F-K Domain Wavefield Continuation with Arbitrary Velocities

Gabriel Alvarez and Brad Artman

SEP120 page: 311-317

Email:gabriel@sep.stanford.edu–brad@sep.stanford.edu

Motivation

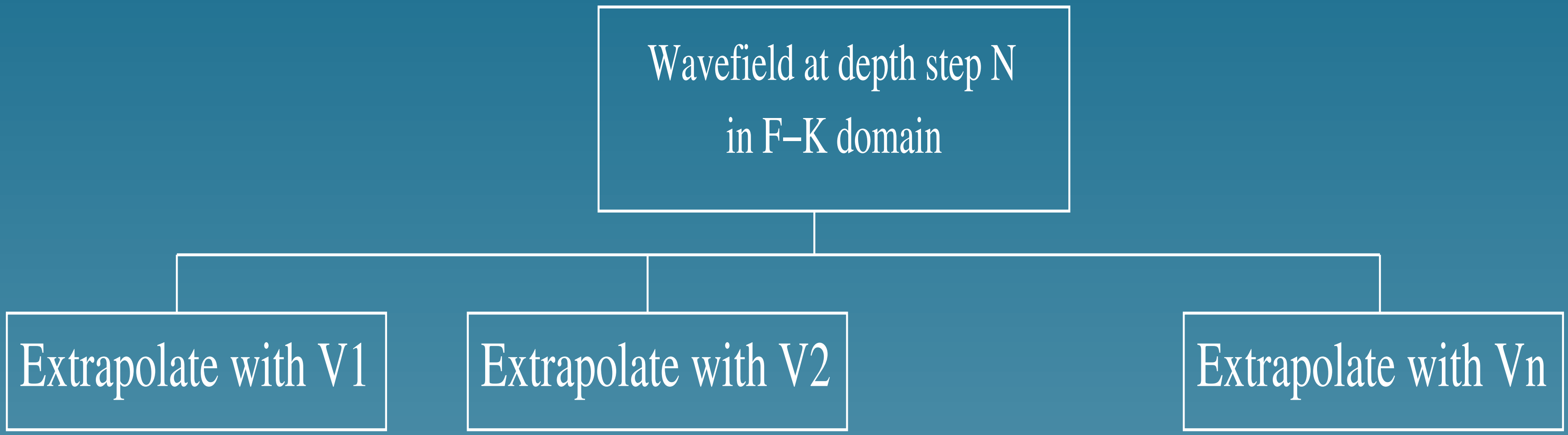
- Mixed-domain methods interpolate wavefields to account for laterally varying velocities. Can we interpolate phases instead?

Motivation

- Mixed-domain methods interpolate wavefields to account for laterally varying velocities. Can we interpolate phases instead?

- Interpolating phases will allow arbitrary velocity variations and a faster and simpler algorithm

Overview of PSPI (I)



Overview of PSPI (II)

In ω-$\mathbf{k}$ space:

$$\mathbf{W}_l^{N+1} = \mathbf{W}^N e^{ik_{z_l}\Delta z}$$

where:

$\mathbf{W}^N$: Wavefield at depth N

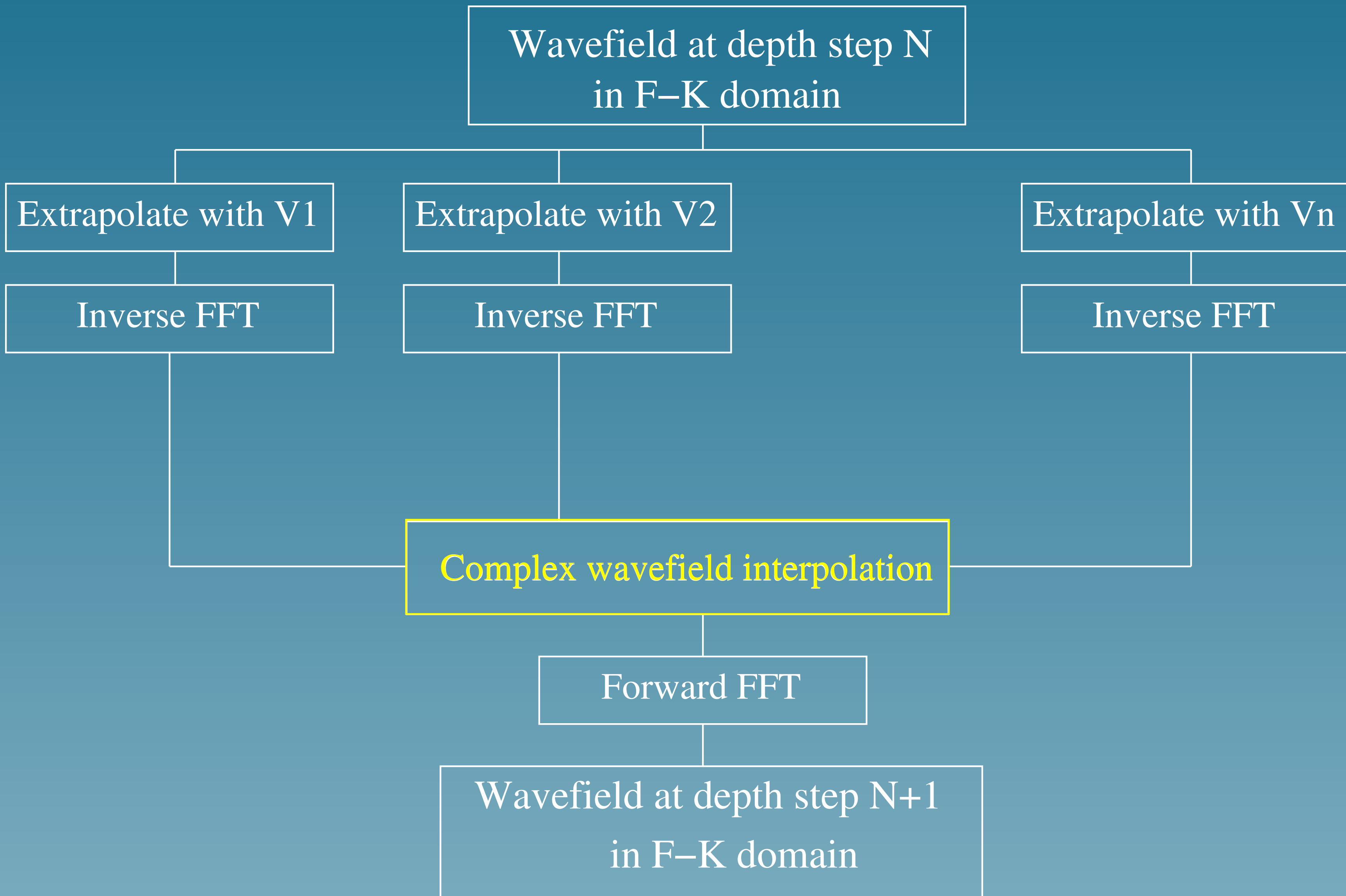
V_l: l-th Reference velocity

$\mathbf{W}_l^{N+1}$: Wavefield at depth $N+1$ continued with V_l

and

$k_{z_l} = \sqrt{\frac{\omega^2}{V_l^2} - |\mathbf{k}|^2}$ is the dispersion relation

Overview of PSPI (III)



Overview of PSPI (IV)

In ω-$\mathbf{x}$ space:

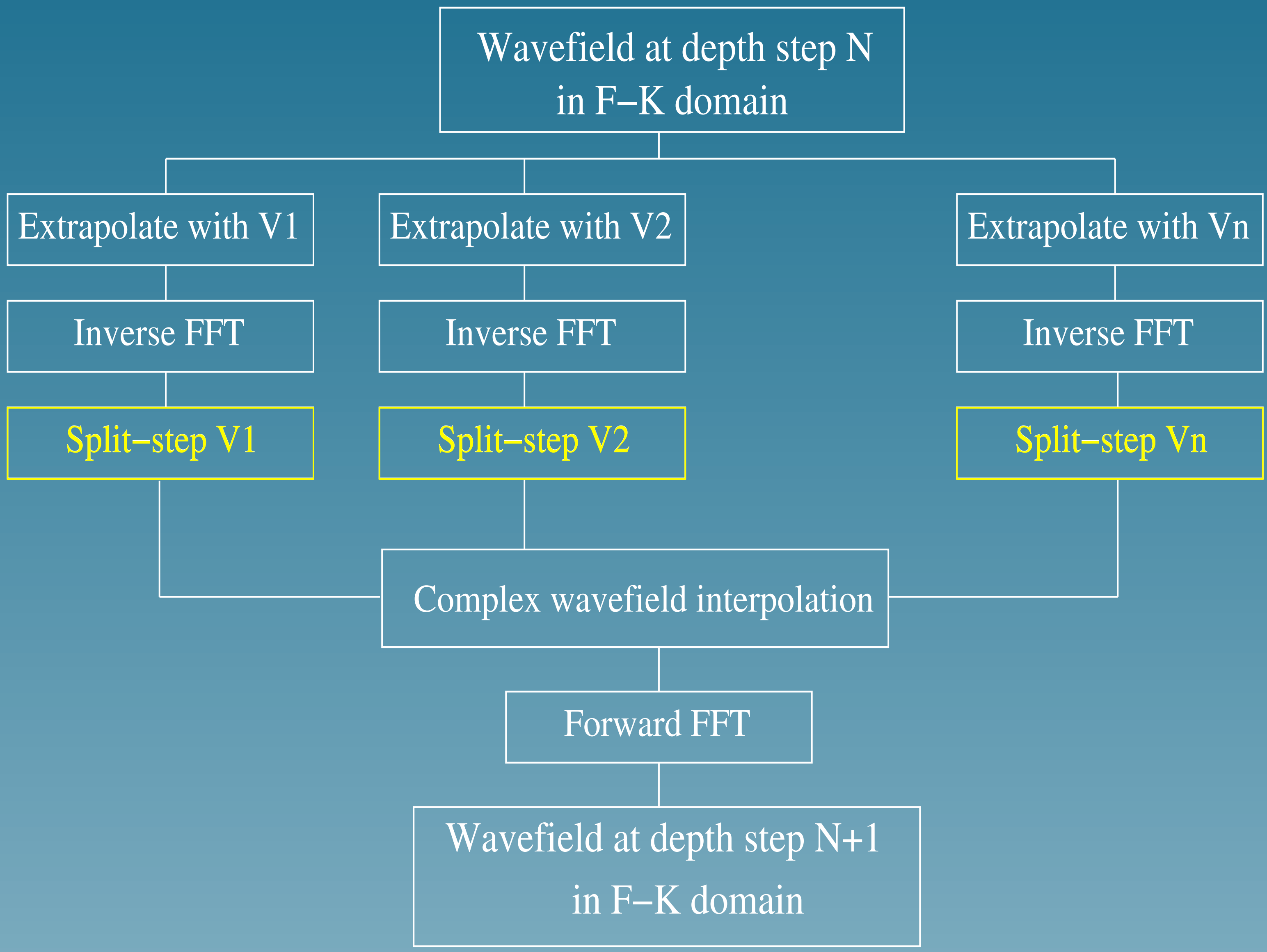
$$\mathbf{w}^{N+1}(j) = \sum_{l=1}^{nv} \sigma_l \mathbf{w}_l^N(j)$$

where:

σ_l: interpolation factor

$\mathbf{w}_l^{N+1}(j)$: wavefield in ω-$\mathbf{x}$ at location j

Overview of Extended Split-step



Overview of Split-step Correction

The split-step correction is given by:

$$e^{i\left(\frac{\omega}{V}-\frac{\omega}{V_l}\right)\Delta z},$$

where V is the true velocity is applied before the interpolation and is intended to compensate, to a first order, for the difference between V and V_l.

The Idea of Interpolating Phases (I)

In Phase shift extrapolation for $V(z)$:

$$\mathbf{W}^{N+1} = \mathbf{W}^{N} e^{i\theta_z}$$

In $V(\mathbf{x}, z)$ find an “equivalent” phase such that:

$$\mathbf{W}^{N+1} = \mathbf{W}^{N} e^{i\theta_{z_{\mathbf{eq}}}}$$

The Idea of Interpolating phases (II)

For any two complex numbers $z_1 = A\theta_1$ and $z_2 = A\theta_2$:

$$\Phi(\frac{z_1 + z_2}{2}) = \frac{\theta_1 + \theta_2}{2}$$

The Idea of Interpolating phases (II)

For any two complex numbers $z_1 = A\theta_1$ and $z_2 = A\theta_2$:

$$\Phi(\frac{z_1 + z_2}{2}) = \frac{\theta_1 + \theta_2}{2}$$

and

$$Amp(\frac{z_1 + z_2}{2}) = \frac{A}{\sqrt{2}}\sqrt{1 + \cos(\theta_2 - \theta_1)} \neq A$$

The Proposed Algorithm

Wavefield Extrapolation with Arbitrary Velocities in ω-$\mathbf{K}$

Basic Concept

- Assume that as many reference velocities as spatial locations are used at each depth step.

Basic Concept

- Assume that as many reference velocities as spatial locations are used at each depth step.
- $nv = n\mathbf{x}$.

Basic Concept

- Assume that as many reference velocities as spatial locations are used at each depth step.
- $nv = n\mathbf{x}$.
- No split-step correction is required.

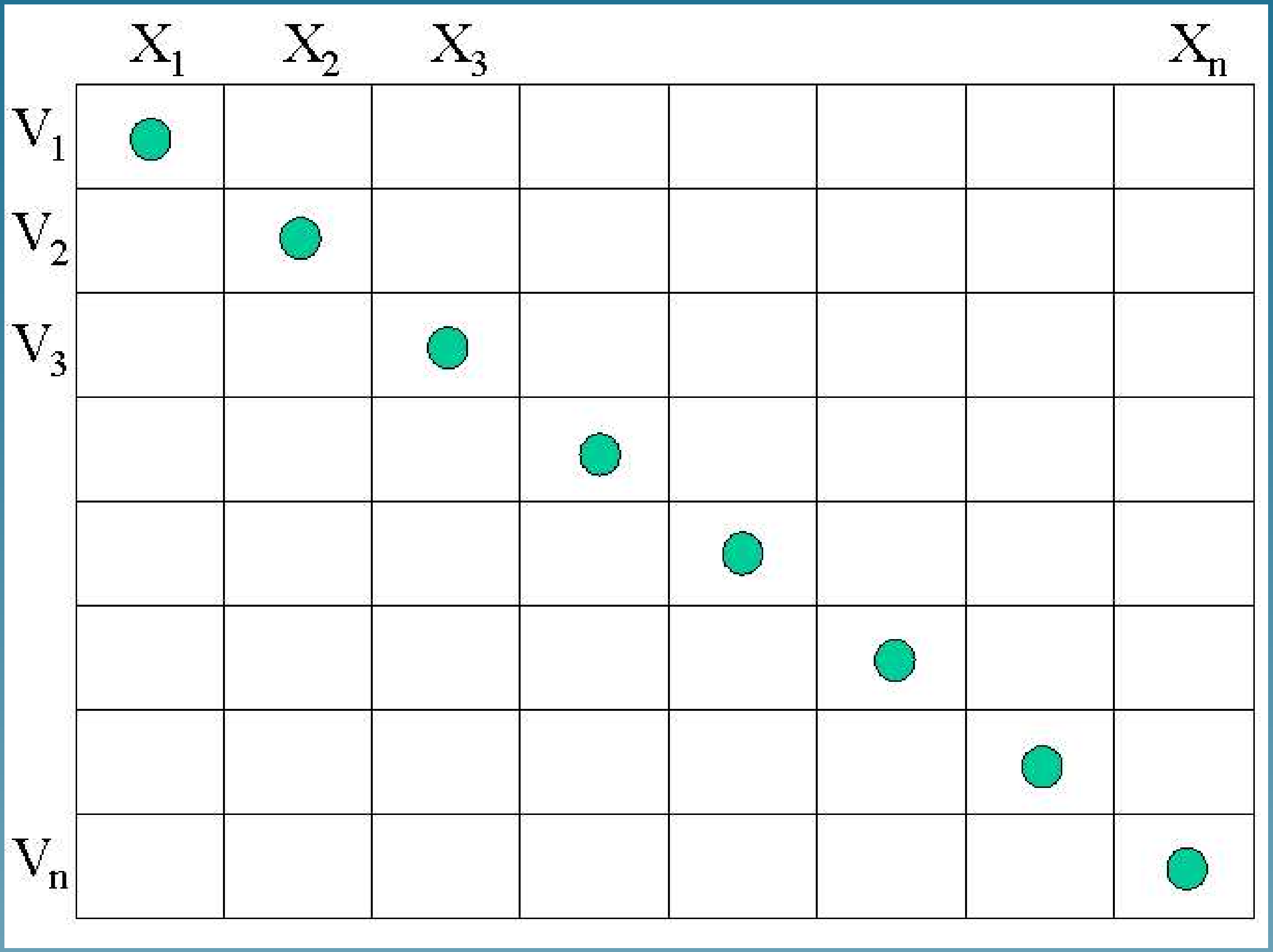
Basic Concept

- Assume that as many reference velocities as spatial locations are used at each depth step.
- $nv = n\mathbf{x}$.
- No split-step correction is required.
- No need for high-order approximation of the dispersion relation.

Basic Concept

- Assume that as many reference velocities as spatial locations are used at each depth step.
- $nv = n\mathbf{x}$.
- No split-step correction is required.
- No need for high-order approximation of the dispersion relation.
- Wavefield interpolation is replaced by selection.

Wavefield Selection



Each row is a wavefield extrapolated with the indicated velocity.

Extrapolated Wavefield in ω-$\mathbf{x}$

The selection process in the ω-$\mathbf{x}$ is given by:

$$\mathbf{w}^{N+1}(j) = \sum_{l=1}^{nv} \mathbf{w}_l^{N+1}(j)\delta_{lj}$$

where

$\mathbf{w}_l^{N+1}$: lth row in the array of extrapolated wavefields.

δ_{lj}: Kronecker delta to select the $j = l$ component.

Extrapolated Wavefield in ω-$\mathbf{K}$

The equivalent equation in the ω-$\mathbf{K}$ domain is:

$$\mathbf{W}^{N+1} = \sum_{l=1}^{nv} \mathbf{W}_l^{N+1} \otimes e^{-ik_x \Delta x_l}$$

where

$\Delta x_l = (l-1)\Delta x / nx$

$\otimes$: circular convolution

One spatial index is used to simplify the notation.

Extrapolated Wavefield in ω-$\mathbf{K}$ (II)

The extrapolated wavefield in ω-$\mathbf{K}$ is then:

$$\mathbf{W}^{N+1}(j) = \sum_{l=1}^{nv} \sum_{m=\langle nx \rangle} \mathbf{W}_l^{N+1}(m) e^{-ik_x(j-m)\Delta x_l}$$

Extrapolated Wavefield in ω-$\mathbf{K}$ (II)

The extrapolated wavefield in ω-$\mathbf{K}$ is then:

$$\mathbf{W}^{N+1}(j) = \sum_{l=1}^{nv} \sum_{m=\langle nx \rangle} \mathbf{W}_l^{N+1}(m) e^{-ik_x(j-m)\Delta x_l}$$

Replacing $\mathbf{W}_l^{N+1}$ in terms of $\mathbf{W}^N$ and rearranging terms:

$$\mathbf{W}^{N+1}(j) = \sum_{m=1}^{nx} \mathbf{W}^N(m) \sum_{l=1}^{nv} e^{-ik_{z_l}(m)\Delta z + k_x(\tilde{m}_j)\Delta x_l}$$

Extrapolated Wavefield in ω-$\mathbf{K}$ (II)

The extrapolated wavefield in ω-$\mathbf{K}$ is then:

$$\mathbf{W}^{N+1}(j) = \sum_{l=1}^{nv} \sum_{m=\langle nx \rangle} \mathbf{W}_l^{N+1}(m) e^{-ik_x(j-m)\Delta x_l}$$

Replacing $\mathbf{W}_l^{N+1}$ in terms of $\mathbf{W}^N$ and rearranging terms:

$$\mathbf{W}^{N+1}(j) = \sum_{m=1}^{nx} \mathbf{W}^N(m) \sum_{l=1}^{nv} e^{-ik_{z_l}(m)\Delta z + k_x(\tilde{m}_j)\Delta x_l}$$

where $\tilde{m}_j = mod(j-m, nx)$

Extrapolated Wavefield in ω-$\mathbf{K}$ (III)

Written as a dot product:

$$\mathbf{W}^{N+1}(j) = \sum_{m=1}^{nx} \mathbf{W}^{N}(m)\mathbf{f}_j(m) = \mathbf{W}^{N} \cdot \mathbf{f}_j.$$

Extrapolated Wavefield in ω-$\mathbf{K}$ (III)

Written as a dot product:

$$\mathbf{W}^{N+1}(j) = \sum_{m=1}^{nx} \mathbf{W}^{N}(m)\mathbf{f}_j(m) = \mathbf{W}^{N} \cdot \mathbf{f}_j.$$

The vector $\mathbf{f}_j$ is independent of the data and contains the velocity information:

$$\mathbf{f}_j = \sum_{l=1}^{nv} e^{-ik_{z_l}(m)\Delta z + k_x(\tilde{m}_j)\Delta x_l}.$$

Practical Implementation (I)

- The algorithm as described is cubic in the model dimensions.

Practical Implementation (I)

- The algorithm as described is cubic in the model dimensions.

$$\mathbf{W}^{N+1}(j) = \sum_{m=1}^{nx} \mathbf{W}^{N}(m) \sum_{l=1}^{nv} e^{-ik_{z_l}(m)\Delta z + k_x(\tilde{m}_j)\Delta x_l}$$

Practical Implementation (I)

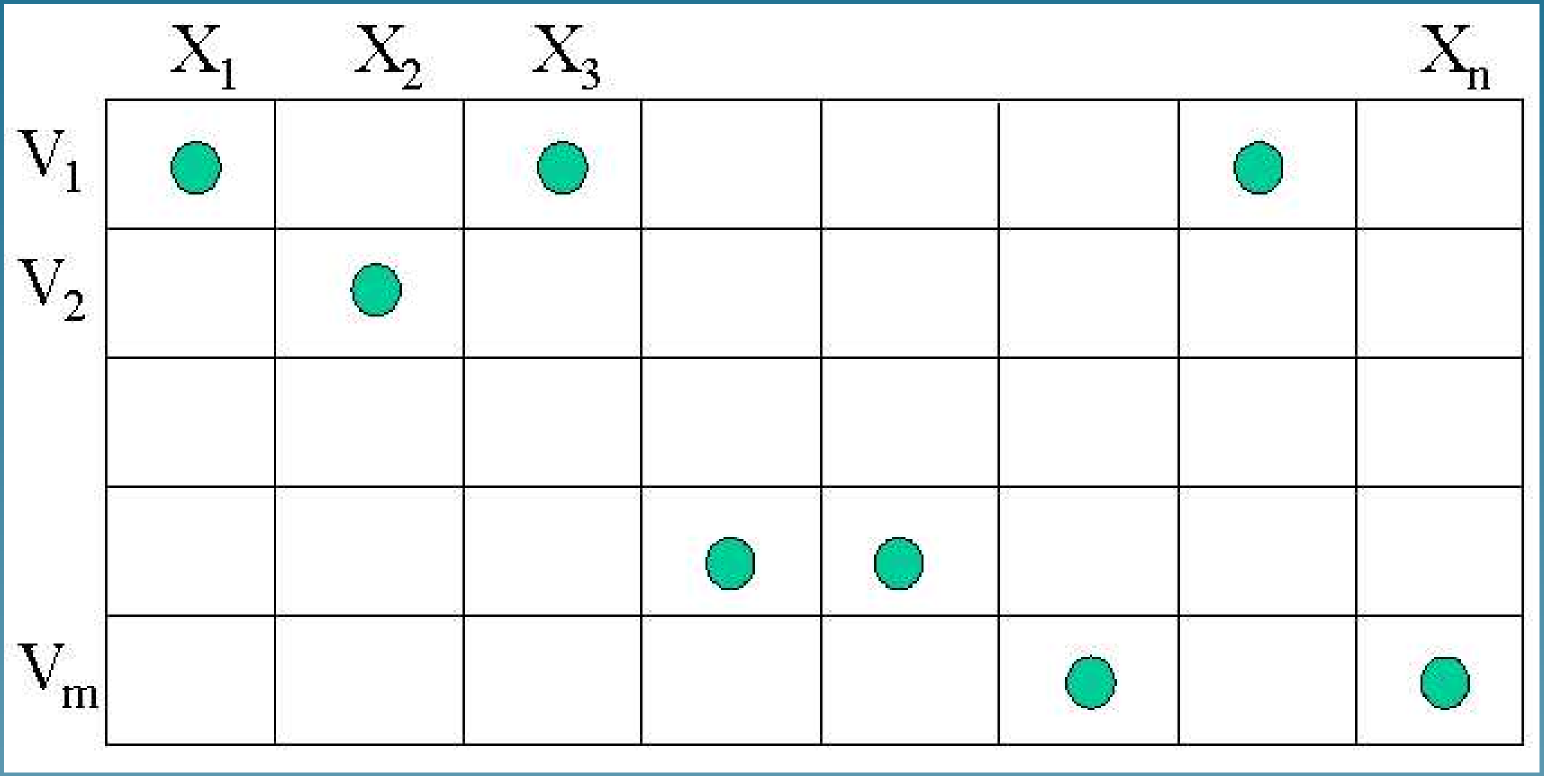
- The algorithm as described is cubic in the model dimensions.

$$\mathbf{W}^{N+1}(j) = \sum_{m=1}^{nx} \mathbf{W}^{N}(m) \sum_{l=1}^{nv} e^{-ik_{z_l}(m)\Delta z + k_x(\tilde{m}_j)\Delta x_l}$$

Practical Implementation (II)

- The algorithm can be made essentially quadratic by realizing that:
  - ⋆ Velocities can be binned to within their assumed accuracy.
  - ⋆ The vertical wavenumber can be precomputed, since it does not depend on velocity.

Modified Wavefield Selection



This time each row is an extrapolated wavefield with the indicated binned velocity.

Modified Wavefield in ω-$\mathbf{x}$

The selection process to calculate the wavefield in ω-$\mathbf{x}$ is now:

$$\mathbf{w}^{N+1} = \sum_{l=1}^{nv} \mathbf{w}_l^{N+1} \sum_p \delta_{pl}.$$

Modified Wavefield in ω-$\mathbf{x}$

The selection process to calculate the wavefield in ω-$\mathbf{x}$ is now:

$$\mathbf{w}^{N+1} = \sum_{l=1}^{nv} \mathbf{w}_l^{N+1} \sum_p \delta_{pl}.$$

l: velocity index.

p: index to select spatial locations with the same velocity.

Modified Wavefield in ω-$\mathbf{K}$

In the ω-$\mathbf{K}$ domain:

$$\mathbf{W}^{N+1}(j) = \sum_{m=1}^{nx} \mathbf{W}^{N}(m) \sum_{l=1}^{nv} \left(e^{-ik_{z_l}(m)\Delta z} \sum_{p} e^{-ik_x(\tilde{m}_j)\Delta x_p} \right)$$

Extrapolated Wavefield

Conceptually, the result is the same that we obtained before:

$$\mathbf{W}^{N+1}(j) = \mathbf{W}^{N} \cdot \mathbf{f}_j,$$

Extrapolated Wavefield

Conceptually, the result is the same that we obtained before:

$$\mathbf{W}^{N+1}(j) = \mathbf{W}^{N} \cdot \mathbf{f}_j,$$

only this time the vector $\mathbf{f}_j$ is given by

$$\mathbf{f}_j = \sum_{l=1}^{nv} \left(e^{-ik_{z_l}(m)\Delta z} \sum_p e^{-ik_x(\tilde{m}_j)\Delta x_p} \right).$$

Remarks

- The algorithm, as presented, is essentially quadratic in the model dimensions. Too slow for 3-D prestack depth migration.

Remarks

- The algorithm, as presented, is essentially quadratic in the model dimensions. Too slow for 3-D prestack depth migration.
- No significant approximations have been made.

Remarks

- The algorithm, as presented, is essentially quadratic in the model dimensions. Too slow for 3-D prestack depth migration.
- No significant approximations have been made.
- The cost comes from considering every wavefield trace in the computation of every other one.

Speculative Ideas on Improving Efficiency

- Subsample the wavefield used for the computation of each wavefield trace at the next depth step.

Speculative Ideas on Improving Efficiency

- Subsample the wavefield used for the computation of each wavefield trace at the next depth step.
- Compute only a subsampled version of the wavefield and interpolate.

Speculative Ideas on Improving Efficiency

- Subsample the wavefield used for the computation of each wavefield trace at the next depth step.
- Compute only a subsampled version of the wavefield and interpolate.
- Interpolate phases two by two.

Subsample the Input Wavefield

- For the computation of each wavefield trace at the $N + 1$ depth step use only, say, the even wavefield traces of the wavefield at the N depth step.

Subsample the Input Wavefield

- For the computation of each wavefield trace at the $N+1$ depth step use only, say, the even wavefield traces of the wavefield at the N depth step.

$$\mathbf{W}^{N+1}(j) = \sum_{m=1,2}^{nx} \mathbf{W}^{N}(m)\mathbf{f}_j(m),$$

Subsample the Input Wavefield

- For the computation of each wavefield trace at the $N+1$ depth step use only, say, the even wavefield traces of the wavefield at the N depth step.

$$\mathbf{W}^{N+1}(j) = \sum_{m=1,2}^{nx} \mathbf{W}^{N}(m)\mathbf{f}_j(m),$$

- Subsampling in wavenumber domain implies windowing in the space domain.

Subsample the Input Wavefield

- For the computation of each wavefield trace at the $N+1$ depth step use only, say, the even wavefield traces of the wavefield at the N depth step.

$$\mathbf{W}^{N+1}(j) = \sum_{m=1,2}^{nx} \mathbf{W}^{N}(m)\mathbf{f}_j(m),$$

- Subsampling in wavenumber domain implies windowing in the space domain.

- May be a better approximation at shallow than a at deeper depths.

Subsample the Computed Wavefield

- Compute only, say, every other wavefield trace at depth step $N+1$ using all traces of the wavefield at the N depth step.

Subsample the Computed Wavefield

- Compute only, say, every other wavefield trace at depth step $N+1$ using all traces of the wavefield at the N depth step.

$$\mathbf{W}^{N+1}(2j) = \sum_{m=1}^{nx} \mathbf{W}^{N}(m)\mathbf{f}_j(m),$$

Subsample the Computed Wavefield

- Compute only, say, every other wavefield trace at depth step $N+1$ using all traces of the wavefield at the N depth step.

$$\mathbf{W}^{N+1}(2j) = \sum_{m=1}^{nx} \mathbf{W}^{N}(m)\mathbf{f}_j(m),$$

- Linearly interpolate for the wavefield traces not computed.

Subsample the Computed Wavefield

- Compute only, say, every other wavefield trace at depth step $N+1$ using all traces of the wavefield at the N depth step.

$$\mathbf{W}^{N+1}(2j) = \sum_{m=1}^{nx} \mathbf{W}^{N}(m)\mathbf{f}_j(m),$$

- Linearly interpolate for the wavefield traces not computed.

- This implies that the wavefield is somewhat smooth in the spatial direction.

Interpolating Phases Two-by-Two

For any two phases θ_1 and θ_2:

$$\Phi\left(\frac{e^{i\theta_1}+e^{i\theta_2}}{2}\right) = \frac{\theta_1+\theta_2}{2}$$

Interpolating Phases Two-by-Two

For any two phases θ_1 and θ_2:

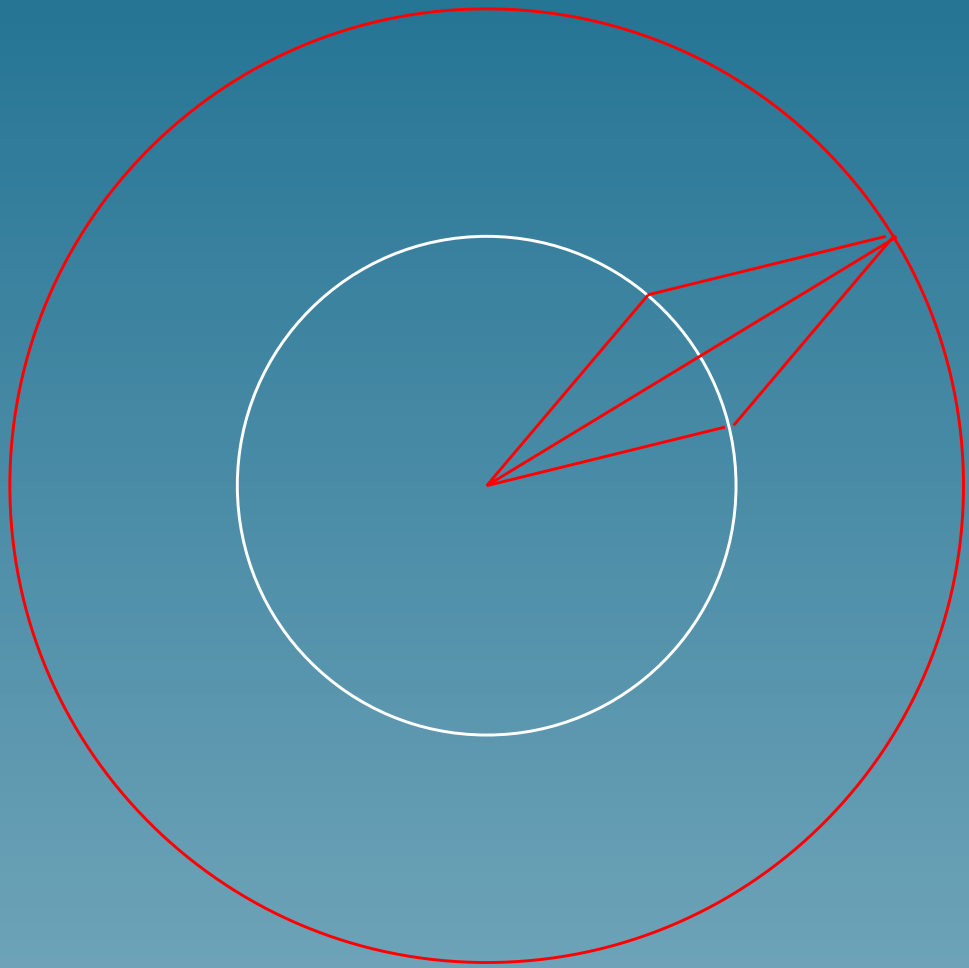
$$\Phi\left(\frac{e^{i\theta_1}+e^{i\theta_2}}{2}\right) = \frac{\theta_1+\theta_2}{2}$$

But

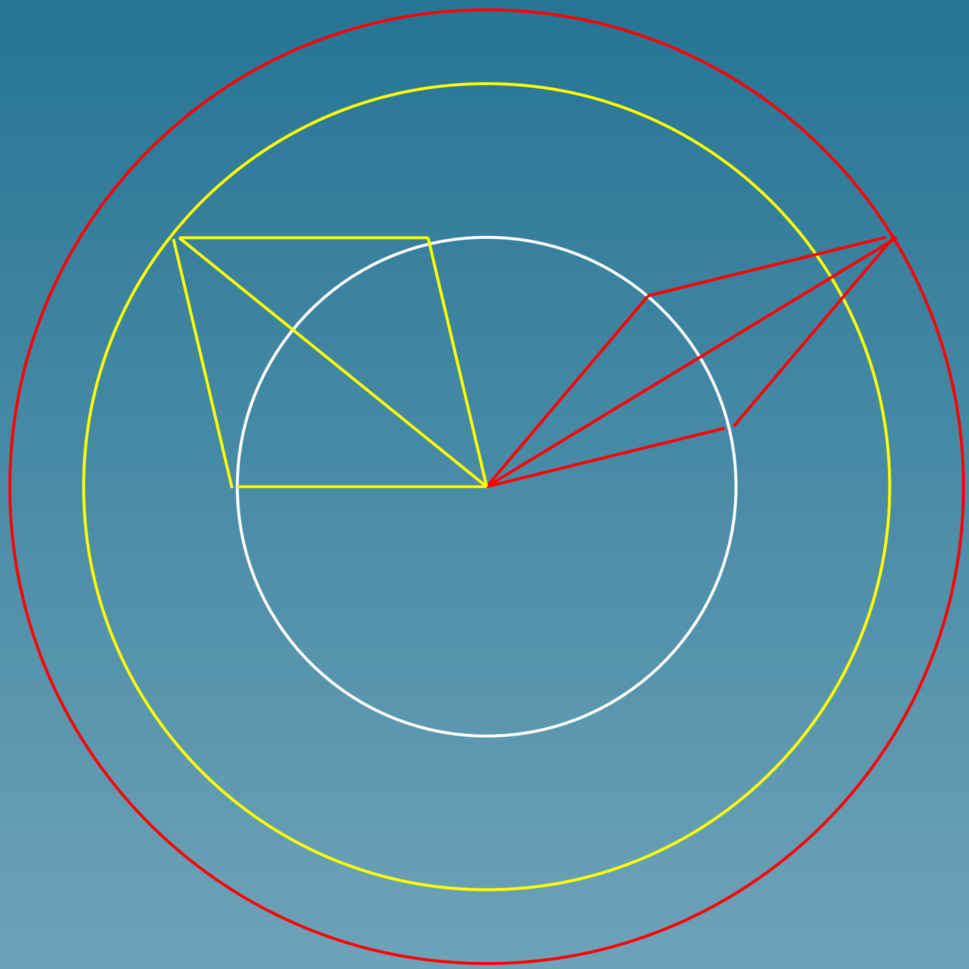
$$Amp\left(\frac{e^{i\theta_1}+e^{i\theta_2}}{2}\right) = \frac{1}{\sqrt{2}}\sqrt{1+\cos(\theta_2-\theta_1)} \neq 1$$

The question is: can we pair-up the sum of exponentials such that the amplitude term becomes a normalization?

Interpolating Phases Sketch



Interpolating Phases Sketch



Conclusions and Future Work

- It is possible to do F-K wavefield continuation with arbitrary spatial velocity variations.

Conclusions and Future Work

- It is possible to do F-K wavefield continuation with arbitrary spatial velocity variations.
- The resulting algorithm is quadratic in the spatial dimensions so needs to be made more efficient.

Conclusions and Future Work

- It is possible to do F-K wavefield continuation with arbitrary spatial velocity variations.
- The resulting algorithm is quadratic in the spatial dimensions so needs to be made more efficient.
- We have given some untested ideas on how to overcome the high cost. Testing those ideas is the next step.