Statoil

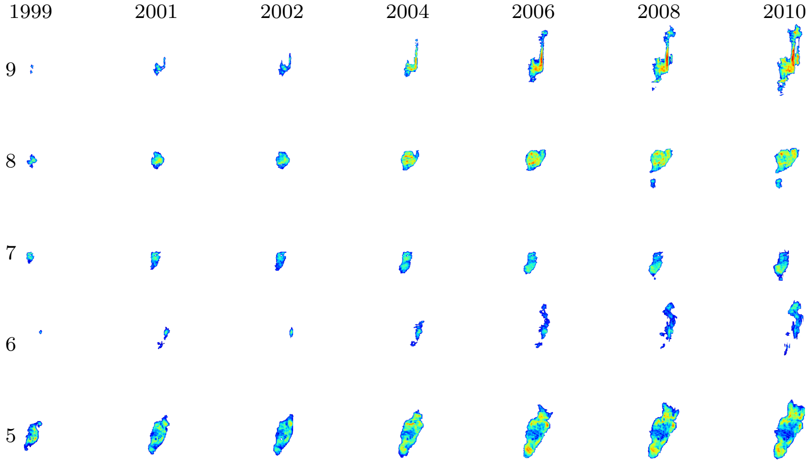
NTNU
Innovation and Creativity

# Interpolating subsequent 3D seismic data sets at the Sleipner $CO_2$ storage site

Anders Kiær

April 24th 2012

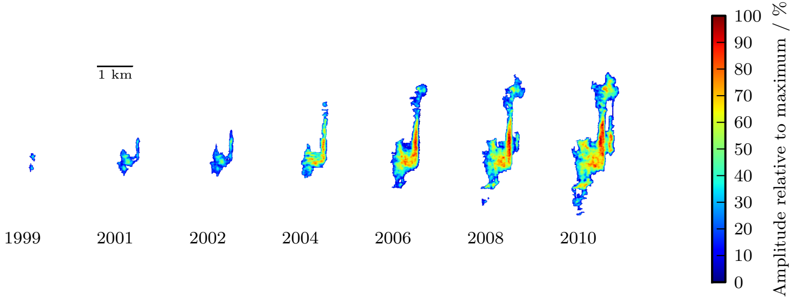
# Sleipner data set



# Problem statement

## Calendar time interpolation

Interpolate between amplitude maps from different calendar times.



# Why interpolate in calendar time?

## Main goals

- Estimates of amplitude maps inbetween surveys
  - at Sleipner there have been multiple gravity surveys and one CSEM survey performed at different times than the seismic surveys.
- Visualization purposes (movie, “cartoon”)
  - easy way to introduce the data set for a new audience
  - makes it easier to get a feeling of the actual front speed

# Existing methods

- History matching tools (e.g., reservoir simulators).
    - Requires a lot of computational time only to complete one forward simulation. In many cases it is also hard to match the observed seismic completely.
- Image morphing techniques (e.g., visual effects in movies)
    - May require considerable user input in order to have a satisfactory interpolation result.
- The level set method
    - Difficult to define the normal front propagation speed. Usually used for front *evolution* and not for interpolation.

# Outline of method

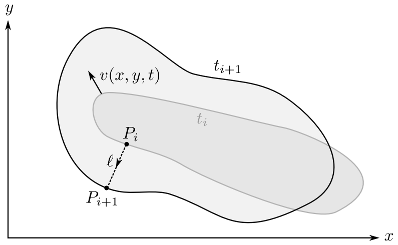
### Two parts

The interpolation method is divided into two parts.

1. The contour is interpolated.
2. The amplitude values are interpolated based on the contour interp.

# Notation

- As input, there are $S$ seismic data sets.
- The surveys have been taken at times $\mathcal{T} = \{t_1, t_2, ..., t_S\}$.
- From the data there are created $S$ different *shapes* $\mathcal{S}_i$.
- The *front* of $\mathcal{S}_i$ is written $\partial\mathcal{S}_i$.
- Normal front propagation speed is denoted by $v(x, y, t)$.



# Main assumption

**No "shape oscillations" during a given time interval $[t_i, t_{i+1}]$.**

In order to make the problem more tractable it is assumed that the front $\partial\mathcal{S}$ passes each point $(x, y)$ at most once during each time interval

$$[t_i, t_{i+1}], \quad i = 1, ..., (S-1).$$

This assumption alone makes it possible to write

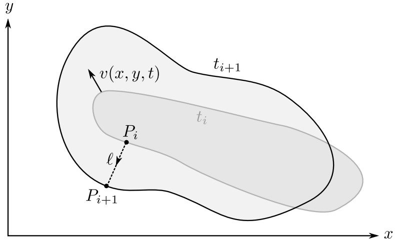
$$v(x, y, t) \equiv \begin{cases} v^{(1)}(x, y) & \text{for } t \in [t_1, t_2), \\ v^{(2)}(x, y) & \text{for } t \in [t_2, t_3), \\ \vdots & \\ v^{(S-1)}(x, y) & \text{for } t \in [t_{S-1}, t_S]. \end{cases}$$

# Traveltime calculation

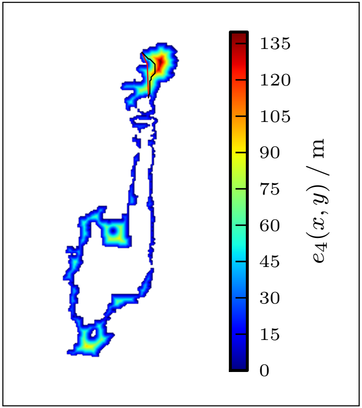
## Traveltime on a given path $\ell$

$$(\Delta t)_\ell = \int_{P_i}^{P_{i+1}} \frac{\mathrm{d}\boldsymbol{\ell}}{v^{(i)}(x,y)} \leq t_{i+1} - t_i \equiv t_i^*.$$

If $(\Delta t)_\ell \ll t_i^*$ then the final point is reached much faster than "necessary". Given the limited amount of information we have, a reasonable approximation for the travel time is $(\Delta t)_\ell \approx t_i^*$.



# Calculation of the paths $\ell$

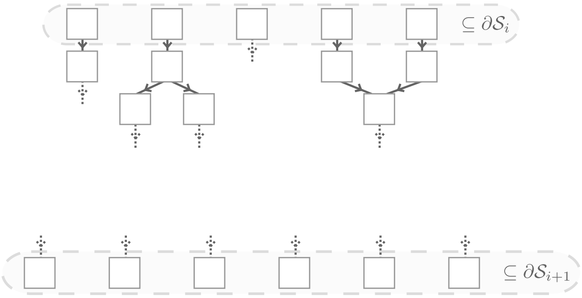



$$\int_{P_i}^{P_{i+1}} \mathrm{d}\ell \quad \boxed{\text{Euclidean}}$$

$$\Downarrow$$

$$\int_{P_i}^{P_{i+1}} \frac{\mathrm{d}\ell}{e_i(x, y)} \quad \boxed{\text{Non} - \text{Euclidean}}$$

# Calculation of the paths $\ell$




From the calculation of the paths we can construct a *directed acyclic graph* (DAG). By traversing this graph it is possible to construct a number of paths each starting out from $\partial\mathcal{S}_i$ and ending on $\partial\mathcal{S}_{i+1}$. By using the time estimates $(\Delta t)_\ell \approx t_i^*$ we can define the final solution

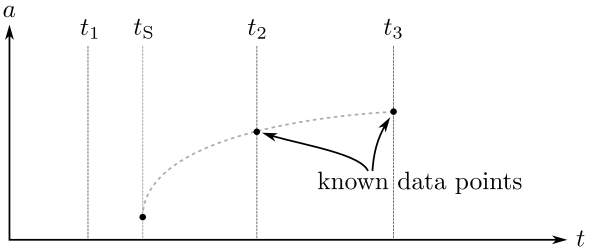
$$\mathbf{w} = \underset{\mathbf{w}}{\operatorname{argmin}} \left\| \begin{pmatrix} \mathsf{P} \\ \lambda \mathsf{T} \end{pmatrix} \mathbf{w} - \begin{pmatrix} \mathsf{t} \\ \mathbf{0} \end{pmatrix} \right\|^2, \quad \left( w_i \equiv v_i^{-1} \right).$$

# Outline of amplitude interpolation part

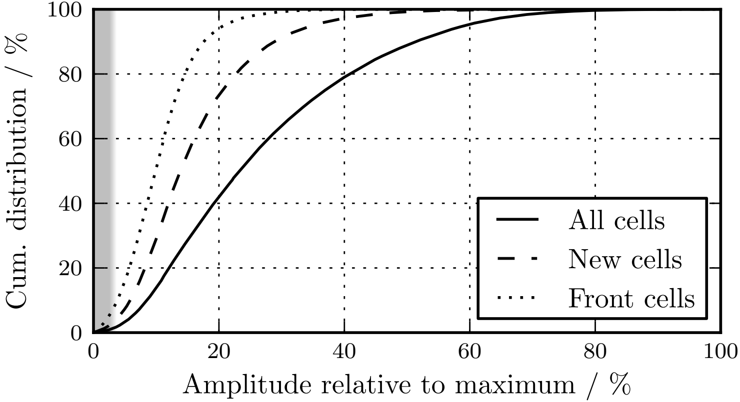
## Two parts

The amplitude interpolation is divided into two parts.

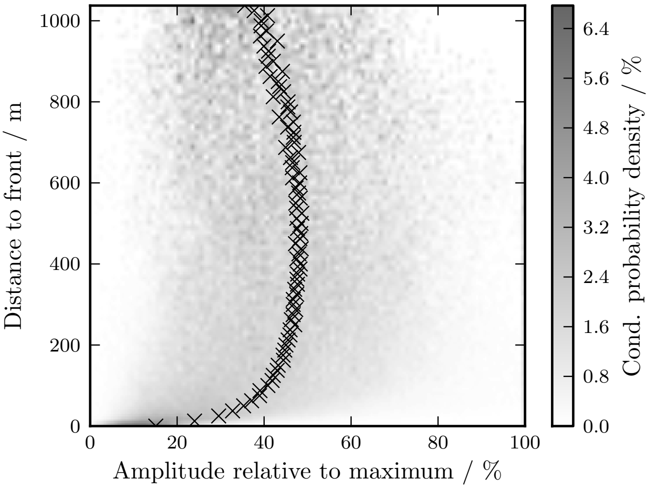
1. Define end point values.
2. Interpolate between end point values and known values from data.



# End point values



# Interpolation scheme



# Interpolation scheme

## A (possibly) non-linear scheme.

Choosing a linear interpolation scheme will introduce large artifacts if the amplitude evolution on average is highly non-linear. In order to take into account any non-linearities we look at the average amplitude among cells a distance $f$ from the front,

$$\mu(f) \equiv \frac{1}{S} \sum_{t_i \in \mathcal{T}} \int_0^\infty a \cdot p(a \,|\, f, t_i) \, \mathrm{d}a.$$

From the result of the contour interpolation part it is straightforward to calculate $f = f(t)$ for the given cell, hence it is possible to scale the composite function $\mu(f(t))$ such that $a(t)$ gets a similar time evolution as $\mu(f(t))$.

# Estimation of interpolation uncertainty

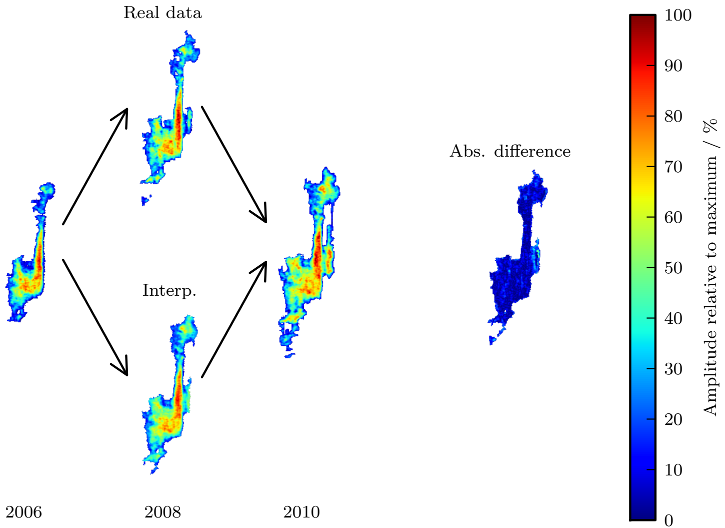
## Number of "error checks"

The interpolation uncertainty can be estimated by excluding data sets, interpolate between the remaining data sets, and then compare the excluded data sets with the interpolated values. It turns out to be

$$E(S) \equiv \sum_{k=1}^{S-1} \sum_{i=k+1}^{S} (i-k-1) = \frac{S(S-1)(S-2)}{6}$$

available "error checks" ($S \geq 2$). For the Sleipner data set, $E(7) = 35$.

# Estimation of interpolation uncertainty



# Acknowledgements


- *Problem statement:* Ola Eiken (Statoil).
- *Supervisors:* Ola Eiken (Statoil), Martin Landrø (NTNU).
- *Prepare Sleipner input data:* Anne-Kari Furre (Statoil).

Statoil

NTNU
Innovation and Creativity