosti vodních útvarů ve scéně může simulace reagovat na všechny rigidní tělesa nebo u případů, ve kterých voda slouží jako pozadí, opominout jakoukoliv interakci. Podle tohoto principu můžeme dělit metody simulace vodní plochy na [2]:

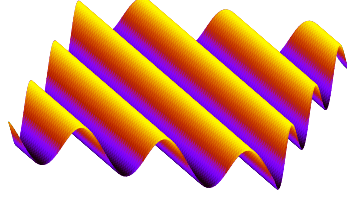
- procedulární
- částicové
- hybridní

#### 3.1.1 Procedulární metody

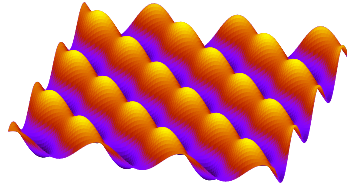
Procedulární metody simulují konečné efekty jako vlnění vodního povrchu, které nejsou vyvolány fyzikální činitely. Největší výhodou této metody je výpočetní rychlost a versetilita, na druhou stranu ale vodní útvary nereagují na dynamické prostředí.

Na základě těchto vlastností se procedulární metody používají pro vizualizaci rozsáhlých vodních ploch, které ve scéně hrají malou roli, např. jako oceán v pozadí scény.





Obrázek 3.2: Simulace vlnění pomocí jedné sinusoidy



Obrázek 3.3: Simulace vlnění pomocí sčítání sinusoid

#### 3.1.1.1 Simulace podle sinusoid

Mezi prvními, kteří zkoumali procedurální metody pro zobrazení vodního povrchu, byl Nelson L. Max ve své práci „Vectorized Procedural Models for Natural Terrain: Wave and Islands in the Sunset,“ ve které modeluje vlnění povrchu pomocí sinusoid. Na základě horizontální pozice a času manipuloval výšku vrcholů roviny [4]. Výšku jednotlivých vrcholů roviny vyjádřil jako funkci

$$h(x, z, t) = -y_0 + A \sin(k_x x + k_z z - \omega t + \varphi), \quad (3.1)$$

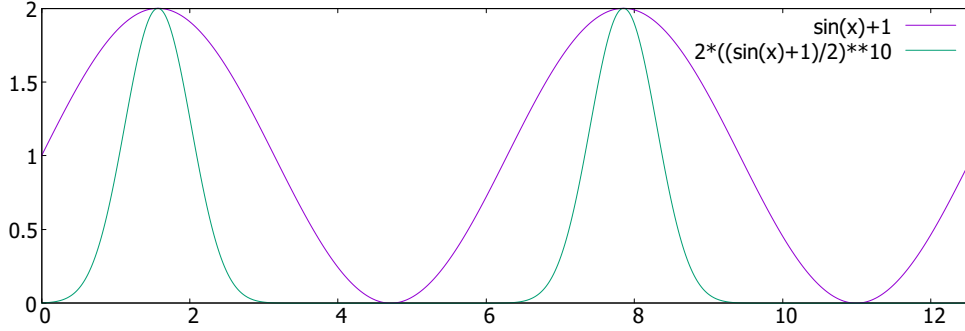
kde  $(x, z)$  je horizontální pozice vrcholu roviny (v celé práci se bude považovat, že kladná osa  $y$  směřuje nahoru),  $t$  je čas,  $y_0$  je výška hladiny vody v klidovém stavu (při nulovém výskytu vln),  $A$  reprezentuje amplitudu roviny,  $\mathbf{k} = (k_x, k_z)$  je vlnový vektor reprezentující směr a rychlost propagace vln,  $\omega$  je úhlová frekvence a  $\varphi$  je fáze vlny.

Tento model se ale vlní jen v jednom směru a výsledné vlny vypadají nerealisticky hladce. K dosažení větších detailů lze k funkci přičíst další sinusoidy s odlišnými parametry amplitudy, vlnového vektoru nebo úhlové frekvence:

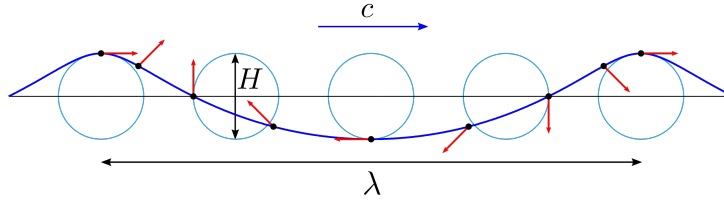
$$h(x, z, t) = -y_0 + \sum_{i=1}^{N_w} A_i \sin(k_{x_i} x + k_{z_i} z - \omega_i t + \varphi_i), \quad (3.2)$$

kde  $N_w$  je celkový počet vln.

### 3. VLASTNOSTI VODNÍHO POVRCHU



Obrázek 3.4: Porovnání mezi vlnami funkcí 3.1 a 3.3



Obrázek 3.5: Gerstnerova vlna s fázovou rychlostí  $c$ , vlnovou délkou  $\lambda$  a rozdílem hřebenu a údolí  $H$  [8]

Realistického řešení však za pomoci jen obyčejných sinusoid nelze dosáhnout. Max si všiml, že vlnění oceánu s vyššími amplitudami mají užší hřeben a mělké údolí, zatímco vrchol a údolí sinusoidy je stejně oblý. Mark Finch a Cyan Worlds přišly s řešením toho problému, které stále využívá jednoduchých sinusoid. Kromě obyčejných funkcí sinusoid přičítá navíc funkci

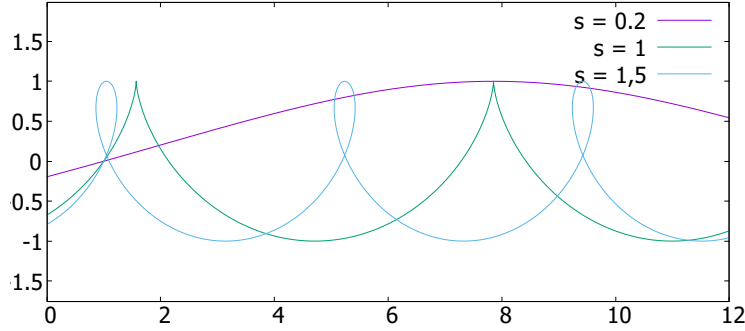
$$f_i(x, z, t) = -y_0 + 2 \left( \frac{\sin(k_{x_i}x + k_{z_i}z - \omega_i t + \varphi_i) + 1}{2} \right)^k, \quad (3.3)$$

kde  $k \in \mathbb{R}^+$  určuje míru, jak má být hřeben úzký [5].

#### 3.1.1.2 Simulace podle Gerstnerovy vlny

Limitující faktor metody založené na transformaci vrcholů pomocí sinusoid je, že manipuluje s jediným parametrem vrcholu, tj. výškou, a pro realističtější výsledky bylo třeba manipulovat vrcholy i v horizontální rovině. Tento problém řeší cyklické křivky, tzv. trochoidy<sup>3</sup>, podle kterých Franz Josef Gerstner, německý fyzik, modeloval vlnění hladiny v hlubokých vodách [7].

<sup>3</sup>Trochoidou nazýváme trajektorii bodu pevně spojeného s „kotálející se“ kružnice po nehybné přímce [6].

Obrázek 3.6: Porovnání hodnot  $s$  ve dvourozměrném prostoru

Dnes v teorii dynamiky tekutin vlnění založené na trochoidních křivkách nazýváme Gerstnerovými vlnami. Mezi prvními, kteří se Gerstnerovými vlnami zabývali, byli Alain Fournier a William T. Reeves, kteří transformovali vrcholy roviny podle těchto parametrických rovnic [9]:

$$x = x_0 + r \cos(k_x x_0 + k_z z_0 - \omega t) \quad (3.4)$$

$$y = r \sin(k_x x_0 + k_z z_0 - \omega t) \quad (3.5)$$

$$z = z_0 + r \cos(k_x x_0 + k_z z_0 - \omega t), \quad (3.6)$$

kde vektor  $(x_0, z_0)$  reprezentuje klidovou pozici vrcholu roviny,  $r$  je délka opisujícího bodu od středu kružnice trochoidní křivky, vektor  $\mathbf{k} = (k_x, k_z)$  určuje rychlost a směr propagace vlnění,  $\omega$  označuje úhlovou frekvenci a  $t$  je čas.

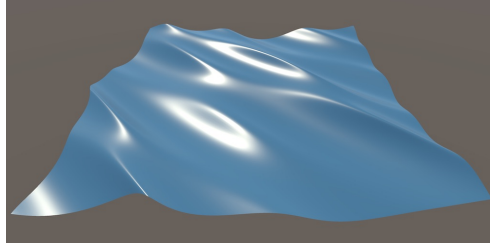
Hodnota  $s = r|\mathbf{k}|$  určuje, jak strmá bude vlna. Pro hodnotu  $s = 0.2$  má vlna tvar jako sinusoida, pro  $s = 1$  má tvar cykloidy a pro  $s > 1$  dochází samoprotínání trochoidy, a proto by se hodnotám vyšší než jedna pro vizualizaci vodního povrchu mělo vyhýbat. V obrázku 3.6 je porovnání hodnot  $s$  ve dvourozměrném prostoru.

Detailnějších výsledků lze získat podobně jako u sinusoid pomocí skládání Gerstnerových vln:

$$x = x_0 + \sum_{i=1}^{N_w} r_i \cos(k_{x_i} x_0 + k_{z_i} z_0 - \omega_i t) \quad (3.7)$$

$$y = \sum_{i=1}^{N_w} r_i \sin(k_{x_i} x_0 + k_{z_i} z_0 - \omega_i t) \quad (3.8)$$

$$z = z_0 + \sum_{i=1}^{N_w} r_i \cos(k_{x_i} x_0 + k_{z_i} z_0 - \omega_i t). \quad (3.9)$$



Obrázek 3.7: Gerstnerovy vlny v simulaci vodního povrchu podle tutoriálu od Jaspera Flicka [10]

### 3.1.1.3 Simulace podle Fourierovy transformace

Předešlé dvě metody jsou kvalitními nástroji pro simulaci vodního povrchu a za jejich pomoci lze i získat fotorealistických výsledků pro vysoký počet vln  $N_w$  v řádech tisíců nebo více. Vysoký počet představuje však výpočetní překážku. Obvykle by výpočet posunutí podle rovnic 3.2 nebo 3.7–3.9 herní engine provedly ve *vertex shaderech*, pro které je však výpočet funkcí sinus a cosinus náročné [11].

Johanson řešil výpočetní náročnost adaptivní metodou, která na základě vzdálenosti kamery a vodní plochy vyřazuje vlny [12]. V případě, že je kamera dostatečně vzdálená od vodní plochy, vyřazuje Johanson vlnění s vysokými frekvencí. Efektivnějšího výsledku dosáhl Lee [12], který před vykreslení vodní plochy navíc zahazuje vrcholy geometrie hladiny, které jsou mimo rozsah kamery.

V současnosti nejlepším řešením tohoto problému je užití rychlé Fourierovy transformace (FFT), resp. inverzní Fourierovy transformace (IFFT) podle Jerryho Tessendorfa [13]. Sčítání sinusoid v rovnici 3.2 je ve podstatě inverzní Fourierova transformace, kde jednotlivé sinusoidy přispívají ke konečnému vlnění. Tessendorf podle stastických dat vlnění oceánů, např. ze satalitních snímků, modeluje výšku vrcholů hladiny jako inverzní Fourierovu transformaci funkce

$$h(\mathbf{x}, t) = \sum_k \tilde{h}(\mathbf{k}, t) e^{i\mathbf{k} \cdot \mathbf{x}}, \quad (3.10)$$

kde  $\mathbf{x}$  je pozice vrcholů roviny, funkce  $\tilde{h}$  závislá na čase  $t$  obsahuje informaci o amplitudě a fázi sinusoidy  $e^{i\mathbf{k} \cdot \mathbf{x}}$  s vlnovým vektorem  $\mathbf{k}$ . Tessendorf následně podle oceánografických dat vhodně volí funkci  $\tilde{h}$ , aby bylo výsledné vlnění nejvíce realistické.

Časová komplexita triviálního sčítání sinusoid v rovnici 3.2, tj. inverzní Fourierovy transformace, je  $\mathcal{O}(n^2)$ , kdežto užitím algoritmů pro IFFT se redukuje časová komplexita na  $\mathcal{O}(n \log n)$  [14].



Obrázek 3.8: Ukázka simulace podle FFT od Asylum DARTH [15]

### 3.1.2 Částicové metody

Procedurální metody jsou stavěny pro vizualizaci rozsáhlých vodních ploch jako oceány a moře. Protože jsou tyto vodní útvary rozlehlé, jakýkoliv zásah buď hráče, či prostředí by měl na výsledné zobrazení hladiny minimální efekt.

V některých případech se ale hladina vody nechová periodicky nebo reakce na dynamické prostředí je právě při zobrazování žádoucí jako u vody fontány nebo kaluží. Tyto detailní vlastnosti vody řeší částicové modely využívající Lagrangeova popisu kontinua. Částicové metody simulují vodní útvary jako systém částic, které reprezentují určitou hmotu kapaliny.

Poprvé, kdo využil částicových systémů pro simulaci tekutiny, byl v roce 1983 Reeves [16]. Reevesův přístup byl však velmi primitivní oproti dnešním variacím, pohyb částic nekorektně modeloval nezávislé na ostatních. V realitě jeho simulace se více blížila k simulaci shlukům jemných částic, např. prachu.

Pro korektní chování je třeba, aby se jednotlivé částice mezi sebou přitahovaly a odpuzovaly. Obecně lze tuto sílu mezi dvěma částicemi vyčíslit jako

$$f(\mathbf{x}_i, \mathbf{x}_j) = F(|\mathbf{x}_i - \mathbf{x}_j|) \cdot \frac{\mathbf{x}_i - \mathbf{x}_j}{|\mathbf{x}_i - \mathbf{x}_j|}, \quad (3.11)$$

kde  $\mathbf{x}_i$ , resp.  $\mathbf{x}_j$  je pozice částice  $i$ , resp. částice  $j$ , funkce  $F$  je velikost síly [1].

Pro simulaci viskózní tekutiny Miller a Pearce modelovali tuto sílu, jako by byly jednotlivé částice spojeny pružinou [17]. Fyzikálně korektnějšího chování lze získat při volbě funkce  $F$  jako hodnotu Lennardovy–Jonesovy síly, která se používá v simulacích molekulární dynamiky,

$$f(\mathbf{x}_i, \mathbf{x}_j) = \left( \frac{k_1}{|\mathbf{x}_i - \mathbf{x}_j|^m} - \frac{k_2}{|\mathbf{x}_i - \mathbf{x}_j|^n} \right) \cdot \frac{\mathbf{x}_i - \mathbf{x}_j}{|\mathbf{x}_i - \mathbf{x}_j|}, \quad (3.12)$$

kde  $k_1, k_2, m, n$  jsou ovládací parametry. Obvyklou volbou je  $k_1 = k_2 = k$ ,  $m = 4$  a  $n = 2$  [1].

Pro real-time simulace může být tento přístup pomalý. V případě, že vodní útvar obsahuje  $n$  částic, musí se funkce  $f$  evaluovat  $\mathcal{O}(n^2)$ . Existují ale optimalizace, které rozdělují částice do pravidelné mřížky, čímž redukuje složitost na  $\mathcal{O}(n)$ . Přestože ji lze vypočítat poměrně rychle, simulace se vůbec neopírá o NSE.

### 3.1.2.1 Simulace podle Smoothed Particle Hydrodynamics

Aktuálně nejrozšířenější částicová metoda založená na výpočtu sil podle NSE vychází z Monaghanova článku „Smoothed Particle Hydrodynamics“ [18]. Monaghan vychází interpolační metody Smoothed Particle Hydrodynamics (SPH), kterou z prva Lucy vyvinul pro astrofyzické problémy [19]. Lucyho metoda je dostatečně obecná, aby ji bylo možné využít v simulaci tekutin pro výpočet skalárních polí hustoty nebo tlaku.

SPH lze v počítačové grafice přirovnat ke konvoluci, kde místo barvy pixelů pracuje s vlastnostmi částic. SPH distribuje obecnou veličinu pro danou částici jako sumu vážených hodnot veličin jejích sousedů. Hodnota obecné veličiny  $A$  pro částici  $i$  je podle SPH [20]

$$A_i(\mathbf{x}_i) = \sum_j m_j \frac{A_j}{\rho_j} W(|\mathbf{x}_i - \mathbf{x}_j|, h), \quad (3.13)$$

kde  $\mathbf{x}_i$ , resp.  $\mathbf{x}_j$ , je pozice částice  $i$ , resp.  $j$ ,  $j$  iteruje přes všechny částice,  $m_j$  je hmotnost částice,  $A_j$  je hodnota veličiny  $A$  a  $\rho_j$  je hustota pro částici  $j$ . Funkce  $W(|\mathbf{x}_i - \mathbf{x}_j|, h)$  se nazývá *smoothing kernel* s poloměrem  $h$ , což je analogií jádra u konvoluce. Obrázek 3.9 ilustruje, jak *smoothing kernel*, které mají obvykle tvar podobné Gaussovu rozdělení, váží hodnoty částic. Největší váhu  $W$  přiřadí vrchlům nejbližší středu, tj. částici  $\mathbf{x}_i$ , a nejnižší částicím vzdálené od středu délkou  $h$ . Validní *smoothing kernels* musí navíc splnit normalizační podmínka [1, 20]:

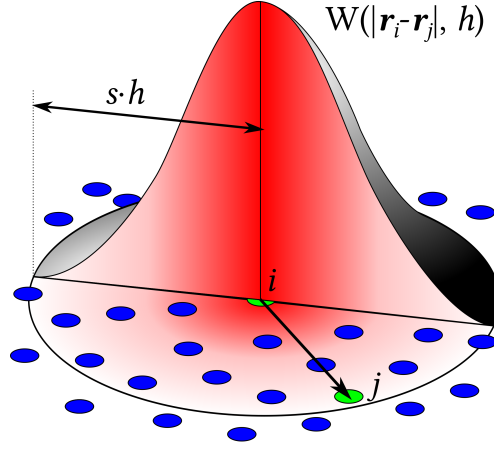
$$\int W(r) dr = 1 \quad (3.14)$$

NSE mimo samotných hodnot veličin dále pracuje s gradientem (operace  $\nabla$ ), jehož aplikace na rovnici 3.13 ovlivní jen *smoothing kernel*

$$A_i(\mathbf{x}_i) = \sum_j m_j \frac{A_j}{\rho_j} \nabla W(|\mathbf{x}_i - \mathbf{x}_j|, h), \quad (3.15)$$

výpočet Laplace (operace  $\nabla^2$ ) obdobně jako u gradientu působí jen na jádro  $W$

$$A_i(\mathbf{x}_i) = \sum_j m_j \frac{A_j}{\rho_j} \nabla^2 W(|\mathbf{x}_i - \mathbf{x}_j|, h). \quad (3.16)$$

Obrázek 3.9: Ilustrace vážení *smoothing kernelem* [21]

Obvyklá volba jádra  $W$ , které lze využít až na dvě výjimky téměř pro každou interpolaci veličin v simulaci tekutin, je *kernel poly6* [1, 20]

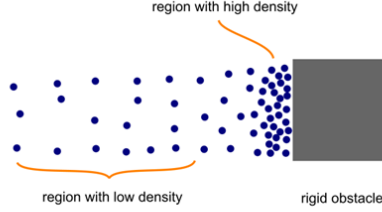
$$W_{poly6}(r, h) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - r^2)^3 & 0 \leq r \leq h \\ 0 & \text{jindy} \end{cases}. \quad (3.17)$$

*Poly6* ale nekoretně aproximuje síly tlakového pole, v simulaci se částice pod vysokým tlakem nefyzikálně shlukují [2], neboť  $\nabla W$ , jež je nutné pro výpočet síly způsobené rozdílem tlaku (rovnice 2.2), se ve středu jádra blíží k nule, což vede k zanedbání odpuzujících sil. Desburn řešil tento problém nahrazením jádra *poly6* za *kernel* [22]

$$W_{spiky}(r, h) = \frac{15}{\pi h^6} \begin{cases} (h - r)^3 & 0 \leq r \leq h \\ 0 & \text{jindy} \end{cases}. \quad (3.18)$$

Poslední nepřesnost jádra *poly6* se vyskytuje při výpočtu silové pole vytvořené viskozitou tekutiny. Výsledek jeho Laplacovy operace, od kterého se odvíjí síla vyvolaná viskozitou (rovnice 2.2), se u středu jádra pohybuje v záporných číslech, které může vést k nekorektnímu zvýšení rychlosti částice a následné nestabilitě simulace [20]. Na základě tohoto nedostatku Müller používá při výpočtu silového pole vyvolané viskozitou jádro

$$W_{viskozita}(r, h) = \frac{15}{2\pi h^3} \begin{cases} -\frac{r^3}{2h^3} + \frac{r^2}{h^2} + \frac{h}{2r} - 1 & 0 \leq r \leq h \\ 0 & \text{jindy} \end{cases}. \quad (3.19)$$



Obrázek 3.10: Demonstrace rozdílů v hustotě tekutiny [23]

Pro konečnou simulaci vody je třeba vyřešit soustavu NSE za pomoci pole rychlosti, pole hustoty a pole tlaku, které SPH interpoluje z vlastností částic. Díky aplikaci Lagrangeova popisu kontinua se některé rovnice výrazně zjednoduší. Rovnice o zachování hmoty 2.1 je automaticky splněna, neboť počet částic zůstane při simulaci konstantní. Dále lze zjednodušit v rovnici o zachování hybnosti 2.2 výraz  $\rho \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}$ , protože se částice pohybují s tekutinou, může se z výrazu vynechat člen  $\mathbf{v} \cdot \nabla \mathbf{v}$ , který reprezentuje změnu rychlosti v případě, že se částice pohybuje *skrz* tekutinu. [20]. Konečná NSE pro SPH částicový systém zní takto

$$\rho \frac{\partial \mathbf{v}}{\partial t} = -\nabla p = \rho \mathbf{g} + \mu \nabla^2 \mathbf{v}. \quad (3.20)$$

Pohyb částice je následně určený silami vyvolané rozdílem tlaků  $-\nabla p$ , viskozitou tekutiny  $\mu \nabla^2 \mathbf{v}$  a gravitací  $\rho \mathbf{g}$  (na vodu působí i externí síly, ty lze zakomponovat s gravitací do  $\mathbf{g}$ ).

Částice si ale udržují informace jen o své pozici, rychlosti a hmotnosti, zbylé vlastnosti si musí simulátor spočítat sám. Každá z nich určuje nějaký objem  $V_i = \frac{m_i}{\rho_i}$ , zatímco hmotnost zůstává během celé simulace konstantní,  $\rho_i$  se musí evaluovat v každé vykreslovací smíčce (viz obrázek 3.10) [20]. Hustota  $\rho_i$  částice  $i$  na pozici  $\mathbf{x}_i$  je pak podle SPH 3.13 určena jako

$$\rho(\mathbf{x}_i) = \sum_j m_j \frac{\rho_j}{\rho_j} W(|\mathbf{x}_i - \mathbf{x}_j|, h) = \sum_j m_j W(|\mathbf{x}_i - \mathbf{x}_j|, h). \quad (3.21)$$

Dále je třeba vypočítat tlak  $p_i$  na pozici  $\mathbf{x}_i$ , který lze aproximovat pomocí stavové rovnice ideálního plynu [20]

$$p_i = k \rho_i, \quad (3.22)$$

kde  $k$  je konstanta plynu závislá na teplotě a  $\rho_i$  hustota částice. Desburn navrhl úpravu rovnice 3.22, která přispěla k větší stabilitě simulace [22]



$$p_i = k(\rho_i - \rho_0), \quad (3.23)$$

kde  $\rho_0$  označuje klidovou hustotu tekutiny. Protože síla vyvolaná rozdílem tlaku závisí na gradientu tlaku, je posun všech hodnot o konstantu stále validním řešením.

S výše uvedenými informacemi lze nakonec spočítat síly působící v tekutině  $-\nabla p$  a  $\mu \nabla^2 \mathbf{v}$ . Na základě SPH 3.13 je tlaková síla působící na částici  $i$  rovna

$$\mathbf{f}_i^{tlak} = -\nabla p(\mathbf{x}_i) = -\sum_j m_j \frac{p_j}{\rho_j} \nabla W(|\mathbf{x}_i - \mathbf{x}_j|, h). \quad (3.24)$$

Bohužel takto vypočítané síly nejsou symetrické (nesplňují Newtonův třetí zákon) [20], tento problém je nejvíce patrný, když simulace obsahuje jen dvě částice. Protože gradient *smoothing kernelu* je v středu, tj. na pozici částice  $i$ , roven nule, částice použije pro výpočet jen tlak částice  $j$ . Müller řeší tento problém jednoduchým průměrováním hodnot  $p_i$  a  $p_j$

$$\mathbf{f}_i^{tlak} = -\nabla p(\mathbf{x}_i) = -\sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(|\mathbf{x}_i - \mathbf{x}_j|, h). \quad (3.25)$$

Podobný problém nastává při výpočtu sil způsobené viskozitou tekutiny

$$\mathbf{f}_i^{viskozita} = \mu \nabla^2 \mathbf{v}(\mathbf{x}_i) = \mu \sum_j m_j \frac{\mathbf{v}_j}{\rho_j} \nabla^2 W(|\mathbf{x}_i - \mathbf{x}_j|, h), \quad (3.26)$$

protože se rychlosti u každé částice liší. Symetrizace, kterou navrhl Müller [20], spočívá v tom, že síla způsobena viskozitou závisí pouze na rozdílech rychlostí částice od svých sousedů, což přirozeně vede k

$$\mathbf{f}_i^{viskozita} = \mu \nabla^2 \mathbf{v}(\mathbf{x}_i) = \mu \sum_j m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla^2 W(|\mathbf{x}_i - \mathbf{x}_j|, h), \quad (3.27)$$

Gravitace a externí síly jsou přímo aplikovány na samotné částice. Se všemi těmito informacemi lze vypočítat NSE, které lze formulovat Newtonovým druhým zákonem

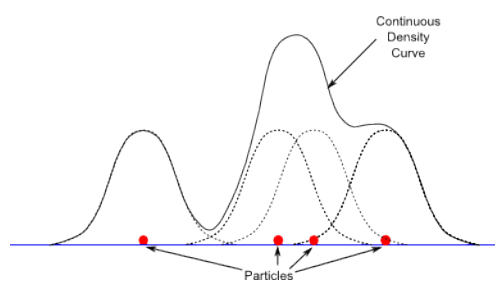
$$\mathbf{a}_i = \frac{D\mathbf{v}_i}{Dt} = \frac{\mathbf{f}_i}{\rho_i}, \quad (3.28)$$

kde  $a_i$  je zrychlení,  $\rho_i$  je hustota a  $\mathbf{f}_i$  součet sil působících na částici  $i$ . Pro získání rychlosti  $\mathbf{v}_i$  stačí integrovat podle času zrychlení  $\mathbf{a}_i$ . Müllerův simulátor [20] pro integraci používal algoritmus Leap-Frog [24], který integruje podle konstantní časových rozestupů, pro lepší výsledky navrhl algoritmy založené na Courant-Friedrichs-Lewy podmínce [22].

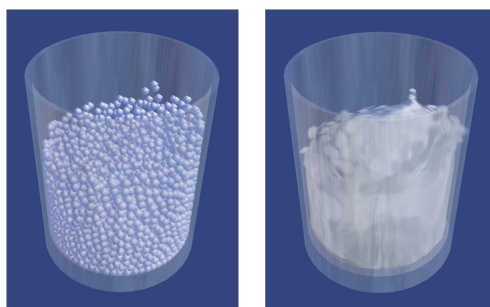
Posledním krokem je vizualizace částic. Bez žádných speciálních metod by částice měly pouhý tvar kuliček, které nejsou nijak spojeny. Řešením je tzv. *point splatting* [25], který na základě množiny bodů vytvoří jednolitý útvar.

### 3. VLASTNOSTI VODNÍHO POVRCHU

---



Obrázek 3.11: Interpolace pole hustoty podle SPH [23]



Obrázek 3.12: Müllerova simulace vody založená na SPH [20]

---

## Závěr

Jedním z cílů práce bylo provést analýzu současných možností pro simulaci vodního povrchu. Rozbor obsahuje několik metod založených na různých přístupech k simulaci kapalin. Na základě typu scén her a míry interakce vodní plochy s prostředím jsou jednotlivé metody různě vhodné.

Dalším cílem bylo vytvořit testovací scénu s vybraným algoritmem pro simulace vodní plochy z analýzy. Testovací scéna je naimplementována pomocí frameworku OpenGL, ale principy simulace jsou lehce přenositelné na různé nástroje. Vodní plocha reaguje jak na prostředí testovací scény, tak i na manipulace hladiny uživatelem.



---

## Bibliografie

1. BRIDSON, Robert; MÜLLER-FISCHER, Matthias. Fluid Simulation: SIGGRAPH 2007 Course Notes<i>Video Files Associated with This Course Are Available from the Citation Page</i>. In: *ACM SIGGRAPH 2007 Courses*. San Diego, California: Association for Computing Machinery, 2007, s. 1–81. SIGGRAPH '07. ISBN 9781450318235. Dostupné z DOI: 10.1145/1281500.1281681.
2. MÜLLER-FISCHER, Matthias. *Fast Water Simulation for Games Using Height Fields* [online]. 2008 [cit. 2022-03-15]. Dostupné z: <https://www.gdcvault.com/play/203/Fast-Water-Simulation-for-Games>. Game Developers Conference.
3. GUENDELMAN, Eran; SELLE, Andrew; LOSASSO, Frank; FEDKIW, Ronald. Coupling Water and Smoke to Thin Deformable and Rigid Shells. In: *ACM SIGGRAPH 2005 Papers*. Los Angeles, California: Association for Computing Machinery, 2005, s. 973–981. SIGGRAPH '05. ISBN 9781450378253. Dostupné z DOI: 10.1145/1186822.1073299.
4. MAX, Nelson L. Vectorized Procedural Models for Natural Terrain: Waves and Islands in the Sunset. *SIGGRAPH Comput. Graph.* 1981, roč. 15, č. 3, s. 317–324. ISSN 0097-8930. Dostupné z DOI: 10.1145/965161.806820.
5. FERNANDO, R. *GPU gems*. Vyd. 1. Boston: Addison-Wesley, 2004. ISBN 0-321-22832-4.
6. FAKULTA ARCHITEKTURY ČVUT V PRAZE. Cyklické křivky. In: *Deskriptivní geometrie* [online]. [B.r.] [cit. 2022-03-15]. Dostupné z: [https://www.fa.cvut.cz/studium/predmety/deskriptivni-geometrie-i-ii/dg\\_elskripta/krivky/cyklicke\\_krivky\\_1.pdf](https://www.fa.cvut.cz/studium/predmety/deskriptivni-geometrie-i-ii/dg_elskripta/krivky/cyklicke_krivky_1.pdf).
7. GERSTNER, František Josef. *Theorie der wellen: samt daraus abgeleiteten theorie der deichprofile*. Gottlieb Haase, 1804.

8. KRAAIENNEST. *Trochoidal wave*. 2015. Dostupné také z: [https://commons.wikimedia.org/wiki/File:Trochoidal\\_wave.svg](https://commons.wikimedia.org/wiki/File:Trochoidal_wave.svg).
9. FOURNIER, Alain; REEVES, William T. A Simple Model of Ocean Waves. *SIGGRAPH Comput. Graph.* 1986, roč. 20, č. 4, s. 75–84. ISSN 0097-8930. Dostupné z DOI: 10.1145/15886.15894.
10. FLICK, Jasper. Waves. In: *Catlike Coding* [online]. [B.r.] [cit. 2022-03-15]. Dostupné z: <https://catlikecoding.com/unity/tutorials/flow/waves/>.
11. On Vertex Shader Performance. In: *GitHub* [online]. [B.r.] [cit. 2022-03-15]. Dostupné z: <https://paroj.github.io/gltut/Positioning/Tut03%20n%20Vertex%20Shader%20Performance.html>.
12. EMMANUELLE, Darles; CRESPIEN, Benoît; GHAZANFARPOUR, Djamchid; GONZATO, Jean-Christophe. A Survey of Ocean Simulation and Rendering Techniques in Computer Graphics. *Computer Graphics Forum - CGF*. 2011, roč. 30. Dostupné z DOI: 10.1111/j.1467-8659.2010.01828.x.
13. TESSENDORF, Jerry et al. Simulating ocean water. *Simulating nature: realistic and interactive techniques. SIGGRAPH*. 2001, roč. 1, č. 2, s. 5.
14. Complexity of FFT algorithms (Cooley-Tukey, Bluestein, Prime-factor). In: *Stack Exchange* [online]. [B.r.] [cit. 2022-03-15]. Dostupné z: <https://math.stackexchange.com/questions/1704788/complexity-of-fft-algorithms-cooley-tukey-bluestein-prime-factor>.
15. OpenGL Ocean Rendering (fast Fourier transform on GPU). In: *YouTube* [online]. [B.r.] [cit. 2022-03-16]. Dostupné z: <https://www.youtube.com/watch?v=CeJCNmI-B7s>.
16. REEVES, William T. Particle Systems—a Technique for Modeling a Class of Fuzzy Objects. *SIGGRAPH Comput. Graph.* 1983, roč. 17, č. 3, s. 359–375. ISSN 0097-8930. Dostupné z DOI: 10.1145/964967.801167.
17. MILLER, Gavin; PEARCE, Andrew. Globular dynamics: A connected particle system for animating viscous fluids. *Computers & Graphics*. 1989, roč. 13, č. 3, s. 305–309. ISSN 0097-8493. Dostupné z DOI: [https://doi.org/10.1016/0097-8493\(89\)90078-2](https://doi.org/10.1016/0097-8493(89)90078-2).
18. MONAGHAN, Joseph. Smoothed Particle Hydrodynamics. *Reports on Progress in Physics*. 2005, roč. 68, s. 1703. Dostupné z DOI: 10.1088/0034-4885/68/8/R01.
19. LUCY, L B. Numerical approach to the testing of the fission hypothesis. *Astron. J.; (United States)*. 1977, roč. 82:12. Dostupné z DOI: 10.1086/112164.

20. MÜLLER, Matthias; CHARYPAR, David; GROSS, Markus. Particle-based fluid simulation for interactive applications. In: *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 2003, s. 154–159.
21. JLCERCOS. *Schematic view of a SPH convolution*. 2018. Dostupné také z: <https://commons.wikimedia.org/wiki/File:SPHInterpolationColorsVerbose.svg>.
22. DESBRUN, Mathieu; GASCUEL, Marie-Paule. Smoothed Particles: A New Paradigm for Animating Highly Deformable Bodies. In: *Proceedings of the Eurographics Workshop on Computer Animation and Simulation '96*. Poitiers, France: Springer-Verlag, 1996, s. 61–76. ISBN 3211828850.
23. SIMON GREEN Richard Tonge, Miguel Sainz. *Fluid Simulation in Alice: Madness Returns*. 2011. Dostupné také z: <https://developer.nvidia.com/content/fluid-simulation-alice-madness-returns>.
24. POZRIKIDIS, C. *Numerical Computation in Science and Engineering*. 2nd ed. New York: Oxford University Press, 2008. ISBN 978-0-19-537611-1.
25. MATTHIAS ZWICKER Hanspeter Pfister, Markus Gross. *SIGGRAPH '01: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. New York, NY, USA: Association for Computing Machinery, 2001. ISBN 158113374X.





## Seznam použitých zkratk

**NSE** Navier–Stokes equations

**FFT** Fast Fourier transform

**IFFT** Inverse fast Fourier transform

**SPH** Smoothed particle hydrodynamics



## **Obsah přiloženého CD**