

Auteursrechterlijke overeenkomst

Opdat de Universiteit Hasselt uw eindverhandeling wereldwijd kan reproduceren, vertalen en distribueren is uw akkoord voor deze overeenkomst noodzakelijk. Gelieve de tijd te nemen om deze overeenkomst door te nemen, de gevraagde informatie in te vullen (en de overeenkomst te ondertekenen en af te geven).

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling met

Titel: Subsurface scattering oplossen m.b.v. radiale basisfuncties

Richting: 2de masterjaar in de informatica - multimedia

Jaar: 2009

in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Niet tegenstaand deze toekenning van het auteursrecht aan de Universiteit Hasselt behoud ik als auteur het recht om de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij te reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

Ik bevestig dat de eindverhandeling mijn origineel werk is, en dat ik het recht heb om de rechten te verlenen die in deze overeenkomst worden beschreven. Ik verklaar tevens dat de eindverhandeling, naar mijn weten, het auteursrecht van anderen niet overtreedt.

Ik verklaar tevens dat ik voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen heb verkregen zodat ik deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal mij als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze overeenkomst.

Ik ga akkoord,

VANMONTFORT, Wouter

Datum: 14.12.2009

Subsurface scattering oplossen m.b.v. radiale basisfuncties

Wouter Vanmontfort

promotor :
Prof. dr. Philippe BEKAERT

Samenvatting

Subsurface scattering treedt op bij voorwerpen die bestaan uit translucente materialen. Om deze voorwerpen realistisch te renderen, zal het effect van subsurface scattering in rekening gebracht moeten worden. In zijn meest algemene vorm wordt subsurface scattering beschreven door de lichttransport vergelijking. Dit is een vrij complexe partiële differentiaal vergelijking. Veel materialen waarin lichtverstrooiing optreedt, zijn echter highly scattering. Doordat het licht in deze materialen fel verstrooid wordt, wordt de richtingsafhankelijkheid van de lichtverstrooiing sterk gereduceerd. Dit laat toe om een aantal vereenvoudigingen door te voeren in de lichttransport vergelijking.

De vereenvoudigde vergelijking die dan bekomen wordt, kan opgelost worden m.b.v. numerieke methodes. Bij de meeste numerieke methodes moet het voorwerp gediscretiseerd worden a.d.h.v. een mesh. Het opstellen van zo'n mesh in een 3D domein is echter niet altijd even gemakkelijk. In deze thesis wordt daarom voorgesteld om de *Boundary Knot Method* te gebruiken om de vergelijking op te lossen. Dit is een numerieke methode om partiële differentiaal vergelijkingen op te lossen waarbij er gebruik gemaakt wordt van radiale basisfuncties. Het grote voordeel van deze methode is dat er geen mesh vereist is.

Dankwoord

Hier zou ik graag van de gelegenheid willen gebruik maken om een aantal mensen te bedanken. Eerst en vooral wil ik mijn begeleider Tom Haber bedanken voor de hulp en input voor deze thesis. Ook wil ik graag prof. dr. Philippe Bekeart bedanken voor het mogelijk maken van deze thesis. Verder wil ik ook nog prof. dr. Wen Chen en Fuzhang Wang bedanken voor het beantwoorden van mijn vragen over de Boundary Knot Method. Natuurlijk gaat er ook nog een woord van dank uit naar mijn ouders en mijn broer, voor de steun die ze mij hebben gegeven en voor het begrip dat ze hebben opgebracht tijdens het maken van deze thesis.

Inhoudsopgave

1	Inleiding	1
2	Licht	3
2.1	Rendering equation	4
2.2	Radiative Transfer Equation	4
3	Subsurface scattering	6
3.1	Interactie van licht met deeltjes	6
3.1.1	Emission	6
3.1.2	Absorption	7
3.1.3	Out-scattering	7
3.1.4	In-scattering	7
3.1.5	Combinatie	8
3.1.6	Fase functie	8
3.2	Opsplitsing	9
3.3	Diffusion approximation	11
3.4	Oplossingsmethodes voor subsurface scattering	13
3.4.1	Diffuse reflectie	13
3.4.2	Reflection from Layered Surfaces	13
3.4.3	Monte Carlo	13
3.4.4	BSSRDF	14
3.4.5	Multigrid	14

4	Wiskundige methodes	16
4.1	Discretisatie en Stelsel van vergelijkingen	16
4.2	Finite Difference Method	17
4.3	Finite Element Method	17
4.4	Boundary Element Method	18
4.5	Boundary Knot Method	18
4.6	Relaxatie	18
4.6.1	Jacobi	19
4.6.2	Gauss-Seidel	19
4.6.3	Successive Overrelaxation	20
4.6.4	Multigrid	20
4.7	Andere methodes	20
4.8	Vergelijking	21
5	Radiale basisfuncties	22
5.1	Radiale basisfuncties	22
5.2	PDE's oplossen met RBF's	25
5.3	Boundary Knot Method	27
6	Uitwerking	30
6.1	Functie bepaling	30
6.2	Bepaling van de source term	34
6.2.1	Photon mapping	34
6.2.2	Irradiance sampling	35
6.3	Bepalen van de BKM coëfficiënten	36
6.4	Visualisatie	38
7	Resultaten	40
8	Conclusie en future work	55
8.1	Conclusie	55
8.2	Future work	55

Hoofdstuk 1

Inleiding

Subsurface scattering, lichtverstrooiing, treedt op in translucente voorwerpen. In de wereld rondom ons zijn er veel materialen die translucient (\approx doorschijnend) zijn. Een aantal bekende voorbeelden hiervan zijn marmer, was, porselein, melk, gezandstraald glas, huid en bladeren. Bij het renderen van scènes waarin er translucente materialen worden gebruikt, is het dus vrij belangrijk dat subsurface scattering in rekening wordt gebracht. Bij rendering algoritmes die enkel gebruik maken van pure oppervlakte reflecties, zullen deze materialen een onrealistisch uitzicht hebben. Subsurface scattering zorgt voor een zachter uitzicht van een voorwerp. Kleine details worden afgezwakt door de lichtverstrooiing. Licht dat aan de voorzijde een object binnendringt kan echter ook, door de verspreiding van het licht in het object, details aan de achterzijde ervan doen oplichten.

Er wordt meestal ook een onderscheid gemaakt tussen homogene en heterogene materialen. Bij homogene materialen is de dichtheid en de grootte van de deeltjes, waaruit het materiaal is opgebouwd, constant. Bij heterogene materialen is de dichtheid en de grootte van deze deeltjes niet constant. Slechts een klein deel van de subsurface scattering algoritmes houdt rekening met heterogene materialen. Het overgrote deel beperkt zich echter tot subsurface scattering in homogene materialen. Een veronderstelling die ook vaak gemaakt wordt, is dat het materiaal *highly-scattering* is. Eenvoudig gezegd, wil dit zeggen dat het licht sterk verstrooid wordt in het materiaal. Aangezien een groot deel van de translucente materialen highly-scattering zijn, is dit een gerechtvaardigde veronderstelling.

Het modelleren van lichtverstrooiing gebeurt zeker niet alleen in het vakgebied van de computer graphics. Lichtverstrooiing is een fenomeen dat ook een belangrijke rol speelt in (bio)medische wetenschappen en in de astronomie. Het is zelfs zo dat een groot deel van de onderliggende principes die in de computer graphics worden gebruikt, voortvloeien uit het onderzoek dat in deze vakgebieden is gebeurd. In literatuur over lichtverstrooiing komt men ook vaak de term *participating media* tegen. Deze term wordt meestal gebruikt als men het heeft over de verstrooiing van licht in gassen en vloeistoffen. In het geval van vaste stoffen gebruikt men meestal de term subsurface scattering. Technieken ontwikkeld voor participating media kunnen ook vaak toegepast worden, mits enkele aanpassingen, voor subsurface scattering, en omgekeerd.

Het volgende hoofdstuk geeft een korte geschiedenis van de verschillende lichtmodellen. Ook worden de *rendering equation* en de *Radiative Transfer Equation* kort besproken. In het derde hoofdstuk komt subsurface scattering meer in detail aan bod. In dit hoofdstuk worden o.a. de verschillende interacties besproken die optreden bij lichtverstrooiing. Verder wordt er ook de

Diffusion Approximation besproken. Als laatste wordt er in dit hoofdstuk een overzicht gegeven van een aantal methodes die gebruikt worden om subsurface scattering te modelleren. In het vierde hoofdstuk komen er een aantal numerieke methodes aan bod die vaak gebruikt worden bij het oplossen van partiële differentiaal vergelijkingen. Het hoofdstuk dat daarop volgt spitst zich toe op radiale basisfuncties en hoe deze kunnen gebruikt worden om partiële differentiaal vergelijkingen op te lossen. In dit hoofdstuk wordt ook de *Boundary Knot Method* uitgelegd. De uitwerking van de ontwikkelde techniek wordt besproken in het zesde hoofdstuk. In hoofdstuk zeven worden de bekomen resultaten gegeven en besproken. Conclusies over de uitgewerkte techniek en mogelijkheden voor de verdere ontwikkeling ervan worden in het laatste hoofdstuk gegeven.

Hoofdstuk 2

Licht

Wat is licht? Dat is ook de vraag die men zich stelde in de tijd van de oude Grieken. In deze periode zijn de grondslagen gelegd voor de geometrische optica. Sommige van de Griekse geleerden dachten toen dat er stralen vanuit het oog vertrokken. Met deze stralen zou dan de omgeving als het ware afgetast worden, waardoor we dingen konden waarnemen. Dit bleek achteraf een foutieve veronderstelling te zijn, maar dit principe is in een zeker opzicht terug te vinden in hedendaagse rendertechnieken. Bijvoorbeeld bij ray tracing en ray casting vertrekken er ook stralen (*rays*) vanuit een virtueel oogpunt in de richting van de te renderen scène.

Tijdens de 17e eeuw is men verder gaan onderzoeken wat licht juist is. Hierbij waren er twee theorieën die gelijktijdig ontwikkeld werden. Langs de ene kant was er Isaac Newton die er vanuit ging dat licht een deeltjeskarakter had [42]. Langs de andere kant ging Christiaan Huygens er vanuit dat licht een golfkarakter had [23]. Doordat interferentie niet kon worden verklaard door het deeltjesmodel maar wel met het golfmodel, werd het deeltjesmodel op de achtergrond geplaatst. In de 19e eeuw toonde James Maxwell met behulp van *De wetten van Maxwell* aan dat licht verklaard kon worden als een elektromagnetische golf [38]. Daarom werd algemeen aangenomen dat licht een golfkarakter had.

Maar tegen het einde van de 19e eeuw zorgde het foto-elektrisch effect ervoor dat de golftheorie in vraag werd gesteld. Dit effect kon niet worden verklaard aan de hand van het golfmodel voor licht. Het deeltjesmodel kon dit fenomeen echter wel verklaren. Gebruikmakende van de kwantummechanica formuleerde Albert Einstein het *golf-deeltje-dualisme* (*wave-particle duality*) [11]. Hierbij wordt verondersteld dat licht bestaat uit fotonen (*photons*). Deze fotonen gedragen zich in sommige opzichten als een golf, maar in andere als een deeltje.

Dit model wordt dikwijls omschreven als *quantum optics*. In de computer graphics is het detail dat mogelijk is met dit model, echter zelden vereist. Ook het golfmodel wordt meestal nog als te gedetailleerd gezien. Tijdens de simulatie van het licht in computer graphics past men daarom meestal de gewone geometrische optica toe. Eén van de hoofdredenen hiervoor is dat de berekeningen van de formules relatief eenvoudig blijven. Een nadeel hiervan is dat een aantal zichtbare effecten zoals interferentie en diffractie, niet gesimuleerd worden.

In het algemeen gaat men bij de lichtsimulatie in de computer graphics uit van de volgende veronderstellingen:

- Licht verplaatst zich volgens een rechte lijn.

- Licht verplaatst zich ogenblikkelijk door een medium.
- Licht ondervindt geen externe invloeden van gravitatie en magnetische velden.

2.1 Rendering equation

Eén van de meest gebruikte formules bij de simulatie van globale belichting is de zogenaamde *rendering equation*. Deze vergelijking werd in 1986 gelijktijdig door Kajiyama [32] en Immel [24] voorgesteld en is gebaseerd op *de wet van het behoud van energie*. De rendering equation is van de volgende vorm:

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{S^2} f_s(x, \omega_o, \omega_i) L_i(x, \omega_i) (\omega_i \cdot n) d(\omega_i) \quad (2.1)$$

Deze vergelijking laat toe om de radiantie L_o^1 te berekenen die uitgestraald wordt volgens de richting ω_o in een punt x , dat op een oppervlak van de scène ligt. Deze radiantie wordt berekend uitgaande van het licht dat gegenereerd wordt in het punt x ($L_e(x, \omega_o)$) en de inkomende radiantie uit de rest van de scène. De inkomende radiantie wordt beschreven door de integraal in het rechterlid van de vergelijking. Het integratiedomein van deze integraal is de bol (*sphere*) S met x als middelpunt. Het integratiedomein wordt dikwijls beperkt tot de halve bol (*hemisphere*) H die gericht is volgens de normaal van het oppervlak in het beschouwde punt x . De functie $f_s(x, \omega_o, \omega_i)$ bepaalt de mate waarin licht uit richting ω_i wordt verstrooid in de richting ω_o . Men noemt deze functie de *Bidirectional Scattering Distribution Function* (BSDF). Indien alleen het gereflecteerde deel van het verstrooide licht beschouwd wordt, spreekt men vaak van de *Bidirectional Reflectance Distribution Function* (BRDF). $L_i(x, \omega_i)$ is de radiantie die uit de richting ω_i toekomt in het punt x . Waarbij ω_i een richting is die tot het integratiedomein behoort. Radiantie is gedefinieerd als de lichtflux per eenheid oppervlak en per eenheid ruimtehoek ($W/steradial\ m^2$). Het oppervlak wordt hierbij loodrecht op de voortplantingsrichting van de lichtflux verondersteld. Het licht valt echter niet altijd loodrecht in op het punt x . Dit richtingsverschil wordt in rekening gebracht door $(\omega_i \cdot n)$. Hierbij is n de normaal van het oppervlak in het punt x en ω_i de richting van het inkomende licht. Voor meer informatie over de rendering equation verwijs ik door naar [10].

Globale belichting kan dus gesimuleerd worden door de rendering equation uit te rekenen. Dit kan op verschillende manieren gebeuren, bijvoorbeeld met ray tracing of met radiosity methodes. De rendering equation vertrekt van het geometrische lichtmodel en beperkt zich o.a. tot licht dat zich in vacuüm verplaatst. Een andere veronderstelling die gemaakt wordt is dat licht alleen op het oppervlak wordt weerkaatst. Hierdoor kan de rendering equation lichtverstrooiing in participerende media en onder het oppervlak van objecten niet simuleren. Subsurface scattering zal dus op een andere manier moeten worden gesimuleerd.

2.2 Radiative Transfer Equation

De Radiative Transfer Equation (RTE), ook wel de Radiative Transport Equation genoemd, beschrijft algemeen hoe licht zich door een medium verplaatst. De vergelijking is ook gebaseerd op het behoud van energie in een punt x .

¹Ook wel specifieke intensiteit genoemd.

$$\frac{\partial L(x, \omega, t)/c}{\partial t} = -\omega \cdot \nabla L(x, \omega, t) + \sigma_t L(x, \omega, t) + \sigma_s \int_{4\pi} L(x, \omega', t) p(x, \omega, \omega') d\omega' + S(x, \omega, t) \quad (2.2)$$

De RTE is een vrij complexe vergelijking om op te lossen. Om rechtstreeks een oplossing van de RTE te bekomen, past men meestal Monte Carlo methodes toe [3, 55, 40]. Om de vergelijking iets of wat te vereenvoudigen wordt meestal verondersteld dat de functie tijdsonafhankelijk is. In het volgende hoofdstuk wordt een simpele afleiding gegeven voor een tijdsonafhankelijke RTE. Voor meer informatie over de RTE verwijs ik door naar [25].

Hoofdstuk 3

Subsurface scattering

Als een lichtstraal het oppervlak van een voorwerp bereikt, zal deze straal voor een deel weerkaatst worden aan het oppervlak en voor een deel verder gaan in het voorwerp. Afhankelijk van de materiaaleigenschappen van het voorwerp zal de verhouding van het weerkaatste en het binnendringende licht anders zijn. Het licht dat binnendringt in het voorwerp zal er voor een deel geabsorbeerd worden. Het licht dat niet geabsorbeerd wordt, zal ofwel zich ongehinderd voortplanten ofwel verstrooid worden. Deze verstrooiing van het licht, onder het oppervlak het voorwerp, is wat men in het Engels *Subsurface scattering* noemt. Het ongehinderd voortplanten van het licht kan ook gezien worden als licht dat volledig wordt verstrooid in de richting van de beschouwde lichtstraal.

Een groot deel van dit hoofdstuk is gebaseerd op [18, 25].

3.1 Interactie van licht met deeltjes

Een lichtstraal die een voorwerp of meer algemeen een medium binnendringt, zal met de deeltjes (particles) van dit medium interageren. Doordat het licht interageert met het medium zal de radiantie van de lichtstraal veranderen. Er zijn vier soorten interacties die kunnen optreden tussen de deeltjes van het medium en de lichtdeeltjes. Deze interacties worden in de volgende secties kort besproken. Hierbij beschouwen we een lichtstraal in het punt x die zich voortplant volgens een richting ω .

3.1.1 Emission

In sommige voorwerpen kunnen de deeltjes zelf energie in de vorm van licht uitstralen. Hierdoor zal de energie van de beschouwde lichtstraal toenemen. De *emittance* functie $\epsilon(x, \omega)$ [W/m^3] beschrijft hoeveel energie er in het punt uitgezonden wordt in een bepaalde richting. Meestal is de emittance functie isotropisch, zodat er geen richtingsafhankelijkheid is. De verandering in radiantie van de beschouwde lichtstraal door de emissie van de deeltjes wordt gegeven door:

$$(\omega \cdot \nabla)L(x, \omega) = \epsilon(x, \omega) \tag{3.1}$$

3.1.2 Absorption

De tegenhanger van emissie is absorptie. Als de fotonen van de lichtstraal zich door het medium verplaatsen, zal een deel van deze fotonen met de deeltjes in het medium botsen. Als een foton met een deeltje van het medium botst, is er een kans dat het foton door dit deeltje wordt geabsorbeerd. Dit absorberen is eigenlijk een omzetting van licht naar een andere vorm van energie (meestal warmte). De kans dat een foton effectief geabsorbeerd wordt, wordt aangegeven met de absorptiecoëfficiënt $\sigma_a(x)$ [1/mm].¹ De absorptiecoëfficiënt σ_a geeft de kans, per mm afgelegde afstand, dat een foton wordt geabsorbeerd. Deze coëfficiënt is afhankelijk van de dichtheid ρ van het medium. Als deze dichtheid plaatsafhankelijk is, zal ook de absorptiecoëfficiënt plaatsafhankelijk zijn. In dit geval noemt men het medium of voorwerp heterogeen. Als de dichtheid ρ constant is, spreekt men van een homogeen medium. Door het optreden van absorptie zal de radiantie van de lichtstraal afnemen, dit wordt beschreven door:

$$(\omega \cdot \nabla)L(x, \omega) = \sigma_a(x)L(x, \omega) \quad (3.2)$$

3.1.3 Out-scattering

De fotonen in de lichtstraal kunnen, zoals eerder al gezegd, botsen met de deeltjes in het medium. Bij zo'n botsing kan het foton, in tegenstelling tot geabsorbeerd worden, ook verstrooid worden. Het foton zal in dat geval dus van baan veranderen. Dit fenomeen noemt men out-scattering. Vergelijkbaar met de absorptiecoëfficiënt geeft de *scattering*-coëfficiënt de kans dat een foton verstrooid wordt door een botsing met het medium. De scattering-coëfficiënt wordt aangeduid met $\sigma_s(x)$. Net zoals de absorptiecoëfficiënt $\sigma_a(x)$ kan de scattering-coëfficiënt $\sigma_s(x)$ in een heterogeen medium ook afhankelijk zijn van de positie x . Als een foton verstrooid wordt als gevolg van een botsing, zal het dus niet meer volgens de richting van de lichtstraal voortbewegen. Hierdoor zal de radiantie van de lichtstraal vergelijkbaar met 3.2 afnemen volgens:

$$(\omega \cdot \nabla)L(x, \omega) = \sigma_s(x)L(x, \omega) \quad (3.3)$$

Absorptie en out-scattering zorgen op een vergelijkbare manier voor de afname van de radiantie van een lichtstraal. Daarom worden de absorptiecoëfficiënt $\sigma_a(x)$ en de scattering-coëfficiënt $\sigma_s(x)$ dikwijls gecombineerd tot de *extinction*-coëfficiënt $\sigma_t(x) = \sigma_a(x) + \sigma_s(x)$. $\sigma_t(x)$ geeft de kans dat een foton wordt geabsorbeerd of verstrooid in het punt x , m.a.w. de kans dat er een botsing optreedt in het punt x . Een andere waarde die bekomen wordt door deze coëfficiënten te combineren is het albedo $\alpha(x) = \frac{\sigma_s(x)}{\sigma_t(x)}$. $\alpha(x)$ geeft aan wat de kans is dat een foton verstrooid wordt als er een botsing optreedt.

3.1.4 In-scattering

De tegenhanger van out-scattering is *in-scattering*. In een punt x zullen niet alleen de fotonen die de richting ω volgen verstrooid worden. Ook de fotonen uit andere richtingen zullen door te botsen met de deeltjes van het medium, verstrooid worden in het punt x . De kans dat deze fotonen worden verstrooid, is hetzelfde als de kans dat de fotonen van de beschouwde lichtstraal worden verstrooid. Deze kans wordt voor deze fotonen ook gegeven door de scattering-coëfficiënt

¹In sommige teksten wordt het symbool μ i.p.v. σ gebruikt om de coëfficiënt aan te duiden.

$\sigma_s(x)$. De kans dat een foton uit een andere richting na een botsing in x zich volgens de richting ω zal voortplanten, wordt gegeven door de fase functie (*phase function*) $p(x, \omega, \omega')$. Deze functie geeft de kans dat een foton komende uit de richting ω' , in het punt x , wordt verstrooid volgens de richting ω . De fase functie is een genormaliseerde functie:

$$\int_{S^2} p(x, \omega, \omega') d\omega' = 1$$

Doordat een aantal fotonen uit andere richtingen verstrooid worden in de richting ω van de beschouwde lichtstraal, zal de radiantie van deze lichtstraal toenemen. Deze toename is afhankelijk van de fase functie en wordt gegeven door:

$$(\omega \cdot \nabla)L(x, \omega) = \sigma_s(x) \int_{S^2} p(x, \omega, \omega') L(x, \omega') d\omega' \quad (3.4)$$

3.1.5 Combinatie

De verschillende interacties zorgen zowel voor een toename als voor een afname van de radiantie in de lichtstraal. Door deze verschillende veranderingen bij elkaar op te tellen bekomen we een tijdsafhankelijke RTE, namelijk:

$$\begin{aligned} (\omega \cdot \nabla)L(x, \omega) &= \epsilon(x, \omega) \\ &\quad - \sigma_a(x)L(x, \omega) \\ &\quad - \sigma_s(x)L(x, \omega) \\ &\quad + \sigma_s(x) \int_{S^2} p(x, \omega, \omega') L(x, \omega') d\omega' \end{aligned} \quad (3.5)$$

Vergelijking 3.5 geeft voor een punt x dus de verandering in radiantie L volgens de richting ω . Hierbij kunnen de coëfficiënten σ_a en σ_s afhankelijk zijn van de positie, maar zijn ze constant in de tijd.

3.1.6 Fase functie

De fase functie bepaalt hoe de fotonen die botsen in een punt x verstrooid worden. $p(x, \omega, \omega')$ geeft dus aan hoeveel van het licht dat in het punt x invalt onder de richting ω wordt weerkaatst in de richting ω' . Voor een fase functie kan men ook de gemiddelde cosinus van de fase functie $g(x)$ berekenen als volgt:

$$g(x) = \int_{S^2} p(x, \omega, \omega') (\omega \cdot \omega') d\omega' \quad (3.6)$$

$g(x)$ geeft het gemiddelde van het verschil tussen *forward scattering* ($\omega \cdot \omega' > 0$) en *backward scattering* ($\omega \cdot \omega' < 0$). Het is dus een aanduiding van de voorkeursrichting van de lichtverstrooiing (*preferred scattering direction*) in het punt x . Om de fase functie enigszins te vereenvoudigen

wordt meestal verondersteld dat de fase functie enkel afhankelijk is van de hoek tussen ω en ω' , dus:

$$p(x, \omega, \omega') = p(\omega \cdot \omega') = p(\cos \theta) = p(\mu) \quad (3.7)$$

Er zijn verschillende modellen die de fase functie van een medium beschrijven. Deze modellen verschillen in complexiteit en toepasbaarheid. Deze toepasbaarheid houdt in dat er afhankelijk van de verhouding tussen de deeltjesgrootte r van het medium en de golflengte λ van het licht, meestal andere modellen worden gebruikt. De eenvoudigste fase functie is de isotropische fase functie:

$$p(\mu) = \frac{1}{4\pi} \quad (3.8)$$

In veel gevallen is deze isotropische fase functie geen goede benadering voor het verstrooiingsgedrag van het medium. Een veel gebruikte fase functie, die een betere benadering mogelijk maakt, is de *Henyeey-Greenstein* fase functie:

$$p(\mu)_{HG} = \frac{1}{4\pi} \frac{1 - g^2}{(1 - 2g\mu + g^2)^{\frac{3}{2}}} \quad (3.9)$$

De isotropische fase functie kan ook gecombineerd worden met de Henyeey-Greenstein fase functie. Of twee Henyeey-Greenstein fase functies met een verschillende parameter g kunnen ook met elkaar gecombineerd worden om zo meer complexe verstrooiingsverdelingen te bekomen. Voor meer informatie over fase functies verwijs ik door naar [50].

$$\begin{aligned} p(\mu) &= \beta \frac{1}{4\pi} + (1 - \beta)p(\mu)_{HG} \\ p(\mu) &= \beta p(\mu)_{HG1} + (1 - \beta)p(\mu)_{HG2} \end{aligned}$$

3.2 Opsplitsing

Om de berekeningen te vergemakkelijken wordt de radiantie meestal in twee delen opgesplitst [25], de *reduced incident intensity* L_{ri} en de *diffuse intensity* L_d . Het licht dat een voorwerp binnendringt zal volgens 3.5 af- en toenemen. Het deel van dit licht dat door absorptie en verstrooiing in sterkte afneemt, is de reduced incident intensity L_{ri} en voldoet aan de vergelijking:

$$(\omega \cdot \nabla)L_{ri}(x, \omega) = -\sigma_t L_{ri}(x, \omega) \quad (3.10)$$

De oplossing voor deze vergelijking wordt gegeven door:

$$L_{ri}(x, \omega) = L_i(x_b, \omega) e^{\int_0^s -\sigma_t(x-r\omega) dr} \quad (3.11)$$

Hierbij is L_i , de zogenaamde *incident intensity*, de radiantie die op het oppervlak in punt x_b invalt. s is de afstand van dit oppervlaktepunt tot het beschouwde punt x [48]. L_{ri} heeft dus

een analytische oplossing. Hierbij komt het er op neer om de extinction coëfficiënt te integreren volgens de richting ω .

Het andere deel van het licht is de diffuse intensity L_d . De diffuse intensity L_d bevindt zich volledig binnenin het voorwerp en is een gevolg van de verstrooiing van het inkomende licht en de emissie in het voorwerp zelf. De totale radiantie $L(x, \omega) = L_{ri}(x, \omega) + L_d(x, \omega)$ voldoet aan vergelijking 3.5. Uit 3.5 en 3.10 volgt dat $L_d(x, \omega)$ moet voldoen aan:

$$\begin{aligned}
(\omega \cdot \nabla)L_d(x, \omega) &= \epsilon(x, \omega) \\
&\quad -\sigma_t(x)L_d(x, \omega) \\
&\quad +\sigma_s(x) \int_{S^2} p(x, \omega, \omega')L_d(x, \omega')d\omega' \\
&\quad +\sigma_s(x) \int_{S^2} p(x, \omega, \omega')L_{ri}(x, \omega')d\omega'
\end{aligned} \tag{3.12}$$

De laatste term in deze vergelijking wordt ook wel de *equivalent source function* ϵ_{ri} genoemd. Dit is de radiantie die gegenereerd wordt door de reduced incident intensity L_{ri} die x bereikt en er voor de eerste keer verstrooid wordt in de richting ω .

$$\epsilon_{ri}(x, \omega) = \sigma_s(x) \int_{S^2} p(x, \omega, \omega')L_{ri}(x, \omega')d\omega' \tag{3.13}$$

Vergelijking 3.12 is een partiële differentiaal vergelijking (*Partial Differential Equation*, PDE). Om een differentiaal vergelijking op te lossen zijn er randvoorwaardes nodig. De randvoorwaarde voor de PDE van de diffuse intensity L_d wordt gegeven door:

$$L_d(x_b, \omega) = 0 \quad | \quad \omega \cdot n < 0 \tag{3.14}$$

Hierbij is x_b een punt op het oppervlak, met als buitenwaarts gerichte normaal n . ω stelt dus alle richtingen voor die in het voorwerp gericht zijn. Deze randvoorwaarde is gebaseerd op het feit dat de diffuse intensity L_d volledig binnenin het voorwerp wordt gegenereerd. Er kan dus geen in-flow zijn van diffuse intensity, wat aangegeven wordt door de randvoorwaarde.

In tegenstelling tot de reduced incident intensity L_{ri} , is er in de meeste praktische gevallen geen analytische oplossing voor de diffuse intensity L_d . De vergelijking zal dus met numerieke methodes opgelost moeten worden. Hierbij zijn er twee manieren waarop het probleem kan geformuleerd worden. De eerste formulering combineert de zojuist gegeven differentiaal vergelijking met de bijhorende randvoorwaardes, om zo een algemene integraal vergelijking op te stellen. Dit leidt tot de zogenaamde *Volume rendering equation*. Een korte afleiding van deze formule is terug te vinden in sectie 7.5 van [25].

De tweede manier van formuleren vertrekt rechtstreeks van de differentiaal vergelijkingen en berekent hiervoor een algemene oplossing. Op deze algemene oplossing worden dan de randvoorwaardes toegepast om zo tot een concrete oplossing te komen. Deze formulering wordt gebruikt bij de *diffusion approximation* die in de volgende sectie wordt besproken.

3.3 Diffusion approximation

De diffusion approximation is een benadering voor de vergelijking van de diffuse intensity L_d . Bij deze benadering wordt er verondersteld dat de verstrooiing van het licht in alle richtingen vrij uniform gebeurt. Dit wil dus zeggen dat de hoek waarin het licht wordt verstrooid zo goed als onafhankelijk is van de invalsricting. Deze veronderstelling geldt alleen in een medium dat *highly scattering* is. Een highly scattering medium is een medium met een hoog albedo en waarin er voldoende deeltjes aanwezig zijn, zodat het licht sterk verstrooid wordt².

De verstrooiing is niet volledig onafhankelijk van de richting, aangezien de lichtflux anders constant zou zijn. Dit zou als gevolg hebben dat er geen energie transport kan plaatsvinden. Daarom wordt er meestal vanuit gegaan dat de diffuse intensity iets groter is in de richting van $F_d(x)$. $F_d(x)$ is een vector die aanduidt hoeveel en in welke richting er het meeste energie (in de vorm van radiantie) vloeit in het punt x . Doordat het licht vrij uniform verstrooid wordt kan de diffuse intensity L_d benaderd worden door de eerste twee termen van de Taylor-reeks van L_d volgens ω . Dit geeft als diffusion approximation voor de diffuse intensity $L_d(x)$:

$$L_d(x, \omega) \approx U_d(x) + \frac{3}{4\pi} F_d(x) \omega \quad (3.15)$$

$$U_d(x) = \frac{1}{4\pi} \int_{S^2} L_d(x, \omega) d\omega \quad (3.16)$$

$$F_d(x) = \int_{S^2} L_d(x, \omega) \omega d\omega \quad (3.17)$$

Hierbij stelt $U_d(x)$ de gemiddelde diffuse intensity L_d voor in het punt x , die proportioneel is met de energiedichtheid in x . Een afleiding van vergelijking 3.12 in termen van $U_d(x)$ en $F_d(x)$ is terug te vinden in [48, 18, 26, 25]. Hierbij wordt vergelijking 3.12 geïntegreerd over alle richtingen, waardoor de bekomen vergelijking in termen van U_d en F_d kan geschreven worden. In de bekomen vergelijking kan dan vergelijking 3.15 ingevuld worden. De uiteindelijke vergelijking³ die hieruit volgt is dan:

$$\nabla \cdot \left[\frac{1}{3\sigma_{tr}(x)} \nabla U_d(x) \right] = 4\pi\sigma_a(x)U_d(x) - 4\pi\sigma_s(x)U_{ri}(x) - \int_{S^2} \epsilon(x, \omega) d\omega + \nabla \cdot \left[\frac{1}{3\sigma_{tr}(x)} Q_{ri}(x) \right] \quad (3.18)$$

In deze vergelijking wordt de *reduced extinction coefficient* $\sigma_{tr}(x)$ ⁴ gebruikt als een afkorting voor:

$$\sigma_{tr}(x) = \sigma_a(x) + (1 - g(x))\sigma_s(x)$$

σ_{tr} brengt via de factor $(1 - g)$ de anisotropie van de fase functie in rekening. $g(x)$ is de preferred scattering direction zoals gegeven in vergelijking 3.6. In het rechterlid van vergelijking 3.18 staan er een aantal termen die onafhankelijk zijn van de diffuse intensity. Deze termen worden meestal gegroepeerd onder de noemer *source term* $S(x)$.

²Met een hoog albedo bedoelen we dat voor het albedo $\alpha(x)$ geldt dat: $\alpha(x) \approx 1$.

³Hierbij wordt de notatie uit [18] gebruikt.

⁴Deze term is vergelijkbaar met de *transport cross section* uit [25, 48].

$$S(x) = -4\pi\sigma_s(x)U_{ri}(x) - \int_{S^2} \epsilon(x, \omega)d\omega + \nabla \cdot \left[\frac{1}{3\sigma_{tr}(x)}Q_{ri}(x) \right] \quad (3.19)$$

Hierin stelt $-4\pi\sigma_s(x)U_{ri}(x)$ alle radiantie voor die op het voorwerp invalt en die voor de eerste keer verstrooid wordt in het punt x . De term $\int_{S^2} \epsilon(x, \omega)d\omega = E(x)$ is de hoeveelheid radiantie die in alle richtingen ω , in het punt x , door het voorwerp zelf wordt uitgezonden. $\nabla \cdot \left[\frac{1}{3\sigma_{tr}(x)}Q_{ri}(x) \right]$ stelt de divergentie voor van $Q_{ri}(x)$ in het punt x . In principe stelt $Q_{ri}(x)$ een vector voor die gericht is volgens de richting waarin er de meeste radiantie wordt verstrooid. Hoe groter deze vector, hoe meer radiantie er uit het infinitesimale gebied rond x vloeit. Q_{ri} is dus vergelijkbaar met F_d . Q_{ri} zelf is hier gedefinieerd als een combinatie van de equivalent source function en de source function:

$$Q_{ri} = \int_{S^2} \epsilon_{ri}(x, \omega)\omega d\omega + \int_{S^2} \epsilon(x, \omega)\omega d\omega \quad (3.20)$$

De factor $\frac{1}{3\sigma_{tr}(x)}$ uit vergelijking 3.18 is de zogenaamde diffusie constante. Deze factor geeft een indicatie van de snelheid waarmee er radiantie wordt uitgewisseld met de omgeving rond x . Hoe groter deze waarde hoe meer radiantie er uitgewisseld wordt.

Aangezien de diffusion approximation een benadering is, zal niet meer exact aan randvoorwaarde 3.14 voldaan kunnen worden. De diffusion approximation veronderstelt dat L_d een eenvoudige hoek verdeling heeft, die bijna isotropisch is. Randvoorwaarde 3.14 veronderstelt echter een sterk anisotropisch karakter op de grenspunten x_b . Daarom zal er voor de diffusion approximation ook een benaderende randvoorwaarde opgesteld moeten worden. Een mogelijke benadering van de randvoorwaarde is dat de totale, naar binnen gerichte, diffuse flux nul is:

$$\int_{2\pi} L_d(x_b, \omega)(\omega \cdot n_b)d\omega = 0$$

Hierbij stellen de punten x_b de boundary points voor. Dit zijn de punten die op de rand van het beschouwde domein liggen. In ons geval zijn dit dus de punten die op het oppervlak van het voorwerp liggen. n_b is de normaal in deze boundary points en is hierbij naar binnen gericht. De 2π is afkomstig van de halve bol (*hemisphere*) die gericht is volgens deze normaal. De nieuwe randvoorwaarde kan ook in termen van U_d uitgedrukt worden [25]. Dit geeft voor de gebruikte notatie:

$$U_d(x_b) - \frac{2}{3\sigma_{tr}(x)} \frac{\partial U_d(x_b)}{\partial n} + \frac{2}{3\sigma_{tr}(x)} (Q_1(x_b) \cdot n_b) = 0 \quad (3.21)$$

Het minteken voor $\frac{\partial U_d(x_b)}{\partial n}$ is een gevolg van het feit dat er verondersteld wordt dat de normalen n_b niet van het object weg zijn gericht maar naar het object toe zijn gericht. De term Q_1 wordt gegeven door:

$$Q_1(x) = \frac{1}{\sigma_{tr}} \int_{S^2} [\epsilon_{ri}(x, \omega) + \epsilon(x, \omega)] \omega d\omega \quad (3.22)$$

We hebben nu een partiële differentiaal vergelijking voor de gemiddelde diffuse intensity $U_d(x)$ en een bijhorende randvoorwaarde. Een oplossing voor $U_d(x)$ kan bekomen worden door de differentiaal vergelijking numeriek op te lossen.

3.4 Oplossingsmethodes voor subsurface scattering

De diffusion approximation uit de vorige sectie is niet de enige methode die kan worden gebruikt om het probleem van subsurface scattering te modelleren. In de volgende secties zullen een aantal methodes overlopen worden die gebruikt worden om het fenomeen van subsurface scattering te modelleren.

3.4.1 Diffuse reflectie

In de computer graphics wordt subsurface scattering in de meeste gevallen vaak herleidt tot een simpele diffuse reflectie. Een diffuse reflectie voldoet aan de cosinuswet van Lambert. Bij een diffuus oppervlak is de uitgaande radiantie in alle kijkrichtingen hetzelfde. Dit fenomeen kan echter geen gevolg zijn van een pure reflectie in het oppervlaktepunt. Het kan o.a. wel verklaard worden door lichtverstrooiing onder het oppervlak. Het licht dat op een punt van een oppervlak invalt, zal onder dit oppervlak verstrooid worden. Hierdoor zal het licht het oppervlak terug verlaten in willekeurige richtingen, waardoor het diffuse gedrag bekomen wordt.

Dit is echter enkel een redelijke benadering als de lichtverstrooiing zeer lokaal is. Bij highly-scattering materialen zal deze benadering dus niet meer correct zijn.

3.4.2 Reflection from Layered Surfaces

Een methode die een betere benadering geeft voor subsurface scattering is *Reflection from Layered Surfaces* [21]. In deze methode wordt er een BRDF opgesteld die de richtingsafhankelijkheid van de lichtverstrooiing in rekening brengt. Hierbij wordt er verondersteld dat het voorwerp uit verschillende vlakke, homogene lagen bestaat die uniform belicht worden. Hierdoor blijft de toepasbaarheid beperkt tot voorwerpen die aan deze eigenschappen voldoen, zoals bijvoorbeeld bladeren. Deze methode houdt ook geen rekening met het lichttransport, als gevolg van subsurface scattering, dat optreedt tussen verschillende punten van het oppervlak.

3.4.3 Monte Carlo

Een veel gebruikte techniek bij het renderen van realistische beelden is de Monte Carlo techniek [28, 44, 9]. Het voordeel van Monte Carlo methodes is dat ze vrij eenvoudig te implementeren zijn, weinig geheugen verbruiken en dat ze kunnen gebruikt worden om complexe problemen op te lossen. Monte Carlo methodes zijn gebaseerd op het willekeurig bemonsteren van kansverdelingsfuncties. Een nadeel dat hieruit volgt is dat de variantie van de methode vrij groot is, waardoor de convergentie traag is. Deze variantie kan echter verlaagd worden door het bemonsteren meer doordacht te doen.

Bij globale belichting wordt de Monte Carlo methode vaak gebruikt in combinatie met *Photon Mapping* [27, 30]. In een eerste fase wordt er dan een photon map opgesteld door fotonen te genereren vanuit de lichtbronnen. De fotonen worden dan in de photon map opgeslaan als ze op een oppervlak inslaan. In een tweede fase wordt de scène dan gerenderd m.b.v. een Monte Carlo path tracing algoritme dat gebruik maakt van de photon map. Door gebruik te maken van een extra volume photon map, kan deze methode ook gebruikt worden om subsurface scattering te modelleren. In deze volume photon map worden de fotonen opgeslaan wanneer ze interageren met het medium [31].

3.4.4 BSSRDF

In [26] wordt er een *Bidirectional Surface Scattering Distribution Function* (BSSRDF) model opgesteld dat subsurface scattering in rekening brengt. Dit is een analytisch model dat vergelijkbaar is met de BRDF. Het model is in principe alleen geldig voor oneindige vlakken en homogene materialen. Aangezien de lichtsterkte exponentieel afneemt, is het BSSRDF model toch vrij correct voor voorwerpen die bestaan uit vlakke oppervlakken. Door de afstand waarop de (virtuele) lichtbronnen bij het dipool model worden geplaatst aan te passen, kan er ook een vrij nauwkeurige benadering bekomen worden voor voorwerpen met gebogen oppervlakken.

Het BSSRDF model maakt gebruik van de *Dipole Approximation*. Een *dipole* bestaat uit twee puntlichtbronnen die dicht bij het beschouwde oppervlak worden geplaatst op het punt waar het licht invalt. Eén lichtbron wordt onder het oppervlak geplaatst, de andere erboven. De lichtbronnen worden zo geplaatst dat er wordt voldaan aan de randvoorwaarden aan het oppervlak. De dipole approximation laat toe om de *diffuse reflectance* R_d analytisch te berekenen.

$$R_d(r) = \frac{\alpha'}{4\pi} \left[(\sigma_{tr}d_r + 1) \frac{e^{-\sigma_{tr}d_r}}{\sigma'_t d_r^3} + z_v (\sigma_{tr}d_v + 1) \frac{e^{-\sigma_{tr}d_v}}{\sigma'_t d_v^3} \right] \quad (3.23)$$

Hierbij is r de afstand tussen het punt x_i waar het licht invalt op het oppervlak en het punt x_o waar het licht het oppervlak verlaat. Voor een beschrijving van de verschillende symbolen verwijs ik door naar [26]. R_d kan gezien worden als de hoeveelheid radiantie van het licht dat in x_i invalt, die in het punt x_o het oppervlak verlaat. Vergelijking 3.23 wordt gebruikt om de diffuse component S_d van het BSSRDF model S te berekenen.

$$S(x_i, \omega_i, x_o, \omega_o) = S_d(x_i, \omega_i, x_o, \omega_o) + S^{(1)}(x_i, \omega_i, x_o, \omega_o) \quad (3.24)$$

$$S_d(x_i, \omega_i, x_o, \omega_o) = \frac{1}{\pi} F_t(\eta, \omega_i) R_d(r) F_t(\eta, \omega_o) \quad (3.25)$$

De functie F_t in vergelijking 3.25 brengt de *Fresnel* refractie in rekening die aan het oppervlak kan optreden. $S^{(1)}$ is de *single scattering term* zoals beschreven in [21] die de richtingsafhankelijkheid in rekening brengt. In highly scattering materialen is de invloed van de single scattering term veel kleiner dan de diffuse term. Hierdoor kan de single scattering term in dit geval genegeerd worden zonder dat er al te grote fouten worden gemaakt.

Sinds het BSSRDF model werd voorgesteld door Jensen et al., is het in een groot aantal technieken toegepast. Een aantal van deze technieken worden beschreven in [29, 35, 39, 53].

3.4.5 Multigrid

In [48] stellen Stam et al. twee numerieke methodes voor om subsurface scattering via de diffusie vergelijking op te lossen. Eén van deze methodes gebruikt de *Finite Differencing Method* om de vergelijking te discretiseren. Hierbij wordt er gebruik gemaakt van de multigrid methode om de convergentie van de oplossing te versnellen. De andere methode gebruikt de *Finite Element Method*, waarbij men de zogenaamde *blobs* gebruikt om het radiantieveld voor te stellen. In het volgende hoofdstuk wordt er meer uitleg gegeven over deze verschillende numerieke methodes. In hun paper gebruiken Stam et al. de multigrid methode enkel op een 2D domein, waarbij er een

homogeen materiaal wordt verondersteld om de rekentijd te beperken. In het drie-dimensionale geval zou volgens hen het geheugengebruik en de rekentijd te groot worden.

Haber et al. stellen in [19] ook een multigrid methode voor. In deze methode maakt men gebruik van een *Adaptive Mesh Refinement* algoritme om het geheugengebruik van de mesh niet onnodig te laten toenemen. Anderzijds maakt men gebruik van de *Embedded Boundary Discretisation* methode om een nauwkeurigere benadering te bekomen op de rand van het object. Met hun methode kan subsurface scattering bij complexe 3D objecten in een paar minuten berekend worden. Hierbij mag het materiaal van het object zowel homogeen als heterogeen zijn.

Hoofdstuk 4

Wiskundige methodes

In het vorige hoofdstuk werd de diffusion approximation besproken. De vergelijking die uit deze benadering volgt is een partiële differentiaal vergelijking. In de meeste gevallen is deze vergelijking niet analytisch op te lossen. Daarom zal de vergelijking numeriek opgelost moeten worden. PDE's komen voor in vrijwel alle wetenschappelijke domeinen. Ook hier moet men de bekomen PDE's meestal numeriek oplossen. Het numeriek oplossen van PDE's is dus een veel voorkomend probleem dat bijgevolg uitgebreid bestudeerd is. Hierdoor zijn er een groot aantal numerieke methodes ontwikkeld om PDE's op te lossen. In dit hoofdstuk zullen een aantal van deze methodes kort besproken worden.

4.1 Discretisatie en Stelsel van vergelijkingen

Meestal is de PDE die men moet oplossen een continue functie die gedefinieerd is in een bepaald domein Ω . Op alle punten van dit domein moet er voldaan worden aan de PDE. Het oplossen van de PDE komt in de meeste gevallen neer op het oplossen van een stelsel van (lineaire) algebraïsche vergelijkingen. Dit stelsel van vergelijkingen kan algemeen in matrixvorm geschreven worden als volgt:

$$\mathbf{Ax} = \mathbf{b} \tag{4.1}$$

Hierbij stelt \mathbf{A} een $N \times N$ matrix voor, waarbij N het aantal punten is die moeten voldoen aan de vergelijkingen. In het continue geval zijn er een oneindig aantal punten, waardoor $N = \infty$. Met als gevolg dat de matrix \mathbf{A} ook oneindig groot zal zijn, en dus praktisch niet op te lossen is. Daarom zal men bij het oplossen van een PDE altijd eerst het beschouwde domein Ω gaan discretiseren. Discretiseren komt grofweg neer op het selecteren van een beperkt aantal punten $\{x_i\}^1$ uit het continue domein. De positie en het aantal punten dat gekozen wordt, is afhankelijk van de gebruikte oplossingsmethode.

Om een stelsel zoals 4.1 op te lossen, kunnen er ook weer verschillende technieken toegepast worden. Eén manier is door gebruikt te maken van relaxatie. Deze methode wordt kort beschreven in § 4.6. Een andere manier is bijvoorbeeld om LU-decompositie te gebruiken om de matrix \mathbf{A} te inverteren en zo de gezochte waardes \mathbf{x} te berekenen.

¹ $i = 1, \dots, N$ waarbij N eindig is.

4.2 Finite Difference Method

De *Finite Difference Method* (FDM) of eindige-differentiemethode is een vrij eenvoudige techniek die veel wordt gebruikt om PDE's op te lossen [47, 1]. De FDM maakt gebruik van differentie quotiënten om de afgeleiden te benaderen. De differentie quotiënten zijn eigenlijk een beperkt deel van de Taylor serie van de afgeleiden. De *Euler* methode is een simpel voorbeeld van een FDM. Hierbij wordt de waarde van de afgeleide van een eenvoudige 1D functie $u'(x)$ gegeven door:

$$u'(x) \approx \frac{u(x+h) - u(x)}{h}$$

In deze vergelijking stelt h de stapgrootte voor. De stapgrootte kan gezien worden als de afstand tot het volgende discrete punt. Deze stapgrootte wordt bepaald door de gekozen grid, ook wel *mesh* genoemd, die gebruikt wordt om het domein te discretiseren. In het geval van een 3D domein, gebruikt men meestal een kubische grid zodat h overal hetzelfde is.

Het nadeel van deze methode is dat de nauwkeurigheid vooral afhangt van de gekozen stapgrootte h . Hoe meer punten men gebruikt in de grid en dus hoe kleiner de stapgrootte h , hoe nauwkeuriger de oplossing. Om domeinen met afgeronde grenzen en scherpe hoeken voldoende nauwkeurig te kunnen benaderen, moet h vrij klein gekozen worden. Hierdoor zullen er veel punten in de grid nodig zijn, waardoor het geheugengebruik sterk toeneemt en de rekentijd vergroot. Een manier om dit probleem enigszins te verminderen is het gebruik van een *adaptive mesh refinement* techniek. Hierbij wordt er enkel op plaatsen waar het nodig is een fijnere grid gebruikt, zodat het geheugengebruik beperkt wordt.

4.3 Finite Element Method

De *Finite Element Method* (FEM) of eindige-elementenmethode is een methode om *Boundary Value Problems* (BVP) op te lossen [57, 4]. Een partiële differentiaal vergelijking met zijn bijhorende randvoorwaardes is een BVP. De FEM wordt veel gebruikt voor de sterkteberekeningen van constructies. Het basisidee van de FEM is dat men een complex probleem opdeelt in een (groot) aantal kleine delen, elementen genoemd, waar het probleem wel gemakkelijk op te lossen is. Achteraf kan men dan de oplossing van deze elementen combineren om zo een oplossing voor het volledige systeem te bekomen.

Bij de FEM discretiseert men het domein dus door het op te delen in verschillende kleine elementen. Bijvoorbeeld in het 2D geval deelt men het domein op in driehoeken of rechthoeken. De oplossing van de PDE wordt dan in ieder element benaderd door gebruik te maken van veeltermen, zogenaamde basisfuncties. Om een nauwkeurige benadering te bekomen, moet men de elementen voldoende klein kiezen. Als er kleinere en dus ook meer elementen worden gebruikt, zal logischerwijze de rekentijd toenemen.

De eindige-elementenmethode vormt ook de basis van de *radiosity* methode. De radiosity methode is een methode om bij globale belichting de diffuse belichting te berekenen [17, 10, 8].

4.4 Boundary Element Method

Nog een andere methode om PDE's op te lossen is de *Boundary Element Method* (BEM) [16, 34]. Bij deze methode wordt de op te lossen partiële differentiaal vergelijking omgevormd tot een integraal die wiskundig gezien equivalent is. Bij het omvormen bekomt men enerzijds een integraal vergelijking die gedefinieerd is op de rand van het domein en een integraal die de oplossing van de randpunten relateert aan de oplossing voor de interne punten. De omvorming gebeurt a.d.h.v. *het tweede theorema van Green*.

$$\int_V \phi \nabla^2 \psi dV = \oint_{\partial V} (\phi \nabla \psi) \cdot dS - \int_V \nabla \phi \cdot \nabla \psi dV$$

De integraalvorm kan echter niet voor alle soorten PDE's bekomen worden, wat wel een nadeel is. De BEM heeft ook problemen bij het oplossen van niet-homogene PDE's. Dit probleem wordt opgelost in de *Dual Reciprocity BEM* [43, 2]. Het grote voordeel van de BEM is dat enkel de rand van het domein gediscetiseerd moet worden. Bij de FEM en FDM moest steeds het volledige domein gediscetiseerd worden. Doordat enkel de rand van het domein moet gediscetiseerd worden is er één dimensie minder nodig om het probleem op te lossen. Hierdoor wordt bijvoorbeeld een 3D probleem gereduceerd tot een 2D probleem.

4.5 Boundary Knot Method

De *Boundary Knot Method* (BKM) is een numerieke methode die gebruikt maakt van radiale basisfuncties [6]. Het grote voordeel van deze methode is dat ze volledig meshless is. Er moet dus geen grid gegenereerd worden, wat bij de vorige methodes wel het geval was. De BKM is een combinatie van de *Dual Reciprocity Method* (DRM), *Radial Basis Functions* (RBF) en het gebruik van een *non-singular general solution*. Deze methode wordt meer in detail besproken in § 5.3.

4.6 Relaxatie

Relaxatie is een iteratieve methode om een stelsel van vergelijkingen op te lossen. Hierbij wordt de matrix \mathbf{A} uit 4.1 opgesplitst als volgt:

$$\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U} \tag{4.2}$$

Hierbij zijn \mathbf{L} , \mathbf{D} en \mathbf{U} respectievelijk de strikte benedendriehoeksmatrix, de diagonaalmatrix en de strikte bovendriehoeksmatrix. De matrix \mathbf{A} wordt hierbij meestal ijl verondersteld. Iteratieve methodes maken gebruik van eenvoudige operaties om een schatting te bekomen van de oplossing. In iedere iteratie wordt er gebruik gemaakt van de vorige schatting om zo een nieuwe en betere schatting van de oplossing te bekomen. In iedere iteratiestap k , convergeert de geschatte oplossing geleidelijk aan naar de exacte oplossing. Als na een iteratiestap k de oplossing voldoende geconvergeerd is, stopt men het itereren. Om de convergentie te controleren, zou op het eerste zicht gebruik gemaakt kunnen worden van de foutwaarde $\mathbf{e}^k = \mathbf{x} - \mathbf{x}^k$. \mathbf{e}^k geeft het verschil tussen de exacte oplossing \mathbf{x} en de geschatte oplossing \mathbf{x}^k bij de k -de iteratie.

Het probleem hierbij is echter dat de exacte oplossing \mathbf{x} juist de onbekende is. Een betere manier om de convergentie te testen, is door \mathbf{x}^k in de vergelijking 4.1 in te vullen en zo de restwaarde (residu) \mathbf{r}^k te berekenen.

$$\mathbf{r}^k = \mathbf{b} - \mathbf{A}\mathbf{x}^k$$

Als de restwaarde \mathbf{r}^k binnen een opgegeven foutmarge ligt, zal de oplossing voldoende nauwkeurig benaderd zijn. De relaxatie van de matrix kan op verschillende manieren gebeuren. In wat volgt zullen een aantal methodes kort besproken worden.

4.6.1 Jacobi

De *Jacobi* methode is de meest eenvoudig relaxatie methode. De matrix definitie ervan is als volgt:

$$\mathbf{x}^{k+1} = \mathbf{D}^{-1}[\mathbf{b} - (\mathbf{L} + \mathbf{U})\mathbf{x}^k] \quad (4.3)$$

De Jacobi methode beschouwt elk van de N vergelijkingen waaruit het stelsel 4.1 is opgebouwd, apart. Voor elke i -de vergelijking wordt een oplossing gezocht voor de bijhorende waarde x_i . Hierbij worden de andere waarden van \mathbf{x} constant verondersteld. In de notatie in matrixelementen geeft dit dan voor x_i :

$$x_i^{k+1} = \frac{b_i - \sum_{j \neq i} a_{ij} x_j^k}{a_{ii}} \quad (4.4)$$

De Jacobi methode vereist dat alle elementen op de diagonaal van \mathbf{A} verschillend zijn van nul. De convergentie van deze methode wordt ook enkel gegarandeerd als de matrix diagonaal dominant² is of symmetrisch en *positive definite*.

4.6.2 Gauss-Seidel

In tegenstelling tot de Jacobi methode worden de N vergelijkingen van het stelsel 4.1 in de *Gauss-Seidel* methode, niet per iteratie apart beschouwd. In de Gauss-Seidel methode worden de N vergelijkingen per iteratie sequentieel doorlopen, waarbij de vorige berekende resultaten direct gebruikt worden als ze beschikbaar zijn. Als ze nog niet beschikbaar zijn, wordt de waarde uit de vorige iteratie gebruikt. In matrix vorm geeft dit:

$$\mathbf{x}^{k+1} = (\mathbf{D} - \mathbf{M})^{-1}(\mathbf{R}\mathbf{x}^k + \mathbf{b}) \quad (4.5)$$

Hierbij is $\mathbf{M} = -\mathbf{L}$ en $\mathbf{R} = -\mathbf{U}$. Doordat de berekeningen sequentieel gebeuren, zullen de waarden van \mathbf{x}^k afhankelijk zijn van de volgorde waarin de vergelijkingen worden doorlopen. De vergelijkingen kunnen dus niet meer parallel uitgerekend worden, wat wel het geval was bij de Jacobi methode. De convergentiesnelheid van de Gauss-Seidel methode is wel beter dan deze van de Jacobi methode. Er worden dezelfde eisen aan de matrix \mathbf{A} gesteld als bij de Jacobi methode. De notatie in matrixelementen geeft voor Gauss-Seidel:

$$x_i^{k+1} = \frac{b_i - \sum_{j < i} a_{ij} x_j^{k+1} - \sum_{j > i} a_{ij} x_j^k}{a_{ii}} \quad (4.6)$$

² $|a_{ii}| > \sum_{i \neq j} |a_{ij}|$

4.6.3 Successive Overrelaxation

De *Successive Overrelaxation* methode kan gezien worden als een veralgemening van de Gauss-Seidel methode. In deze methode wordt er een gewogen gemiddelde genomen tussen de waarde uit de vorige iteratie en de berekende Gauss-Seidel waarde in de huidige iteratie. In matrixvorm geeft dit dan:

$$\mathbf{x}^{k+1} = (\mathbf{D} - \alpha\mathbf{M})^{-1}[\alpha\mathbf{R} + (1 - \alpha)\mathbf{D}]\mathbf{x}^k + \alpha(\mathbf{D} - \alpha\mathbf{M})^{-1}\mathbf{b} \quad (4.7)$$

Hierbij zijn \mathbf{M} en \mathbf{R} weer gedefinieerd als $\mathbf{M} = -\mathbf{L}$ en $\mathbf{R} = -\mathbf{U}$. Het gewicht wordt bepaald door een relaxatiefactor α . Deze factor moet in het interval $[0, 2]$ liggen om de oplossing te laten convergeren. Als we α gelijk nemen aan 1 dan bekomen we de gewone Gauss-Seidel methode. Om een snellere convergentie te bekomen dan bij de Gauss-Seidel methode, moet α groter dan 1 gekozen worden. Door α optimaal te kiezen, zal de snelste convergentie bekomen worden. Een nadeel echter is dat deze optimale waarde alleen uitgerekend kan worden als er een analytische oplossing is voor het gegeven probleem. Als er een gelijkaardig probleem is dat analytisch opgelost kan worden, dan kan men de α benaderen door de α waarde van dat analytische probleem. De matrixelementen notatie is hier:

$$x_i^{k+1} = \alpha \frac{b_i - \sum_{j < i} a_{ij} x_j^{k+1} - \sum_{j > i} a_{ij} x_j^k}{a_{ii}} + (1 - \alpha)x_i^k \quad (4.8)$$

4.6.4 Multigrid

De methodes die tot nu toe besproken zijn convergeren in het begin vrij snel omdat de restwaardes dan hoogfrequent zijn. Maar nadien neemt de convergentiesnelheid af, aangezien de restwaardes dan laagfrequent zijn. Om dit probleem op te lossen kan men een *Multigrid* methode toepassen. Het basis idee van een multigrid methode is dat men een hiërarchische structuur van grovere en fijnere grids gebruikt i.p.v. één vaste grid. De multigrid methode bestaat uit drie stappen: *relax*, *coarsen* en *refine*. Deze stappen vormen een v-cyclus die herhaald wordt tot de gewenste convergentie bereikt is.

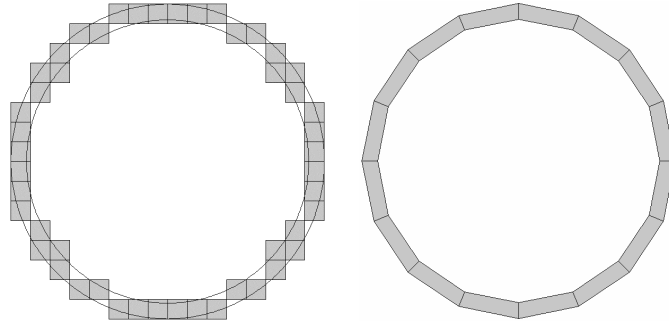
In de relax-stap wordt er een vast aantal iteraties van een relaxatie algoritme, bijvoorbeeld Gauss-Seidel, uitgevoerd op een bepaald grid-niveau. In de coarsen-stap worden dan de bekomen restwaardes uit de de relax-stap geprojecteerd naar een grovere grid. Op deze grid wordt dan opnieuw een vast aantal keren geïtereerd. Indien er nog een grovere grid is, worden de bekomen restwaardes opnieuw geprojecteerd naar deze grovere grid en voert men ook weer een aantal iteraties uit. De benadering die op de meest grove grid bekomen wordt, wordt dan in een refine-stap geïnterpoleerd naar de onderliggende fijnere grid. Hier wordt dan terug een aantal keren geïtereerd, waarna de benadering weer naar de eventuele fijnere grid wordt geïnterpoleerd. Door gebruik te maken van meerdere grids zal de convergentiesnelheid groter zijn dan bij gewone relaxatie. Voor meer informatie over multigrid methodes verwijs ik door naar [20].

4.7 Andere methodes

De verschillende methodes die hier kort besproken werden, zijn zeker niet de enige. Door de jaren heen zijn er vele numerieke methodes ontwikkeld waarvan *Method of Fundamental Solution*, *Finite Volume Method*, *Hybrid FEM-BEM*, *Embedded Boundary Method*, *Spectral Methods* en *Monte Carlo* nog een aantal voorbeelden zijn.

4.8 Vergelijking

In vergelijking met de andere methodes is de FDM, wiskundig gezien, de meest eenvoudige methode. Het nadeel van de FDM is dat het vaak moeilijk is een gegeven domein voldoende nauwkeurig te benaderen. Dit is een gevolg van de eenvoudige grid die gebruikt wordt bij de FDM. Figuur 4.1 geeft dit duidelijk weer. Om de nauwkeurigheid bij de FDM te verhogen, moet er een kleinere grid gekozen worden waardoor het rekenwerk en dus ook de rekestijd sterk zal toenemen.



Figuur 4.1: Benadering van het domein door FDM en FEM [46]

Zoals in figuur 4.1 ook te zien is, kan de FEM met een kleiner aantal elementen het domein beter benaderen dan de FDM. Door gebruik te maken van elementen i.p.v. een vaste grid kan de FEM nauwkeurigere resultaten bekomen dan de FDM. Het discretiseren van het domein in elementen is echter niet altijd even voor de hand liggend. Voor domeinen met meerdere dimensies zal het discretiseren van het domein in elementen redelijk wat rekestijd in beslag nemen. In gebieden in het domein waar er veel variatie is, zullen de elementen klein genoeg gekozen moeten worden om correcte resultaten te bekomen. Hierdoor zal de rekestijd ook weer toenemen. Het grote voordeel van de FEM is dat de methode door zijn algemeen karakter een zeer groot toepassingsgebied heeft.

De BEM heeft als voordeel t.o.v. de FEM dat enkel de rand van het domein moet gediscetiseerd worden. Hierdoor vermindert de dimensionaliteit van het probleem met één dimensie, waardoor de rekestijd ook vermindert. Bijvoorbeeld een 3D object moet niet meer volledig opgedeeld worden in volume elementen, maar enkel het 2D oppervlak moet opgedeeld worden. Een beperking van de BEM is dat de methode minder goed overweg kan met niet-homogene problemen, zoals bijvoorbeeld tijdsafhankelijke problemen. In vergelijking met de FEM en FDM is de BEM ook, wiskundig gezien, ingewikkelder. De matrices bij de BEM zijn volle matrices in tegenstelling tot de ijle matrices bij de FEM en de FDM. Maar de matrices bij de BEM zijn wel kleiner.

Een nadeel van zowel de FEM en de BEM is dat het genereren van de mesh meestal rekenintensief is. Meshless methodes, zoals de MFS en de BKM, zijn methodes waarvoor er geen mesh moet gegenereerd worden. Hierdoor kan er redelijk wat rekestijd bespaard worden. De meshless methodes hebben echter ook hun nadelen. Bijvoorbeeld bij de MFS moet er een artificiële grens opgesteld worden buiten het gegeven domein om singulariteiten te voorkomen. De BKM heeft dit probleem niet. Een nadeel van de meshless methodes, zoals de BKM, is dat het wiskundige toepassingsgebied beperkter is dan dat van de BEM en de FEM. M.a.w. de vergelijking waarvoor een oplossing wordt gezocht, moet aan strengere wiskundige voorwaarden voldoen.

Hoofdstuk 5

Radiale basisfuncties

Radiale basisfuncties (*Radial Basis Function*, RBF) worden pas sinds de laatste jaren gebruikt in de computer graphics. In andere wetenschappelijke domeinen zijn ze echter al langer bekend. Radiale basisfuncties werden oorspronkelijk voornamelijk gebruikt in interpolatie methodes [22]. Later werd aangetoond dat radiale basisfuncties ook gebruikt konden worden om partiële differentiaal vergelijkingen op te lossen [33]. In het eerste deel van dit hoofdstuk zullen radiale basisfuncties algemeen besproken worden. In het tweede deel zal bekeken worden hoe men radiale basisfuncties kan gebruiken om partiële differentiaal vergelijkingen op te lossen. In het laatste deel wordt de Boundary Knot Method meer in detail besproken.

5.1 Radiale basisfuncties

Een radiale basisfunctie $\phi_j(x) = \phi(\|x - x_j\|)$ is een reële functie waarvan de waarde in een punt x alleen afhankelijk is van de afstand van een vast punt $x_j \in \mathbb{R}^d$ tot het punt $x \in \mathbb{R}^d$. Hierbij duidt de index d op het aantal dimensies. Als afstand $r = \|x - x_j\|$ gebruikt men meestal de gewone Euclidische afstand $\sqrt{(r_1)^2 + \dots + (r_d)^2} = \sqrt{(x_1 - x_{j1})^2 + \dots + (x_d - x_{jd})^2}$. Een radiale basisfunctie heeft ook de eigenschap dat de functie radiaal symmetrisch (rotatie-onafhankelijk) is t.o.v. het punt x_j .

Als we beschikken over een verzameling van N datapunten $\{x_i, f(x_i)\}_{i=1}^N$, dan kunnen we de bijhorende (onbekende) functie $f(x)$ globaal benaderen m.b.v. radiale basisfuncties. Dit gebeurt door op elk punt x_i van de gegeven dataset een radiale basisfunctie te plaatsen.

$$\tilde{f}(x) = \sum_{j=1}^N \alpha_j \phi(\|x - x_j\|) = \sum_{j=1}^N \alpha_j \phi_j \quad (5.1)$$

De functie $\tilde{f}(x)$ is een benaderende functie van de functie $f(x)$. Ieder punt x_i uit de gegeven dataset wordt in de som gebruikt als middelpunt (x_j) van een radiale basisfunctie. α_j is het gewicht dat op ieder punt x_j aan de radiale basisfunctie wordt toegekend. De waarde van dit gewicht is echter een onbekende en zal dus berekend moeten worden. Aangezien $\tilde{f}(x)$ een benadering is van $f(x)$ moet voldaan zijn aan de volgende voorwaarde:

$$\tilde{f}(x_i) = f(x_i) \quad \text{waarbij} \quad i = 1, 2, \dots, N \quad (5.2)$$

Aan de hand van voorwaarde 5.2 kan een systeem van lineaire vergelijkingen opgesteld worden. In matrix vorm geeft dit:

$$\mathbf{\Phi} \boldsymbol{\alpha} = \mathbf{f} \quad (5.3)$$

Hierbij zijn $\mathbf{f} = (f_1, f_2, \dots, f_N)^T$ en $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_N)^T$ vectoren die respectievelijk de functiewaardes en de onbekende gewichten voorstellen. De matrix $\mathbf{\Phi}$ wordt gegeven door:

$$\mathbf{\Phi} = \begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1N} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{N1} & \phi_{N2} & \cdots & \phi_{NN} \end{bmatrix} \quad \text{waarbij} \quad \phi_{ij} = \phi(\|x_i - x_j\|)$$

De matrix $\mathbf{\Phi}$ is een symmetrische matrix. Afhankelijk van de gekozen radiale basisfunctie kan er ook voor gezorgd worden dat de matrix *positive definite* ($\mathbf{\Phi} > 0$) is, waardoor er altijd een oplossing zal bestaan [41, 56]. Voor sommige radiale basisfuncties moet er echter een extra veelterm $P(x)$ worden toegevoegd om te verzekeren dat de bekomen matrix positive definite is, dit geeft:

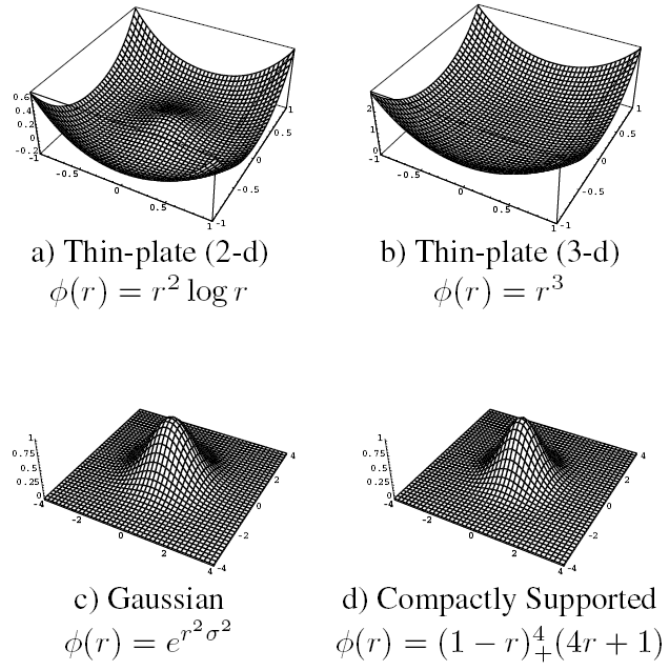
$$\tilde{f}(x) = \sum_{j=1}^N \alpha_j \phi(\|x - x_j\|) + P(x) \quad \tilde{\mathbf{\Phi}} = \begin{bmatrix} \mathbf{\Phi} & \mathbf{P} \\ \mathbf{P}^T & 0 \end{bmatrix}$$

Hier zal echter in deze thesis niet verder op worden ingegaan. Voor meer informatie verwijst ik door naar [56].

Tot nu toe hebben we nog niks gezegd over de vorm van de radiale basisfuncties zelf. Zoals eerder al gezegd is een radiale basisfunctie een functie die enkel gebruik maakt van een afstands-waarde r . Tabel 5.1 geeft een overzicht van de meest gebruikte radiale basisfuncties. Een aantal van deze functies worden weergegeven in figuur 5.1.

$\Phi(r)$	Benaming
r	lineair
r^3	kubisch
$r^2 \log r$	thin-plate
$e^{-\frac{r^2}{c^2}}$	Gaussiaans
$(r^2 + c^2)^{\frac{\beta}{2}}$	Multiquadratisch
$(r^2 + c^2)^{-\frac{\beta}{2}}$	Invers Multiquadratisch
$(1 - r)_+^2$	Compactly supported C^0
$(1 - r)_+^4 (4r + 1)$	Compactly supported C^2
$(1 - r)_+^6 (35r^2 + 18r + 3)$	Compactly supported C^4

Tabel 5.1: Verschillende soorten radiale basisfuncties



Figuur 5.1: Een aantal RBF's grafisch weergegeven [41]

De laatste drie functies uit tabel 5.1 zijn de zogenaamde *compactly supported radial basis functions* [54]. Dit zijn radiale basisfuncties met een lokale basis (support). Deze functies zijn van de vorm:

$$\phi(r) = \begin{cases} p(r) & 0 \leq r \leq 1 \\ 0 & r > 1 \end{cases} \quad (5.4)$$

Zoals uit vergelijking 5.4 blijkt, zijn deze functies vanaf een bepaalde afstand nul. Hierdoor moet er maar een beperkt aantal punten x_i in rekening gebracht worden. Dit heeft een aantal voordelen t.o.v. de gewone radiale basisfuncties. De gewone functies brengen steeds alle datapunten in rekening. Hierdoor hebben deze radiale basisfuncties een globaal karakter. Een voordeel van dit globaal karakter is dat bij het interpoleren de gaten in een dataset vrij goed opgevuld worden. Het globaal karakter heeft echter ook verschillende nadelen. Doordat ieder punt in elke radiale basisfunctie wordt gebruikt, zal een verandering in een punt de volledige oplossing beïnvloeden. Door het globale karakter zullen ook lokale details afgezwakt worden. Een ander nadeel is dat de matrix Φ die bekomen wordt via deze radiale basisfuncties, een volle (*dense*) matrix is. Om volle matrices op te lossen is redelijk wat rekenkracht en geheugen nodig. Als het aantal datapunten N toeneemt zal dit voor problemen zorgen.

Doordat compactly supported radiale basisfuncties een lokale basis hebben, zal de matrix Φ een groot aantal nullen bevatten. De matrix Φ is in dit geval dus een ijle (*sparse*) matrix, zodat één van de algoritmes om stelsel met ijle matrices op te lossen, gebruikt kan worden. Stelsels met ijle matrices zijn minder zwaar om uit te rekenen dan stelsels met volle matrices. Compactly supported radiale basisfuncties maken gebruik van de punten die zich in de buurt

bevinden van het middelpunt van de radiale basisfunctie. Om deze punten snel te kunnen vinden, kan de dataset opgeslaan worden in een efficiënte datastructuur zoals bijvoorbeeld een kd-tree. Tabel 5.2 geeft een vergelijking tussen de complexiteit bij het oplossen van stelsels m.b.v. radiale basisfuncties met een globale basis en een lokale basis. Een bijkomend voordeel van de compactly supported radial basisfuncties is dat ze *positive definite* zijn en dus ook geen extra veelterm nodig hebben.

	Globaal	Lokaal
Stelsel opstellen	$O(n^2)$	$O(n \log n)$
Stelsel oplossen	$O(n^2)$	$O(n^{1.5})$
Geheugen	$O(n^2)$	$O(n)$
Evaluatie	$O(n)$	$O(\log n)$
Invloedsgebied van één punt	globaal	lokaal

Tabel 5.2: Verschil tussen een globale en een lokale basis [41]

De compactly supported radiale basisfuncties zijn gedefinieerd voor een eenheidsafstand r . Om de basis van deze radiale basisfuncties uit te breiden kan de afstand met een schaalfactor $1/\delta$ vermenigvuldigd worden. Hierdoor wordt de basis uitgebreid tot een bol met diameter δ . δ zal voor een groot deel bepalen hoe ijel de matrix Φ is. Hoe kleiner deze waarde, hoe ijler de matrix Φ . Maar als de matrix Φ ijler wordt, zal de nauwkeurigheid van de benadering afnemen.

$$\phi^\delta(r) = \phi\left(\frac{r}{\delta}\right) \quad \delta > 0$$

5.2 PDE's oplossen met RBF's

In het vorige deel werd aangetoond hoe radiale basisfuncties gebruikt kunnen worden om een bepaalde functie te benaderen. De vraag is nu echter hoe deze techniek gebruikt kan worden om een partiële differentiaal vergelijking op te lossen. Een simpele methode om PDE's op te lossen m.b.v. radiale basisfuncties is de *Collocation by RBF's* methode [56]. Deze methode heeft een aantal nadelen, maar zal hier kort besproken worden om aan te geven hoe een PDE m.b.v. radiale basisfuncties kan opgelost worden.

Om te beginnen wordt de gegeven dataset $\{x_i\}_{i=1}^N$ van N datapunten opgesplitst in twee delen. Het eerste deel bevat M interne punten $(x_i, i = 1, \dots, M)$. Het tweede deel bevat de $N - M$ resterende randpunten $(x_i, i = M + 1, \dots, N)$.

Vervolgens beschouwen we een algemene PDE met de volgende vorm:

$$L\{u(x)\} = f(x) \quad x \in \Omega \tag{5.5}$$

$$B\{u(x)\} = g(x) \quad x \in \partial\Omega \tag{5.6}$$

In vergelijking 5.5 stelt Ω een domein voor in \mathbb{R}^d . $\partial\Omega$ uit 5.6 duidt de rand van dit domein aan. L en B zijn respectievelijk de interne en de rand operators die gedefinieerd zijn op de functie $u(x)$. De functie $u(x)$ wordt benaderd door $\tilde{u}(x)$ m.b.v. gewone radiale basisfuncties via:

$$\tilde{u}(x) = \sum_{j=1}^N \alpha_j \phi(\|x - x_j\|) \quad (5.7)$$

De punten x_j zijn de N verschillende datapunten in het domein Ω en α_j zijn de te bepalen coëfficiënten. De afgeleiden van de functie $\tilde{u}(x)$ kunnen nu op de volgende manier gedefinieerd worden:

$$\begin{aligned} \frac{\partial \tilde{u}(x)}{\partial x} &= \sum_{j=1}^N \alpha_j \phi_x(\|x - x_j\|) \\ \frac{\partial^2 \tilde{u}(x)}{\partial x^2} &= \sum_{j=1}^N \alpha_j \phi_{xx}(\|x - x_j\|) \end{aligned} \quad (5.8)$$

Hierbij wordt de notatie ϕ_x, ϕ_{xx}, \dots gebruikt om de eerste, tweede, \dots afgeleide van ϕ voor te stellen. We kunnen nu vergelijkingen 5.7 en 5.8 in 5.5 en 5.6 substitueren. Rekening houdend met de N datapunten x_i , kunnen we dan de volgende voorwaardes opstellen:

$$L\{\tilde{u}(x_i)\} = f(x_i) \quad x_i \in \Omega \quad (5.9)$$

$$B\{\tilde{u}(x_i)\} = g(x_i) \quad x_i \in \partial\Omega \quad (5.10)$$

Vergelijkbaar met 5.3 kan nu de volgende matrix opgesteld worden:

$$\hat{\Phi} \alpha = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} \quad \text{waarbij} \quad \hat{\Phi} = \begin{bmatrix} L[\Phi] \\ B[\Phi] \end{bmatrix} \quad (5.11)$$

De delen $L[\Phi]$ en $B[\Phi]$ die samen de matrix $\hat{\Phi}$ vormen zijn gedefinieerd als:

$$\begin{cases} L[\Phi] = L[\phi](x_i - x_j) & i, j = 1, \dots, M \\ B[\Phi] = B[\phi](x_i - x_j) & i, j = M + 1, \dots, N \end{cases} \quad (5.12)$$

Uiteindelijk kan de benaderende functie $\tilde{u}(x)$ bekomen worden door:

$$\tilde{u}(x) = \mathbf{S}^T \hat{\Phi}^{-1} \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} \quad (5.13)$$

\mathbf{S}^T is hierbij gegeven door $\mathbf{S}^T = [\phi(\|x - x_1\|), \phi(\|x - x_2\|), \dots, \phi(\|x - x_N\|)]$. Vergelijking 5.13 is alleen maar geldig als $\hat{\Phi}$ niet-singulier, of m.a.w. inverteerbaar, is. Een nadeel van deze methode is dat deze niet-singulariteit niet gegarandeerd kan worden [12]. In de meeste gevallen is de matrix echter wel inverteerbaar. Er zijn nog een aantal andere nadelen zoals het niet symmetrisch zijn van de resulterende matrix en het feit dat deze matrix meestal *ill-posed*¹ is. Een verbetering van deze methode is de *Hermite-type Collocation* methode waarbij er wel een symmetrische, en dus inverteerbare, matrix bekomen wordt [12].

¹Een ill-posed of slecht gesteld probleem, is een probleem dat geen of meerdere oplossingen kan hebben en waarvan de oplossing sterk afhankelijk kan zijn van de input gegevens.

5.3 Boundary Knot Method

De *Boundary Knot Method* (BKM) is een numerieke methode om PDE's op te lossen en werd ontwikkeld door W. Chen en M. Tanaka [6, 7]. De methode bestaat uit twee delen. Eerst wordt er gezocht naar een benadering van de particuliere oplossing van het probleem. Daarna wordt de homogene oplossing berekend. Als we terug de algemene PDE

$$L\{u(x)\} = f(x) \quad x \in \Omega \quad (5.14)$$

$$B\{u(x)\} = g(x) \quad x \in \partial\Omega \quad (5.15)$$

veronderstellen, kunnen we $u(x) = u_h(x) + u_p(x)$ opdelen in een homogene vergelijking $u_h(x)$ en particuliere vergelijking $u_p(x)$. Deze twee vergelijkingen moeten dan voldoen aan:

$$L\{u_p(x)\} = f(x) \quad (5.16)$$

$$L\{u_h(x)\} = 0 \quad (5.17)$$

$$B\{u_h(x)\} = g(x) - B\{u_p(x)\} \quad (5.18)$$

Zoals uit 5.16 blijkt moet de particuliere vergelijking $u_p(x)$ niet voldoen aan de randvoorwaarden. Een methode die vaak gebruikt wordt om een oplossing voor de particuliere vergelijking te bekomen, is de *Dual Reciprocity method* (DRM). Bij de BKM worden er radiale basisfuncties gebruikt in de DRM. Een oplossing voor $u_p(x)$ kan dan bekomen worden door eerst en vooral de source term $f(x)$ te benaderen door een aantal basisfuncties. Bij radiale basisfuncties geeft dit dan:

$$f(x) \approx \tilde{f}(x) = \sum_{j=1}^N \alpha_j \phi_j(x) \quad (5.19)$$

Als we, zoals in het vorige deel, veronderstellen dat $f(x_i) = \tilde{f}(x_i)$ voor de N verschillende datapunten, dan kunnen we terug de volgende matrixvergelijking voor de coëfficiënten α_j opstellen:

$$\alpha = \Phi^{-1} f \quad (5.20)$$

Hierbij worden de elementen Φ_{ij} gegeven door $\phi(\|x_i - x_j\|)$. Als we dan de benadering \tilde{f} invullen in vergelijking 5.16 bekomen we:

$$L\{u_p(x)\} \approx L\{\tilde{u}_p(x)\} = \tilde{f}(x) = \sum_{j=1}^N \alpha_j \phi_j(x) \quad (5.21)$$

Om $\tilde{u}_p(x)$ uit 5.21 te halen definiëren we eerst ψ_j als volgt:

$$L\{\psi_j\} = \phi_j \quad (5.22)$$

De benadering $\tilde{u}_p(x)$ voor de particuliere vergelijking kan nu bekomen worden door:

$$u_p(x) \approx \tilde{u}_p(x) = \sum_{j=1}^N \alpha_j \psi_j(x) = \mathbf{S}^T \boldsymbol{\alpha} = \mathbf{S}^T \boldsymbol{\Phi}^{-1} \mathbf{f} \quad (5.23)$$

Hierbij is $\mathbf{S}^T = [\psi_1(x), \psi_2(x), \dots, \psi_N(x)]$. De enige moeilijkheid hierbij is het bepalen van ψ_j uit vergelijking 5.22. Voor een gegeven ϕ_j kan ψ_j namelijk slechts in een beperkt aantal gevallen van de operator L bepaald worden². Om dit probleem te omzeilen wordt in de BKM een omgekeerde strategie toegepast. Hier kiest men eerst de functie ψ_j en daarna berekent men de bijhorende ϕ_j a.d.h.v. vergelijking 5.22.

Bij het bepalen van de homogene vergelijking $u_h(x)$ maakt men bij de BKM gebruik van de niet-singuliere, algemene oplossing voor de differentiaal operator. Een aantal niet-singuliere, algemene oplossingen voor veelgebruikte differentiaal operators wordt gegeven in [7]. Als we veronderstellen dat $G(r)$ een algemene oplossing is, kunnen we $u_h(x)$ als volgt benaderen:

$$u_h(x) \approx \tilde{u}_h(x) = \sum_{k=M+1}^N \beta_k G(r_k) \quad (5.24)$$

Zoals voorheen is $r_k = \|x - x_k\|$ en vergelijkbaar met α_j stelt β_k de onbekende coëfficiënt voor. k is een index die itereert over de $N - M$ boundary points uit de dataset $\{x_i\}$. Als we 5.24 invullen in 5.18 krijgen we:

$$\sum_{k=M+1}^N \beta_k B\{G(r_k)\} = g(x) - B\{u_p(x)\} \quad (5.25)$$

Afhankelijk van de gegeven randvoorwaardes zal de operator B verschillen. Twee veel voorkomende types van randvoorwaardes zijn de *Neumann* en *Dirichlet* randvoorwaardes. Vergelijking 5.25 heeft in deze gevallen de volgende vorm:

$$\sum_{k=M+1}^N \beta_k \frac{\partial G(r_{mk})}{\partial n} = N(x_m) - \frac{\partial u_p(x_m)}{\partial n} \quad \text{Neumann} \quad (5.26)$$

$$\sum_{k=M+1}^N \beta_k G(r_{dk}) = D(x_d) - u_p(d) \quad \text{Dirichlet} \quad (5.27)$$

Hierbij geven m en d respectievelijk de Neumann en Dirichlet randpunten aan. n is de normaal op de rand van het domein. Als er ook interne punten worden gebruikt, moeten de volgende vergelijkingen ook worden toegevoegd:

$$\sum_{k=M+1}^N \beta_k G(r_{lk}) = u_l - u_p(x_l) \quad l = 1, \dots, M \quad (5.28)$$

²In het geval dat L rotatie symmetrisch is.

l duidt hier de index aan van de M gebruikte interne datapunten. Als we de coëfficiënten α en β voor u_h en u_p gevonden hebben, kunnen we $u(x)$ eenvoudig uitrekenen als volgt:

$$u(x) = u_h(x) + u_p(x) = \sum_{k=M+1}^N \beta_k G(r_k) + \sum_{j=1}^N \alpha_j \psi_j(r_k) \quad (5.29)$$

Hoofdstuk 6

Uitwerking

6.1 Functie bepaling

Om de Boundary Knot Method te kunnen toepassen bij de diffusion approximation, zullen eerst de gebruikte radiale basisfuncties gekozen moeten worden. Op de gekozen functies moeten ook de gepaste differentiaal operators L en B uit 5.14 en 5.15 toegepast worden. De vorm van de differentiaal operator L bepaalt ook de keuze van de gebruikte niet-singuliere, algemene oplossing (non-singular general solution, NSGS). L kan eenvoudig afgeleid worden door de vergelijking van de diffusion approximation (3.18) in een meer algemene vorm op te schrijven:

$$\nabla \cdot [a \nabla U_d(x)] = b U_d(x) + S(x) \quad (6.1)$$

Bij homogene materialen is de coëfficiënt a een constante. Door de coëfficiënt a in dat geval weg te werken en alle termen met U_d naar het linkerlid te brengen, bekomen we de volgende vergelijking:

$$\nabla^2 U_d(x) - \lambda^2 U_d(x) = f(x) \quad (6.2)$$

Uit vergelijking 6.2 volgt dat de differentiaal operator L voor homogene materialen de volgende vorm heeft:

$$L = [\nabla^2 - \lambda^2] \quad (6.3)$$

Hierbij is de coëfficiënt $\lambda = \sqrt{3\sigma_a(x)\sigma_{tr}(x)}$.¹ Bij heterogene materialen kan de coëfficiënt a uit vergelijking 6.1 niet zomaar weggewerkt worden. In dat geval beschrijft vergelijking 6.1 een vereenvoudigd Berger convectie-diffusie probleem. In het homogene geval waarbij λ een constante is, heeft L de vorm van een aangepaste Helmholtz operator. In het homogene gedeelte van de BKM wordt een NSGS als radiale basisfunctie gebruikt. De NSGS is hierbij een oplossing van de homogene vorm van vergelijking 6.1. Deze homogene vorm wordt bekomen door de bronterm gelijk te stellen aan nul. Voorbeelden van een NSGS voor de Berger en de Helmholtz operator worden gegeven door [43, 36]:

¹ λ is hier te vergelijken met κ_d uit [25].

$$U^\#(r, x) = I_0\left(\frac{\sqrt{2}}{|x|}r\right) \quad (6.4)$$

$$U_m^\#(r) = A_m (\lambda r)^{-\frac{n}{2}+1+m} I_{\frac{n}{2}-1+m}(\lambda r) \quad : \quad n \geq 2 \quad (6.5)$$

In de rest van deze thesis zullen we ons beperken tot objecten bestaande uit homogene materialen. Dit wil dus zeggen dat we ons beperken tot de gevallen waarbij de partiële differentiaal vergelijking van de diffusion approximation de vorm heeft van een aangepaste Helmholtz operator. In vergelijking 6.5 stelt I_m een aangepaste Bessel functie van de eerst soort² en van orde m voor. n is de dimensionaliteit van het probleem. Aangezien we de diffusion approximation proberen op te lossen in de 3D ruimte is $n = 3$. De parameter r is de Euclidische afstand tussen de punten die gebruikt worden in de BKM. λ is de parameter uit van de Helmholtz operator en de constante A_m wordt gegeven door:

$$\begin{aligned} A_0 &= 1 \\ A_m &= \frac{A_{m-1}}{2m\lambda^2} \end{aligned}$$

Deze vergelijking voor A_m kan omgezet worden in de volgende niet-recursieve vorm:

$$A_m = \frac{1}{(2\lambda^2)^m m!} \quad (6.6)$$

De aangepaste Bessel functie kan ook omgezet worden naar de aangepaste sferische Bessel functie van de eerste soort via de volgende eigenschap³:

$$I_{m+\frac{1}{2}}(\lambda r) = \sqrt{\frac{2\lambda r}{\pi}} i_m(\lambda r) \quad (6.7)$$

Door 6.6 en 6.7 toe te passen op 6.5 en door $n = 3$ te nemen, bekomen we de vergelijking van de NSGS voor de BKM:

$$U_m^\#(r) = \sqrt{\frac{2}{\pi}} \frac{1}{m!} \left(\frac{r}{2\lambda}\right)^m i_m(\lambda r) \quad (6.8)$$

Het omzetten van de aangepaste Bessel functie naar de aangepaste sferische Bessel functie heeft als voordeel dat de laatstgenoemde eenvoudiger te evalueren is. Voor gehele ordes en positieve waarden van λr geldt namelijk dat:

$$i_m(\lambda r) = (\lambda r)^m \left(\frac{1}{(\lambda r)} \frac{d}{d(\lambda r)}\right)^m \frac{\sinh(\lambda r)}{(\lambda r)} \quad (6.9)$$

Dit geeft voor de nulde en eerste orde van de aangepaste sferische Bessel functie:

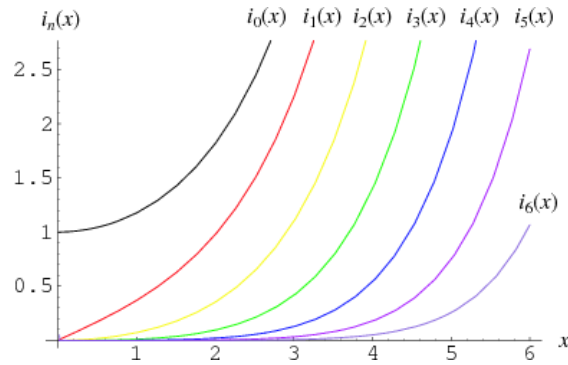
²Meer informatie over de *modified Bessel function of the first kind* is terug te vinden op [51].

³Deze eigenschap geldt enkel als de orde m een geheel getal is.

$$i_0(\lambda r) = \frac{\sinh(\lambda r)}{(\lambda r)}$$

$$i_1(\lambda r) = \frac{\cosh(\lambda r)}{(\lambda r)} - \frac{\sinh(\lambda r)}{(\lambda r)^2}$$

Hierbij dient opgemerkt te worden dat er een deling door nul kan optreden als $r = 0$. Daarom moet de limiet waarde gebruikt worden als dit optreedt. In figuur 6.1 wordt voor een aantal ordes het verloop van de aangepaste sferische Bessel functies weergegeven. Hieruit valt af te leiden dat $i_0(0) = 1$ en $i_m(0) = 0$ voor $m \neq 0$. Wat in de figuur ook valt op te merken is dat de aangepaste sferische Bessel functies een exponentieel verloop tonen. Aangezien er op een computer meestal een beperking is op de grootte van de voor te stellen getallen, kan dit bij grote functie argumenten voor problemen zorgen. Een mogelijke oplossing hiervoor is een negatieve, exponentiële schaalfactor toe te passen.



Figuur 6.1: Verschillende ordes van aangepaste sferische Bessel functies van de eerste soort [52].

Bij het bepalen van de homogene oplossing in de BKM moet de differentiaal operator B van de randvoorwaarde ook op de gekozen functies worden toegepast. Om deze operator te bepalen, schrijven we de randvoorwaarde van de diffusion approximation (3.21) neer in zijn algemene vorm:

$$U_d(x_b) - k \frac{\partial U_d(x_b)}{\partial n} = G(x_b) \quad (6.10)$$

Hieruit is af te leiden dat de differentiaal operator B van de randvoorwaarde de volgende vorm heeft:

$$B = \left[1 - k \frac{\partial}{\partial n} \right] \quad (6.11)$$

Hierbij is $k = \frac{2}{3\sigma_{tr}}$ en n is de normaal in het randpunt x_b die gericht is naar het object. Deze operator moet bij het benaderen van de homogene oplossing in de BKM, zowel op de NSGS als op de radiale basisfuncties van de particuliere oplossing toegepast worden. De differentiaal operator B kan niet rechtstreeks op de radiale basisfuncties toegepast worden. Dit komt omdat

de radiale basisfuncties niet rechtstreeks afgeleid kunnen worden naar de normaal. Dit probleem kan opgelost worden door de kettingregel toe te passen. Hierdoor wordt de differentiaal operator B voor de NSGS $U_0^\#$:

$$B = \left[1 - k \frac{\partial}{\partial(\lambda r)} \frac{\partial(\lambda r)}{\partial n} \right] \quad (6.12)$$

Toepassen van 6.12 op 6.8 geeft uiteindelijk⁴:

$$\left[1 - k \frac{\partial}{\partial(\lambda r)} \frac{\partial(\lambda r)}{\partial n} \right] U_0^\#(\lambda r) = \sqrt{\frac{2}{\pi}} i_0(\lambda r) - k \left[\sqrt{\frac{2}{\pi}} i_1(\lambda r) \frac{\partial(\lambda r)}{\partial n} \right] \quad (6.13)$$

De afgeleide van de afstand λr naar de normaal n wordt hierbij gegeven door:

$$\frac{\partial(\lambda r)}{\partial n} = \lambda \frac{n_x(p_{b_x} - p_{i_x}) + n_y(p_{b_y} - p_{i_y}) + n_z(p_{b_z} - p_{i_z})}{r} \quad (6.14)$$

r is hier dus de afstand tussen het beschouwde randpunt p_b en een ander randpunt p_i . Als p_i samenvalt met p_b zal deze afstand nul worden, waardoor ook de noemer nul wordt. In dat geval zal vergelijking 6.13 gewoon $\sqrt{\frac{2}{\pi}}$ als resultaat hebben.

Ook voor het particuliere deel van de BKM moeten er radiale basisfuncties gekozen worden. De differentiaal operator L moet ook op de gekozen radiale basisfunctie ψ toegepast worden, zoals aangegeven in vergelijking 5.22. Ook hier kan de differentiaal operator niet zomaar toegepast worden op de radiale basisfunctie. Dit komt omdat de aangepaste Helmholtz operator die gegeven werd in vergelijking 6.3 gedefinieerd is voor een Cartesisch coördinatenstelsel. Een radiale basisfunctie daarentegen is in principe een functie die gedefinieerd is in een sferisch coördinatenstelsel. De differentiaal operator L moet dus omgevormd worden naar een vorm in sferische coördinaten. De aangepaste 3D Helmholtz operator voor radiale basisfuncties wordt dan gegeven door:

$$L = \left[\frac{\partial^2}{\partial r^2} + \frac{2}{r} \frac{\partial}{\partial r} - \lambda^2 \right] \quad (6.15)$$

Deze operator L moet dan op de radiale basisfunctie ψ toegepast worden om ϕ te bekomen. Voor ψ kan een keuze gemaakt worden uit verschillende soorten radiale basisfuncties. Mogelijke keuzes zijn de compactly supported, Gaussiaanse of multiquadratische radiale basisfuncties, zoals al werd aangegeven in § 5.1. Ook de hogere ordes van de NSGS (6.5) zouden als radiale basisfunctie voor de particuliere oplossing gebruikt kunnen worden. In de huidige implementatie maken we gebruik van de Gaussiaanse radiale basisfunctie. Het toepassen van de aangepaste Helmholtz operator op deze Gaussiaanse radiale basisfunctie geeft:

$$L \left\{ e^{-\frac{r^2}{c^2}} \right\} = \left(\frac{4r^2}{c^4} - \frac{6}{c^2} - \lambda^2 \right) e^{-\frac{r^2}{c^2}} \quad (6.16)$$

⁴Hierbij wordt gebruik gemaakt van de volgende eigenschap van de aangepaste sferische Bessel functie: $\frac{\partial i_m(x)}{\partial x} = \frac{m i_{m-1}(x) + (m+1) i_{m+1}(x)}{2m+1}$.

Met de parameter c kan de breedte van de Gauss curve worden aangepast. De differentiaal operator B van de randvoorwaarde blijft onveranderd. Toegepast op de Gaussiaanse radiale basisfunctie geeft dit:

$$B \left\{ e^{-\frac{r^2}{c^2}} \right\} = e^{-\frac{r^2}{c^2}} - k \left(\frac{-2r}{c^2} \right) e^{-\frac{r^2}{c^2}} \frac{\partial r}{\partial n} \quad (6.17)$$

6.2 Bepaling van de source term

Vergelijking 6.1 bevat de source term $S(x)$ die gegeven werd in 3.19. Deze source term moet bepaald worden vooraleer de coëfficiënten voor de particuliere oplossing van de BKM berekend kunnen worden. De source term is afhankelijk van de reduced incident intensity L_{ri} en het licht dat gegenereerd wordt in het object zelf, de emissie ϵ . In [25] worden er een aantal voorbeelden gegeven voor het uitrekenen van de diffusion approximation voor verschillende gevallen van evenwijdige lichtstralen die invallen op een plaat met een bepaalde dikte. Hierbij wordt ook afgeleid hoe de source term in deze gevallen kan worden uitgerekend.

Evenwijdige lichtstralen, of anders gezegd gecollimeerd licht komt voor bij lasers en bij lichtbronnen op een oneindige afstand. De formules uit de voorbeelden in [25] zijn dus maar in een relatief beperkt aantal situaties correct. In veel gevallen is het licht echter niet gecollimeerd. Om de source term onder meer algemene belichtingsomstandigheden uit te rekenen, kunnen technieken zoals *photon mapping* of *irradiance sampling* gebruikt worden.

6.2.1 Photon mapping

Om de source term te bepalen met behulp van photon mapping wordt er een volume photon map opgesteld. Fotonen die na refractie een object binnendringen worden bij interactie (verstrooiing of absorptie) met het object opgeslaan in de volume photon map. De afstand waarop de interactie optreedt wordt gesampled op basis van de transmittance τ .

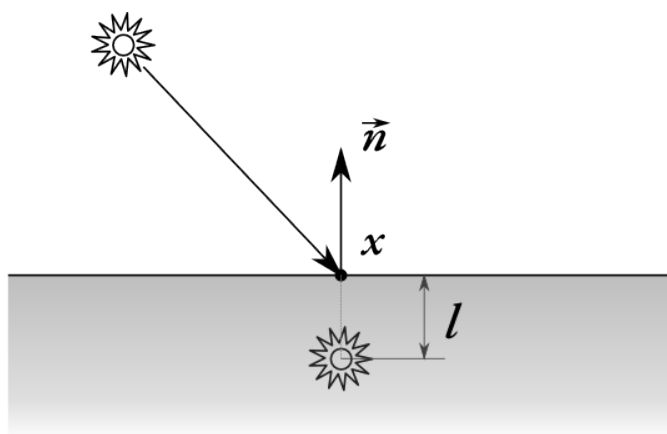
$$\tau(x, x_s) = e^{-\int_{x_s}^x \sigma_t(x_s+u(x-x_s))du} = 1 - \zeta \quad (6.18)$$

Hierbij is x het punt in het object waar de interactie optreedt, x_s is het punt waar het foton het object binnentreedt of het vorige punt waar er interactie optrad en ζ is een uniform verdeeld, willekeurig getal. Aan de hand van Russian roulette wordt bepaald als het foton verstrooid of geabsorbeerd wordt. Het albedo α bepaalt de kans dat het foton wordt verstrooid. Nadat de volume photon map is opgesteld kan deze gebruikt worden om de source term te bepalen. U_{ri} en Q_{ri} voor een bepaald punt x worden bijvoorbeeld gegeven door:

$$U_{ri}(x) \approx \frac{1}{4\pi} \sum_{p=0}^n \frac{\Delta\Phi(x_p, \omega_p)}{\frac{4}{3}\pi r^3} \quad (6.19)$$

$$Q_{ri}(x) = \sum_{p=0}^n \frac{\Delta\Phi(x_p, \omega_p)}{\frac{4}{3}\pi r^3} g\omega_p \quad (6.20)$$

In deze formules is n het aantal fotonen dat omvat wordt door het volume $\frac{4}{3}\pi r^3$. ω_p is de inkomende richting van het p -de foton, die ook in de volume foton map werd opgeslaan. Meer informatie over deze aanpak is terug te vinden in [18, 31].



Figuur 6.2: Schematische voorstelling van irradiance sampling.

6.2.2 Irradiance sampling

Irradiance sampling is de methode die in de huidige implementatie wordt gebruikt. Het idee achter deze methode is zeer eenvoudig. De irradiantie wordt gesampled op een aantal punten op het oppervlak van het object. Daarna wordt de irradiantie in ieder punt vervangen door een isotrope puntlichtbron⁵ die op een kleine afstand onder het oppervlak wordt geplaatst. Dit wordt schematisch weergegeven in figuur 6.2. In deze figuur is n de normaal in het punt x waarop de irradiantie wordt gesampled. De afstand l komt overeen met het *mean free path*. Dit is de gemiddelde afstand die een foton aflegt zonder dat er interactie optreedt. Deze afstand wordt gegeven door:

$$l = \frac{1}{\sigma_{tr}} \quad (6.21)$$

Zoals blijkt uit vergelijking 6.22 wordt de sterkte van de lichtbron L_b bepaald door de irradiantie E aan het oppervlak, vermenigvuldigd met een attenuatiefactor. Voor de eenvoud wordt er verondersteld dat de invloed van de Fresnel refractie te verwaarlozen is.

$$L_b(x) = e^{-\sigma_i l} \frac{E(x)}{4\pi} \quad (6.22)$$

Doordat het invallende licht vervangen wordt door interne lichtbronnen, zal de reduced incident intensity nul zijn. De deelterm $-4\pi\sigma_s U_{r_i}$ uit de source term zal daardoor ook nul worden. Ook de deelterm met Q_{r_i} valt volledig weg als gevolg van het wegvallen van de reduced incident intensity en het isotroop zijn van de lichtbronnen⁶. De deelterm met Q_1 uit de randvoorwaarde wordt omwille van dezelfde reden geëlimineerd. Door gebruik te maken van irradiance sampling worden de source term en de randvoorwaarde dus herleid tot:

$$S(x) = - \int_{S^2} \epsilon(x, \omega) d\omega \quad (6.23)$$

⁵Een isotrope puntlichtbron is een puntlichtbron die in alle richtingen dezelfde hoeveelheid energie uitstraalt.

⁶Voor isotrope lichtbronnen geldt: $\int_{4\pi} \epsilon \omega d\omega = 0$

$$U_d(x_b) - \frac{2}{3\sigma_{tr}} \frac{\partial U_d(x_b)}{\partial n} = 0 \quad (6.24)$$

6.3 Bepalen van de BKM coëfficiënten

Om de coëfficiënten α en β uit 5.29 te bepalen, moeten de matrices die voor de particuliere en de homogene oplossing opgesteld werden, geïnverteerd worden. Het inverteren van deze matrices zorgt echter voor problemen bij de BKM. De BKM heeft namelijk als nadeel dat de matrices die gevormd worden door de radiale basisfuncties zeer slecht geconditioneerd (*ill conditioned*) zijn. Een stelsel met een slecht geconditioneerde matrix, heeft als negatieve eigenschap dat een kleine verandering in één van de elementen van de source term, grote veranderingen kan teweeg brengen in de oplossing van de coëfficiënten. Dus ook afrondingsfouten die een gevolg zijn van de beperkte nauwkeurigheid van digitale systemen, kunnen het eindresultaat sterk beïnvloeden. Voor zeer slecht geconditioneerde matrices kunnen deze afrondingsfouten zelfs zorgen voor een volledig verkeerd resultaat.

De iteratieve algoritmes uit § 4.6 zullen in de meeste gevallen falen bij slecht geconditioneerde matrices. Bovendien veronderstellen deze meestal ook een symmetrische matrix. De matrix in het homogene gedeelte van de BKM is echter niet meer symmetrisch als een gevolg van de *Robin* randvoorwaarde⁷. Omwille van deze redenen zal er geen iteratieve methode worden gebruikt om de matrices te inverteren.

Ook inversie met behulp van LU-decompositie zal geen goed resultaat geven. Figuur 6.3 geeft het verschil aan tussen het gebruik van de *inv()* functie en de *pinv()* functie om de particuliere matrix te inverteren in MATLAB. De *inv()* functie maakt intern gebruik van LU-decompositie. *pinv()* is een functie die de pseudoinverse van een matrix berekent. Uit deze figuur blijkt duidelijk dat door LU-decompositie toe te passen op de slecht geconditioneerde particuliere matrix er uiteindelijk een foutief resultaat bekomen wordt voor de particuliere oplossing. De pseudoinverse, meer bepaald de *Moore-Penrose pseudoinverse*, geeft daarentegen wel een aanvaardbaar resultaat. In de huidige implementatie wordt daarom ook deze pseudoinverse gebruikt om de inverse van de slecht geconditioneerde matrices van de BKM te benaderen.

Het gebruik van de Moore-Penrose pseudoinverse geeft een *least squares* oplossing voor een stelsel van lineaire vergelijkingen. Een least squares oplossing is een oplossing die er naar streeft dat de Euclidische norm van de vector $\|\mathbf{Ax} - \mathbf{b}\|^2$ minimaal is. Als een matrix goed geconditioneerd is zal de pseudoinverse ervan overeenkomen met de gewone inverse van de matrix. Voor slecht geconditioneerde en singuliere matrices is dit niet het geval en is de pseudoinverse dus slechts een benadering van de gewone inverse. Het berekenen van de pseudoinverse van een matrix \mathbf{A} kan gebeuren met behulp van de *singular value decomposition* (SVD). De singular value decomposition van de matrix \mathbf{A} wordt gegeven door⁸:

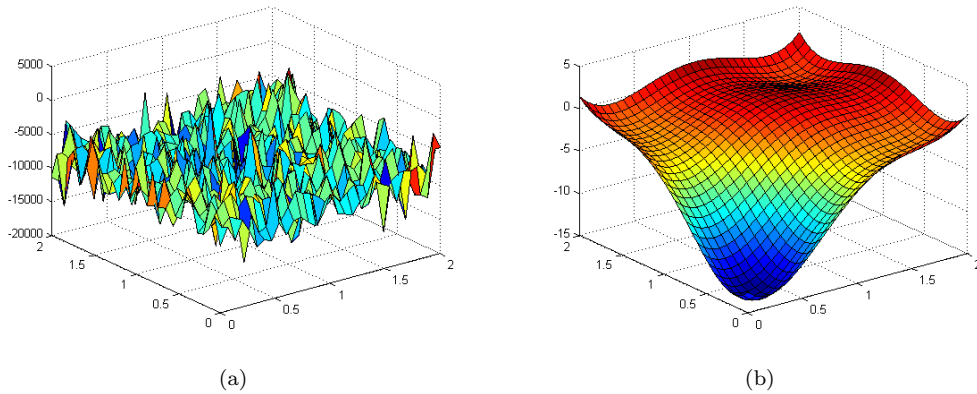
$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (6.25)$$

\mathbf{V}^T stelt hierbij de getransponeerde van \mathbf{V} voor. De matrices \mathbf{U} en \mathbf{V} zijn beide unitaire matrices⁹. De matrix \mathbf{S} is een diagonaal matrix waarvan de diagonaal elementen de singuliere

⁷Een *Robin* randvoorwaarde is een randvoorwaarde van de derde soort. Dit is een randvoorwaarde die bestaat uit een combinatie van een Dirichlet en Neumann randvoorwaarde en heeft dus de vorm: $au(x) + b\frac{\partial u(x)}{\partial n} = g(x)$.

⁸Als \mathbf{A} een $M \times N$ matrix is, dan is \mathbf{U} een $M \times M$ matrix, \mathbf{S} een $M \times N$ matrix en \mathbf{V} een $N \times N$ matrix.

⁹Een unitaire matrix voldoet aan de eigenschap $\mathbf{A}^*\mathbf{A} = \mathbf{I}$. Hierbij stelt \mathbf{A}^* de hermitisch toegevoegde van \mathbf{A} voor. \mathbf{I} is de eenheidsmatrix.



Figuur 6.3: Het resultaat van de particuliere oplossing van de diffusion approximation in een horizontale snede van een kubus. In figuur (a) werd de *inv()* functie van MATLAB gebruikt, in (b) de *pinv()* functie. Merk ook op dat de schaal van de z-as sterk verschilt in de twee figuren. Het conditiegetal van de particuliere matrix is hierbij $2.277876e^{20}$.

waarden worden genoemd. Deze singuliere waarden worden meestal geordend van groot naar klein. Met behulp van deze drie matrices kan de Moore-Penrose pseudoinverse \mathbf{A}^+ berekend worden als volgt:

$$\mathbf{A}^+ = \mathbf{V}\mathbf{S}^+\mathbf{U}^T \quad (6.26)$$

Zoals uit vergelijking 6.26 blijkt moet de pseudoinverse van de matrix \mathbf{S} berekend worden om \mathbf{A}^+ uit te rekenen. \mathbf{S}^+ kan gemakkelijk berekend worden door de singuliere waarden te vervangen door hun omgekeerde¹⁰, dus:

$$s_{ii}^+ = \begin{cases} \frac{1}{s_{ii}} & s_{ii} > tol \\ 0 & s_{ii} \leq tol \end{cases} \quad (6.27)$$

Om de stabiliteit te verhogen worden de waarden die kleiner zijn dan een bepaalde kleine tolerantie op nul gezet. In MATLAB wordt deze tolerantie standaard gegeven door $tol = \max(M, N) \|\mathbf{A}\| eps$ [37]. *eps* is hierbij de relatieve floating-point precisie¹¹. De norm van \mathbf{A} komt overeen met de maximale waarde van \mathbf{S} en M en N zijn de afmetingen van de matrix.

Om de stabiliteit tijdens het berekenen van de pseudoinverse nog lichtjes te verbeteren, kunnen zeer kleine waarden in de particuliere en homogene matrix op voorhand op nul gezet worden. In de huidige implementatie worden waarden die kleiner zijn dan 10^{-30} op nul gezet.

¹⁰Indien \mathbf{A} en dus ook \mathbf{S} niet vierkant zijn, moet \mathbf{S} ook eerst nog getransponeerd worden

¹¹Voor een double is dit 2^{-52} in MATLAB.

6.4 Visualisatie

Als de coëfficiënten van de particuliere en de homogene oplossing uitgerekend zijn, kan de waarde van $U_d(x)$ via vergelijking 5.29 gemakkelijk uitgerekend worden voor willekeurige punten in en op het object. Aan de hand van een ray tracer kunnen dan stralen naar het object geschoten worden. Op de positie x waar deze straal het object raakt, kan dan de waarde van $U_d(x)$ uitgerekend worden om zo de bijdrage aan de radiantie $L(x, \omega)$ op dat punt te bepalen. Voor de verhouding tussen U_d en \mathbf{F}_d geldt dat [25]:

$$U_d \gg |\mathbf{F}_d| \quad (6.28)$$

De bijdrage van \mathbf{F}_d in vergelijking 3.15 zal dus aanzienlijk kleiner zijn dan die van U_d . Daarom wordt in onze implementatie de term met \mathbf{F}_d genegeerd. Dit vereenvoudigt ook de berekeningen omdat de gradiënt van U_d niet meer moet uitgerekend worden. Aangezien we het invallende licht vervangen hebben door interne puntlichtbronnen valt de reduced incident intensity L_{ri} ook weg, zodat we de radiantie in een punt kunnen benaderen door:

$$L(x, \omega) \approx U_d(x) \quad (6.29)$$

Tijdens het testen in MATLAB hebben we de functies van MATLAB zelf gebruikt om de testresultaten te visualiseren. De kubus die gebruikt werd tijdens de testen bestond uit een verzameling van uniform verdeelde punten. Om het 3D resultaat van de testen te visualiseren hebben we gebruik gemaakt van de *delaunayn()* en de *patch()* functies van MATLAB. De visualisatie mogelijkheden van MATLAB zijn in zekere zin echter beperkt. Daarom hebben we in een latere fase *PBRT* gebruikt om het resultaat van de diffusion approximation te visualiseren.

PBRT staat voor *Physically Based Ray Tracer* en is een open source rendering systeem dat geschreven is in C++. De opbouw, werking en achterliggende principes van dit rendering systeem worden uitvoerig beschreven in het bijhorende boek *Physically Based Rendering: From Theory to Implementation* [45]. De verschillende componenten van PBRT zijn geïmplementeerd als plugins, waardoor het systeem vrij eenvoudig kan worden uitgebreid. Om subsurface scattering met de BKM in PBRT te visualiseren hebben we een plugin toegevoegd die gebaseerd is op de *SurfaceIntegrator* interface. Deze interface voorziet de nodige methodes om de belichting uit te rekenen op het punt waar dat een bepaalde ray botst op een oppervlak uit de scène. Om de implementatie eenvoudig te houden is de nieuwe surface integrator gebaseerd op de *Whitted* surface integrator uit PBRT. De Whitted surface integrator, en dus ook onze, houdt enkel rekening met de directe belichting. Bovendien wordt er verondersteld dat de scattering-coëfficiënten voor alle golflengtes hetzelfde zijn, zodat er maar één set van BKM-coëfficiënten uitgerekend moet worden.

De *SurfaceIntegrator* interface voorziet ook een *preprocess()* functie. Deze functie wordt door PBRT opgeroepen nadat de beschrijving van de scène is ingeladen, maar voordat er begonnen wordt met het renderen van de scène. Deze functie maakt het dus mogelijk om de coëfficiënten van de BKM op voorhand uit te rekenen. Dit gebeurt in drie deelstappen. In de eerste stap wordt ieder subsurface scattering object recursief verfijnd tot zijn basiselementen. Met een basiselement wordt hier een object bedoeld dat in PBRT gesampled kan worden. De tweede stap verdeelt het aantal punten dat gesampled moet worden over de verschillende basiselementen. Deze verdeling gebeurt op basis van de verhouding tussen de oppervlakte van het basiselement en de totale oppervlakte van het subsurface scattering object. Voor ieder sample punt wordt dan

de hoeveelheid invallend licht berekend. In de laatste stap worden er op basis van de samples interne punten gegenereerd en wordt de source term opgesteld. Daarna worden de coëfficiënten van de BKM uitgerekend.

Als een ray op een oppervlak van een subsurface scattering object invalt, zal onze surface integrator de belichting berekenen op basis van de BKM in plaats van de gewone berekening uit te voeren. Een probleem dat bij het schrijven van de plugin optrad, was dat PBRT standaard geen mogelijkheid voorziet om een onderscheid te maken tussen subsurface scattering objecten en gewone objecten. Daarom hebben we ook code moeten toevoegen aan een aantal basisklassen. Deze extra code maakt het mogelijk om in het bestand waarin de te renderen scène beschreven staat, een *id* toe te kennen aan de objecten. Op basis van deze *id* kan dan wel het onderscheid gemaakt worden tussen een gewoon en een subsurface scattering object.

Hoofdstuk 7

Resultaten

In eerste instantie hebben we de BKM getest op een analytisch verifieerbare vergelijking zodat we de correctheid van het resultaat konden controleren. De gebruikte testvergelijking wordt gegeven door 7.1. Het resultaat van het toepassen van de aangepaste Helmholtz operator op deze vergelijking en de bijhorende randvoorwaarde worden respectievelijk gegeven door 7.2 en 7.3.

$$x^2 \sin(x) \cos(y) \cos(z) \tag{7.1}$$

$$((2 - (3 + \lambda^2)x^2) \sin(x) + 4x \cos(x)) \cos(y) \cos(z) \tag{7.2}$$

$$x^2 \sin(x) \cos(y) \cos(z) - \frac{\partial(x^2 \sin(x) \cos(y) \cos(z))}{\partial n} \tag{7.3}$$

De λ waarde van de Helmholtz operator werd tijdens de eerste testen gelijk gesteld aan $\sqrt{3}$ en voor de randvoorwaarde was $k = 1$. Figuur 7.1 toont een aantal testresultaten. In figuur 7.1(i) is te zien dat het verschil tussen de exacte oplossing en de oplossing van de BKM vrij klein is. De absolute fout ($\|\hat{x} - x\|_\infty$) van de oplossing bij deze resultaten was 4.8332×10^{-5} . Voor de testvergelijking gaf de BKM dus zeer goede resultaten.

De punten of *knots* die gebruikt worden bij de BKM, kunnen op verschillende manieren worden gegenereerd. De punten kunnen bijvoorbeeld door een functie worden gegenereerd. Dit is ook de methode die wordt toegepast voor de kubus die wordt gebruikt in de MATLAB testen. Het voordeel van deze aanpak is dat de structuur van de punten gemakkelijk kan bepaald worden. De toepasbaarheid van deze aanpak is echter beperkt tot objecten met een vrij eenvoudige vorm. Een andere aanpak is de vertices van een model te gebruiken als randpunten. Nadeel van deze methode is echter dat het aantal punten dan voor een bepaald model altijd vast ligt. In onze techniek maken we gebruik van de sampling functionaliteit van PBRT om de randpunten te genereren. PBRT voorziet namelijk voor de meeste van zijn basiscomponenten de mogelijkheid om ze op een uniforme manier te samplen. Hierbij wordt gebruik gemaakt van de *StratifiedSample2D()* functie om de sample posities te genereren. Met deze functie is het ook mogelijk om te zorgen voor een kleine, willekeurige variatie (*jitter*) op de sample posities. In figuur 7.2 wordt het verschil getoond tussen samples met en zonder jitter. Merk op dat bij de samples zonder jitter er geen punten zijn op de uiterste z-waardes. Dit zorgt voor de afbuiging die te zien is in het midden van de bol in figuur 7.11(b).

De gegenereerde sample punten worden als randpunten voor de BKM gebruikt. Bij het particuliere deel van de BKM worden zowel randpunten als interne punten gebruikt. De lichtbronnen die bij een sample punt op de mean free path afstand onder het oppervlak werden geplaatst, worden gebruikt als interne punten. Er kunnen nog bijkomende interne punten worden gebruikt. Figuur 7.4 en figuur 7.5 geven het verschil aan voor een aantal verschillende opstellingen van bijkomende interne punten. De opstelling van de punten wordt getoond in figuur 7.6. De source term van deze extra interne punten is nul. Er is duidelijk een verschil tussen de oplossing met een reguliere grid van interne punten en geen extra interne punten. Bij de reguliere grid is de verspreiding van het licht minder sterk. De verspreiding van het licht wordt dus in zekere zin beperkt door de extra interne punten. Het genereren van een grid van punten in een willekeurig object is geen triviale zaak. Daarom hebben we er in onze implementatie voor gekozen om enkel extra interne punten te plaatsten op $\frac{3}{2}$ van het mean free path, zoals ook werd toegepast om de resultaten in figuur 7.4(b) te bekomen. De resultaten uit 7.4(b) en 7.4(c) bevinden zich tussen deze van 7.4(a) en 7.4(d). Hoewel er bij 7.4(b) geen extra punten op $\frac{1}{2}$ van het mean free path worden geplaatst, en er dus minder punten worden gebruikt dan bij 7.4(c), levert dit toch een resultaat dat beter aansluit bij dat uit 7.4(d).

Zoals al werd aangehaald in §6.3 zijn de matrices van de BKM zeer slecht geconditioneerd. Tijdens het omzetten van de MATLAB testcode naar C++ werd duidelijk hoe onstabiel de methode hierdoor eigenlijk is. Om de punten voor te stellen maken we gebruik van de *Point* klasse van PBRT. In deze klasse worden de coördinaten als *float* opgeslaan. De *Point* klasse van PBRT voorziet ook een functie om de afstand tussen twee punten te berekenen. Gebruik makend van deze functie wordt het resultaat uit figuur 7.3(a) bekomen. Het licht lijkt hierbij naar de bovenzijde van de kubus toe te nemen. Het licht bestaat in de testopstelling echter uit een eenvoudige puntlichtbron die op de helft van de hoogte van de kubus werd geplaatst. De lichtverstrooiing zou dus in de boven- en onderhelft van de kubus gelijk moeten zijn, zoals in figuur 7.3(b). In deze figuur wordt de afstand berekend met een zelfgeschreven afstandsfunctie waarbij de coördinaten van de punten eerst omgezet worden naar *doubles*. Hoewel het verschil tussen de afstand berekend met *floats* en met *doubles* op zich vrij klein is, toont het eindresultaat toch grote afwijkingen. Bij het omzetten van de MATLAB code naar C++ hebben we ook gemerkt dat de resultaten van de radiale basisfuncties ook niet helemaal hetzelfde zijn, hoewel ze toch op dezelfde manier worden uitgerekend. De waardes komen overeen tot op ongeveer zes cijfers na de komma, maar daarna verschillen ze licht van elkaar. Tijdens de berekeningen zijn er dus kleine afrondingsverschillen tussen MATLAB en C++. De berekende pseudoinverses zullen dus ook verschillen.

Tijdens het berekenen van de pseudoinverses worden bij de SVD de waardes van de diagonaal matrix \mathbf{S} , de singuliere waardes, die kleiner zijn dan een bepaalde tolerantie op nul gezet. Dit wordt gedaan om de numerieke stabiliteit van de pseudoinverse te verbeteren. Maar hoe meer singuliere waardes op nul worden gezet, hoe grover de benadering van de pseudoinverse is. In tabel 7.1 wordt voor een aantal particuliere matrices aangegeven hoeveel van deze singuliere waardes worden gebruikt om de pseudoinverse te berekenen en wat het conditiegetal is van de matrix. Hieruit blijkt dat er maar een zeer klein aantal waardes van \mathbf{S} effectief worden gebruikt. In de bijhorende figuur 7.7 worden de singuliere waardes op een logaritmische schaal uitgezet. Hierin valt te bemerken dat de grootte van de singuliere waardes zeer snel afneemt, wat erop wijst dat de matrices bijna singulier zijn.

Om de SVD uit te rekenen die gebruikt wordt bij de pseudoinverse hebben we de *LAPACK++* library gebruikt [49]. LAPACK++ is een C++ implementatie van de Fortran LAPACK (*Linear Algebra PACKage*, [15]) library die ook gebruik maakt van de BLAS (*Basic Linear Algebra Subprograms*, [14]) routines. MATLAB maakt intern ook gebruik van deze libraries. De SVD van

Aantal singuliere waarden $\neq 0$	Conditiegetal	Tolerantie
10	2.643×10^{20}	2.357×10^{-9}
11	6.874×10^{20}	2.357×10^{-9}
53	2.519×10^{20}	2.357×10^{-9}
10	5.022×10^{20}	2.144×10^{-21}

Tabel 7.1: Deze tabel geeft het aantal singuliere waarden van de matrix \mathbf{S} die gebruikt worden bij het berekenen van de pseudoinverse. De grootte van de matrix bedraagt in alle gevallen 1296. In de eerste drie gevallen zijn Gaussiaanse RBF's gebruikt met respectievelijk $c = 500$, $c = 100$ en $c = 10$. In het laatste geval is een Invers Multiquadratische RBF gebruikt met parameters $c = 1000$ en $\beta = 3$.

	Aantal randpunten		
	108	432	972
sgesvd	0.44s	54.03s	684.80s
sgesdd	0.28s	27.20s	356.42s

Tabel 7.2: De tijd die nodig is om de coëfficiënten van de BKM uit te rekenen met respectievelijk de *sgesvd* en de *sgesdd* routine. De matrix voor de particuliere oplossing is hierbij de grootste matrix en heeft een grootte van drie maal het aantal randpunten.

MATLAB maakt gebruik van de *sgesvd* routine van LAPACK. In LAPACK++ wordt standaard de *sgesdd* routine gebruikt die sneller is. Het verschil in snelheid in LAPACK++ tussen deze twee routines wordt gegeven in tabel 7.2. Uit deze tabel blijkt dat de *sgesdd* routine maar ongeveer de helft van de tijd van de *sgesvd* routine nodig heeft, hoewel het resultaat nagenoeg hetzelfde is. Daarom wordt in onze huidige implementatie de *sgesdd* routine gebruikt.

Zoals eerder al werd gezegd, zijn de matrices van de BKM zeer gevoelig voor kleine afwijkingen. Doordat de inverse van de matrix bij de particuliere oplossing door de pseudoinverse benaderd wordt, zal er altijd een afwijking zijn op de coëfficiënten van de particuliere oplossing. Deze coëfficiënten worden dan gebruikt bij het berekenen van de particuliere oplossing bij het opstellen van de bronterm voor de homogene oplossing. De afwijkingen die bij de particuliere oplossing werden geïntroduceerd zullen dus ook gevolgen hebben op het resultaat van de homogene oplossing. Het resultaat dat uiteindelijk bekomen wordt, is daardoor zeer onstabiel en ook foutief. De hoogfrequente oscillatie die optrad door de gewone inversie zoals in figuur 6.3(a), lijkt door de pseudoinverse in de uiteindelijke resultaten te zijn vervangen door een laagfrequente zoals in figuur 7.8 te zien is.

In figuur 7.9 wordt het verschil in resultaat getoond tussen een Gaussiaanse en een Invers Multiquadratische RBF. De resultaten met beide RBF's zijn gelijkaardig. Dit komt omdat beide functies, voor de kleinere waarden en op een schaalfactor na, min of meer een gelijkaardig verloop hebben. De parameter c van de Gaussiaanse RBF is hierbij 500. Voor de Invers Multiquadratische RBF zijn de parameters c en β respectievelijk 1000 en 3. Figuur 7.10 toont de resultaten van het aanpassen van de parameter c van de Gaussiaanse RBF. Hoewel geen enkele parameterwaarde een correct resultaat levert, is het wel duidelijk dat de waarde van de parameter een sterke invloed heeft op het resultaat.

Eerder werd al aangehaald dat ook het aantal interne punten en de plaatsing ervan invloed heeft op het resultaat. In figuur 7.11 wordt voor vier verschillende opstellingen van interne punten

Aantal randpunten	BKM	Rendering
100	0.235s	6.2s
484	40.516s	25.8s
961	349.907s	50.2s
1764	2378.219s	90.0s

Tabel 7.3: De tijd om de BKM coëfficiënten te berekenen en de rendertijd voor de verschillende afbeeldingen uit figuur 7.12. De rendertijd met de gewone *Whitted* surface integrator was 1.1s voor deze scène.

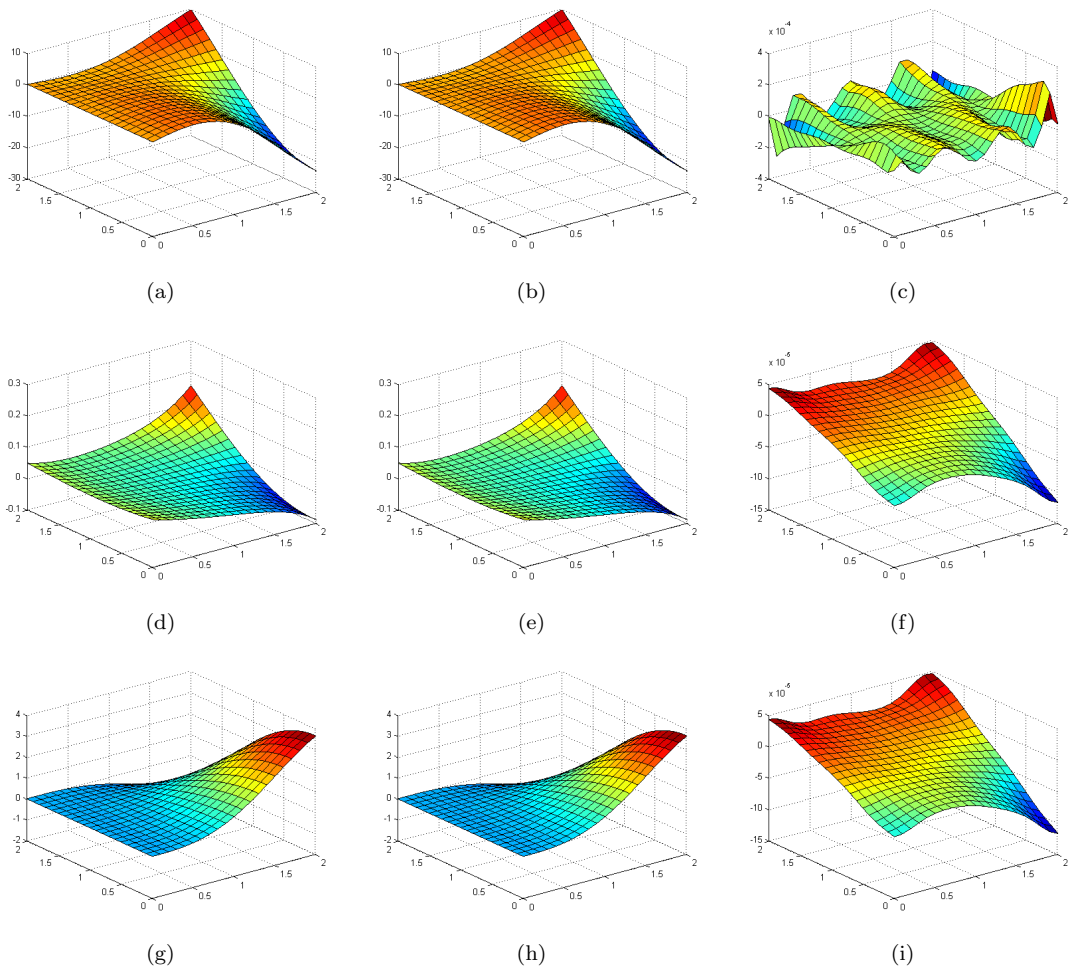
het resultaat gegeven. Ook hier valt weer te bemerken dat geen van de resultaten correct is, maar dat de interne punten wel degelijk hun invloed hebben. De extra punten in figuur 7.11(d) hebben schijnbaar geen invloed meer in vergelijking met figuur 7.11(c). Dit valt te verklaren doordat deze punten zich kort bij het centrum van de bol bevinden. Hun invloed aan de rand van de bol is door de grotere afstand echter sterk afgezwakt. Ook het aantal randpunten heeft zijn invloed op het resultaat zoals te zien is in figuur 7.12. Een van de eigenschappen van de BKM die werd gegeven in [7], is dat de bekomen oplossing vrij snel convergeert. Deze convergentie is ook te zien in figuur 7.12 want in 7.12(d) is het aantal randpunten bijna dubbel zoveel als in 7.12(c), maar het resultaat verschilt nog nauwelijks. Als het aantal randpunten te laag is, is het resultaat volledig foutief.

In tabel 7.3 wordt ook de tijd die nodig is om de BKM coëfficiënten te berekenen en de rendertijd geven voor de figuren uit 7.12. Het vergroten van het aantal randpunten heeft ook een logisch nadeel, namelijk dat het geheugen dat nodig is tijdens de berekeningen ook toeneemt. Afhankelijk van het gebruikte systeem is er dus een grenswaarde voor het maximale aantal punten dat kan gebruikt worden. Voor het gebruikte systeem lag deze grens kort bij de 2000 randpunten¹. Hierbij moet er in het achterhoofd gehouden worden dat de grootte van de particuliere matrix drie maal zo groot is aangezien er ook twee lagen van interne punten worden gegenereerd.

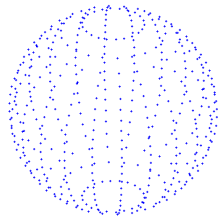
In figuur 7.13 wordt het resultaat gegeven van de diffusion approximation in MATLAB. Doordat het berekenen van de pseudoinverse in MATLAB iets stabielier is zullen de resultaten in bepaalde gevallen ook iets of wat stabielier zijn. Uit de testen bleek echter dat de resultaten echter snel onstabiel werden als de eigenschappen van het materiaal of de virtuele afmetingen van het object veranderden. Indien de afmetingen van het object zeer klein zijn, zal de mean free path afstand relatief groot zijn. Daardoor zullen de lichtbronnen van de irradiance sampling techniek buiten het object geplaatst worden in plaats van binnenin het object, waardoor het resultaat foutief is. Ook als de afmetingen te groot worden, wordt het resultaat onstabiel. In de laatste figuur uit 7.13 zijn alle parameters hetzelfde als die in figuur 7.13(b), alleen werd de coëfficiënt g aangepast van 0.85 naar 0.0. Hierdoor wordt het resultaat ook weer onstabiel. Als de virtuele afmetingen veranderen zullen σ_s en σ_a van het materiaal ook veranderen. Samen met de coëfficiënt g zorgt dit ervoor dat de waarde van λ uit de Helmholtz operator verandert. Hieruit kan afgeleid worden dat de stabiliteit van de BKM ook beïnvloed wordt door deze λ .

We sluiten dit hoofdstuk af met een aantal resultaten waarbij de techniek werd toegepast op een aantal interessantere modellen. De bekomen resultaten worden weergegeven in figuur 7.14.

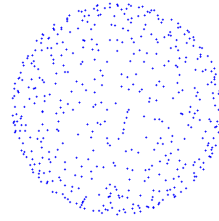
¹Het gebruikte systeem is een laptop met een Intel Core 2 Duo processor van 2.00GHz en 2.0GB geheugen.



Figuur 7.1: Deze figuur toont de resultaten van het toepassen van de BKM op een analytisch verifieerbare vergelijking. Van links naar rechts wordt eerst de exacte oplossing gegeven. Daarna de oplossing met de BKM en als laatste het verschil tussen deze beide oplossingen. Van boven naar onder worden respectievelijk de particuliere, de homogene en de gecombineerde oplossing gegeven.

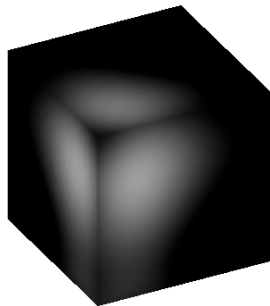


(a)

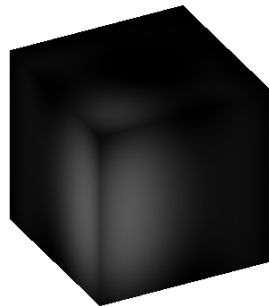


(b)

Figuur 7.2: Het resultaat van een bol ($r = 1$) te samplen op 484 locaties in PBRT. In figuur (a) worden de samples zonder jitter getoond. In figuur (b) de samples met jitter.

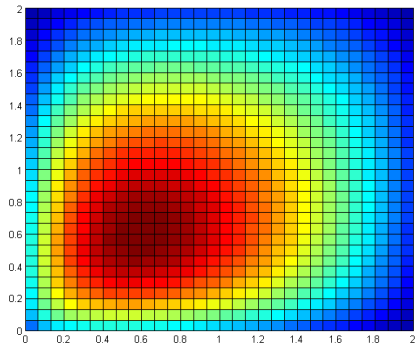


(a)

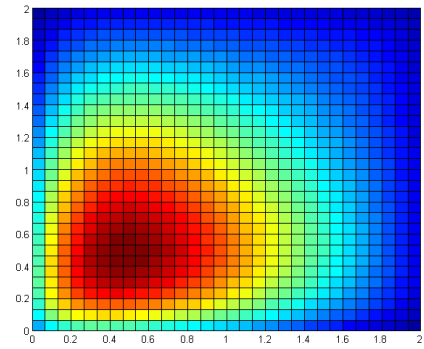


(b)

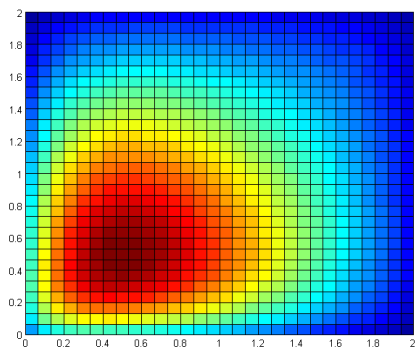
Figuur 7.3: Figuur (a) toont het resultaat waarbij de afstand met de op *floats* gebaseerde *Distance()* functie van de *Point* klasse van PBRT wordt berekend. Figuur (b) geeft het resultaat van de zelfgeschreven afstandsfunctie waarbij de coördinaten eerst worden omgezet naar *double* waardes. De lichtbron bevindt zich in het punt $(-1, -1, 1)$ en de uiterste punten van de kubus zijn respectievelijk $(0, 0, 0)$ en $(2, 2, 2)$.



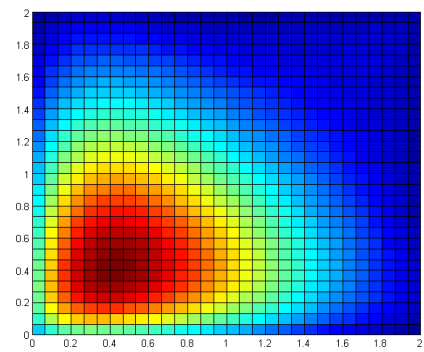
(a)



(b)

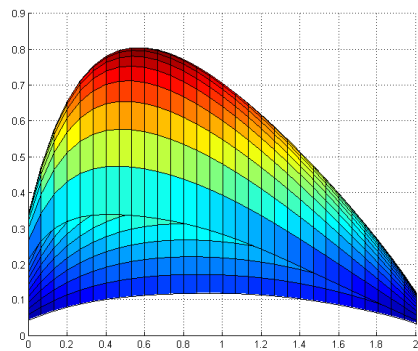


(c)

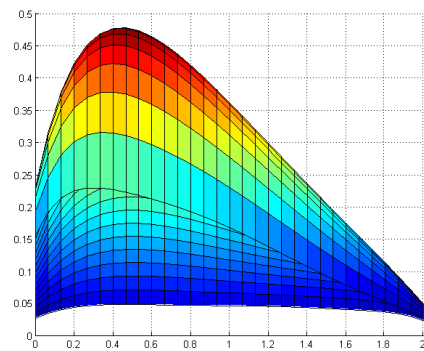


(d)

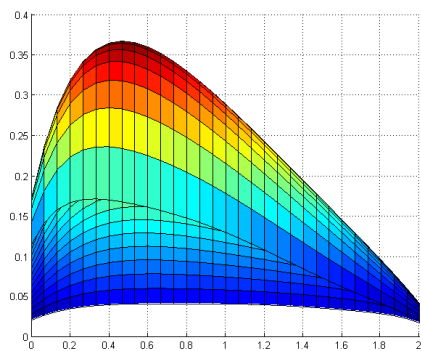
Figuur 7.4: De resultaten van de diffusion approximation met de BKM in een doorsnede van een kubus in MATLAB. In (a) werden er geen interne punten gebruikt. In (b) werden er bij de randpunten op een afstand van $\frac{3}{2}$ maal het mean free path extra punten geplaatst en in (c) zowel op $\frac{1}{2}$ als op $\frac{3}{2}$ maal het mean free path. In (d) werd er een grid van interne punten gebruikt.



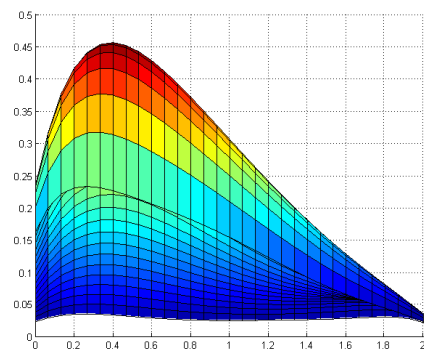
(a)



(b)

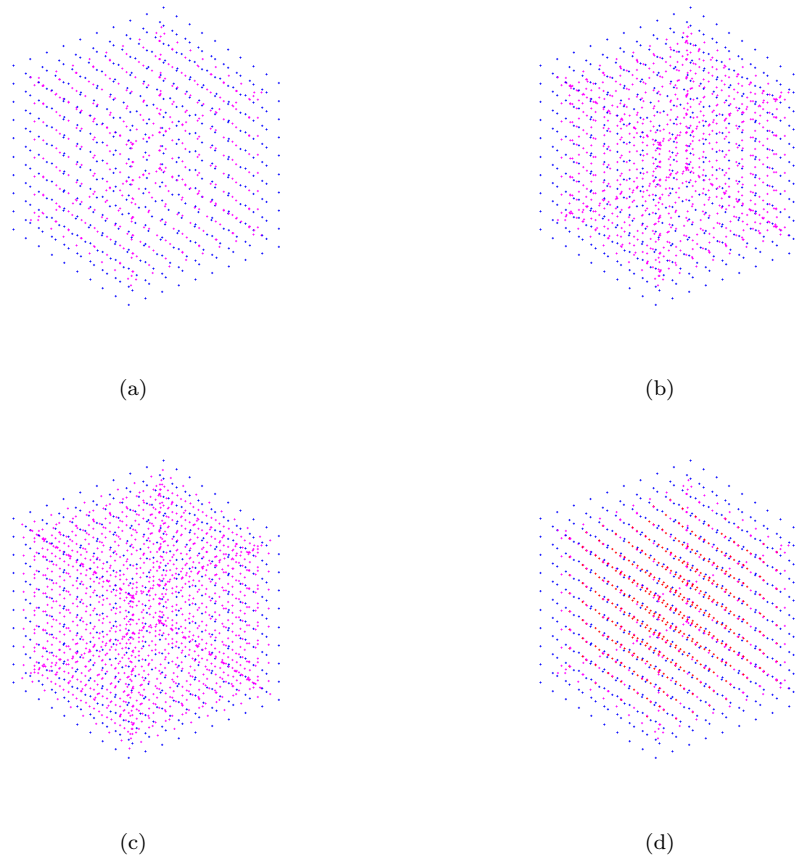


(c)

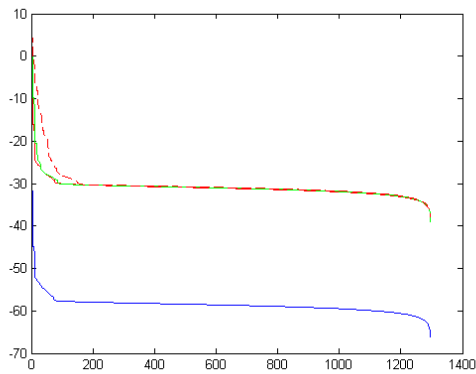


(d)

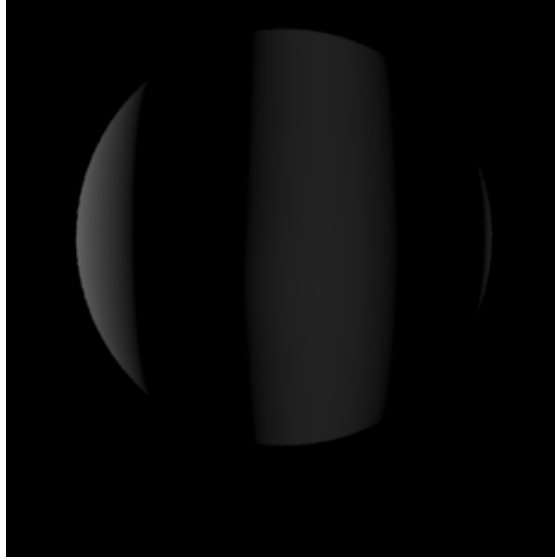
Figuur 7.5: Dezelfde resultaten als in figuur 7.4 maar dan bekenen in het XZ vlak in plaats van het XY vlak. Merk hierbij ook op dat de maximale waarden verschillend zijn.



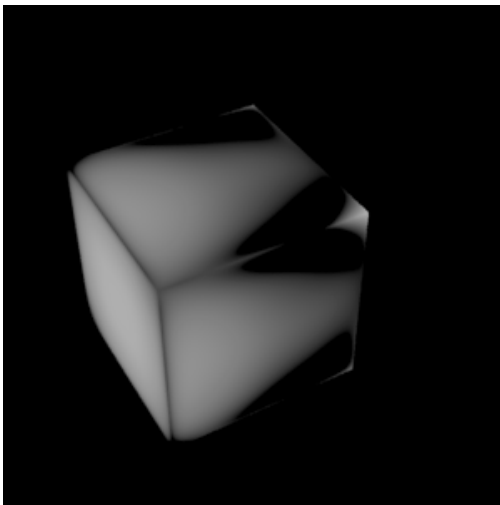
Figuur 7.6: De verschillende opstellingen van punten die gebruikt werden om de resultaten uit de figuren 7.4 en 7.5 te bekomen.



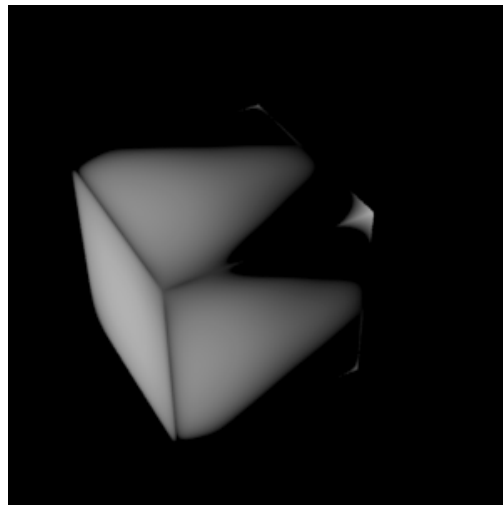
Figuur 7.7: Deze figuur toont het logaritme van de singuliere waarden van \mathcal{S} afkomstig van de SVD van de matrices die gebruikt werden om tabel 7.1 op te stellen. De rijen in deze tabel komen respectievelijk overeen met de rode, de groene, de rood gestreepte en de blauwe grafiek.



Figuur 7.8: In deze figuur werd een bol langs links belicht met een puntlichtbron. Als een gevolg van de onstabieliteit ontstaan er een soort oscillatieringen.

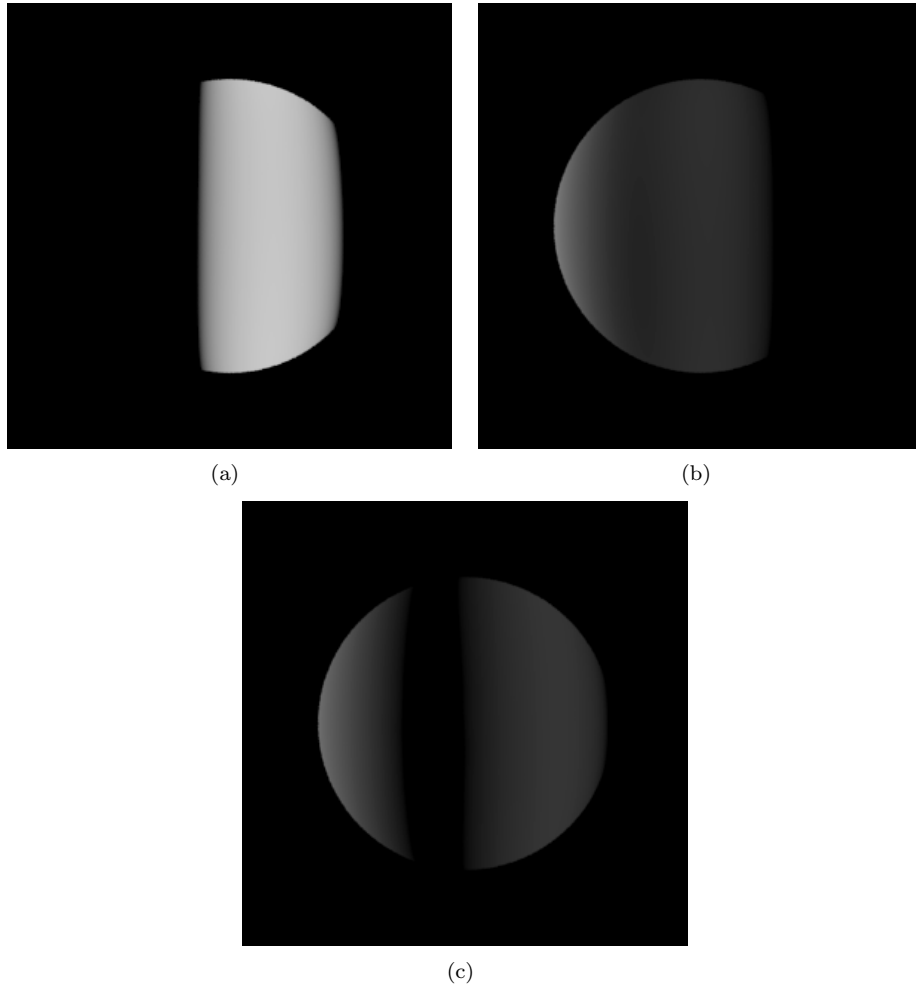


(a)

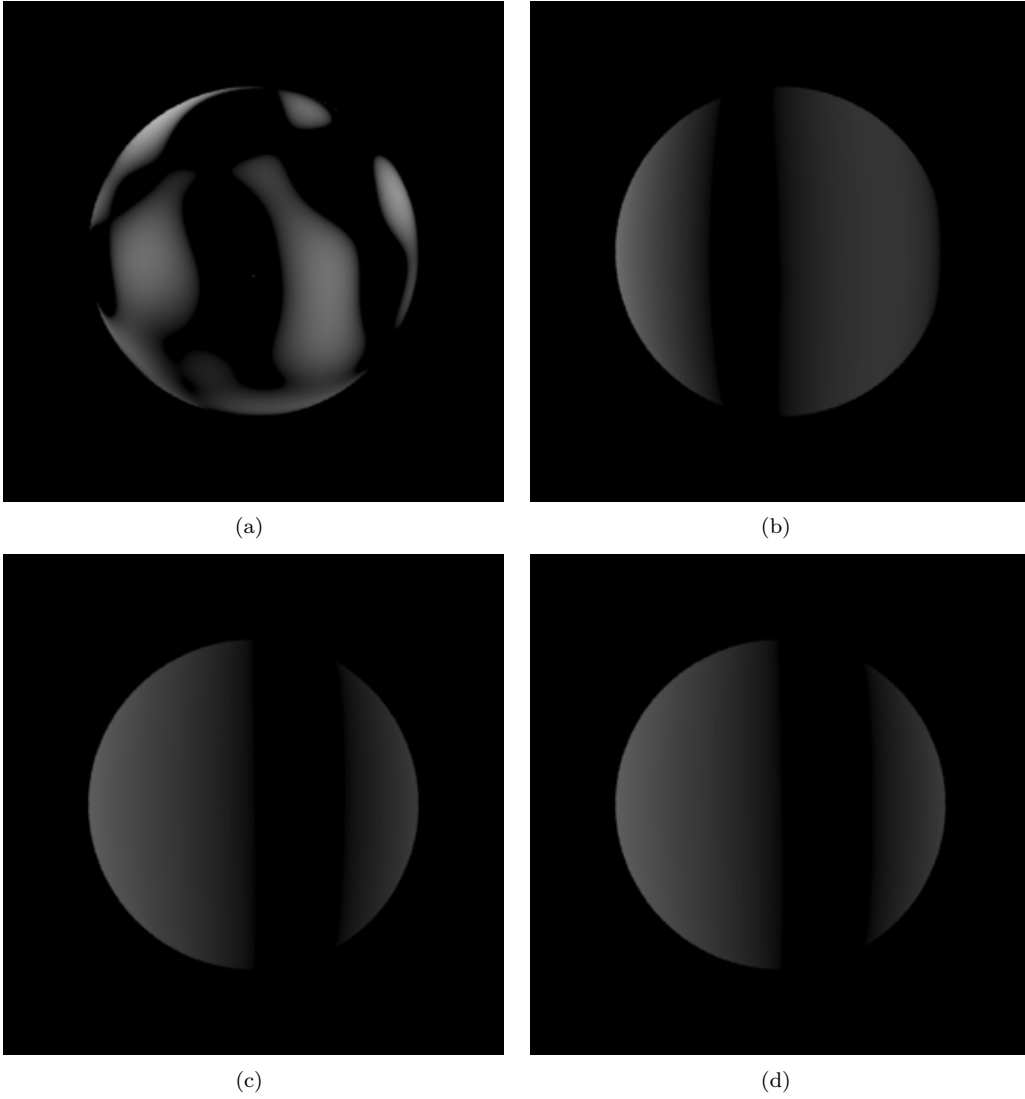


(b)

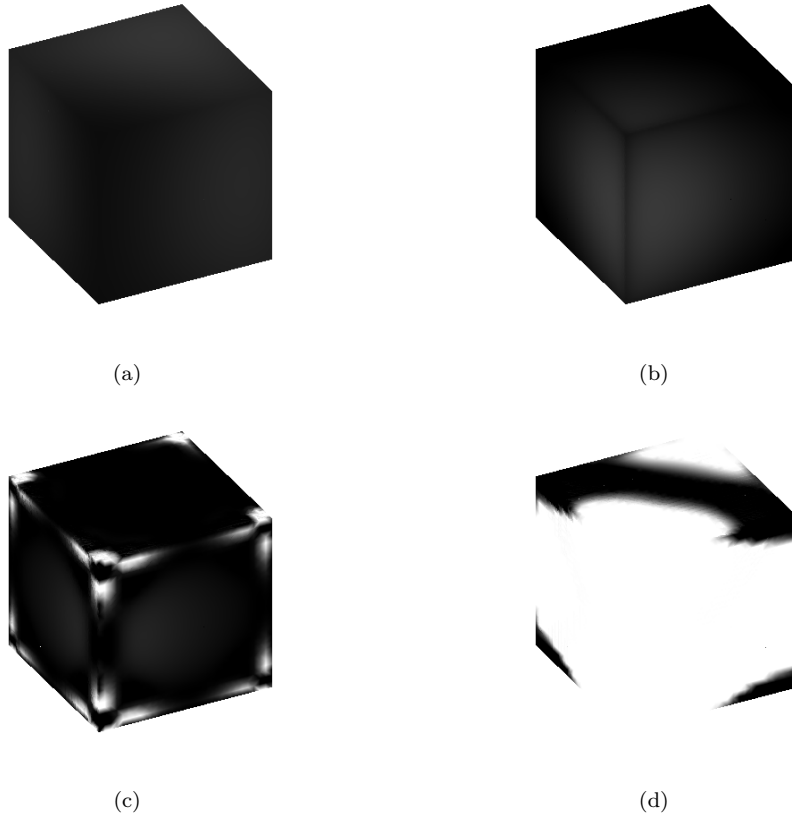
Figuur 7.9: Figuur (a) toont het resultaat waarbij een Gaussiaanse RBF met parameter $c = 500$ werd gebruikt. In figuur (b) werd een Invers Multiquadratische RBF met parameters $c = 1000$ en $\beta = 3$ gebruikt.



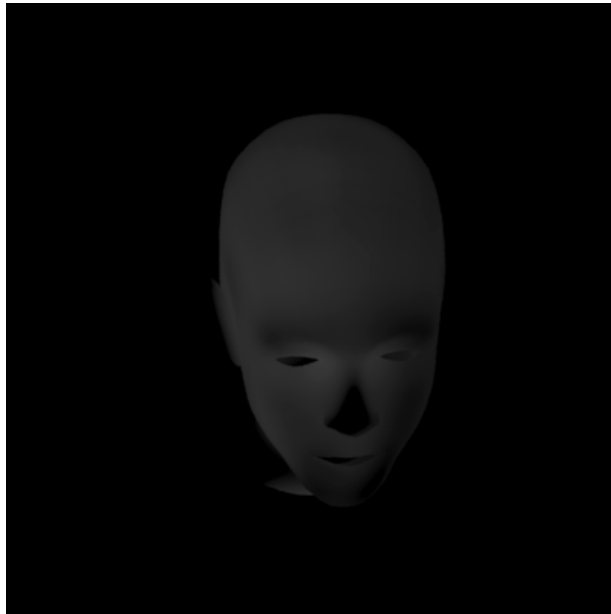
Figuur 7.10: In deze figuren wordt het effect van het veranderen van de parameter c van de Gaussiaanse RBF weergegeven. In figuur (a) is $c = 10$, in figuur (b) is $c = 100$ en in figuur (c) is $c = 500$.



Figuur 7.12: In deze figuren neemt het aantal gebruikte randpunten steeds toe. Voor de figuren (a), (b), (c) en (d) is het aantal randpunten respectievelijk 100, 484, 961 en 1764. Er kan gezien worden dat de oplossing convergeert naarmate het aantal randpunten toeneemt.



Figuur 7.13: In de figuren (a), (b) en (c) zijn de scattering-coëfficiënten vermenigvuldigd met een schaalfactor van respectievelijk 1.0, 10.0 en 100.0. In figuur (d) is de coëfficiënt g aangepast van 0.85 naar 0.0. De andere parameters zijn hetzelfde als die in figuur 7.13(b). Van deze figuren geeft enkel het resultaat bij een schaalfactor van 10.0 en $g = 0.85$ een redelijk resultaat.



(a)



(b)

Figuur 7.14: Het resultaat van het toepassen van de techniek op een hoofd model en het *killeroo* model.

Hoofdstuk 8

Conclusie en future work

8.1 Conclusie

Het idee dat er bij de BKM geen discretisatie nodig was van het model, maakte deze methode interessant om te onderzoeken. Bij de eerste testen van de BKM op een analytisch verifieerbare testfunctie waren de resultaten van deze methode veelbelovend en was het resultaat ook redelijk stabiel. Maar de combinatie van de Dirichlet en Neumann randvoorwaarden die voorkomen bij de partiële differentiaal vergelijking van de diffusion approximation, zorgden er echter voor dat de resultaten van de BKM onstabiel werden. De matrices die bij de diffusion approximation opgesteld werden, waren een stuk slechter geconditioneerd dan de matrices van de testfunctie. Dit had tot gevolg dat de matrices niet meer met de gebruikelijke methodes konden geïnverteerd worden. Ook het benaderen van de inverse met de pseudoinverse leverde in de meeste gevallen een onvoldoende correct resultaat. Als gevolg van deze grote onstabieleit moet geconcludeerd worden dat de uitgewerkte methode, zoals ze in deze thesis beschreven werd, niet geschikt is om het fenomeen van subsurface scattering van licht op te lossen.

8.2 Future work

Het grootste probleem van de BKM bij het oplossen van de partiële differentiaal vergelijking van de diffusion approximation, is dat de matrices moeilijk te inverteren zijn doordat ze zeer slecht geconditioneerd zijn. Indien dit probleem overwonnen kan worden, zou de BKM wel stabiele resultaten kunnen leveren. Daarom kan er verder gezocht worden naar meer geavanceerde methodes om slecht geconditioneerde matrices te inverteren. Een potentiële techniek hiervoor is deze die beschreven wordt in [13].

Omdat de parameter λ van de Helmholtz operator een grote invloed heeft op de stabiliteit van de oplossing, kan er geprobeerd worden om de waarde van λ in een beperkt, stabiel gebied te houden. Een methode die hiervoor misschien gebruikt kan worden is de schaal van de materiaaleigenschappen aan te passen. Door een juiste schaalfactor te gebruiken kunnen de materiaaleigenschappen en bijgevolg ook de waarde van λ , in een stabiel gebied gebracht worden. Maar het bepalen van deze schaalfactor is echter geen triviale zaak. Om een correct resultaat te

kunnen bekomen moet ook het licht geschaald worden. De effectiviteit van deze techniek moet eerst nader onderzocht worden.

In principe kan de BKM ook voor heterogene materialen gebruikt worden. Voor heterogene materialen veranderen de gebruikte formules, aangezien de aangepaste Helmholtz operator dan verandert in een Berger operator. Bijgevolg zal ook een andere niet-singuliere, algemene oplossing gebruikt moeten worden. Er kan ook nog een andere methode, zoals bijvoorbeeld de photon mapping methode, uitgetoetst worden in plaats van de irradiance sampling techniek. Hierbij moet er dan ook rekening gehouden worden dat de bepaling van de bronterm verandert. Het effect van de photon mapping methode te gebruiken in plaats van de irradiance sampling methode bij de uitgewerkte methode, kan dus ook nog onderzocht worden.

In [5] wordt de *Boundary Particle Method* (BPM) beschreven. Deze methode kan gezien worden als een soort opvolger van de *Boundary Knot Method*. De BPM is ook een meshless techniek, maar er zijn geen interne punten meer nodig bij inhomogene problemen. Ook wordt er maar één interpolatie matrix opgesteld, wat de rekentijd ten goede komt. Verder onderzoek moet uitwijzen als deze methode betere en stabielere resultaten oplevert dan de BKM bij de diffusion approximation.

Bibliografie

- [1] William F. Ames. *Numerical Methods for Partial Differential Equations*. Academic Press, New York, 1977.
- [2] W. T. Ang, David Laurence Clements, and N. Vahdata. A dual-reciprocity boundary element method for a class of elliptic boundary value problems for non-homogenous anisotropic media. In *Engineering Analyses With Boundary Elements*, 27, (1), pages 49–55. Elsevier, 2003.
- [3] A. Bilenca, A. Desjardins, B. Bouma, and G. Tearney. Multicanonical Monte-Carlo simulations of light propagation in biological media. In *Optics Express*, Vol. 13, Issue 24, pages 9822–9833. OSA, 2005.
- [4] D. S. Burnett. *Finite Element Analysis: From Concepts to Applications*. Addison Wesley, 1987.
- [5] Wen Chen. Meshfree boundary particle method applied to helmholtz problems. In *Engineering Analysis with Boundary Elements*, Volume 26, pages 577–581. Elsevier, 2002.
- [6] Wen Chen and Masataka Tanaka. New insights in boundary-only and domain-type RBF methods. In *International Journal of Nonlinear Sciences and Numerical Simulation*, pages 145–152. Freund Publishing House Ltd., 2000.
- [7] Wen Chen and Masataka Tanaka. A meshless, integration-free, and boundary-only RBF technique. In *Computers & Mathematics with Application*, 43, pages 379–391. Elsevier, 2002.
- [8] Michael F. Cohen and John R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press Profesional, Boston, MA, 1993.
- [9] Julie Dorsey, Alan Edelman, Justin Legakis, Hendrik Wann Jensen, and Hans Kohling Pedersen. Modeling and rendering of weathered stone. In *Siggraph 1999, Computer Graphics Proceedings*, pages 225–234, Los Angeles, 1999. Addison Wesley Longman.
- [10] Phil Dutre, Philippe Bekaert, and Kavita Bala. *Advanced Global Illumination, 1st Edition*. A K Peters, Natick, MA, 2003.
- [11] Albert Einstein. Über einen die erzeugung und verwandlung des liches betreffenden heuristischen gesichtspunkt. In *Annalen der Physik* 17, pages 132–148, Germany, 1905. Wiley-VCH Verlag GmbH & Co. KGaA.

- [12] Gregory E. Fasshauer. Solving partial differential equations by collocation with radial basis functions. In *Proceeding of Chamonix*, pages 1–8, Nashville, 1996. Vanderbilt University Press.
- [13] Bengt Fornberg and Grady Wright. Stable computation of multiquadric interpolants for all values of the shape parameter. In *Computer & Mathematics with Applications, Volume 48*, pages 853–867. Elsevier, 2004.
- [14] National Science Foundation. Blas (basic linear algebra subprograms). <http://www.netlib.org/blas/>, 2009.
- [15] National Science Foundation. Lapack – linear algebra package. <http://www.netlib.org/lapack/>, 2009.
- [16] Lothar Gaul, Martin Kögl, and Marcus Wagner. *Boundary Element Methods for Engineers and Scientists*. Springer, 2003.
- [17] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modeling the interaction of light between diffuse surfaces. In *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 213–222, New York, NY, USA, 1984. ACM.
- [18] Tom Haber. Rendering subsurface scattering for heterogeneous materials. Master’s thesis, School voor Informatie Technologie, Transnationale Universiteit Limburg, 2004.
- [19] Tom Haber, Tom Mertens, Philippe Bekaert, and Frank Van Reeth. A computational approach to simulate subsurface light diffusion in arbitrarily shaped objects. In *GI '05: Proceedings of Graphics Interface 2005*, pages 79–86, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005. Canadian Human-Computer Communications Society.
- [20] Wolfgang Hackbusch. Multi-grid methods and applications. In *Springer Series in Computational Mathematics, vol 4*, Berlin, 1985. Springer-Verlag.
- [21] Pat Hanrahan and Wolfgang Krueger. Reflection from layered surfaces due to subsurface scattering. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 165–174, New York, NY, USA, 1993. ACM.
- [22] Rolland L. Hardy. Multiquadric equations of topographic and other irregular surfaces. In *Journal of Geophysical Research*, 76, pages 1905–1915, 1971.
- [23] Christiaan Huygens. *Traité de la lumière*. 1690.
- [24] David S. Immel, Michael F. Cohen, and Donald P. Greenberg. A radiosity method for non-diffuse environments. In *SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 133–142, New York, NY, USA, 1986. ACM.
- [25] Akira Ishimaru. *Wave Propagation and Scattering in Random Media*, volume 1 and 2. Academic Press, New York, 1978.
- [26] Hendrik W. Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 511–518, New York, NY, USA, 2001. ACM.

- [27] Hendrik Wann Jensen. *Realistic image synthesis using photon mapping*. A. K. Peters, Ltd., Natick, MA, USA, 2001.
- [28] Hendrik Wann Jensen. Monte Carlo ray tracing. Technical report, SIGGRAPH 2003 Course 44, 2003.
- [29] Hendrik Wann Jensen and Juan Buhler. A rapid hierarchical rendering technique for translucent materials. In *SIGGRAPH '00: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 576–581, New York, NY, USA, 2002. ACM.
- [30] Hendrik Wann Jensen and Niels Jørgen Christensen. Photon maps in bidirectional Monte Carlo ray tracing of complex objects. In *Computers & Graphics, vol 19, 2*, pages 215–224, 1995.
- [31] Hendrik Wann Jensen and Per H. Christensen. Efficient simulation of light transport in scenes with participating media using photon maps. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 311–320, New York, NY, USA, 1998. ACM.
- [32] James Thomas Kajiya. The rendering equation. In *SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150, New York, NY, USA, 1986. ACM.
- [33] Ed J. Kansa. Multiquadrics – a scattered data approximation scheme with applications to computational fluid-dynamics – i surface approximations and partial derivative estimates. In *Computer & Mathematics with Applications, Volume 19, Issues 8-9*, pages 127–145. Elsevier, 1990.
- [34] Stephen Kirkup. *The Boundary Element Method in Acoustics*. Integrated Sound Software, 2007.
- [35] Hendrik P. A. Lensch, Michael Goesele, Philippe Bekaert, Jan Kautz, Marcus A. Magnor, Jochen Lang, and Hans-Peter Seidel. Interactive rendering of translucent objects. In *PG '02: Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, pages 214–224, Washington, DC, USA, 2002. IEEE Computer Society.
- [36] Itagaki M. Higher order three-dimensional fundamental solutions to the Helmholtz and the modified Helmholtz equations. 15:289–293, 1995.
- [37] The MathWorks Inc. MATLAB - Documentation, pinv. <http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/ref/pinv.html>, 2009. Moore-Penrose pseudoinverse of matrix.
- [38] James Clerk Maxwell. *A Treatise on Electricity and Magnetism*. 1873.
- [39] Tom Mertens, Jan Kautz, Philippe Bekaert, Hans-Peter Seidel, and Frank Van Reeth. Interactive rendering of translucent deformable objects. In *EGRW '03: Proceedings of the 14th Eurographics workshop on Rendering*, pages 130–140, Aire-la-Ville, Switzerland, 2003. Eurographics Association.
- [40] M. F. Modest. Backward Monte Carlo simulations in radiative heat transfer. In *Journal of heat transfer, vol. 125*, pages 57–62, New York, 2003. American Society of Mechanical Engineers.

- [41] Bryan S. Morse, Terry S. Yoo, Penny Rheingans, David T. Chen, and Kalpathi R. Subramanian. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In *Shape Modeling International*, pages 89–98. IEEE Computer Society, 2001.
- [42] Isaac Newton. *Opticks, or a Treatise of Reflections, Refractions, Inflections, and Colours of Light*. 1704.
- [43] P. W. Partridge, C. A. Brebbia, and L. W. Wrobel. *The Dual Reciprocity Boundary Element Method*. Computational Mechanics Publications, Southampton and Elsevier, London, 1992.
- [44] Matt Pharr and Pat Hanrahan. Monte Carlo evaluation of non-linear scattering equations for subsurface reflection. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 75–84, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
- [45] Matt Pharr and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
- [46] RWP-Simtec. FEM vs. FDM/FVM some essential differences. http://www.optimal-tech.com/download/publications/FEM-FDM_eng.pdf.
- [47] G. D. Smith. *Numerical Solution of Partial Differential Equations: Finite Difference Methods*. Oxford University Press, USA, 1986.
- [48] Jos Stam. Multiple scattering as a diffusion process. In P. M. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, pages 41–50, Dublin, Ireland, 1995. Springer-Verlag.
- [49] Christian Stimming. Lapack++. <http://sourceforge.net/projects/lapackpp/>, 2009.
- [50] H. C. van de Hulst. *Light Scattering by Small Particles*. Dover Publications, 1981.
- [51] Weisstein Eric W. Modified Bessel function of the first kind. <http://mathworld.wolfram.com/ModifiedBesselFunctionoftheFirstKind.html>, 04 2009. From MathWorld—A Wolfram Web Resource.
- [52] Weisstein Eric W. Modified spherical Bessel function of the first kind. <http://mathworld.wolfram.com/ModifiedSphericalBesselFunctionoftheFirstKind.html>, 04 2009. From MathWorld—A Wolfram Web Resource.
- [53] Rui Wang, John Tran, and David P. Luebke. All-frequency interactive relighting of translucent objects with single and multiple scattering. In *ACM Transaction on Graphics TOG, Volume 24*, pages 1202–1207, New York, NY, USA, 2005. ACM.
- [54] Holger Wendland. Piecewise polynomial, positive definite and compactly supported radial basis functions of minimal degree. In *Advances in Computational Mathematics 4*, pages 389–396. Springer, 1995.
- [55] S. Wolf, Th. Henning, and B. Stecklum. Multidimensional self-consistent radiative transfer simulations based on the Monte-Carlo method. In *Astronomy and Astrophysics, v.349*, pages 839–850. A&A, 1999.
- [56] Mao Xian Zhong. Radial basis function method for solving differential equations. Master’s thesis, City University of Hong Kong, Department of Mathematics, 2000.

- [57] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals, Sixth Edition*. Butterworth-Heinemann, 2005.