Chapter 4

Accurate interpolation using derivatives

A new technique for accurate interpolation using derivative information is presented. It is a hybrid of extrapolation to arbitrary order and linear interpolation, and combines the advantages of both methods. Through a modification of the coefficients of the Taylor expansion, extrapolations from a number of locations can be combined to obtain a polynomial order of accuracy that is one higher than that of a single Taylor expansion.

The formulation of the method is very general: it can be used both in regular and irregular data distributions in spaces of arbitrary dimension. The technique is expected to be useful in many applications. It is used at various stages in the algorithms described in the following chapters.

4.1 Introduction

The theory developed in Chapter 3 shows that ray field maps in position/angle coordinates are useful tools for seismic imaging in complex media. To be relevant for practical applications it is important that these maps are calculated efficiently. Therefore, two algorithms for the calculation of ray field maps are investigated in Chapters 5 and 6. In preparation, this chapter introduces a new and accurate local interpolation technique that is used at various stages in these algorithms.

Interpolation methods approximate a function using data that are available at a distribution of data points. The data consist of the function values and possibly some of its derivatives. The data points may be distributed in a space of arbitrary dimension. Both regular and irregular data distributions are relevant for this work, and hence both are discussed here.

Interpolation on a *regular*, rectangular grid is essential for the practical application of the ray field maps. In the common practice of Kirchhoff migration, for

example, travel time maps are first calculated on a coarse grid, but during the actual migration the travel time maps are interpolated to a denser grid (e.g., Epili and McMechan, 1996). The main purpose of this approach is to limit storage space and disk traffic, because in 3-D applications the maps become far too voluminous to keep them in working memory. The more accurately the interpolation can be performed, the more computer time is saved (e.g., Vanelle and Gajewski, 2002).

Interpolation on *irregular* distributions comes into play in the construction of ray field maps. In algorithms such as wave front construction (Vinje et al., 1993; Lambaré et al., 1996) and related methods (Chapters 5 and 6) ray fields are first calculated in terms of ray field coordinates, and subsequently mapped onto spatial coordinates. In the spatial domain the points at which the ray field is evaluated form an irregular distribution, which has to be regularised for practical application. One way to perform this regularisation is by means of interpolation. Another approach, based on averaging integrals, is used in Chapter 6.

This chapter introduces a general approach to enhancing the accuracy of local interpolation by using derivative information. The approach can be applied to both regular and irregular data distributions in arbitrary dimension.

Interpolation methods are widely used in many fields of science. Usually classical techniques as described in textbooks such as Press et al. (1992) and Ralston and Rabinowitz (2001) are used. Recent work has been devoted to the smooth interpolation of irregular data distributions (Sibson, 1981; Sambridge et al., 1995). The use of derivative information is usually limited to Hermite interpolation in a single variable (Ralston and Rabinowitz, 2001) or to bi-cubic interpolation in 2-D rectangular grids (Press et al., 1992).

The method proposed here combines aspects of interpolation and extrapolation. Using the available derivatives the function is extrapolated from the data points to the point of evaluation. Linear interpolation is used to weight the extrapolations from a number of data points. The best known extrapolation formula that uses the function values and derivatives from a single point is the Taylor expansion. Using the Taylor expansion for the extrapolation, however, can be shown to be sub-optimal. Instead, a a modification of the Taylor expansion is proposed, in which the expansion terms are multiplied by a set of coefficients. This makes the accuracy of the interpolation one order higher than that of each individual Taylor expansion. Since the proposed interpolation method is a hybrid of interpolation and extrapolation the term *intrapolation* is coined to refer to it. The modified Taylor expansion is given the name *Dutch Taylor expansion*.

The basic principles of intrapolation and the Dutch Taylor expansion are explained in Section 4.2. Application of intrapolation depends on the availability of the derivatives as independent data. If the derivatives are not available they can, under some accuracy conditions, be estimated from the available data. In regular grids finite difference approximation of derivatives is adequate, as explained in Section 4.3. The accuracy and convergence rates of intrapolation are compared to a number of alternative interpolation methods in Section 4.4.1. An example of intrapolation in the context of ray field maps is provided in Section 4.4.2.

4.2 Accurate interpolation using derivatives

The objective of a numerical interpolation method is to approximate a function using data known at a set of isolated points. The data may include the function values and various derivatives. In the current context only local interpolation by means of low order polynomials is considered. The adjective local indicates that data is used only from a neighbourhood of the point at which the function value is estimated.

The general approach is to construct an interpolating function, the *interpolant*, which fits the available data perfectly. However, this assumes that the available data are exact, which in many applications is not the case. Methods exist for reducing the effect of errors by exploiting redundancy in the available data, and by relaxing the requirement of perfect fit, but these methods are inherently less local and computationally more expensive.

Even in the presence of errors, local interpolation is, in many cases, the most practical method for approximation. It is then important to choose an interpolant that does not enhance the errors. In case of first order (i.e. linear) interpolation the interpolated error is bounded by the errors in the contributing data. In general, however, the risk of enhancing errors increases with increasing polynomial order of the interpolant, especially if derivatives are used in its construction.

The reason to choose higher order interpolation may either be to enhance the smoothness or to enhance the accuracy (Press et al., 1992). Here, the main interest lies in enhancing the accuracy. This can be accomplished by using independent derivative information or data from a larger neighbourhood. A larger neighbourhood can be used either to define interpolants with a larger support, or to obtain derivatives estimates for the data points within original support, which can subsequently be used to constrain the higher order interpolant. Since it is more practical, the latter approach is followed in Section 4.3. For the remainder of this section it is assumed that the derivatives are known exactly or with sufficient accuracy.

4.2.1 Interpolation of Taylor expansions

The classical approach for incorporating derivative information in interpolation schemes is to use interpolants that fit the derivatives as well as the function values. Construction of such interpolants is relatively easy in 1-D (Hermite interpolation, e.g., Ralston and Rabinowitz, 2001), but becomes more complicated even for 2-D rectangular grids (e.g., Press et al., 1992). Generalisation to irregular data distributions and to rectangular grids in higher dimensions leads to complex calculations.

An alternative approach is to use a hybrid of extrapolation and interpolation. The function values and the available derivatives at each data point are used to construct an extrapolating function. At the point of evaluation the extrapolations

from neighbouring data points are then weighted with a standard linear interpolant.

In the following it will first be shown how this hybrid strategy works for extrapolation using the Taylor expansion, which is the most intuitive choice for extrapolation. It turns out that for this choice the interpolation does not improve on the accuracy of the individual extrapolations. A major improvement is achieved by adopting another expansion that renders the order of accuracy of the interpolation equal to its polynomial order. The approach is outlined below in 1-D; generalisations to higher dimensions are discussed in Section 4.2.4.

The n-th order Taylor expansion of a function $f(x)$ around $x = \xi$ is given by

$$\mathcal{T}_n[f(x)](x;\xi) = \sum_{k=0}^{n} \frac{1}{k!}(x-\xi)^k f^{(k)}(\xi), \tag{4.1}$$

where $f^{(k)}$ stands for the k-th derivative of $f(x)$. A two-point interpolation formula based on x_0 and x_1 may then be constructed as follows:

$$\mathcal{I}_n^*[f(x)](x;x_0,x_1) = \frac{(x_1-x)}{(x_1-x_0)}\mathcal{T}_n[f(x)](x;x_0) + \frac{(x-x_0)}{(x_1-x_0)}\mathcal{T}_n[f(x)](x;x_1). \tag{4.2}$$

This hybrid interpolation formula may in some sense be regarded as a generalisation of linear interpolation. If no derivative information is available, the extrapolation is of zeroth order ($n = 0$) and the usual linear interpolation is obtained:

$$\mathcal{I}_0^*[f(x)](x;x_0,x_1) = \frac{(x_1-x)}{(x_1-x_0)}f(x_0) + \frac{(x-x_0)}{(x_1-x_0)}f(x_1). \tag{4.3}$$

Despite its intuitive character, the interpolation of Taylor expansions has a disadvantage that is best illustrated for interpolation using first order derivatives. The Taylor expansion (4.1) is in this case a linear function, or first order polynomial. The interpolation of both Taylor expansions (4.2) therefore produces a second order polynomial. It is interesting to see what second order interpolation formula (4.2) does to an arbitrary second order polynomial sampled at x_0 and x_1. Figure 4.1(a) clearly illustrates that the quadratic function is not reconstructed.

To understand what happens it is instructive to look at the general quadratic function $p_2(x)$:

$$p_2(x) = c_0 + c_1 x + c_2 x^2. \tag{4.4}$$

The first order Taylor expansion of this function around ξ is

$$\begin{aligned}\mathcal{T}_1[p_2(x)](x;\xi) &= p_2(\xi) + (x-\xi)p_2'(\xi)\\ &= c_0 + c_1 x + c_2(2x-\xi)\xi,\end{aligned} \tag{4.5}$$

which, after some algebra, leads to the interpolant

$$\mathcal{I}_1^*[p_2(x)](x;x_0,x_1) = p_2(x) + c_2(x-x_0)(x-x_1). \tag{4.6}$$

The interpolant differs from the function itself by a second order polynomial. This quadratic function has coefficient c_2 as a multiplier, which indicates that if this coefficient is zero, the function is perfectly reconstructed. This is no surprise, because in that case the function is linear, and both Taylor approximations are exact.

For comparison, the result of using a standard linear interpolant is

$$\mathcal{I}_0^*[p_2(x)](x;x_0,x_1) = p_2(x) - c_2(x-x_0)(x-x_1). \tag{4.7}$$

Apart from the sign, the error is exactly the same as in Equation (4.6). Hence, the use of derivative information using Equation (4.2) does not improve the accuracy of the interpolation.

4.2.2 Intrapolation and the Dutch Taylor expansion

Here, a hybrid of extrapolation and interpolation is proposed that is similar to the Taylor-based approach of Equation (4.2), but with a major improvement in accuracy. The interpolants are constructed in such a way that the order of accuracy is equal to the order of interpolation. In order to stress that the resulting interpolation formula is a hybrid of extrapolation and interpolation, the name *intrapolation* is introduced to refer to it. The corresponding interpolant is analogously called the *intrapolant*.

In analogy of Equation (4.2) the intrapolant is defined as

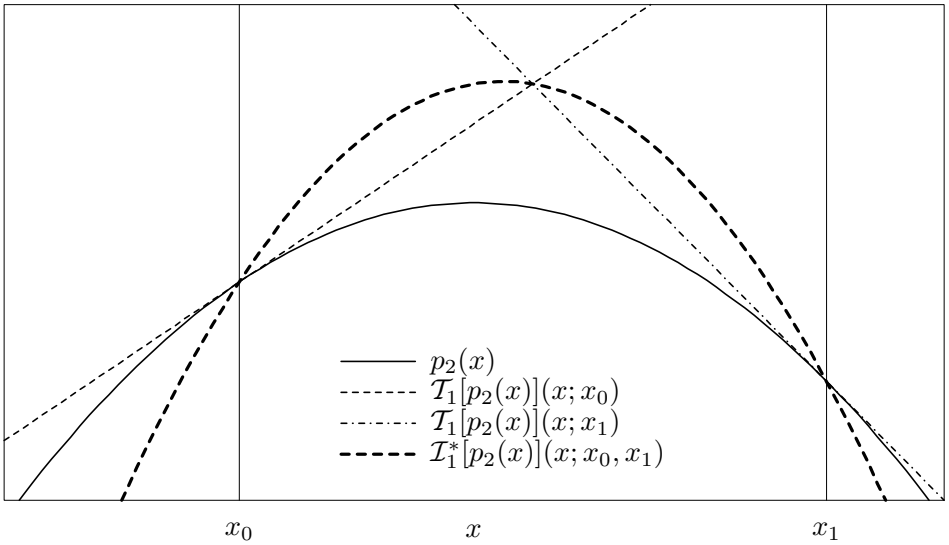
$$\mathcal{I}_n[f(x)](x;x_0,x_1) = \\ \frac{(x_1-x)}{(x_1-x_0)}\mathcal{D}_n[f(x)](x;x_0) + \frac{(x-x_0)}{(x_1-x_0)}\mathcal{D}_n[f(x)](x;x_1), \tag{4.8}$$

with $\mathcal{D}_n$ an extrapolation formula defined as

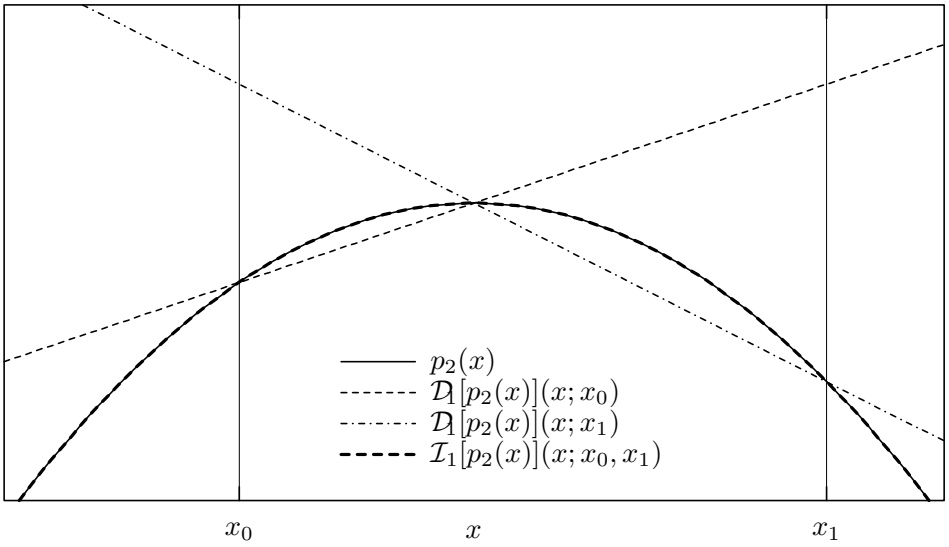
$$\mathcal{D}_n[f(x)](x;\xi) = \sum_{k=0}^{n} \frac{a_k^n}{k!}(x-\xi)^k f^{(k)}(\xi). \tag{4.9}$$

At this point the expansion is expressed as an ansatz in terms of unknown coefficients a_k^n, where the superscript n should be interpreted as an index rather than a power. Note that the parametric form of (4.9) reduces to the Taylor expansion with $a_k^n = 1$.

The coefficients introduced in extrapolation formula $\mathcal{D}_n$ provide the degrees of freedom that can be used to force the intrapolant (4.8) to satisfy some accuracy requirements. Since an n-th order Taylor expansion gives n-th order accuracy, interpolation of n-th order expansions is useful if it can enhance the order of



(a) Quadratic interpolation between points x_0 and x_1 based on the Taylor expansion, as in Equation (4.2). An arbitrary quadratic function is not reproduced by the interpolation.



(b) Quadratic interpolation between points x_0 and x_1 based on the Dutch Taylor expansion, as in Equation (4.21). An arbitrary quadratic function is reproduced exactly.

Figure 4.1: Approximation of a quadratic function $p_2(x)$, based on interpolation of first order extrapolations. For Figure (a) the extrapolation is the conventional Taylor expansion $\mathcal{T}$, as in Equation (4.1), for Figure (b) it is the Dutch Taylor expansion $\mathcal{D}$ (4.9) introduced in this paper.

accuracy to $n+1$. It should be noted here that the polynomial order of the intrapolant $\mathcal{I}_n$ is $n+1$. This means that the goal is to choose coefficients a_k^n in such a way that the order of accuracy of the intrapolant is equal to its polynomial order, which is the highest order of accuracy that can be obtained. For the extrapolation formula thus obtained the term *Dutch Taylor expansion* is coined, for reasons that will become clear later on.

A general algorithm for finding coefficients in linear functionals such as (4.9) is the method of undetermined coefficients (e.g., Ralston and Rabinowitz, 2001). As a first, trivial, example determine the coefficient a_0^0, that makes intrapolant (4.8) exact for first order polynomials. This should result in the well known linear interpolant. Since any first order polynomial in x can be written as a linear combination of the monomials 1 and x, it is sufficient to solve the system of equations

$$\begin{cases} \mathcal{I}_0[1](x;x_0,x_1) = 1 \\ \mathcal{I}_0[x](x;x_0,x_1) = x, \end{cases} \tag{4.10}$$

which, using (4.8), leads to the more explicit system

$$\begin{cases} a_0^0 \left(\dfrac{(x_1-x)}{(x_1-x_0)} + \dfrac{(x-x_0)}{(x_1-x_0)} \right) = 1 \\ a_0^0 \left(\dfrac{(x_1-x)}{(x_1-x_0)} x_0 + \dfrac{(x-x_0)}{(x_1-x_0)} x_1 \right) = x. \end{cases} \tag{4.11}$$

Although there are two equations for one coefficient the solution is consistent:

$$a_0^0 = 1. \tag{4.12}$$

Hence the first order intrapolant is given by

$$\mathcal{I}_0[f(x)](x;x_0,x_1) = \frac{(x_1-x)}{(x_1-x_0)} f(x_0) + \frac{(x-x_0)}{(x_1-x_0)} f(x_1), \tag{4.13}$$

which is indeed equal to the usual linear interpolant.

A similar system of equations can be composed to determine the coefficients a_k^1 for the second order intrapolant:

$$\begin{cases} \mathcal{I}_1[1](x;x_0,x_1) = 1 \\ \mathcal{I}_1[x](x;x_0,x_1) = x \\ \mathcal{I}_1[x^2](x;x_0,x_1) = x^2. \end{cases} \tag{4.14}$$

After some algebra this yields the coefficients

$$a_0^1 = 1, \quad a_1^1 = \frac{1}{2}, \tag{4.15}$$

and the intrapolant

$$\mathcal{I}_1[f(x)](x;x_0,x_1) = \frac{(x_1-x)}{(x_1-x_0)}\left[f(x_0)+\frac{1}{2}(x-x_0)f'(x_0)\right] + \frac{(x-x_0)}{(x_1-x_0)}\left[f(x_1)+\frac{1}{2}(x-x_1)f'(x_1)\right]. \quad (4.16)$$

This intrapolant is exact for second order polynomials, which can be checked by inserting p_2 as defined in Equation (4.4):

$$\mathcal{I}_1[p_2(x)](x;x_0,x_1) = p_2(x). \quad (4.17)$$

Note the difference with Equation (4.6). Figure 4.1(b) shows graphically that intrapolation with first derivatives is exact to second order.

It is not immediately obvious what the coefficients a_k^n for the Dutch Taylor expansion should be for higher values of n. An intrapolant that certainly has practical relevance (e.g. Section 4.4.2) is that for $n = 2$, which is based on both first and second derivatives. Another system of equations, similar to (4.10) and (4.14), yields coefficients

$$a_0^2 = 1, \quad a_1^2 = \frac{2}{3}, \quad \text{and} \quad a_2^2 = \frac{1}{3}. \quad (4.18)$$

Figure 4.2 compares intrapolation of a cubic function using these coefficients with the less accurate interpolation based on the Taylor expansion.

While the solutions for higher n are not very likely to be useful in practice (see Section 4.2.3), it is interesting to see what happens. The system of equations gets increasingly complex and more difficult to solve, but a pattern soon becomes clear. For general n the coefficients form the series

$$a_k^n = \frac{n+1-k}{n+1} = 1 - \frac{k}{n+1} \quad (4.19)$$

where $n+1$ is the polynomial order of the intrapolant, and n the maximum order of derivatives used.

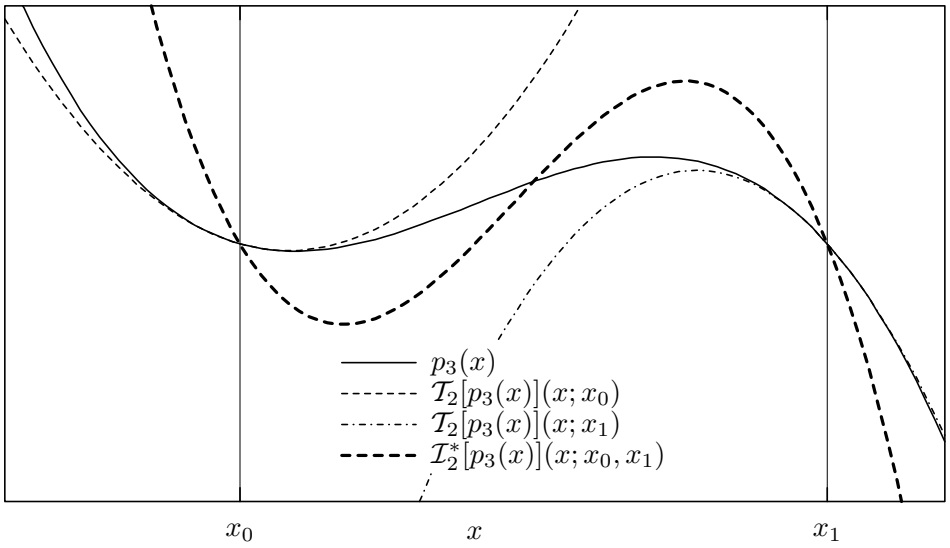
Using coefficients (4.19) the Dutch Taylor expansion (4.9) can be expressed explicitly as

$$\mathcal{D}_n[f(x)](x;\xi) = \sum_{k=0}^{n}\left(1-\frac{k}{n+1}\right)\frac{1}{k!}(x-\xi)^k f^{(k)}(\xi), \quad (4.20)$$

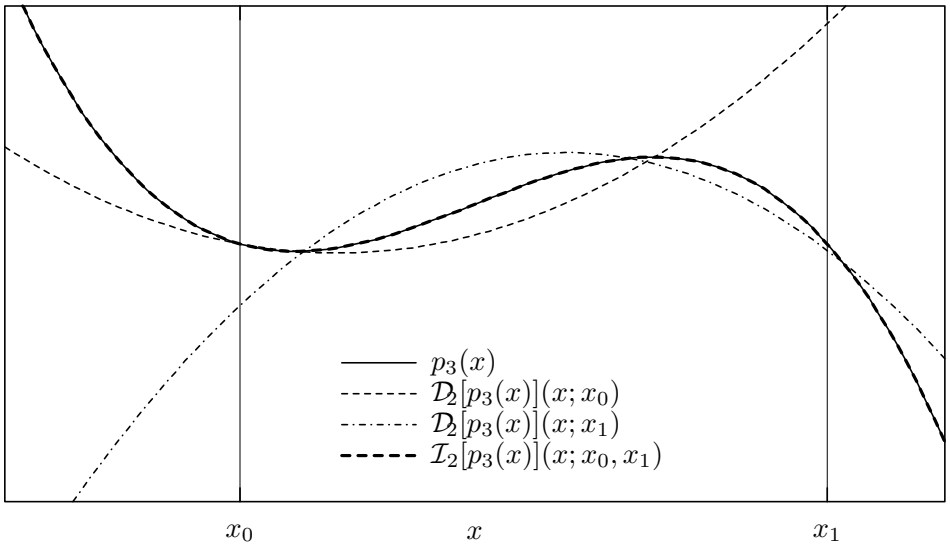
or alternatively as a normal Taylor expansion and a correction term:

$$\mathcal{D}_n[f(x)](x;\xi) = \mathcal{T}_n[f(x)](x,\xi) - \frac{1}{n+1}\sum_{k=1}^{n}\frac{k}{k!}(x-\xi)^k f^{(k)}(\xi). \quad (4.21)$$

The correction term sums over k, starting at $k = 1$ simply because the argument vanishes at $k = 0$. Appendix 4.A.1 proves the correctness of 4.19, by showing that expansion (4.21) makes the intrapolant (4.8) exact to order $n+1$ for arbitrary n.



(a) Cubic interpolation between points x_0 and x_1 based on the Taylor expansion, as in Equation (4.2). A arbitrary cubic function is not reproduced by the interpolation.



(b) Cubic interpolation between points x_0 and x_1 based on the Dutch Taylor expansion, as in Equation (4.21). An arbitrary cubic function is reproduced exactly.

Figure 4.2: The equivalent of Figure 4.1, now for a cubic function $p_3(x)$, with interpolation based on second order extrapolations.

4.2.3 Some interpretative remarks

The high order of accuracy achieved by intrapolation is primarily due to the Dutch Taylor expansion used for the extrapolation. In order to understand why this expansion is so successful it is instructive to look at it in some more detail and compare it with the Taylor expansion.

First, the case of first order expansion is examined. Coefficient a_1^1 in (4.15) is equal to 1/2. Hence, the first order term in the Dutch Taylor expansion is only half the size of the first order term of the Taylor expansion. Why this is better for the sake of interpolation can be understood qualitatively by realising that the interpolation of the zeroth-order terms alone results already in a linear function. This linear function accommodates the average of the derivative over the interval. Hence, in order to improve the interpolation one should not use the full derivative information in the extrapolation. In the first order expansion using only half the derivative gives the correct result.

From the coefficients of the Dutch Taylor expansion given in Equation (4.19) it can be concluded that the same observation holds more generally. The coefficients of the Taylor expansion are equal to 1, regardless of n and k. All terms ($k > 0$) of the Dutch Taylor expansion are smaller, in an absolute sense, than the equivalent terms in the Taylor expansion.

The Dutch Taylor expansion may thus be regarded as an "economical" [1] alternative to the Taylor expansion. For extrapolation alone, it is less accurate than the Taylor expansion. When used in combination with a linear interpolation, however, it provides greater accuracy. This is possible because the interpolation joins the information coming from two sides. The combination of economy and the fact that in the interpolation each side brings in its own share of information provides ample motivation for the choice of the name *Dutch* Taylor expansion to refer to extrapolation formulas (4.20) and (4.21).

One of the key observations of Appendix 4.A.1 is that the n-th order Dutch Taylor expansion (4.21) is, for a fixed point of evaluation x, a linear function of the data point position ξ for polynomials up to order $n + 1$. At $\xi = x$ this linear function passes through the correct value $f(\xi = x)$. Hence, the n-th order Dutch expansion alone gives an exact approximation only for constant functions, but the errors are such that they cancel by linear interpolation for polynomials up to order $n + 1$.

In contrast, the n-th order Taylor expansion is exact for polynomials up to order n. For a polynomial of order $n+1$, however, the error is another polynomial of order $n + 1$. This type of error is not cancelled by linear interpolation, which explains why an interpolation of Taylor expansions is less accurate than an interpolation of Dutch Taylor expansions.

It is important to note that although intrapolation and the Dutch Taylor expansion are discussed here for general order n, practical applications are usually restricted to low values of n. The range of practical applicability is similar to that

[1] Here "economical" should not be interpreted in the sense of computational efficiency.

of Taylor expansions. High order Taylor expansions are often used in theoretical formulations. In practical applications, however, their use is limited because the convergence with increasing order is typically very slow, and the region of convergence very small. The Dutch Taylor expansion and intrapolation are to be used in small regions where the function to be approximated is well represented by a low order polynomial, i.e. where the Taylor expansion coefficients decrease quickly for increasing order.

4.2.4 Generalisation to higher dimensions

Since extrapolation to arbitrary order and first order interpolation are both well defined in spaces of arbitrary dimension, the intrapolation method can easily be generalised to higher dimensions.

The generalisation of the extrapolation formula is straightforward. It is defined analogous to a multivariate Taylor expansion:

$$\mathcal{T}_n[f(\boldsymbol{x})](\boldsymbol{x};\boldsymbol{\xi}) = \sum_{k=0}^{n} \frac{1}{k!}(\boldsymbol{x}-\boldsymbol{\xi})^{[k]} : \nabla^{[k]} f(\boldsymbol{\xi}), \tag{4.22}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots)^T$ is now a vector variable. Superscript $[k]$ indicates the iterated tensor product with k indices, as in

$$\boldsymbol{x}^{[k]} = x_{i_1} x_{i_2} \ldots x_{i_k}, \tag{4.23}$$

and the double dot (:) indicates contraction (summation) over all k indices. The multivariate analogue of (4.21) is

$$\mathcal{D}_n[f(\boldsymbol{x})](\boldsymbol{x};\boldsymbol{\xi}) = \mathcal{T}_n[f(\boldsymbol{x})](\boldsymbol{x};\boldsymbol{\xi}) - \frac{1}{n+1}\sum_{k=1}^{n} \frac{k}{k!}(\boldsymbol{x}-\boldsymbol{\xi})^{[k]} : \nabla^{[k]} f(\boldsymbol{\xi}). \tag{4.24}$$

As argued in Section 4.2.1 for intrapolation in a single dimension, the accuracy of order $n+1$ can be explained by the fact that the Dutch Taylor expansion (4.21) is a linear function of data point ξ for polynomials up to order $n+1$, passing through the correct value at $\xi = x$. The same argument holds for multiple variables.

For multivariate polynomials up to order $n+1$, the Dutch Taylor expansion (4.24) yields a first order polynomial in terms of the data point variables $\boldsymbol{\xi}$. Hence, in order to do intrapolation with an accuracy of order $n+1$ any type of linear interpolation will do. Natural generalisations of linear interpolation to higher dimensions include the N-linear interpolation on rectangular grids, and barycentric interpolation for general data distributions organised by N-dimensional triangulation. Both types of interpolation are discussed in Appendix D.

4.3 Calculation of derivatives

The interpolation methods introduced in the previous sections assume the presence of derivative information on the data points. The alternative is to calculate

the derivative information from the available data. In essence this means introducing data available at neighbouring data points which do not participate in the interpolation itself.

In order to maintain the accuracy of intrapolation it is important that the accuracy of the derivatives match the order of the interpolating polynomial. That is to say, if the aim is a second order interpolant, which requires only first derivatives, these derivatives should be accurate to second order.

Derivative information on arbitrary data distributions can be obtained using finite difference (FD) methods (e.g., Fornberg, 1996, 1998). The calculation of these FD estimates is another example of the method of undetermined coefficients. The derivatives are expressed as a sum over neighbouring data values, multiplied by undetermined coefficients. The requirement that the derivatives are exact for polynomials up to a given order leads to a system of equations similar to those discussed in Section 4.2.1.

The number of data points used in the summation, and thus the size of the neighbourhood, depends on the number of coefficients needed to parameterise a polynomial of the required order. The combination of the (relative) data locations and the corresponding weights is called the FD stencil.

The complexity of the system of equations leading to the weights in an FD stencil strongly depends on the dimension of the data space, on the order of accuracy that is required, and on the geometry of the data points. A particular advantage of rectangular, equidistant grids is that the stencils only depend on relative data point positions, which means that the same stencil can be used for almost every data point in the grid. Only the grid points near the edges require adapted stencils.

In arbitrary data distributions the situation is a lot more complicated. The definition of a neighbourhood around a data point is much less obvious, and the relative positions of neighbouring data points change from place to place. The result is that each data point requires a dedicated FD stencil to obtain accurate derivatives. For most practical applications, including ours, this is computationally too expensive.

If the objective for the calculation of derivatives is smooth rather than accurate interpolation, one can think of many alternatives for the calculation of reasonable derivative estimates (e.g., Sibson, 1981). The author is, however, not aware of a practical approach that meets the accuracy requirements. For irregular data distributions it is assumed that the derivative information is independently available.

Centred finite differences

In the following the calculation of low order FD estimates in equidistant grids is summarised. The derivations are not be discussed here but can be found for example in Fornberg (1996, 1998). First, consider the one dimensional FD estimate for the derivative $f'(x)$ of a function $f(x)$. A three-point centred FD is exact for

polynomials up to second order:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2). \tag{4.25}$$

The coefficient multiplying $f(x)$ itself happens to be equal to zero, which effectively makes it a two-point formula.

The second derivative estimate requires all three function values:

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + O(h^2). \tag{4.26}$$

This formula is actually exact for polynomials up to third order, due to the symmetry of the FD stencil. This suggests the use of three-point centred FD for third order accurate intrapolation. The first order derivatives obtained using (4.25), however, are only accurate to second order. The application of (4.25) to a third order monomial cx^3 yields

$$\frac{c(x+h)^3 - c(x-h)^3}{2h} = 3cx^2 + ch^2. \tag{4.27}$$

This shows that the three-point centred FD estimate of the first derivative of any third order polynomial is wrong by a constant value. Fortunately, the intrapolant (4.8) only depends on the difference of the first derivatives at the two data points. This can be seen by looking at the interpolation of the first order expansion terms only:

$$\begin{aligned} a_1^n \frac{(x_1 - x)}{(x_1 - x_0)}(x - x_0)f'(x_0) + a_1^n \frac{(x - x_0)}{(x_1 - x_0)}(x - x_1)f'(x_1) \\ = a_1^n \frac{(x - x_0)(x - x_1)}{(x_1 - x_0)}(f'(x_1) - f'(x_0)) \end{aligned} \tag{4.28}$$

This means that the constant is cancelled in the interpolant, and it is indeed possible to do third order accurate interpolation using only three-point FD estimates.

One-sided finite difference formulas

At the boundaries of a grid it is necessary to compute the derivatives using one-sided rather than centred FD formulas. Depending on the desired order of accuracy the following formulas are available (from Fornberg, 1996, table 3.1-2):

$$f'(x) = \frac{-f(x) + f(x+h)}{h} + O(h) \tag{4.29}$$

$$= \frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h} + O(h^2) \tag{4.30}$$

$$= \frac{-11f(x) + 18f(x+h) - 9f(x+2h) + 2f(x+3h)}{6h} + O(h^3), \tag{4.31}$$

where the latter is exact for third order polynomials. Similarly, for the second derivatives

$$f''(x) = \frac{f(x) - 2f(x+h) + f(x+2h)}{h^2} + O(h) \tag{4.32}$$

$$= \frac{2f(x) - 5f(x+h) + 4f(x+2h) - f(x+3h)}{2h^2} + O(h^2) \tag{4.33}$$

can be used, the latter being exact for third order polynomials. Note that the errors are different for one-sided and centred FD estimates, and cancellation of errors in the first derivative does not occur when both are combined.

Multivariate finite differences

The one dimensional finite difference formulas are easily generalised to higher dimensions. For the calculation of derivatives in a single variable, the other variables may be regarded as constant, while mixed derivatives can be calculated by repeated differencing. For example, the mixed second derivative of a bivariate function $f(x, y)$ is given by:

$$\begin{aligned}\frac{\partial^2 f(x,y)}{\partial x \partial y} =& \frac{f(x+h, y+h) - f(x+h, y-h)}{4h^2} \\ &- \frac{f(x-h, y+h) - f(x-h, y-h)}{4h^2} + O(h^2),\end{aligned} \tag{4.34}$$

which is exact for bivariate polynomials of second order in both x and y.

4.4 Examples

4.4.1 Comparison of 1D interpolation methods

To assess the accuracy and convergence behaviour of intrapolation, it is compared with a number of alternative interpolation methods. The experiment is performed only in 1-D, but, qualitatively, the result will be the same in higher dimensions.

Experimental setup

Interpolation and extrapolation methods are typically designed to be exact for polynomials up to a given order (see Section 4.2.1). For an assessment of these methods it is useful to look at their performance when applied to a different class of test function. Fourier functions are ideal for this purpose because they form a complete basis for continuous functions, and they relate to familiar concepts as sampling intervals and length scales of variation.

The interpolation of the complete range of real-valued Fourier functions is studied in the form

$$\sin(kx + \phi), \tag{4.35}$$

where x is the variable to be interpolated, k the frequency (or wave number) and ϕ an arbitrary phase shift.

Without loss of generality the first data point may be chosen at $x_0 = 0$ (NB arbitrary phase shifts are accommodated by ϕ), and the second data point at $x_1 = 1$, because variations in the width of the data point interval are mathematically equivalent to variations in k. In this light k may be interpreted as a relative frequency that can be expressed as a product of the absolute frequency $\hat{k}$ and the width of the interpolation interval h:

$$k = \hat{k}h. \tag{4.36}$$

Variations in k may thus be attributed to variations in interval width (h), or to variations in absolute frequency ($\hat{k}$).

For an arbitrary interpolant $\mathcal{I}$, the error E may be expressed as a function of k, x, and ϕ:

$$E(k, x, \phi) = \mathcal{I}\left[\sin(kx + \phi)\right](x, 0, 1) - \sin(kx + \phi). \tag{4.37}$$

For fixed frequency k and fixed position x the error is a periodic function of ϕ. To obtain insight in the behaviour of the error as a function of k and x, the RMS average over the range of ϕ is calculated:

$$\left[E_{\phi}^{\mathrm{rms}}(k, x)\right]^2 = \frac{1}{2\pi} \int_0^{2\pi} \left[E(k, x, \phi)\right]^2 d\phi. \tag{4.38}$$

To isolate the frequency dependence the error is averaged over the range of x:

$$\left[E_{x\phi}^{\mathrm{rms}}(k)\right]^2 = \frac{1}{2\pi} \int_0^1 \int_0^{2\pi} \left[E(k, x, \phi)\right]^2 d\phi dx. \tag{4.39}$$

The error functions are calculated for a number of different interpolation methods.

Interpolation methods

The interpolation methods discussed here are subdivided into three groups. The first group consists of the intrapolation methods $\mathcal{I}_n$ (4.8), for $n = 0, 1, 2$. The derivatives used in the extrapolation may either be exact or obtained by finite difference. For the FD estimators an equidistant grid is assumed, which means that data points at $x = -1$ and $x = 2$ come into play. In total, five different intrapolation methods are tested.

The second group consists of two variations of Hermite interpolation: one using exact and one using FD derivatives. In two-point Hermite interpolation (e.g., Ralston and Rabinowitz, 2001) a third order polynomial interpolant is constructed

that fits both the function values and its first derivatives exactly. The Hermite interpolant for a function $f(x)$ may be expressed as

$$\mathcal{I}^H[f(x)](x,0,1) = f(0)a_0(x) + f(1)a_1(x) + f'(0)b_0(x) + f'(1)b_1(x), \quad (4.40)$$

with basis functions $a_0(x)$, $a_1(x)$, $b_0(x)$, and $b_1(x)$ defined by

$$a_0(x) = 2x^3 - 3x^2 + 1, \quad (4.41)$$
$$a_1(x) = -2x^3 + 3x^2, \quad (4.42)$$
$$b_0(x) = x^3 - 2x^2 + x, \qquad \text{and} \quad (4.43)$$
$$b_1(x) = x^3 - x^2, \quad (4.44)$$

Two-point Hermite interpolation does not have a simple analogue in dimensions higher than one.

The third group are nearest neighbour interpolations $\mathcal{I}_n^N$. In fact, these are Taylor expansions from the data point that is nearest to the point of interpolation:

$$\mathcal{I}_n^N[f(x)](x,0,1) = \begin{cases} \mathcal{T}_n[f(x)](x,0) & \text{for} \quad 0 \le x < \frac{1}{2} \\ \mathcal{T}_n[f(x)](x,1) & \text{for} \quad \frac{1}{2} \le x \le 1. \end{cases} \quad (4.45)$$

Nearest neighbour interpolations are also studied for $n = 0, 1, 2$. The latter two use derivatives that can either be exact or FD estimates, which makes a total of five different nearest neighbour interpolants. Nearest neighbour interpolation with $n = 2$ and FD derivatives is used by Vanelle and Gajewski (2002) for the interpolation of travel time maps (see Section 4.4.2).

Results

For all types of interpolation, the error functions (4.37) to (4.39) can be calculated analytically. The error curves in Figures 4.3- 4.5 are made using high precision arithmetic in *Mathematica*.

Figure 4.3 shows the phase shift averaged error (4.38) for a fixed frequency of 0.1 as a function of x. For each type of interpolation a single line style is chosen; the distinction between different orders for intrapolation and nearest neighbour interpolation is easily made from the height of the curves. (The lowest curve represents the smallest error and corresponds to the highest order.)

Due to the symmetry of the interpolations, all error curves are symmetrical about $x = 0.5$. Most of the curves, viz. those that represent an interpolation that uses derivatives, are coloured differently on the left and the right side of the graph. Black and grey make a distinction between the methods that use exact derivatives (black on the left side, grey on the right), and those that use FD derivatives (grey on the left, black on the right).

For the more accurate interpolations (lower curves) it obvious that using exact derivatives yield a higher accuracy then FD derivatives. If derivatives are available, Hermite interpolation and third order intrapolation ($n = 2$) contend for the

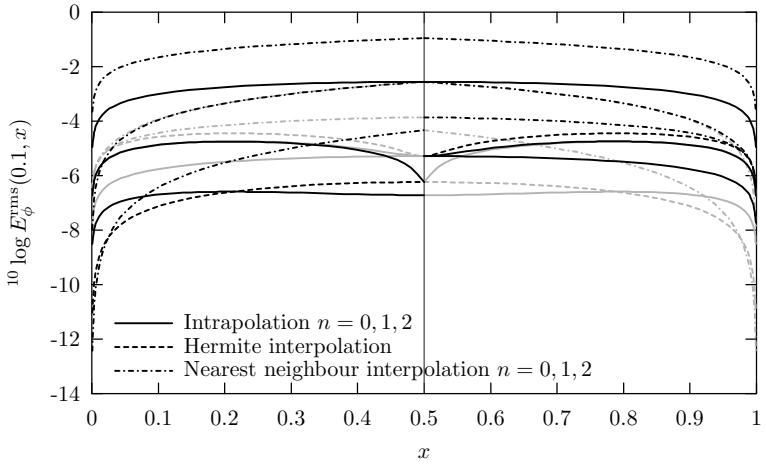


Figure 4.3: The RMS error made in the interpolation of a sine (4.35) with frequency $k = 0.1$, averaged over the full range of possible phase shifts (4.38). All curves are symmetric about $x = 0.5$. Some curves, however, are plotted black on one side, and grey on the other; curves that are black on the left side use either no or exact derivatives, those that are black on the right side use either no or FD derivatives. The line styles of different orders n of an interpolation type are the same, but the lowest curves (least error) correspond to the highest order.

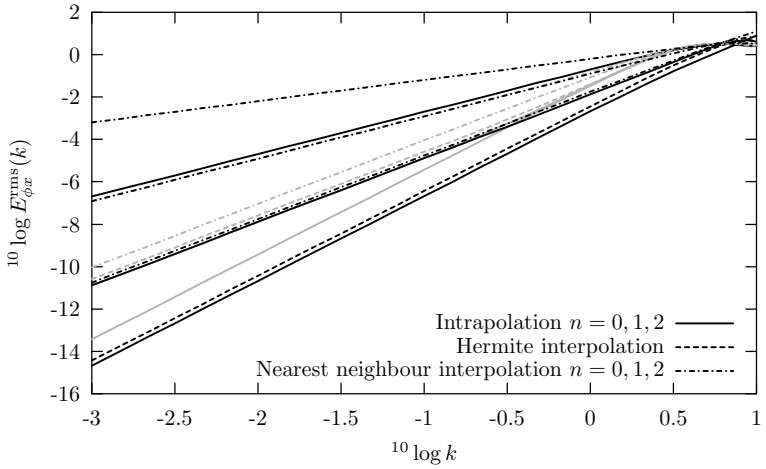


Figure 4.4: The RMS error of interpolating sines (4.35) as a function of frequency, averaged over the full range of possible phase shifts and over the range of x (4.39). The values at $^{10}\log k = -1$ correspond to the RMS averages of the curves of Figure 4.3. Bold black curves correspond to interpolation types that use either no or exact derivatives, while grey curves use FD estimates. The plot clearly indicates the different rates of convergence for the different interpolation types, as well as their relative RMS accuracy.

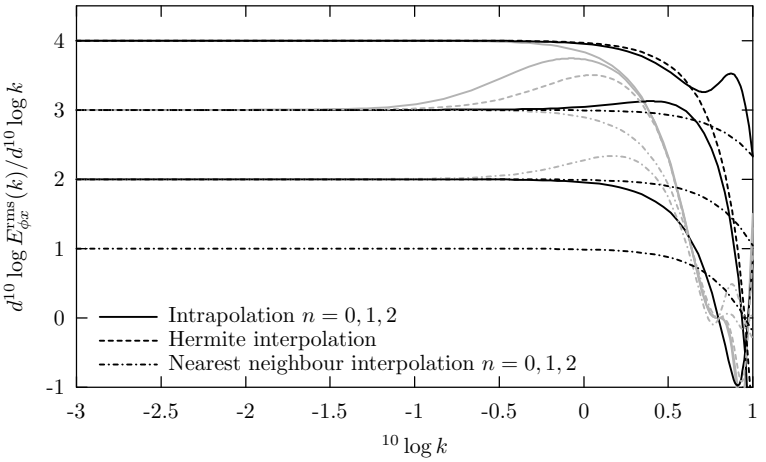


Figure 4.5: The derivatives of the curves in Figure 4.4. The rates of convergence are expressed more clearly. For low frequencies the third order accurate methods have a dominant error term of the fourth order, which gives a logarithmic derivative of 4.0. The only zeroth order method is the nearest neighbour interpolant with $n = 0$.

highest accuracy. Closer to the data points Hermite is more accurate because it fits the derivatives exactly. In the central region, however, intrapolation is more accurate, probably because it uses second order derivatives, while Hermite uses only derivatives of first order. The main disadvantage of Hermite interpolation is that it does not generalise to higher dimension, while intrapolation does.

Looking at the lowest error curves on the right side of the graph, it is clear that in the case of FD derivatives, Hermite interpolation is not as accurate as third order intrapolation. Even if only first derivatives are used, the second order intrapolation is more accurate than the FD Hermite interpolation. From the viewpoint of computational efficiency there is no difference between intrapolation and Hermite interpolation of the same polynomial order. The second order intrapolant is calculated more efficiently, simply because of its lower polynomial order.

The nearest neighbour interpolations are not as accurate as the intrapolations that use the same amount of derivative information. The fact that nearest neighbour interpolation is basically an extrapolation shows from the relatively high error near the centre of the interval.

The convergence behaviour of the interpolation methods is best appreciated by looking at the phase and interval averaged error function (4.39). Figure 4.4 shows the logarithm of this error as a function of the logarithm of the frequency. The linear behaviour of the error functions indicate convergence with polynomial rates. The derivatives of these curves correspond to the order of the error. The order of the error is one higher than the order of accuracy of the interpolations. Figure 4.5 clearly shows that the order of accuracy of the interpolations ranges

from 0 to 3.

In both convergence plots the black lines correspond to the interpolation methods that use exact derivatives, while the grey lines correspond to those that use FD derivative estimates. The intrapolations and the nearest neighbour interpolations show the expected convergence behaviour. In general, the order of accuracy of the nearest neighbour interpolations are equal to the order of the derivatives that are used. For the intrapolations, the order of accuracy is always one higher than the order of derivatives (n) used.

Hermite interpolation with exact derivatives shows the expected result of being accurate to third order. If FD estimates are used for the derivatives, however, Hermite interpolation turns out to be accurate only to second order, because the FD estimates themselves are only accurate to second order (4.25). For intrapolation the latter does not pose a limitation on the order of accuracy because of the fortuitous cancellation of errors discussed in Section 4.3. Even intrapolation based only on first order FD derivatives (n=1) is slightly more accurate than FD Hermite interpolation, while both methods use the same data.

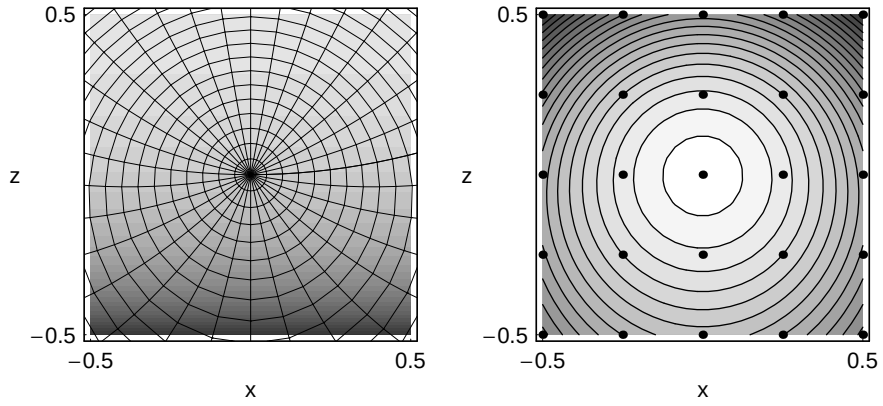
If no exact derivatives are available, intrapolation should be preferred over Hermite interpolation. Also, if both first and second order exact derivatives are available, Figure 4.4 indicates that intrapolation is the method of choice for interpolation, because it gives the lowest error curve in the plot. If only first order exact derivatives are known Hermite interpolation is preferred. In higher dimensions, however, Hermite interpolation is not available. Finally, intrapolation is always to be preferred over nearest neighbour interpolations.

4.4.2 Travel time interpolation

An important application of interpolation in the context of this thesis is the interpolation of travel time maps for Kirchhoff-type imaging. Coarse gridded travel time maps in depth are commonly calculated for the full range of source and receiver locations, prior to the actual imaging. Upon imaging these coarse maps are interpolated to a finer grid. Higher order accuracy in the interpolation allows for coarser maps to start with, which saves storage space and data traffic overhead in practical implementations.

The advantages of travel time interpolation with higher order accuracy were shown by Vanelle and Gajewski (2002). They proposed the second order ($n = 2$) nearest neighbour interpolant discussed in Section 4.4.1, using FD estimates of the derivatives. Here, it is shown that using the same amount of information a greater accuracy can be achieved using intrapolation. The second order method proposed by Vanelle and Gajewski (2002) may therefore be replaced by third order intrapolation, which enhances the advantages at negligible additional cost. Another advantage is that the interpolated travel times are continuous.

To demonstrate that intrapolation works in higher dimensions, a single example of travel time interpolation in a regular grid is shown here. Other examples will be provided in later chapters.



(a) Rays and wave fronts (b) Contours of squared travel time

Figure 4.6: Figure (a) shows the rays and wave fronts in a medium with V satisfying $V^{-2} = 1 - z$. The grey tones in the background indicates the velocity (darker means faster). Figure (b) shows a contour plot of the corresponding squared travel time. On top a rectangular grid of points is drawn, which contain the data used in the interpolation

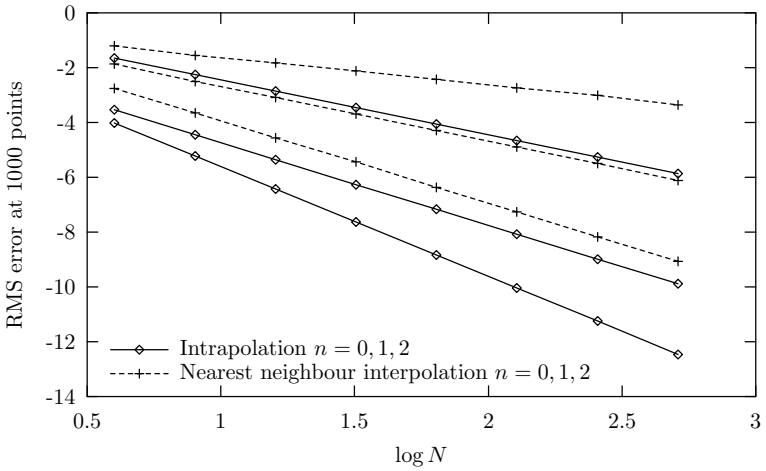


Figure 4.7: Convergence of RMS interpolation error for the square of travel time at 1000 random samples in the domain of figure 4.6. The number N corresponds to the number of grid intervals in both dimensions. Each interpolation type has three curves, arranged in increasing order from top to bottom.

Vanelle and Gajewski (2002) argue that it is better to interpolate the square of the travel time (*hyperbolic interpolation*) than the travel time itself (*parabolic interpolation*), because in a homogeneous medium the square of the travel time is a quadratic function of the position relative to the source. Hence, in a homogeneous medium any interpolation method of second or higher order accuracy will reconstruct the squared travel time perfectly. In inhomogeneous media, the interpolation will not be exact anymore, but close to the source the hyperbolic interpolation will always be superior.

The travel time map to be interpolated is one that can be obtained analytically from the analytic ray solution. The medium, shown in Figure 4.6(a), along with a number of rays and wave fronts emitted by a point source in the origin, has a constant gradient of the square of the slowness (reciprocal velocity) of magnitude 1, and velocity of 1 at the origin. Figure 4.6(b) shows some contours of the squared travel time, as well as an example of a rectangular grid defined in the medium.

Because the travel times on the grid are calculated analytically, the grid spacing determines the accuracy of interpolation. For a range of grid spacings, the RMS interpolation error is calculated from a set of 1000 random locations. For N the number of grid intervals in each dimension the error is plotted in Figure 4.7.

As in Section 4.4.1, the nearest neighbour interpolations as well as the intrapolations show the expected convergence rates. Intrapolation based on first and second order derivatives ($n = 2$) proves to be the most accurate. Surprisingly, even intrapolation with only first order derivatives ($n = 1$), is more accurate than the nearest neighbour interpolation that uses both first and second derivatives.

In practical applications one is usually less interested in the order of convergence of a given interpolation method then in the grid spacing corresponding to a given error. The results of Figure 4.7 can be fitted with linear functions. In this way the number of grid intervals needed for each interpolation method to obtain an acceptable error of, say 10^{-4} can be deduced. Rounded to the first integer above, this gives 60, 6, and 5 for intrapolation with $n = 0, 1, 2$, respectively, and 2393, 46, and 11 for the respective nearest neighbour interpolations. The higher the desired accuracy, the greater the advantage of intrapolation.

4.5 Discussion and conclusions

A new technique for accurate interpolation using derivative information is presented. The technique is called intrapolation, because it is a hybrid of linear interpolation and extrapolation. The extrapolation is done by a modification of the Taylor expansion, referred to as the Dutch Taylor expansion. The order of accuracy of intrapolation is one higher than that of an individual Taylor expansion that uses the same amount of derivative information.

Intrapolation is generally applicable in spaces of arbitrary dimension, in regular as well as irregular data distributions. For interpolation inside an irregular data distribution, the presence of a geometric structure such as an N-dimensional

triangulation is assumed. For interpolation in rectangular grids, intrapolation may also be applied without the presence of independent derivative information. Instead, finite difference estimates for the derivatives may be used. Using only three point FD stencils, an interpolation of third order accuracy is obtained.

Intrapolation does not attempt to fit the available derivative data perfectly. It only uses this data to "steer" the interpolant in the right direction, resulting in an enhanced accuracy. It should be noted, however, that the derivatives are in general discontinuous across the edges of the interpolants. If smoothness rather than accuracy is the reason for higher order interpolation it will be advisable to use a different method (e.g., Press et al., 1992; Sambridge et al., 1995).

An important advantage of not attempting to fit the derivatives perfectly is that it is not necessary to use high polynomial orders for the interpolants. Higher order polynomials are more expensive to evaluate, and are more sensitive to errors in the data. In intrapolation, the influence of the derivatives at all of the data points is (in a way) averaged – note the reduced size of the coefficients in the Dutch Taylor expansion – and therefore the interpolants are less sensitive to errors and more stable than methods that fit the derivatives perfectly.

In order to check the accuracy of intrapolation the method is subjected to two tests. The first test in a 1-D setting shows that the theoretical order of accuracy is indeed obtained in practice. For two-point interpolation, intrapolation outperforms all other interpolation methods. Only if the first derivatives are known exactly, Hermite interpolation yields a third order accuracy, while intrapolation needs second derivatives as well to improve on that. In higher dimensions, however, Hermite interpolation is not available.

The second test involves the interpolation of travel times for a point source in 2-D, where intrapolation is tested against nearest neighbour interpolation, both using FD derivative estimates. It turns out that in this case intrapolation using only a single derivative already outperforms the nearest neighbour interpolation that uses second derivatives as well.

It seems fair to conclude that intrapolation is a very attractive method for accurate interpolation both for regular and – if independent derivative data is available – for irregular data distributions. Implementation is straightforward. The method should be useful in a wide variety of applications. Examples of its use in ray field calculations are provided in Chapter 5, while the applicability of the Dutch Taylor expansion in a broader context is further investigated in Chapter 6.

4.A Appendix

4.A.1 Dutch Taylor expansion coefficients

The intrapolant of polynomial order $n+1$ is constructed using the the function derivatives up to order n. This appendix provides the proof that the Dutch Taylor expansion coefficients a_k^n, as stated in Equation (4.19), render the intrapolant (4.8)

exact for polynomials up to order $n+1$.

Point of departure is the Dutch Taylor expansion 4.21. For a given function $f(x)$, and a data point ξ at which the derivatives up to order n are known, this formula gives the extrapolated value as a function of the point of evaluation x.

Alternatively one can interpret (4.21) as a function of ξ, for a fixed point of evaluation x. This perspective is very useful here, because it clearly shows the conditions under which the intrapolant (4.8) is exact. Since the intrapolant is defined as a linear interpolation, it is exact if and only if extrapolation formula (4.21) is a linear function of data point ξ, passing through the correct value at $\xi = x$. Here it is shown that this is the case for polynomials up to order $n+1$. The proof is divided in two parts, one for polynomials up to order n, and one for the polynomial of order $n+1$. First, however, some binomial identities are shown that are useful in the derivations.

Binomial identities

The binomial 'n over k' is defined as

$$\binom{n}{k} = \frac{n!}{(n-k)!k!}. \tag{4.46}$$

The binomial occurs in the expansion of the n-th power of a sum:

$$(A+B)^n = \sum_{k=0}^{n} \binom{n}{k} A^k B^{n-k}. \tag{4.47}$$

This relation will be used in the form

$$\sum_{k=0}^{n} \binom{n}{k} (x-\xi)^k \xi^{n-k} = x^n. \tag{4.48}$$

Finally,

$$k\binom{n}{k} = n\binom{n-1}{k-1} \tag{4.49}$$

is another binomial identity that will be used.

Proof for polynomials of order $m \leq n$.

For a monomial x^m, with $0 \leq m \leq n$, the n-th order Taylor expansion is exact, so the extrapolation formula (4.21) becomes

$$\mathcal{D}_n[x^m](x;\xi) = x^m - \frac{1}{n+1} \sum_{k=1}^{m} k\binom{m}{k} (x-\xi)^k \xi^{m-k}, \tag{4.50}$$

where the limit of the sum has changed from n to m, simply because higher terms are zero.

Using binomial identity (4.49) and the index $K = k - 1$ gives

$$\begin{aligned}\mathcal{D}_n[x^m](x;\xi) &= x^m - \frac{m}{n+1}\sum_{K=0}^{m-1}\binom{m-1}{K}(x-\xi)^{K+1}\xi^{m-1-K} \\ &= x^m - \frac{m}{n+1}(x-\xi)\sum_{K=0}^{m-1}\binom{m-1}{K}(x-\xi)^{K}\xi^{m-1-K}.\end{aligned} \tag{4.51}$$

Application of (4.48) finally leads to

$$\begin{aligned}\mathcal{D}_n[x^m](x;\xi) &= x^m - \frac{m}{n+1}(x-\xi)x^{m-1} \\ &= \left(1 - \frac{m}{n+1}\right)x^m + \frac{m}{n+1}x^{m-1}\xi.\end{aligned} \tag{4.52}$$

The extrapolation formula is apparently a linear function of data point ξ for polynomials up to order n, while it has the exact value x^m for $\xi = x$. Hence, the intrapolant is exact at least up to order n.

Proof for polynomial of order $n+1$

For a monomial x^{n+1} the n-th order Taylor expansion is no longer exact, but the error is easy to find:

$$\mathcal{T}_n[x^{n+1}](x;\xi) = x^{n+1} - (x-\xi)^{n+1}. \tag{4.53}$$

The Dutch Taylor expansion becomes

$$\begin{aligned}\mathcal{D}_n[x^{n+1}](x;\xi) = x^{n+1} - (x-\xi)^{n+1} \\ - \sum_{k=1}^{n}\frac{k}{n+1}\binom{n+1}{k}(x-\xi)^k\xi^{n+1-k}.\end{aligned} \tag{4.54}$$

Using binomial identities (4.48) and (4.49), and the index $K = k-1$ this expression can be simplified to

$$\begin{aligned}\mathcal{D}_n[x^{n+1}](x;\xi) &= x^{n+1} - (x-\xi)^{n+1} - \sum_{K=0}^{n-1}\binom{n}{K}(x-\xi)^{K+1}\xi^{n-K} \\ &= x^{n+1} - (x-\xi)^{n+1} - (x-\xi)\left[x^n - (x-\xi)^n\right] \\ &= x^n\xi.\end{aligned} \tag{4.55}$$

Again, this is an expression that is linear in ξ. In fact, it is of the same form as the expression obtained for lower order polynomials in Equation (4.52), with $m = n+1$. Linear interpolation of this expression using the intrapolant (4.8) indeed returns the exact polynomial x^m for $0 \le m \le n+1$. Hence, the intrapolant is accurate to order $n+1$.