*Stanford Exploration Project, Report SERGEY, November 9, 2000, pages 101–**??***

Short Note

On the general theory of data interpolation

Sergey Fomel[1]

INTRODUCTION

Data interpolation is one of the most important tasks in geophysical data processing. Its importance is increasing with the development of 3-D seismics, since most of the modern 3-D acquisition geometries carry non-uniform spatial distribution of seismic records. Without a careful interpolation, acquisition irregularities may lead to unwanted artifacts at the imaging step (Gardner and Canning, 1994; Chemingui and Biondi, 1996).

The interpolation problem in geophysics implies interpolating irregularly sampled data to a regular grid. In general, this problem requires a regularized inversion scheme, such as the method of inversion to zero offset (Ronen et al., 1991, 1995). Theoretically, it is easier to consider a different (in a sense, the opposite) problem: to find a continuous interpolation function for the data, given on a regular grid. The latter problem has been a traditional subject in computational mathematics. Though its solution is not directly applicable to the handling of irregular acquisition geometries, it can give us some insights into the possible ways of approaching geophysical interpolation.

In this paper, I present a simple mathematical theory of interpolation *from* a regular grid. I derive the main formulas from a very general idea of function bases. In conclusion, I discuss possible applications of the general theory in geophysics and, in particular, its relation to azimuth moveout (Biondi et al., 1996).

PROBLEM FORMULATION

Mathematical interpolation theory considers a function f, defined on a regular grid N. The problem is to find f in a continuum, which includes N. I am not defining the dimensionality of N and f here because it is not essential for the derivations. Most of the examples in this paper use one-dimensional functions, but the general theory applies equally well to a higher number of dimensions. Furthermore, I am not specifying the exact meaning of "regular grid", since it will become clear from the further analysis. The function f is assumed to belong to a Hilbert space with a defined dot product.

[1]**email:** sergey@sep.stanford.edu

If we restrict our consideration to a linear case, the desired solution will take the following general form

$$f(x) = \sum_{n \in N} W(x,n)\, f(n)\,, \tag{1}$$

where x is a point from the continuum, and $W(x,n)$ is a linear weight. If the grid N itself is considered as continuous, the sum in formula (1) transforms to an integral in dn. Two general properties of the linear weighting function $W(x,n)$ are evident from formula (1).

Property 1

$$W(n,n) = 1\,. \tag{2}$$

Equality (2) is necessary to assure that the interpolation of a single spike at some point n does not change the value $f(n)$ at the spike.

Property 2

$$\sum_{n \in N} W(x,n) = 1\,. \tag{3}$$

This property is the normalization condition. Formula (3) assures that interpolation of a constant function $f(n)$ remains constant.

One classic example of the interpolation weight $W(x,n)$ is the Lagrange polynomial, which has the form

$$W(x,n) = \prod_{i \neq n} \frac{(x-i)}{(n-i)}\,. \tag{4}$$

The Lagrange interpolation provides a unique polynomial, which goes exactly through the data points $f(n)$. The known numerical instabilities of Lagrange's interpolation have been overcome by various types of spline interpolation (de Boor, 1978). It is curious to note that the interpolation and finite-difference filters, developed by Karrenbach (1995) from a general approach of self-similar operators, reduce to a localized form of Lagrange polynomials. The local 1-point Lagrange interpolation is equivalent to the nearest-neighbor interpolation, defined by the formula

$$W(x,n) = \begin{cases} 1, & \text{for} \quad n-1/2 \leq x < n+1/2 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Likewise, the local 2-point Lagrange interpolation is equivalent to the linear interpolation, defined by the formula

$$W(x,n) = \begin{cases} 1-|x-n|, & \text{for} \quad n-1 \leq x < n+1 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

The Lagrange interpolators of higher order correspond to more complicated polynomials.

Function basis

To obtain the solution (1), let us assume the existence of a function basic $\{\psi_k(x)\}$, $k \in K$, such that the function $f(x)$ can be represented by a linear combination of the basis functions, as follows:

$$f(x) = \sum_{k \in K} c_k \psi_k(x) . \tag{7}$$

The linear coefficients c_k can be found by multiplying both sides of equation (7) by one of the basis functions (e.g. $\psi_j(x)$). Inverting the equality

$$\left(\psi_j(x), f(x)\right) = \sum_{k \in K} c_k \Psi_{jk} , \tag{8}$$

where the parentheses denote the dot product, and

$$\Psi_{jk} = \left(\psi_j(x), \psi_k(x)\right) , \tag{9}$$

gives us the following explicit expression for the coefficients c_k:

$$c_k = \sum_{j \in K} \Psi_{kj}^{-1} \left(\psi_j(x), f(x)\right) . \tag{10}$$

Here Ψ_{kj}^{-1} refers to the kj component of the matrix, inverse to Ψ. The matrix Ψ is invertible as long as the basis set of functions is linearly independent. In the special case of an orthonormal basis, Ψ reduces to the identity matrix:

$$\Psi_{jk} = \Psi_{kj}^{-1} = \delta_{jk} .$$

Equation (10) is a least-square estimate of the coefficients c_k. For a given set of basis functions, it approximates the function f in formula (1) in the least-square sense.

SOLUTION

The usual (although not unique) mathematical definition of the continuous dot product is

$$(f_1, f_2) = \int \bar{f_1}(x) f_2(x) dx , \tag{11}$$

where the bar over f_1 stands for complex conjugate (in the case of complex-valued functions.) Applying definition (11) to the dot product in formula (10) and approximating the integral by a finite sum on the regular grid N, we arrive at the approximate equality

$$(\psi_j(x), f(x)) = \int \bar{\psi_j}(x) f(x) dx \approx \sum_{n \in N} \bar{\psi_j}(n) f(n) . \tag{12}$$

We can consider equation (12) not only as a useful approximation, but also as an implicit *definition* of the regular grid. The grid regularity means that approximation (12) is possible. According to this definition, the more regular the grid is, the more accurate is the approximation.

Substituting equality (12) into formulas (10) and (7) gives us a solution to the interpolation problem. The solution takes the form of equation (1) with

$$W(x,n) = \sum_{k\in K}\sum_{j\in K} \Psi_{kj}^{-1}\psi_k(x)\bar{\psi_j}(n)\,. \qquad (13)$$

We have found a constructive way of creating the linear interpolation operator from a specified set of basis functions.

It is important to note that the adjoint of the linear operator in formula (1) is the continuous dot product of functions $W(x,n)$ and $f(x)$. This simple observation follows from the definition of the adjoint operator and the simple equality

$$\left(f_1(x), \sum_{n\in N} W(x,n)\, f_2(n)\right) = \sum_{n\in N} f_2(n)\,(f_1(x), W(x,n)) = \\ ((W(x,n),\, f_1(x)),\, f_2(n))\ , \qquad (14)$$

where we have assumed that the discrete dot product is defined by the sum

$$(f_1(n),\, f_2(n)) = \sum_{n\in N} \bar{f_1}(n)\, f_2(n)\,. \qquad (15)$$

Applying the adjoint interpolation operator to the function f, defined with the help of formula (13), and employing formulas (7) and (10), we discover that

$$(W(x,n),\, f(x)) = \sum_{k\in K}\sum_{j\in K} \Psi_{kj}^{-1}\bar{\psi_j}(n)(\psi_k(x),\, f(x)) = \\ \sum_{j\in K}\bar{\psi_j}(n)\sum_{k\in K}\Psi_{jk}^{-1}(\psi_k(x),\, f(x)) = \sum_{j\in K} c_j\,\psi_j(n) = f(n)\,. \qquad (16)$$

This remarkable result shows that although the forward linear interpolation is based on approximation (12), the adjoint interpolation produces an exact value of $f(n)$! The approximate nature of equation (13) reflects the fundamental difference between adjoint and inverse linear operators (Claerbout, 1992). When adjoint interpolation is applied to a constant function $f(x) \equiv 1$, it is natural to require the constant output $f(n) = 1$. This requirement leads to yet another general property of the interpolation functions $W(x,n)$:

Property 3

$$\int W(x,n)dx = 1\,. \qquad (17)$$

INTERPOLATION WITH FOURIER BASIS

To illustrate the general theory with familiar examples, I consider in this section the most famous example of an orthonormal function basis, the Fourier basis of trigonometric functions. What kind of linear interpolation does this basis lead to?

Continuous Fourier basis

For the continuous Fourier transform, the set of basis functions is defined by

$$\psi_\omega(x) = \frac{1}{\sqrt{2\pi}} e^{i\omega x} , \tag{18}$$

where ω is the continuous frequency. For a 1-point sampling interval, the frequency is limited by the Nyquist condition: $|\omega| \leq \pi$. In this case, the interpolation function W can be computed from formula (13) to be

$$W(x,n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{i\omega(x-n)} d\omega = \frac{\sin[\pi(x-n)]}{\pi(x-n)} . \tag{19}$$

The interpolation function (19) is well-known as the Shannon sinc interpolator. A known problem with its practical implementation is the slow decay with $(x-n)$. This problem is solved in practice with heuristic tapering (Hale, 1980), such as Harlan's triangle tapering (Harlan, 1982). While the function W from equation (19) automatically satisfies properties (3) and (17), where both x and n range from $-\infty$ to ∞, its tapered version may require additional normalization.

Discrete Fourier basis

Assuming that the range of the variable x is limited in the interval from $-N$ to N, the discrete Fourier basis (*Fast Fourier Transform*) employs a set of orthonormal periodic functions

$$\psi_k(x) = \frac{1}{\sqrt{2N}} e^{i\pi \frac{k}{N} x} , \tag{20}$$

where the discrete frequency index k also ranges, according to the Nyquist sampling criterion, from $-N$ to N. The interpolation function is computed from formula (13) to be

$$\begin{aligned} W(x,n) \; = \; & \frac{1}{2N} \sum_{k=-N}^{N-1} e^{i\pi \frac{k}{N}(x-n)} = \frac{1}{2N} e^{-i\pi(x-n)} \left[1 + e^{i\pi \frac{x-n}{N}} + \cdots + e^{i\pi \frac{2N-1}{N}(x-n)}\right] = \\ & \frac{1}{2N} e^{-i\pi(x-n)} \frac{e^{2i\pi(x-n)} - 1}{e^{i\pi \frac{x-n}{N}} - 1} = \frac{1}{2N} e^{-i\pi \frac{x-n}{2N}} \frac{e^{i\pi(x-n)} - e^{-i\pi(x-n)}}{e^{i\pi \frac{x-n}{2N}} - e^{-i\pi \frac{x-n}{2N}}} = \\ & e^{-i\pi \frac{x-n}{2N}} \frac{\sin[\pi(x-n)]}{2N \sin\left[\pi(x-n)/2N\right]} . \end{aligned} \tag{21}$$

An interpolation function, equivalent to (21), has been found by Muir (Popovici et al., 1993; Lin et al., 1993). It can be considered as a tapered version of the sinc interpolator (19) with the smooth tapering function

$$\frac{\pi(x-n)/2N}{\tan\left[\pi(x-n)/2N\right]} .$$

Unlike triangle-tapered sinc interpolator, Muir's interpolator (21) satisfies not only the obvious property (2), but also properties (3) and (17), where the interpolation function $W(x,n)$ should be set to zero for x outside the range from $n-N$ to $n+N$. The form of this function is shown in Figure 1.

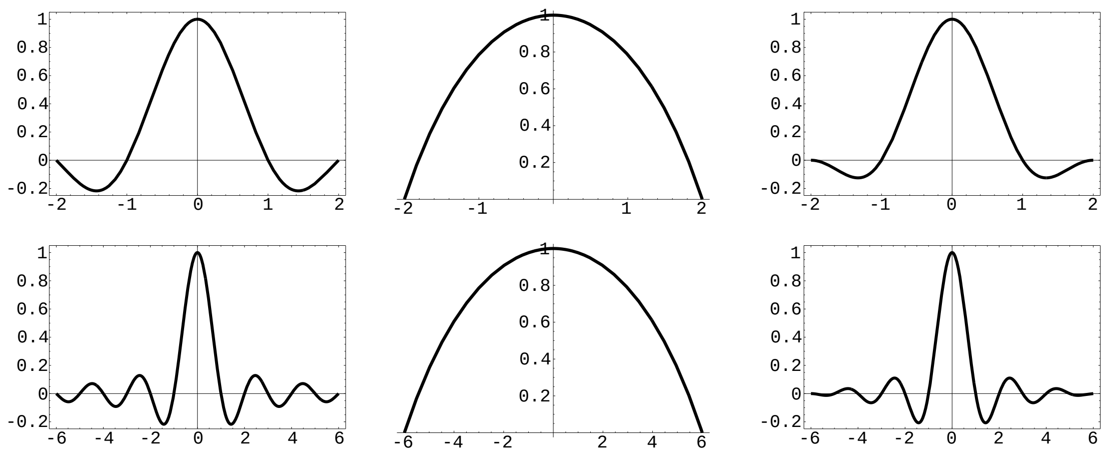

Figure 1: The left plots show the sinc interpolation function. Note the slow decay in x. The middle shows the effective tapering function of Muir's interpolation; the right is Muir's interpolator. The top is for $N=2$ (5-point interpolation); the bottom, $N=6$ (13-point interpolation). genint-ma-sinc [CR]

The development of the mathematical wavelet theory (Daubechies, 1992) has opened the door to a whole universe of orthonormal function bases, different from the Fourier basis. The wavelet theory should find many useful applications in geophysical data interpolation, but exploring this interesting opportunity goes beyond the scope of this paper.

The next section carries the analysis to the continuum and compares the mathematical interpolation theory with seismic imaging operators.

CONTINUOUS CASE AND SEISMIC IMAGING

Of course, the linear theory is not limited to discrete grids. It is interesting to consider the continuous case, because of its connection to the linear integral operators commonly used in seismic imaging. Indeed, in the continuous case, linear decomposition (7) takes the form of the nonstationary convolution integral

$$f(y) = \int m(x) G(y;x) dx \; , \tag{22}$$

where x is a continuous analog of the discrete coefficient k in (7), the continuous function $m(x)$ is analogous to the coefficient c_k, and $G(y;x)$ is analogous to one of the basis functions $\psi_k(x)$. The linear integral operator in (22) has a mathematical form similar to the form of well-known integral imaging operators, such as Kirchhoff migration or "Kirchhoff" DMO. Function $G(y;x)$ in this case represents the Green function (impulse response) of the imaging operator. Linear decomposition of the data into basis functions means decomposing it into the combination of impulse responses ("hyperbolas".)

In the continuous case, formula (13) transforms to

$$W(y,n) = \int\int \Psi^{-1}(x_1,x_2) G(y;x_1) \bar{G}(n;x_2) dx_1\, dx_2 \; , \tag{23}$$

where $\Psi^{-1}(x_1,x_2)$ refers to the inverse of the "matrix" operator

$$\Psi(x_1,x_2) = \int G(y;x_1) \bar{G}(y;x_2) dy \; . \tag{24}$$

When the linear operator, defined by formula (22), is *unitary*,

$$\Psi^{-1}(x_1,x_2) = \delta(x_1 - x_2) \; , \tag{25}$$

and formula (23) simplifies to the single integral

$$W(y,n) = \int G(y;x) \bar{G}(n;x) dx \; . \tag{26}$$

With respect to seismic imaging operators, one can recognize in the interpolation operator (26) the generic form of azimuth moveout (Biondi et al., 1996), which is derived either as a cascade of adjoint ($\bar{G}(n;y)$) and forward ($G(x;y)$) DMO or as a cascade of migration ($\bar{G}(n;y)$) and modeling ($G(x;y)$) (Fomel and Biondi, 1995). In the first case, the intermediate variable y corresponds to the space of zero-offset data cube. In the second case, it corresponds to a point in the subsurface.

Asymptotic pseudo-unitary operators as orthonormal bases

It is interesting to note that a wide class of integral operators, routinely used in seismic data processing, have the form of operator (22) with the "Green" function

$$G(t,\mathbf{y};z,\mathbf{x}) = \left|\frac{\partial}{\partial t}\right|^{m/2} A(\mathbf{x};t,\mathbf{y}) \delta\left(z - \theta(\mathbf{x};t,\mathbf{y})\right) \; . \tag{27}$$

where we have split the variable x into the one-dimensional component z (typically depth or time) and the m-dimensional component $\mathbf{x}$ (typically a lateral coordinate with m equal 1 or 2.)

Similarly, the variable y is split into t and $\mathbf{y}$. The function θ represents the *summation path*, which captures the kinematic properties of the operator, and A is the amplitude function.

Impulse response (27) is typical for different forms of Kirchhoff migration and datuming as well as for velocity transform, integral offset continuation, DMO, and AMO. Integral operators of that class rarely satisfy the unitarity condition, with Radon transform (slant stack) being a notable exception. In an earlier paper (Fomel, 1996), I have shown that it is possible to define the amplitude function A for each kinematic path θ so that the operator becomes *asymptotic pseudo-unitary*. This means that the adjoint operator coincides with the inverse in the high-frequency (stationary-phase) approximation. Consequently, equation (25) is satisfied to the same asymptotic order.

Using asymptotic pseudo-unitary operators, we can apply formula (26) to find an explicit analytic form of the interpolation function W, as follows:

$$W(t,\mathbf{y};t_n,\mathbf{y}_n) = \int\!\!\int G(t,\mathbf{y};z,\mathbf{x})\,G(t_n,\mathbf{y}_n;z,\mathbf{x})\,dz\,d\mathbf{x} =$$
$$\left|\frac{\partial}{\partial t}\right|^{m/2}\left|\frac{\partial}{\partial t_n}\right|^{m/2}\int A(\mathbf{x};t,\mathbf{y})\,A(\mathbf{x};t_n,\mathbf{y}_n)\,\delta\left(\theta(\mathbf{x};t,\mathbf{y})-\theta(\mathbf{x};t_n,\mathbf{y}_n)\right)\,d\mathbf{x}\,. \tag{28}$$

Here the amplitude function A is defined according to the general theory of asymptotic pseudo-inverse operators as

$$A = \frac{1}{(2\pi)^{m/2}}\,\left|F\,\widehat{F}\right|^{1/4}\left|\frac{\partial\theta}{\partial t}\right|^{(m+2)/4}\,, \tag{29}$$

where

$$F = \frac{\partial\theta}{\partial t}\,\frac{\partial^2\theta}{\partial\mathbf{x}\,\partial\mathbf{y}} - \frac{\partial\theta}{\partial\mathbf{y}}\,\frac{\partial^2\theta}{\partial\mathbf{x}\,\partial t}\,, \tag{30}$$
$$\widehat{F} = \frac{\partial\widehat{\theta}}{\partial z}\,\frac{\partial^2\widehat{\theta}}{\partial\mathbf{x}\,\partial\mathbf{y}} - \frac{\partial\widehat{\theta}}{\partial\mathbf{x}}\,\frac{\partial^2\widehat{\theta}}{\partial\mathbf{y}\,\partial z}\,, \tag{31}$$

and $\widehat{\theta}(\mathbf{x};t,\mathbf{y})$ is the *dual* summation path, obtained by solving equation $z=\theta(x;t,y)$ for t (assuming that an explicit solution is possible).

For a simple example, let us consider the case of zero-offset time migration with a constant velocity v. The summation path θ in this case is an ellipse

$$\theta(\mathbf{x};t,\mathbf{y}) = \sqrt{t^2-\frac{(\mathbf{x}-\mathbf{y})^2}{v^2}}\,, \tag{32}$$

and the dual summation path $\widehat{\theta}$ is a hyperbola

$$\widehat{\theta}(\mathbf{y};z,\mathbf{x}) = \sqrt{z^2+\frac{(\mathbf{x}-\mathbf{y})^2}{v^2}}\,. \tag{33}$$

The corresponding pseudo-unitary amplitude function is found from formula (29) to be (Fomel, 1996)

$$A = \frac{1}{(2\pi)^{m/2}}\,\frac{\sqrt{t/z}}{v^m z^{m/2}}\,. \tag{34}$$

Substituting formula (34) into (28), we derive the corresponding interpolation function

$$W(t,\mathbf{y};t_n,\mathbf{y}_n) = \frac{1}{(2\pi)^m} \left|\frac{\partial}{\partial t}\right|^{m/2} \left|\frac{\partial}{\partial t_n}\right|^{m/2} \int \frac{\sqrt{t\,t_n}}{v^{2m} z^{m+1}}\, \delta(z - z_n)\, d\mathbf{x}\,, \tag{35}$$

where $z = \theta(\mathbf{x};t,\mathbf{y})$, and $z_n = \theta(\mathbf{x};t_n,\mathbf{y}_n)$. For $m = 1$ (the two-dimensional case), we can apply the known properties of the delta function to simplify formula (35) further to the form

$$W = \frac{v}{\pi} \left|\frac{\partial}{\partial t}\right|^{1/2} \left|\frac{\partial}{\partial t_n}\right|^{1/2} \frac{\sqrt{t\,t_n}}{\sqrt{\left[(\mathbf{y}-\mathbf{y}_n)^2 - v^2(t-t_n)^2\right]\left[v^2(t+t_n)^2 - (\mathbf{y}-\mathbf{y}_n)^2\right]}}\,. \tag{36}$$

The result is an interpolator for zero-offset seismic sections. Like the sinc interpolator in formula (19) that is based on decomposing the signal into sinusoids, interpolation (36) is based on decomposing the zero-offset section into hyperbolas.

DISCUSSION

A simple linear interpolation theory can be derived from the sole principle of function bases. The choice of a function basis for the interpolated data uniquely defines a linear interpolation operator.

In application to seismic data interpolation, the basis set of functions can be given by the Green functions of an imaging operator, such as prestack migration or DMO. The linear interpolation operator in this case is intimately related to the general formulation of azimuth moveout (AMO). Some of the conclusions that the general theory can supply for AMO are

- In interpolation problems, the accuracy of operators (e.g. taking into account anisotropy, velocity variations, etc.) is of minor importance as long as the operator provides a complete basis set for describing the data.

- Formula (26) stresses the importance of using unitary operators (orthonormal bases) to construct linear interpolation. It suggests that unitary operators are even more important in interpolation problems than "true-amplitude" operators. Though applying non-orthogonal bases in interpolation problems is theoretically possible, it requires an intrinsic inversion of the matrix operator Ψ, defined in formulas (9) or (24). Such an inversion is rarely feasible in practice. The theory of asymptotic pseudo-unitary operators (Fomel, 1996) supplies a useful tool for constructing asymptotically orthonormal bases.

- It is also important to seek the most compact set of basis functions, e.g., the fewest number of frequencies in the spectrum. The Green functions may correspond to the solutions of a partial differential equation. The frequencies, actually present in the data, may correspond to the zeroes of the prediction-error filter. More challenging research needs to be done in relating differential equations, prediction filters, and function bases.

How is the mathematical theory of interpolation related to the problem of interpolating irregularly sampled data? The theory provides a linear interpolation operator **W**, defined in formula (1) and evaluated in formula (13). What we actually need to consider is a linear equation

$$\mathbf{f}_i = \mathbf{S}\mathbf{W}\mathbf{f}_n \; , \tag{37}$$

where $\mathbf{f}_n$ represents the desired regularly sampled output, $\mathbf{f}_i$ denotes the recorded irregularly spaced data, and **S** is the *sampling* operator. Estimating $\mathbf{f}_n$ from (37) requires the art and science of linear inversion, which includes such tools as regularization and preconditioning.

ACKNOWLEDGMENTS

I thank Biondo Biondi and Yalei Sun for stimulating discussions. Matthias Schwab provided a thorough proofreading.

REFERENCES

Biondi, B., Fomel, S., and Chemingui, N., 1996, Azimuth moveout for 3-D prestack imaging: SEP–**93**, 15–44.

Chemingui, N., and Biondi, B., 1996, Handling the irregular geometry in wide azimuth surveys: 66th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 32–35.

Claerbout, J. F., 1992, Earth Soundings Analysis: Processing Versus Inversion: Blackwell Scientific Publications.

Daubechies, I., 1992, Ten lectures on wavelets: SIAM, Philadelphia, Pennsylvania.

de Boor, C., 1978, A practical guide to splines: Springer-Verlag.

Fomel, S., and Biondi, B. L., 1995, The time and space formulation of azimuth moveout: 65th Ann. Internat. Meeting, Soc. Expl. Geophys., Expanded Abstracts, 1449–1452.

Fomel, S., and Claerbout, J., 1996, Simple linear operators in Fortran 90: SEP–**93**, 317–328.

Fomel, S., 1996, Stacking operators: Adjoint versus asymptotic inverse: SEP–**92**, 267–292.

Gardner, G. H. F., and Canning, A. J., 1994, Effects of irregular sampling on 3-D prestack migration: 64th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1553–1556.

Hale, I. D., 1980, Resampling irregularly sampled data: SEP–**25**, 39–58.

Harlan, W. S., 1982, Avoiding interpolation artifacts in Stolt migration: SEP–**30**, 103–110.

Karrenbach, M., 1995, Elastic tensor wave fields: Ph.D. thesis, Stanford University.

Lin, J., Teng, L., and Muir, F., 1993, Comparison of different interpolation methods for Stolt migration: SEP–**79**, 255–260.

Popovici, A. M., Blondel, P., and Muir, F., 1993, Interpolation in Stolt migration: SEP–**79**, 261–264.

Ronen, S., Sorin, V., and Bale, R., 1991, Spatial dealiasing of 3-D seismic reflection data: Geophysical Journal International, pages 503–511.

Ronen, S., Nichols, D., Bale, R., and Ferber, R., 1995, Dealiasing DMO: Good-pass, bad-pass, and unconstrained: 65th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 743–746.

APPENDIX A

A GENERIC INTERPOLATION PROGRAM

Module `interp1` on page 113 implements a generic 1-D interpolation program. Analogous modules also exist for the 2-D and 3-D cases. The module includes the initialization (constructor) subroutine `interp1_init`, which takes an array `coord` to define the data coordinates and the SEPlib-style values `o1`, `d1`, and `n1` to define the model grid. Additionally, it accepts an external function `interp` to compute the interpolation filter $W(x,n)$. The size of the filter is defined by the integer parameter `nfilt`. An example of a function with the interface of `interp` is `lagrange` on the following page, which implements the local Lagrange interpolation[2]. The initialization program allocates and computes three *private* arrays: the integer array `x` that defines the mapping from the data coordinates to the model grid, the logical array `m` that masks the data values outside the grid, and the real-value array `w` that stores coefficients of the interpolation filter. Computing these arrays outside the actual interpolation program `interp1_op` not only complies with the object-oriented design of linear operators (Fomel and Claerbout, 1996), but also significantly improves the efficiency when the interpolation operator is applied more than once (e.g., in iterative least-square optimization.) The arrays are deallocated by the "destructor" program `interp1_close`.

To illustrate the forward interpolation operator, I chose a regularly sampled chirp function $\exp -t^2/\sigma^2 \cos \omega t$ as the input model (Figure A-1). Figures A-2, A-3, and A-4 show the result of forward interpolation with different interpolators.

[2]The implementation is not as efficient as it could be, but sufficiently fast for testing purposes.

```
function lagrange (x, w) result (stat)
  integer             :: stat
  real, intent (in)   :: x
  real, dimension (:) :: w

  integer             :: i, j, nf
  real                :: f, xi

  nf = size (w)
  do i = 1, nf
     f = 1.
     xi = x + 1. - i
     do j = 1, nf
        if (i /= j) f = f * (1. + xi / (i - j))
     end do
     w (i) = f
  end do
  stat = 0
end function lagrange
```

```
module interp1
  use adj_mod
  integer,                               private :: nd, nf
  integer, dimension (:),   allocatable, private :: x
  logical, dimension (:),   allocatable, private :: m
  real,    dimension (:,:), allocatable, private :: w

contains
  subroutine interp1_init (coord, o1, d1, n1, interp, nfilt)
    real, dimension (:), intent (in) :: coord
    real,                intent (in) :: o1, d1
    integer,             intent (in) :: n1, nfilt
    interface
       integer function interp (x, w)
         real, intent (in)   :: x
         real, dimension (:) :: w
       end function interp
    end interface
    integer                          :: id, ix, stat
    real                             :: rx

    nd = size (coord) ; nf = nfilt
    if (.not. allocated (x)) allocate (x (nd), m (nd), w (nf,nd))
    do id = 1, nd ; rx = (coord (id) - o1)/d1 ; ix = rx
                    rx = rx - ix ; x (id) = ix + 1 - 0.5*nf
       m (id) = .true. ; w (:, id) = 0.
       if ((x (id) + 1 >= 1) .and. (x (id) + nf <= n1)) then
          m (id) = .false. ; stat = interp (rx, w (:,id))
       end if
    end do
  end subroutine interp1_init

  function interp1_op (adj, add, mod, ord) result (stat)
    integer              :: stat
    logical, intent (in) :: adj, add
    real, dimension (:)  :: mod, ord
    integer              :: id, i1, i2

    call adjnull (adj, add, mod, ord)
    do id = 1, nd ; if (m (id)) cycle
       i1 = x (id) + 1 ; i2 = x (id) + nf
       if (adj) then
          mod (i1:i2) =   mod (i1:i2) + w (:,id)  * ord (id)
       else
          ord (id) = sum (mod (i1:i2) * w (:,id)) + ord (id)
       end if
    end do
    stat = 0
  end function interp1_op

  subroutine interp1_close ()
    deallocate (x, m, w)
  end subroutine interp1_close
end module interp1
```

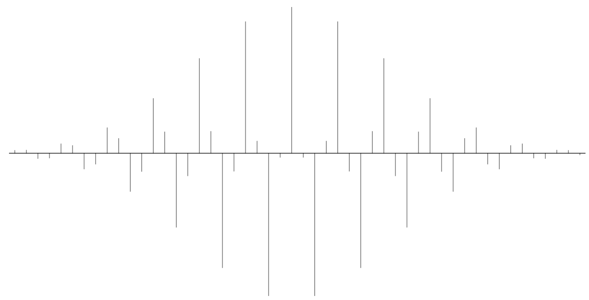

Figure A-1: The input is a regularly sampled chirp function. genint-alias [ER]

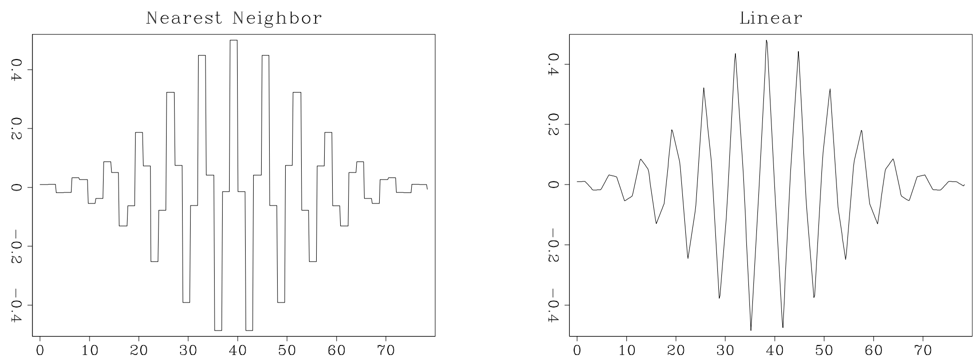

Figure A-2: Left is the result of forward nearest-neighbor interpolation; right, linear interpolation. genint-bin [ER]

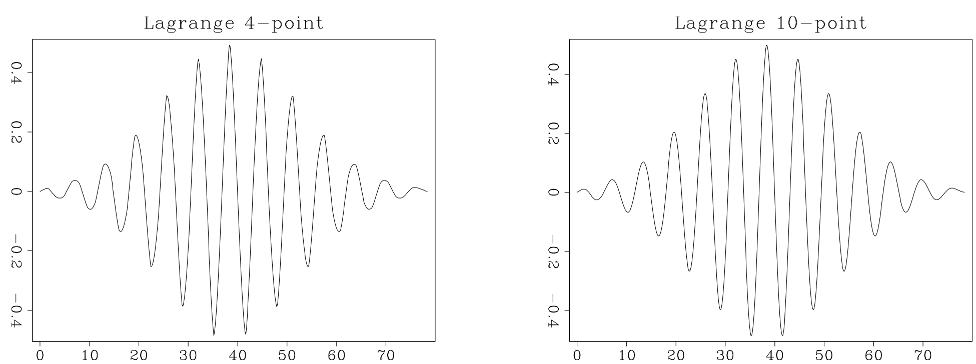

Figure A-3: Result of forward Lagrange interpolation. Left is the 4-point interpolator; right, 10-point. Increasing the number of coefficients may lead to instabilities. genint-lgg [ER]

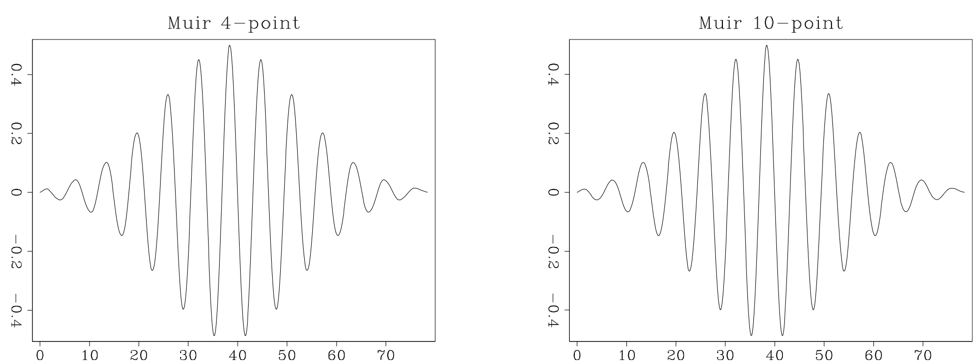

Figure A-4: Result of forward Muir interpolation. Left is the 4-point interpolator; right, 10-point. genint-sinc [ER]