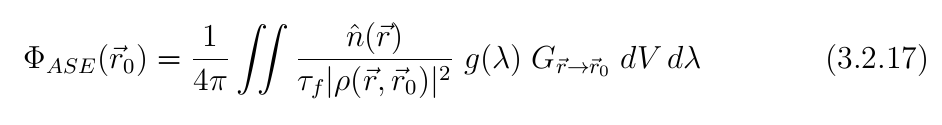
Beschreibung des bestehenden Codes - Einführung:

Eine Einführung in die Grundlagen findet man in der Dissertation von D. Albach im Kapitel 3 „Amplified Spontaneous Emission Management“. Die Dissertation ist frei unter

<http://tel.archives-ouvertes.fr/tel-00504915/>

verfügbar.

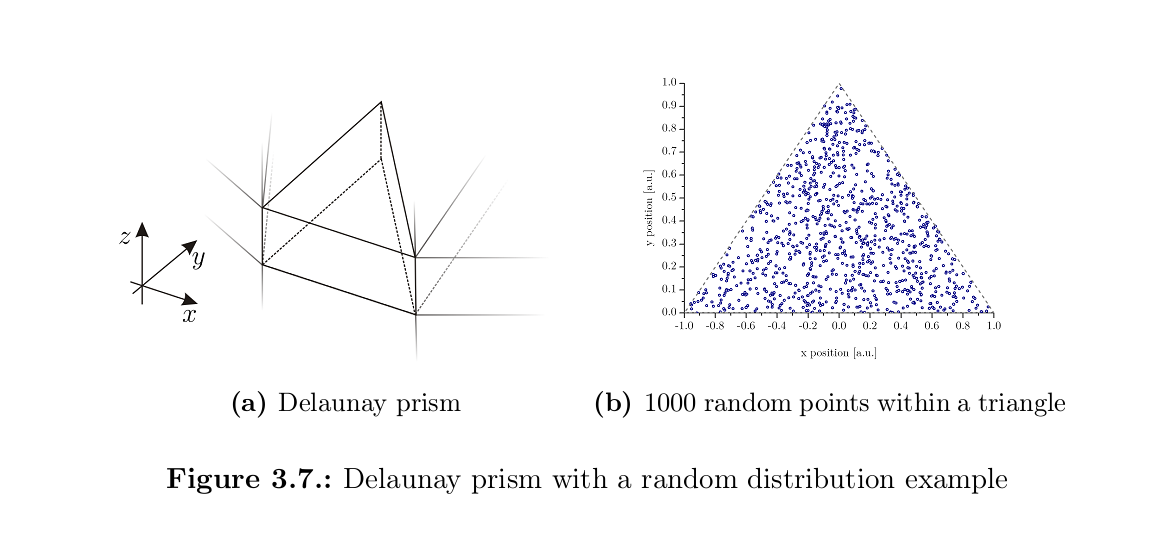
Insbesondere interessiert uns die Formel 3.2.17 (ASE-Flux im Punkt r0)



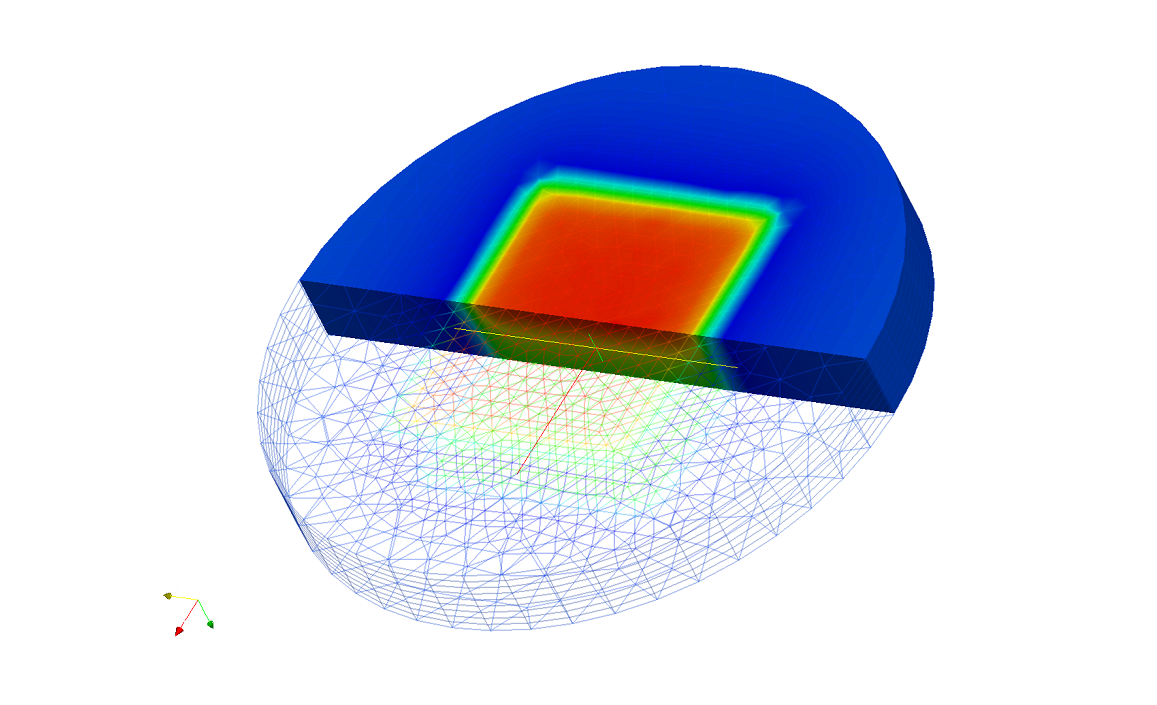
welche wir mittels numerischer Methoden für eine vorgegebene Geometrie zu lösen. Der Hauptgegenstand der Rechnung ist der Faktor G, welche die Verstärkung der spontanen Emission vom Punkt r zum Beobachtungspunkt r0 beschreibt. Dieses ist ein Linienintegral auf dem Weg r->r0.

Ein Ansatz ist es, dieses Integral mittels Monte-Carlo zu lösen. Eine Beschreibung befindet sich insbesondere in Kapitel 3.3.

Zusammengefasst geschieht dies folgendermaßen: Die Gleichung wird von einem Integral in eine Abschätzung auf Basis einer Summierung umgeschrieben (Integral zu einem Erwartungswert). Da nun die Geometrie numerisch erfasst wird, wird diese auch in Einzelpunkte und Zellen zerlegt. Im bestehenden Fall wird die Geometrie in Delaunay-Dreiecke zerlegt, welche danach extrudiert werden (also zu Prismen), womit man eine dreidimensionale Punktverteilung mit einem zugehörigen Gitter (Mesh) erhält.



Zufällig im Raum verteilte Emission propagiert von (Test-)Punkten (r) zum Beobachtungspunkt (r0) und ergibt schlussendlich eine Abschätzung von 3.2.17 für einen Punkt. Dies wird für alle Punkte des dreidimensionalen Gitters (Mesh) durchgeführt. Aus den daraus gegeben Punkten wird der Einfluss der verstärkten spontanen Emission (ASE) berechnet und in die Differentialgleichung für die Anregung in jedem der Gitterpunkte gelöst. Als Ergebnis findet man die Energieverteilung in einem dreidimensionalen Körper wieder.



Bestehender Code – Beschreibung der momentanen Probleme:

Im Moment wird aus rechenzeitlichen Gründen die Formel 3.2.17 als monochromatisch angenommen. Die Integration über die Wellenlänge findet nicht statt. Dies ist für den Fall einer sehr hohen Anregung in einem recht schmalbandigen Verstärkermaterial in Ordnung. Für breitbandige Materialien ist dies aber nicht mehr so einfach in Übereinstimmung zu bringen. Ebenso gibt es keinerlei Mehrfachreflexionen in dem Material.

Programmierungstechnisch sind dieses zwar keine Probleme, allerdings dauert die Berechnung sehr lange, was nicht im Verhältnis zwischen Nutzen und Aufwand für einige spezifische Anwendungen steht. Als Lösung bietet sich hier die Parallelisierung mittels GPUs an.

Umzusetzende Modifikationen am Code:

* Umsetzung der rechenintensiven Berechnung von einem seriellen C-Code in einen parallelen Code (GPU)
* Erweiterung auf eine polychromatische Rechnung
* Bonus: Reflexionen
* Superbonus : Rückreflexionen vom Rand

Kurze Diskussion des Codes:

Der Code (for\_loops\_clad.cpp) wird mittels MEX-Fähigkeit direkt von Matlab als Funktion aufgerufen. Zum Kompilieren wird der Mersenne Twister benötigt (mt19937ar).

Momentan befinden sich neben der Hauptfunktion (mexFunction) drei weitere Funktionen im Code:

1. LineIntegralMCRK4\_S: momentan nicht benutzt
2. propagation: der Propagations-Code, welcher Optimiert werden soll
3. importf: zur Berechnung des importance samplings (ruft propagation auf)

Der Hauptteil der bestehenden Rechnung findet in „propagation“ statt, welcher die Propagation in der dreidimensionalen Prismenaufteilung beschreibt.

Anhand der Start- und Endposition wird ein Vektor erzeugt (Zeile 299-308), welcher entweder eine Ebene (Zeile 396-426) nach Oben oder Unten, oder in ein über die lateralen Seiten benachbartes Prisma (Zeile 349-392) propagieren kann. Die sich bewegte Strecke wird ermittelt, die nächste Zelle ermittelt (switch-Anweisung Zeile 432-467), bestimmt ob das durchpropagierte Material ein Absorber oder Aktives Medium ist und die Verstärkung ermittelt (Zeile 476-481). Am Ende der Rechnung wird die Verstärkung zurückgegeben.

Dieser Bereich ist sehr rechenaufwändig und sollte gegen einen Raytracer, welcher das Linienintegral ausgibt ersetzt werden.

Bei der Abarbeitung aller Test-Strahlen für einen Punkt wird der nächste Punkt berechnet. Ist dies beendet, gibt die Routine den Wert für den ASE-Flux an Matlab (ASE\_calc\_clad.m) zurück.

Als Ausgabe stehen neben den gesicherten .mat-Informationen auch exportierte .vtk-Daten zur Visualisierung (z.B. mittels Paraview) bereit.