

# 1 Innledning

## Problembeskrivelse

- Simulering av lydbølger under vann
  - Mange anvendelser: Alt fra å finne ubåter til å legge kabler
- Svært beregningskrevende
  - Et utall modeller finnes, med varierende kompleksitet, ytelse og nøyaktighet
  - Mange anvendelser krever flere kjøringar med varierende parametre
    - for eksempel med forskjellig lydkilde-frekvens
- Mulig å bruke kraften i et moderne grafikkort?
  - Enormt ytelsespotensiale, men mange nye utfordringer

## Implementasjon: GPUwave

- Tittelen sier det meste
  - **wave**: Simulering av lydbølger under vann
  - **GPU**: ...på grafikkortet
- Proof-of-concept
  - Fokus på GPU-siden  $\Rightarrow$  kun én modell vurdert
  - Mangler brukervennlig grensesnitt og en del “polish”
  - Potensiale for ytelsesforbedring
- Fri programvare
  - Full kildekode tilgjengelig på <http://gpuwave.sesse.net/>

# 2 Split-step Fourier-metoden

## Split-step Fourier-metoden: Et kræsjskurs

- Bølgeligningen i sylinderkoordinater:

$$\frac{\partial^2 p}{\partial r^2} + \frac{1}{r} \frac{\partial p}{\partial r} + \frac{\partial^2 p}{\partial z^2} + k_0^2 \tilde{n}^2 p = 0$$

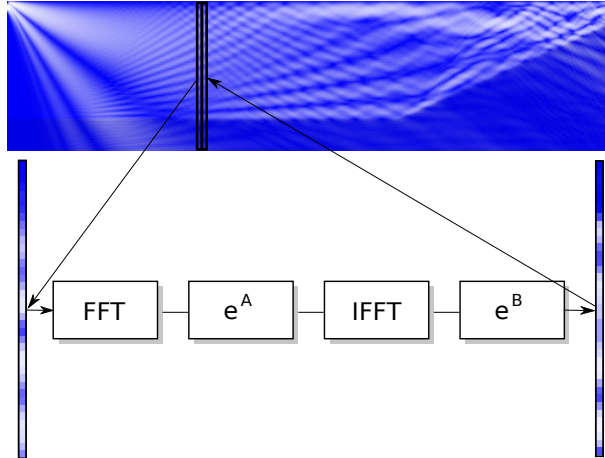
- “Vi ser lett at”

$$\psi(r + \Delta r, z) = e^{i(\tilde{n}^2(r_m, z) - 1) \frac{k_0}{2} \Delta r} \mathcal{F}^{-1} \left\{ e^{-ik_z^2 \frac{\Delta r}{k_0}} \mathcal{F} \{ \psi(r, z) \} \right\}$$

der

$$\psi(r, z) = \frac{p(r, z)}{H_0^{(1)}(k_0 r)}$$

...litt mer lettforståelig



### 3 GPU-programmering

Hva er spesielt med GPU-programmering?

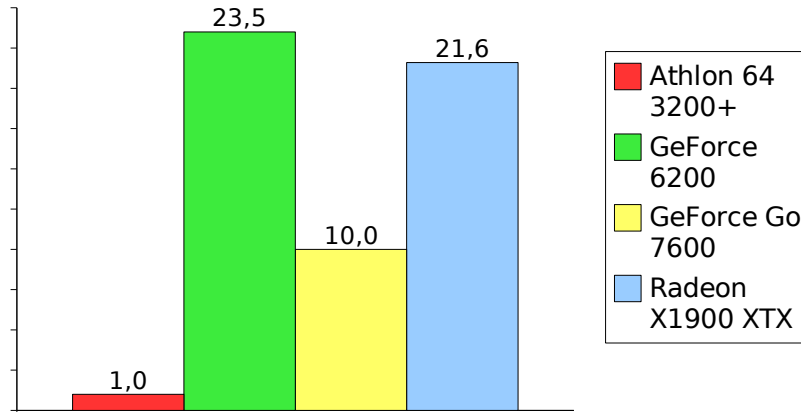
- “Streaming”-modell
  - Opprinnelig designet for spill — nå nyttig til det utroligste
  - En eller flere inn-vektorer  $\Rightarrow$  én ut-vektor fra  $\mathbb{R}^4$  (et **fragment**)
  - Kjører samme programsnitt mange ganger, bare med varierende inn- og utdata
  - Veldig begrensede muligheter for kontrollflyt
- GPUen er **massivt parallell**
  - Ikke uvanlig å beregne flere tusen datasett i samme gruppe
  - Utfordringen er å tilpasse problemet til GPU-modellen

Utfordringer i GPU-implementasjon

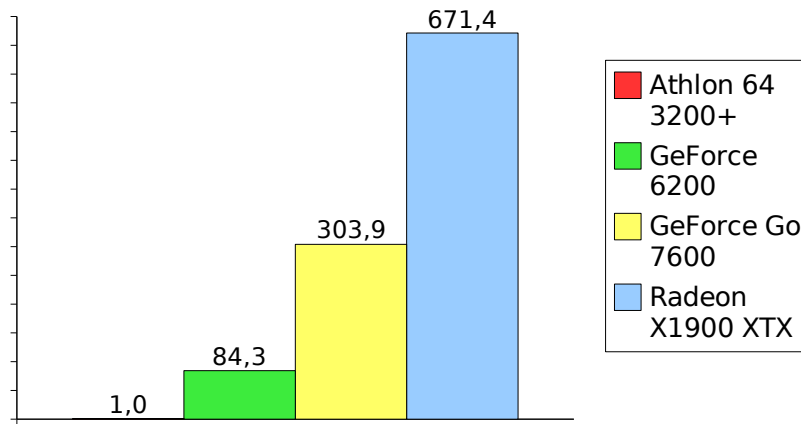
- Ønsker å maksimalisere ytelse
  - Pakking av data i minnet
  - Effektiv implementasjon av FST (Fast Sine Transform)
  - ...og en mengde andre praktiske småproblemer som til sammen betyr mye
- Må møte et visst minimumskrav for nøyaktighet
  - GPUen støtter generelt sett lavere nøyaktighet enn en typisk CPU
  - Kan gjøre en del avveininger mellom ytelse og nøyaktighet

## 4 Resultater

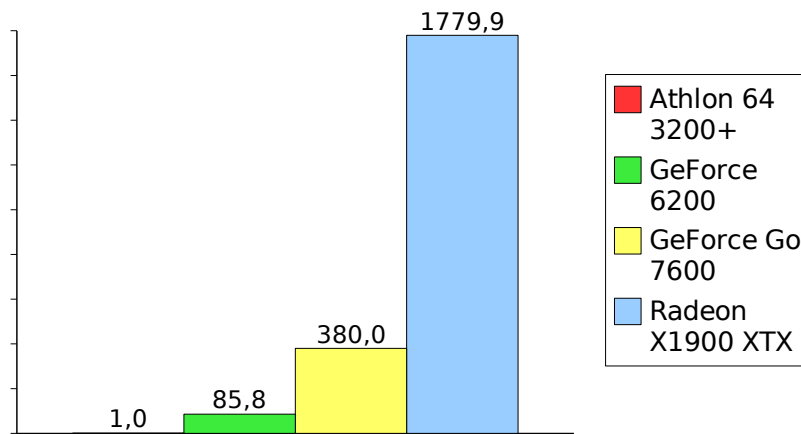
Relativ ytelse, enkeltfrekvens



Relativ ytelse, 16 frekvenser i parallell

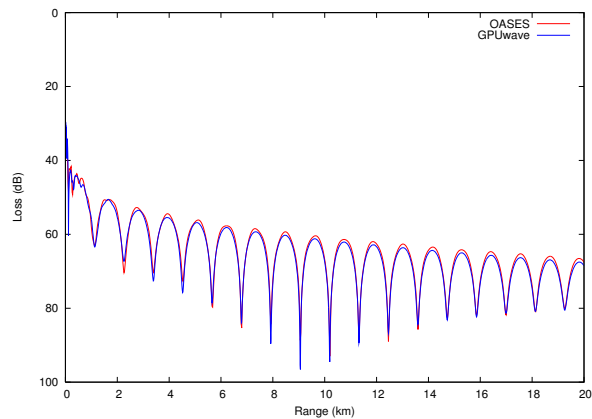


Relativ ytelse, 512 frekvenser i parallell



### Resultater: Nøyaktighet i forhold til andre modeller

- Sammenligning mot **Ocean Acoustics and Seismic Exploration Synthesis (OASES)**



### Konklusjoner

- Teknikk med enormt potensiale
  - Tre størrelsesordener forbedring i ytelse!
  - ...og fortsatt antageligvis mye å hente
- Nøyaktighet kan bli et problem i noen anvendelser
- Framtidig arbeid
  - Burde undersøke muligheten for implementasjon av mer sofistikerte metoder enn Fourier-step-metoder

- Integrasjon i større modeller
- Bruk av andre programmeringsmodeller

**Spørsmål?**

Prosjektrapport, programvare m.m. på <http://gpuwave.sesse.net/>