APPROXIMATION THEORY

INTRODUCTION

Approximation theory is a specialized discipline of mathematics that has become indispensable to the computer and computing sciences. The approximation of magnitudes and functions describing some physical behavior is an integral part of scientific computing, queueing problems, neural networks, graphics, robotics, network traffic, financial trading, antenna design, floating-point arithmetic, image processing, speech analysis, and video signal filtering, to name just a few.

The sense of finding a simple mathematical function that describes some behavior approximately is twofold. Either the exact behavior that one is studying cannot be expressed in a closed mathematical formula or its exact description is far too complicated for practical use, for instance, in an implementation. In both cases, an approximate and preferably simple formula is required. Now the question arises which mathematical functions are considered simple.

The simplest and fastest functions for implementation are polynomials, because they only use additions and multiplications. Then come rational functions, which also need the hardware divisions, one or more depending on their representation. Rational functions offer the clear advantage that they can reproduce asymptotic behavior (vertical, horizontal, slant), which is something polynomials are incapable of. For periodic phenomena, linear combinations of trigonometric functions make good candidates. For rapidly decaying magnitudes, linear combinations of exponentials can be used.

In approximation theory, one distinguishes between interpolation and so-called least-squares problems. In the former, one wants the approximate model to take exactly the same values as prescribed by given data at some argument values. In the latter, a set of data is regarded as a trend and is approximated by a simple model in the best sense. The difference will be formalized in the next section.

In the sequel of the presentation, we limit ourselves to a description of one-dimensional approximation problems. Although the growth in computer power allows for the study of more and more complex models and simulations, the theory of several more-dimensional approximation problems is still not sufficiently complete. We indicate where up-to-date literature on the more-dimensional generalizations can be found.

NUMERICAL INTERPOLATION

Let some data f_i be given at some points $x_i \in [a, b]$ where $i = 0, \ldots, n$. If points x_i are repeated, then in fact not only the value of some underlying function $f(x)$ is given (or measured) at x_i but also one or more higher derivatives $f^{(j)}(x_i)$. In this section, we deal with the two extreme cases: either the value of the function and that of the first n derivatives are all given at one single point x_0 or all points x_i are mutually distinct and no derivative information is available. In the next two subsections polynomials are used, whereas in the subsections on Padé approximation and on rational interpolation, rational functions are used. When n grows large, polynomial interpolation is not suitable because of the Runge phenomenon, which is explained below. Another possibility is the use of piecewise polynomial functions as interpolant, a technique that is also presented below.

Taylor Series Approximation

We explore the possibility to approximate a function $f(x)$ by a polynomial of the form

$$p_n(x) = \sum_{j=0}^{n} d_j (x - x_0)^j \tag{1}$$

It is clear, by successive substitution and derivation, that

$$\begin{aligned} p_n(x_0) &= d_0 \\ p_n'(x_0) &= d_1 \\ p_n''(x_0) &= 2d_2 \\ &\vdots \\ p_n^{(n)}(x_0) &= n! d_n. \end{aligned}$$

Hence, if the value of $f(x)$ and its derivatives $f^{(i)}(x)$ for $i = 1, \ldots, n$ are given at the point x_0, then the polynomial $p_n(x)$ with $d_j = f^{(j)}(x_0)/j!$ satisfies

$$p_n^{(i)}(x_0) = f^{(i)}(x_0), \ i = 0, \ldots, n. \tag{2}$$

This polynomial is called the Taylor polynomial of degree n and is the partial sum of degree n of the Taylor series representation of $f(x)$,

$$f(x) = \sum_{j=0}^{\infty} \frac{f^{(j)}(x_0)}{j!} (x - x_0)^j.$$

The question where the series actually equals the function $f(x)$ is not within the scope of this exposition.

Newton and Lagrange Interpolation

For $n + 1$ given $f_i = f(x_i)$ at mutually distinct points x_i, the polynomial interpolation problem of degree n,

$$p_n(x) = \sum_{j=0}^{n} a_j x^j, \ p_n(x_i) = f_i, \quad i = 0, \ldots, n$$

has a unique solution for the coefficients a_j. Now let us turn to the computation of $p_n(x)$. Essentially two approaches can be used, depending on the use of the polynomial interpolant afterward. If one is interested in easily updating the polynomial interpolant by adding an extra data point and consequently increasing the degree of $p_n(x)$, then Newton's formula for the interpolating polynomial is most useful. If one wants to use the interpolant for several sets of values f_i while keeping the points x_i fixed, then Lagrange's formula is appropriate.

Wiley Encyclopedia of Computer Science and Engineering, edited by Benjamin Wah.

In the Newton form one writes the interpolating polynomial $p_n(x)$ as

$$p_n(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1) + \ldots + b_n(x - x_0)\cdots(x - x_{n-1})$$

The coefficients b_j are then obtained as the divided differences $b_j = f[x_0,\ldots,x_j]$ from the recursive scheme

$$f[x_j] = f_j, \quad j = 0,\ldots,n$$
$$f[x_0,x_j] = \frac{f_j - f_0}{x_j - x_0}, \quad j = 1,\ldots,n$$
$$f[x_0,x_1,\ldots,x_{k-1},x_k,x_j] = \frac{f[x_0,x_1,\ldots,x_{k-1},x_j] - f[x_0,x_1,\ldots,x_{k-1},x_k]}{x_j - x_k}, \quad k,j = 2,\ldots,n.$$

We remark that the divided differences $f[x_0,\ldots,x_j]$ for $j = 0,\ldots,n$ do not depend on the ordering of the data (x_i,f_i). Newton's form for the interpolating polynomial is very handy when one wants to update the interpolation with an additional point (x_{n+1}, f_{n+1}). It suffices to add the term

$$b_{n+1}(x - x_0)\cdots(x - x_n)$$

to $p_n(x)$, which does not destroy the previous interpolation conditions since it evaluates to zero at all the previous x_i, and to complement the recursive scheme for the computation of the divided differences with the computation of

$$f[x_0,x_1,\ldots,x_k,x_{n+1}], \ k = 0,\ldots,n.$$

In the Lagrange form, which is especially suitable if the interpolation needs to be repeated for different sets of f_i at the same points x_i, another form for $p_n(x)$ is used. We write

$$p_n(x) = \sum_{j=0}^{n} c_j \beta_j(x), \ \beta_j(x) = \prod_{k=0,k\neq j}^{n} \frac{(x - x_k)}{(x_j - x_k)}.$$

The basis functions $\beta_j(x)$ satisfy a simple interpolation condition themselves, namely

$$\beta_j(x_i) = \begin{cases} 0 & \text{for} \quad j \neq i \\ 1 & \text{for} \quad j = i \end{cases}.$$

Hence, the choice $c_j = f_j$ for the coefficients solves the interpolation problem. So when altering the f_i, without touching the x_i that make up the basis functions $\beta_j(x)$, it takes no extra computation scheme to get the new coefficients c_j.

The Runge Phenomenon

What happens if we continue updating the interpolation problem with new data, in other words, if we let the degree n of the interpolating polynomial $p_n(x)$ increase? Will the interpolating polynomial of degree n become better and better? Surprisingly enough not! At least not for freely chosen points x_i. The next counterexample illustrates this. Consider

$$f(x) = \frac{1}{1 + 25x^2}, \ -1 \le x \le 1$$

and take equidistant interpolation points $x_i = -1 + 2i/n$, $i = 0,\ldots,n$. Then the error $(f - p)(x)$ increases with n, toward the endpoints of the interval. Take a look at the bell-shaped $f(x)$ and the interpolating polynomial $p_n(x)$ for $n = 10$ and $n = 20$ in the Figs. 1 and 2.

This phenomenon is called Runge's phenomenon as he was the first to discover this behavior for real-valued interpolation. An explanation for it can be found in the fundamental theorem of algebra, which states that a polynomial has as many zeroes as its degree. Each of these zeroes can be real or complex. So if n is large and in case the zeroes are all real, the polynomial under consideration displays a very oscillatory behavior.

On the other hand, under certain simple conditions for $f(x)$, it can be proved that if the interpolation points x_i equal

$$x_i = \frac{a+b}{2} + \frac{b-a}{2}\cos\left(\frac{i\pi}{n}\right), \quad i = 0,\ldots,n$$

then

$$\lim_{n\to\infty} \max_{x\in[-1,1]} |(f - p_n)(x)| = 0.$$

The effect of this choice of interpolation points, if it is at all possible to control the choice of the x_i, is illustrated in the Figs. 3 and 4.

If a lot of accurate datapoints have to be used in an interpolation scheme, the next subsection offers a better

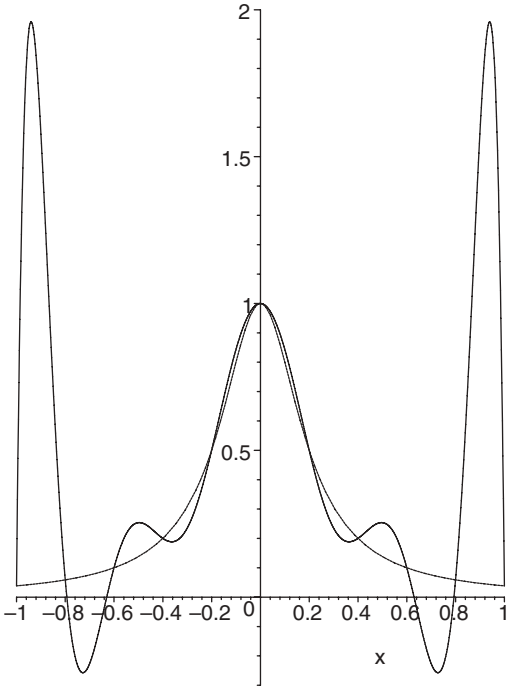

Figure 1. Degree 10 equidistant interpolation.

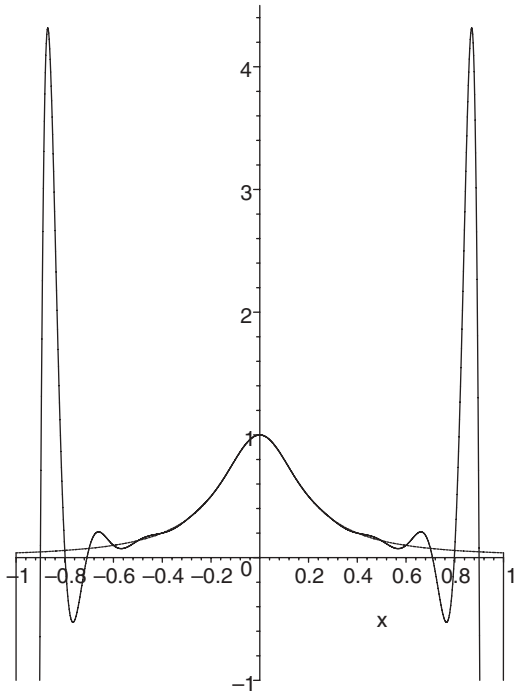

Figure 2. Degree 20 equidistant interpolation.

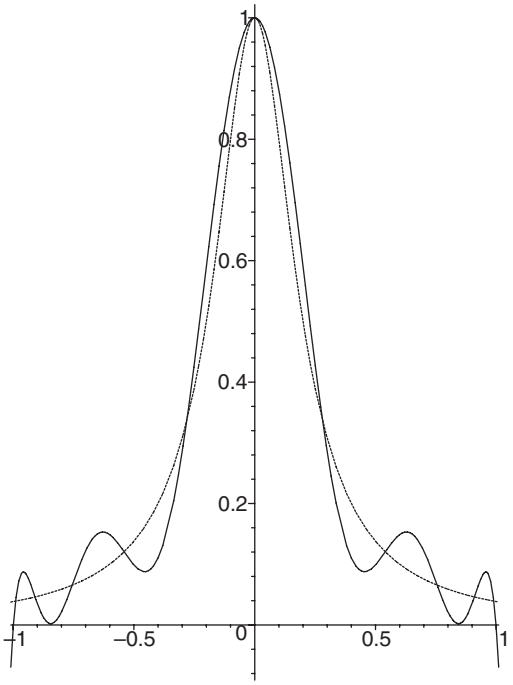

Figure 3. Degree 10 nonequidistant interpolation.

alternative than a monolithic high-degree polynomial interpolant.

Spline Interpolation

To avoid the Runge phenomenon when interpolating large data sets, piecewise polynomials, also called splines, can be used. To this end we divide the data set of $n+1$ points into smaller sets each containing 2 data points. Rather than interpolating the full data set by one n-degree polynomial, we interpolate each of the smaller data sets by a low-degree polynomial. These separate polynomial functions are then glued together as continuously differentiable as possible.

Take, for instance, the data set (x_i, f_i) with $x_0 < x_1 < \ldots x_n$, and consider linear polynomials interpolating every two consecutive (x_i, f_i) and (x_{i+1}, f_{i+1}). These linear polynomial pieces can be glued together in the data points (x_i, f_i) to result in a piecewise linear continuous function or polygonal curve. Remark that the function is continuous but not differentiable at the interpolation points since it is polygonal.

If we introduce two parameters, Δ and D, to respectively denote the degree of the polynomial pieces and the differentiability of the overall function, then for the polygonal curve, $\Delta = 1$ and $D = 0$. With $\Delta = 2$ and $D = 1$, a piecewise quadratic and smooth (meaning continuously differentiable in the entire interval $[x_0,\ldots,x_n]$) function is constructed. With $\Delta = 3$ and $D = 2$, a piecewise cubic and twice continuously differentiable function is obtained. The slope of a smooth function is a continuous quantity. Twice continuously differentiable functions also enjoy continuous curvature. Can the naked eye distinguish between continuous and discontinuous curvature in a function? The untrained eye certainly cannot! As an example we take the cubic polynomial pieces

$$c_1(x) = x^3 - x^2 + x + 1, \quad x \in [-1, 0]$$
$$c_2(x) = x^3 + x^2 + x + 1, \quad x \in [0, 1]$$

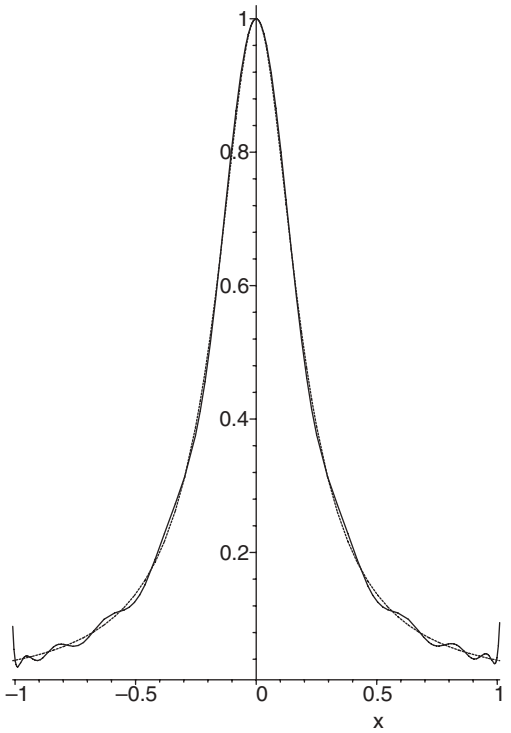

Figure 4. Degree 20 nonequidistant interpolation.

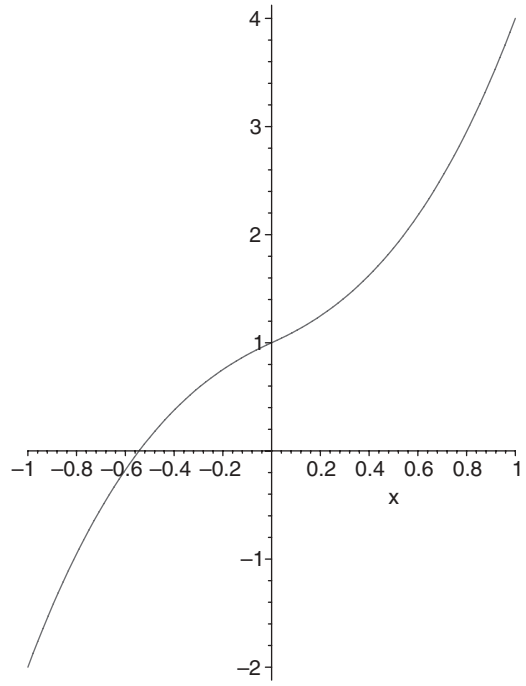

Figure 5. Piecewise cubic function that is not twice continuously differentiable.

and glue these together in 0. The result is a function that is continuous and differentiable in the origin, but for the second derivatives at the origin, we have $c_1^{(2)}(0) = -2$, whereas $c_2^{(2)}(0) = +2$. Nevertheless the result of the gluing procedure shown in Fig. 5 is a very pleasing function that at first sight looks smooth enough. But although Δ equals 3, D is only 1.

Since a trained eye can spot these discontinuities, the most popular choice for piece-wise polynomial interpolation in industrial applications is $\Delta = 3$ and $D = 2$. For manufacturing, the smoothness of the curvature is not unimportant.

Let us take a look at the general situation where $\Delta = m$ and $D = m-1$. Assume we are given the interpolation points $x_0,\ldots,x_n$. With these $n+1$ points, we can construct n intervals $[x_i,x_{i+1}]$. The points x_0 and x_n are the endpoints, and the other $n-1$ interpolation points are called the internal points. If $\Delta = m$, then per interval $[x_i, x_{i+1}]$ we have to determine $m + 1$ coefficients, because the explicit formula for the spline on $[x_i, x_{i+1}]$ is an m-degree polynomial:

$$S(x) = s_i(x), \quad x \in [x_i, x_{i+1}], \quad i = 0, \ldots, n-1$$
$$s_i(x) = \sum_{j=0}^{m} a_j^{(i)} x^j.$$

So in total $n(m + 1)$ unknown coefficients $a_j^{(i)}$ have to be computed. From which conditions? There are the $n + 1$ interpolation conditions $S(x_i) = f_i$, and we have the smoothness or continuity requirements at the internal points, expressing that several derivatives of $s_{i-1}(x)$ evaluated at the right endpoint of the domain $[x_{i-1}, x_i]$ should coincide with the derivatives of $s_i(x)$ when evaluated at the left endpoint of the domain $[x_i, x_{i+1}]$,

$$s_{i-1}^{(k)}(x_i) = s_i^{(k)}(x_i), \; i = 1, \ldots, n-1, \quad k = 0, \ldots, m-1.$$

The latter add another $(n-1)m$ continuity conditions, which brings us in total to $n + 1 + (n - 1)m = n(m + 1) - m + 1$ conditions for $n(m+1)$ unknowns. In other words, we lack $m - 1$ conditions to determine the m-degree piecewise polynomial interpolant with all-over smoothness of order $m - 1$. So when $m = 1$, which is the case of the piecewise linear spline or polygonal curve, no conditions are lacking. When $m = 2$, usually a value for $s_0'(x_0)$ is given as an additional piece of information. When $m = 3$, which is the case of the widely used cubic spline, values for $s_0''(x_0)$ and $s_{n-1}''(x_n)$ are provided (cubic spline with clamped end conditions) or they are set to zero (natural cubic spline).

Padé Approximation

The rational equivalent of the Taylor polynomial (1) satisfying Eq. (2) is the irreducible rational function $r_{k,\ell}(x)$ of numerator degree at most k and denominator degree at most ℓ, which satisfies

$$r_{k,\ell}^{(i)}(x_0) = f^{(i)}(x_0), \; i = 0, 1, \ldots, n \tag{3}$$

with n as large as possible. It is also called the Padé approximant of degree k over ℓ. The aim is to have $n = k + \ell$. Two questions originate immediately from the definition. Why impose at least one condition less than the total number $k + \ell + 2$ of coefficients in a rational function of degree k in the numerator and degree ℓ in the denominator? Can n actually be less than $k + \ell$, and when does that occur or not occur?

The answer to the first question is simple. A rational function

$$\frac{p_{k,\ell}(x)}{q_{k,\ell}(x)} = \frac{a_0 + a_1(x - x_0) + \ldots a_k(x - x_0)^k}{b_0 + b_1(x - x_0) + b_\ell(x - x_0)^\ell}$$

is only unique up to a scalar multiple in numerator and denominator. Hence, not all coefficients in numerator and denominator must be fixed by the approximation conditions (3). One coefficient can be determined by the form in which we want to write down $r_{k,\ell}(x)$, such as the requirement to make the numerator monic, or the denominator monic, or have a normalized constant term in the denominator. The Padé approximant is still uniquely determined by Eq. (3).

The answer to the second question requires some analysis. Computing the numerator and denominator coefficients of $r_{k,\ell}(x)$ from Eq. (3) gives rise to a nonlinear system of equations. So let us explore whether the Padé approximant can also be obtained from the linearized approximation conditions

$$(fq_{k,\ell} - p_{k,\ell})^{(i)}(x_0) = 0, \; i = 0, 1, \ldots, k + \ell. \tag{4}$$

Indeed, the linearized conditions (4) always have at least one nontrivial solution for the coefficients $a_0, \ldots, a_k, b_0, \ldots, b_\ell$, because they are a homogeneous linear system of $k + \ell + 1$ conditions in $k + \ell + 2$ unknowns,

$$\begin{cases} d_0 b_0 = a_0 \\ d_1 b_0 + d_0 b_1 = a_1 \\ \vdots \\ d_k b_0 + \ldots + d_{k-\ell} b_\ell = a_\ell \end{cases}, \begin{cases} d_{k+1} b_0 + \ldots + d_{k-\ell+1} b_\ell = 0 \\ \vdots \\ d_{k+\ell} b_0 + \ldots + d_k b_\ell = 0 \end{cases}$$

where the d_j are given by Eq. (1) with $d_j = 0$ for $j < 0$. Moreover, all solutions $p_{k,\ell}(x)$ and $q_{k,\ell}(x)$ of Eq. (4) are equivalent in the sense that they have the same irreducible form. Hence every solution of Eq. (3) with $n = k + \ell$ also satisfies Eq. (4) but not vice versa. From Eq. (4) for $p_{k,\ell}(x)$ and $q_{k,\ell}(x)$, we find for their unique irreducible form $r_{k,\ell}(x) = p^*_{k,\ell}(x)/q^*_{k,\ell}(x)$ that

$$\begin{aligned} &(f - r_{k,\ell})^{(i)}(x_0) = 0,\ i = 0, \ldots, k' + \ell' + r, \\ &k' = \partial p^*_{k,\ell}, \ell' = \partial q^*_{k,\ell}, r \geq 0. \end{aligned} \tag{5}$$

In some textbooks, the Padé approximation problem of degree k over ℓ is said to have no solution if $k' + \ell' + r < k + \ell$. In others the Padé approximant $r_{k,\ell}$ is identified with $r_{k',\ell'} = p^*_{k,\ell}/q^*_{k,\ell}$ in that case. This kind of complication does not occur when using the polynomial model (1). But then a polynomial model cannot reproduce asymptotic behavior.

Let us illustrate the situation with a simple example. Take $x_0 = 0$ with $d_0 = 1, d_1 = 0, d_2 = 1$ and $k = 1 = \ell$. Then the linearized conditions (4) are

$$\begin{cases} b_0 = a_0 \\ b_1 = a_1 \\ b_0 = 0. \end{cases}$$

A solution is given by $p_{1,1}(x) = x$ and $q_{1,1}(x) = x$. Hence we find $r_{1,1}(x) = 1$, $k' = 0$, $\ell' = 0$ and

$$(f - r_{1,1})^{(2)}(x_0) = 2 \neq 0.$$

Since $r = 1$, we have $k' + \ell' + r = 1 < k + \ell = 2$.

Rational Interpolation

The rational equivalent of polynomial interpolation at mutually distinct interpolation points x_i consists in finding an irreducible rational function $r_{k,\ell}(x)$, of numerator degree at most k and denominator degree at most ℓ, that satisfies

$$r_{k,\ell}(x_i) = f_i,\ i = 0, \ldots, k + \ell \tag{6}$$

where $f_i = f(x_i)$. Instead of solving Eq. (6), one considers the linearized equations

$$(fq_{k,\ell} - p_{k,\ell})(x_i) = 0,\ i = 0, \ldots, k + \ell \tag{7}$$

where $p_{k,\ell}(x)$ and $q_{k,\ell}(x)$ are polynomials of respective degree k and ℓ. Condition (7) is a homogeneous linear system of $k+\ell+1$ equations in $k+\ell+2$ unknowns and, hence, always has a nontrivial solution. Moreover, as in the Padé approximation case, all solutions of Eq. (7) are equivalent in the sense that they deliver the same unique irreducible rational function.

By computing the irreducible form $r_{k,\ell}(x)$ of $p_{k,\ell}(x)/q_{k,\ell}(x)$, common factors in numerator and denominator are canceled and it may well be that $r_{k,\ell}$ does not satisfy the interpolation conditions (6) anymore although $p_{k,\ell}$ and $q_{k,\ell}$ are solutions of Eq. (7), because one or more of the canceled factors may be of the form $(x - x_i)$ with x_i an interpolation point. A simple example illustrates this. Let $x_0 = 0$, $x_1 = 1$, $x_2 = 2$ with $f_0 = 0, f_1 = 3, f_2 = 3$, and take $k = 1 = \ell$. Then the homogenous linear system of interpolation conditions is

$$\begin{cases} a_0 = 0 \\ 3(b_0 + b_1) - (a_0 + a_1) = 0 \\ 3(b_0 + 2b_1) - (a_0 + 2a_1) = 0 \end{cases}$$

A solution is given by $p_{1,1}(x) = 3x$ and $q_{1,1}(x) = x$. Hence, $r_{1,1}(x) = 3$ and clearly $r_{1,1}(x_0) \neq f_0$. The interpolation point x_0 is then called unattainable. This problem can only be fixed by increasing the degrees k and/or ℓ until the interpolation point is not unattainable anymore. Note that unattainable interpolation points do not occur in polynomial interpolation ($\ell = 0$).

MULTIVARIATE INTERPOLATION

Several multivariate generalizations of the above interpolation problems have been studied in the past decades:

- A nice state of the art on multivariate polynomial interpolation can be found in Ref. 1, listing different definitions for divided differences and Newton or Lagrange forms. In particular we refer to Refs. 2 and 3.
- Reviews of multivariate polynomial spline techniques for scattered data or data on triangulations are, respectively, given in Refs. 4 and 5.
- Information on how to generalize the concept of Padé approximation to functions of more variables is bundled in Ref. 6. Some approaches are closer to the univariate theory than others.
- For the subject of multivariate rational interpolation we refer to Ref. 7, with many references to computer science and engineering applications therein.

LEAST-SQUARES APPROXIMATION

When the quality of the data does not justify that we impose an exact match on the model, or when the quantity of the data is just overwhelming and rather depicts a trend than very precise measurements, then interpolation techniques are of no use. It is far better to find a linear combination of suitable basis functions that approximates the data in some best sense. The discrete linear least squares problem is introduced next. How the bestness or nearness of the approximation is measured is explained. Different

measures lead to different approximants and are to be used in different contexts. The importance of the use of orthogonal basis functions is also illustrated. Then the continuous linear least-squares problem is formulated. Finally, a way to deal with periodic data is given.

Discrete-Least Squares Approximation

Let us consider a large data set of values f_i at points x_i that we want to approximate by a linear combination of some linearly independent basis functions $b_j(x)$:

$$\lambda_1 b_1(x_i) + \ldots + \lambda_n b_n(x_i) = f_i, \quad i = 1, \ldots, m \gg n. \tag{8}$$

This $m \times n$ linear system can be written compactly as

$$A\lambda = f, \quad \lambda = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{pmatrix}, \quad f = \begin{pmatrix} f_1 \\ \vdots \\ f_m \end{pmatrix},$$
$$A = (a_{ij})_{m\times n} = (b_j(x_i))_{m\times n}. \tag{9}$$

Unless at least $m - n$ linearly dependent equations can be found among the m linear equations, the system cannot be solved exactly. The residual vector is given by

$$r = f - A\lambda, \quad r \in \mathrm{R}^m$$

and the solution λ we are looking for is the one that solves the system best, in other words, that makes the magnitude (or norm) of the residual vector minimal (it will soon become apparent why the problem is called a least-squares problem). This optimization problem usually translates to

$$(A^T A)\lambda = A^T f$$

which is now a square linear system of equations and is called the system of normal equations. In practice, the system of normal equations is never solved since different numerical techniques, which are applicable directly to the overdetermined system of linear conditions (9), are more suitable. It is important to note that if the matrix A of the overdetermined linear system has maximal column rank, then the matrix $A^T A$ is nonsingular.

Choice of Norm

If the optimal solution to the overdetermined linear system is the one that makes the norm $\|r\|$ of the residual minimal, then we must decide which norm to use to measure r. Although norms are in a way equivalent, because they equal one another up to a scalar multiple, it makes quite a difference to minimize $\|r\|_1$, $\|r\|_2$ or $\|r\|_\infty$. Let us perform the following experiment.

Using a Gaussian random number generator with mean μ and standard deviation σ, we generate m numbers f_i. The approximation problem we consider is the computation of an estimate for μ from the datapoints f_i. Compare this with a real-life situation where the data f_i are collected by performing some measurements of a magnitude μ and σ represents in a way the accuracy of the measuring tool used to obtain the f_i. In the terminology of the previous subsection, we want to fit the f_i by a multiple of the basis function $b_1(x) = 1$ because we are looking for the constant μ. The overdetermined linear system takes the form

$$\lambda_1 \cdot 1 = f_i, \quad i = 1, \ldots, m.$$

It is clear that this linear system does not have an exact solution. The residual vector is definitely nonzero. We shall see that different criteria or norms can be used to express the closeness of the estimate λ_1 for μ to the datapoints f_i, or in other words the magnitude of the residual vector r with components $f_i - \lambda_1$, and that the standard deviation σ will also play a role.

If the Euclidean norm or ℓ_2-norm $\|r\|_2 = ({r_1}^2 + \ldots + {r_m}^2)^{1/2}$ is used, then the optimal estimate ${\lambda_1}^{(2)}$ is the mean of the m measurements f_i,

$$\lambda_1^{(2)} = \frac{1}{m}\sum_{i=1}^{m} f_i.$$

If we choose the ℓ_1-norm $\|r\|_1 = \sum_{i=1}^m |r_i|$ as a way to measure distances, then the value $\lambda_1^{(1)}$, which renders the ℓ_1-norm minimal is the median of the values f_i. When choosing as the distance function the ℓ_∞-norm $\|r\|_\infty = \max_{i=1,\ldots,m}|r_i|$, then the optimal solution $\lambda_1^{(\infty)}$ to the problem is given by

$$\lambda_1^{(\infty)} = \frac{1}{2}\left(\min_{i=1,\ldots,m} f_i + \max_{i=1,\ldots,m} f_i\right)$$

This can also be understood intuitively. The value for λ_1 that makes $\|r\|_\infty$ minimal is the one that makes the largest deviation minimal, so it should be right in the middle between the extremal values.

Let us consider an actual example. Take $\mu = 5$, $\sigma = 0.1$, and $m = 10$ with f_i shown in Fig. 6. Then

$$\lambda_1^{(2)} = 5.00299, \ \lambda_1^{(1)} = 5.04832, \ \lambda_1^{(\infty)} = 4.97042.$$

The ℓ_2-norm solution is clearly the most adequate in this case, with the ℓ_∞-norm solution as first runner-up. When introducing a typo in the values f_i, the situation changes

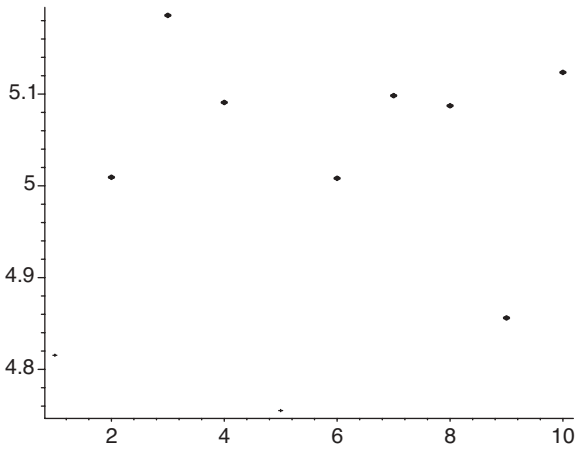

Figure 6. $\mu = 5$, $\sigma = 0.1$, $m = 10$.

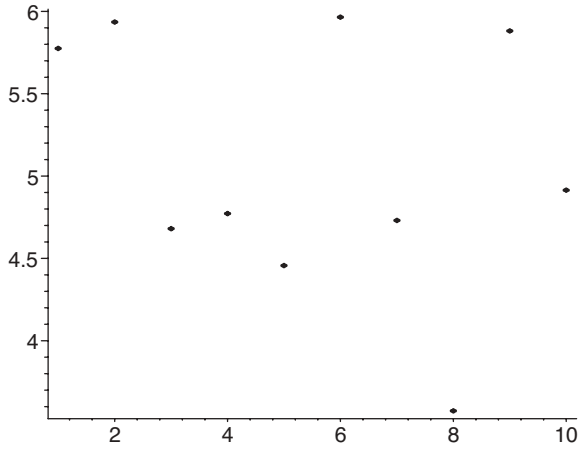

Figure 7. $\mu = 5$, $\sigma = 0.1$, $m = 10$.

completely. Change for instance the third measurement $f_3 = 5.185\ldots$ to $f_3 = 8.185\ldots$ to obtain

$$\lambda_1^{(2)} = 5.30299,\ \lambda_1^{(1)} = 5.04832,\ \lambda_1^{(\infty)} = 6.47042.$$

This has a dramatic effect on $\lambda_1^{(\infty)}$, a large effect on $\lambda_1^{(2)}$, but apparently the ℓ_1-technique is much more stable and can better cope with outliers (the median here does not change). This conclusion does not only hold for our little test problem, but it is valid in general. If one suspects outliers in the dataset, then the ℓ_1-norm is a more appropriate choice as a distance function than the ℓ_2-norm. The latter is the general recommendation in the case of normally distributed noise on the measurements.

When increasing the standard deviation σ from 0.1 to 0.7, which in terms of measurements means that the measuring tool is less accurate, then another important conclusion can be drawn. Using the dataset given in Fig. 7, we find

$$\lambda_1^{(2)} = 5.0685,\ \lambda_1^{(1)} = 4.84329,\ \lambda_1^{(\infty)} = 4.76951.$$

The ℓ_2-norm is still the best choice, but the ℓ_∞-norm has lost its second place to another distance function. The following conclusion again holds in general. The ℓ_∞-norm criterion performs well only in the context of accurate data suffering relatively small input (round-off) errors. When outliers or additional errors (such as from manual data input) are suspected, use of the ℓ_1-norm is recommended. If the measurement errors are believed to be normally distributed with mean zero, then the ℓ_2-norm is the usual choice. And therefore approximation problems of this type are called least-squares problems. When using the ℓ_2-norm as a criterion, the overdetermined linear system (9) translates to the system of normal equations (10). The usual way to solve Eq. (9) though is not via Eq. (10) but using techniques to rewrite the matrix A in a more suitable form.

Orthogonal Basis Functions

In the same way that we prefer to draw a graph using an orthogonal set of axes (the smaller the angle between the axes, the more difficult it becomes to make a clear drawing), it is preferred to use a so-called orthogonal set of basis functions $b_j(x)$ in Eq. (8). The use of orthogonal basis functions $b_j(x)$ can tremendously improve the conditioning of the problem (9).

The notion of orthogonality in a function space parallels that of orthogonality in the vector space $\mathbb{R}^k$; two elements of the space are called orthogonal if their inner product $\langle \cdot, \cdot \rangle$ equals zero. For two vectors $A = (a_1,\ldots,a_k)$ and $B = (b_1,\ldots,b_k)$, this translates to

$$(a_1,\ldots,a_k) \perp (b_1,\ldots,b_k) \Leftrightarrow \langle A,B\rangle = \sum_{j=1}^{k} a_j b_j = 0$$

The continuous analogon, where functions replace vectors and integrals replace sums, is given by

$$f(x) \perp g(x),\ x \in [a,b] \Leftrightarrow \langle f,g\rangle = \int_a^b f(x)g(x)dx = 0$$

A more advanced definition of w-orthogonality in addition uses discrete weights $W = (w_1,\ldots,w_k)$ or a weight function $w(x)$ defined on the interval $[a, b]$:

$$(a_1,\ldots,a_k) \perp w(b_1,\ldots,b_k) \Leftrightarrow \sum_{j=1}^{k} w_j a_j b_j = 0$$

$$f(x) \perp_{w(x)} g(x),\ x \in [a,b] \Leftrightarrow \langle f,g\rangle_w = \int_a^b f(x)g(x)w(x)dx = 0.$$

The function $w(x)$ can assign a larger weight to certain parts of the interval $[a, b]$. For instance, the function $w(x) = 1/\sqrt{1-x^2}$ on $[-1,1]$ assigns more weight toward the endpoints of the interval. For $w(x) = 1$ and $[a, b] = [-1,1]$, a sequence of orthogonal polynomials $L_i(x)$ satisfying

$$\int_{-1}^{1} L_j(x)L_k(x)dx = 0, \qquad j \neq k$$

is given by

$$\begin{aligned} L_0(x) &= 1 \\ L_1(x) &= x \\ L_{i+1}(x) &= \frac{2i+1}{i+1}xL_i(x) - \frac{i}{i+1}L_{i-1}(x), \qquad i \geq 1. \end{aligned}$$

The polynomials $L_i(x)$ are called the Legendre polynomials. For $w(x) = 1/\sqrt{1-x^2}$ and $[a,b] = [-1,1]$, a sequence of orthogonal polynomials $T_i(x)$ satisfying

$$\int_{-1}^{1} T_j(x)T_k(x)\frac{1}{\sqrt{1-x^2}}dx = 0, \quad j \neq k$$

is given by

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_{i+1}(x) &= 2xT_i(x) - T_{i-1}(x), \quad i \geq 1. \end{aligned}$$

The polynomials $T_i(x)$ are called the Chebyshev polynomials. They are also very useful in continuous (versus discrete) least-squares problems, as discussed below.

When the polynomials are to be used on an interval $[a, b]$ different from $[-1, 1]$, then the simple change of variable

$$x \to \left(x - \frac{a+b}{2}\right) \Big/ \frac{b-a}{2}$$

transforms the interval $[a, b]$ into the interval $[-1, 1]$, on which the orthogonal polynomials are defined.

Chebyshev Series

Let us choose the basis functions $b_j(x) = T_j(x)$ and look for the coefficients λ_j that make the ℓ_2-norm of

$$f(x) - \sum_{j=0}^{n} \lambda_j T_j(x), \quad -1 \le x \le 1$$

minimal. This is a continuous least-squares problem because the norm of a function is minimized instead of the norm of a finite-dimensional vector. Since

$$\begin{aligned} \left\| f - \sum_{j=0}^{n} \lambda_j T_j \right\|_2^2 &= \left\langle f - \sum_{j=0}^{n} \lambda_j T_j,\ f - \sum_{j=0}^{n} \lambda_j T_j \right\rangle \\ &= \|f\|_2^2 - \sum_{j=0}^{n} \langle f, T_j \rangle^2 + \sum_{j=0}^{n} (\langle f, T_j \rangle - \lambda_j)^2 \end{aligned} \tag{11}$$

in which only the last sum of squares depends on λ_j, the minimal is attained for the so-called Chebyshev coefficients $\lambda_j = \langle f, T_j \rangle$. Apparently the partial sum of degree n of the Chebyshev series development of a function,

$$f(x) = \sum_{j=0}^{\infty} \langle f, T_j \rangle T_j(x)$$

is the best polynomial approximation of degree n to $f(x)$ in the ℓ_2-sense. Since

$$\left| f(x) - \sum_{j=0}^{n} \langle f, T_j \rangle T_j(x) \right| \le \sum_{j=n+1}^{\infty} |\langle f, T_j \rangle|$$

this error can be made arbitrarily small when the series of Chebyshev coefficients converges absolutely. It suffices to choose n sufficiently large.

Minimax Approximation

Instead of minimizing the ℓ_2-distance (11) between a function $f(x)$ and a polynomial model for $f(x)$, we can also consider the problem of minimizing the ℓ_∞-distance (12). Every continuous function $f(x)$ defined on a closed interval $[a, b]$ has a unique so-called minimax polynomial approximant of degree n. This means that a unique polynomial $p_n = p_n^*$ of degree at most n exists that minimizes $\|f - p_n\|_\infty$, which is given by

$$\|f - p_n\|_\infty = \max_{x \in [a,b]} \left| f(x) - \sum_{j=0}^{n} \lambda_j x^j \right|. \tag{12}$$

The minimum is attained and is not an infimum. It is computed using the Remes algorithm, which is based on its characterization being the typical alternation property of the function $(f - p_n^*)(x)$ when $p_n^*(x)$ equals the minimax approximation:

$$\begin{aligned} &\|f - p_n^*\|_\infty = \min_{p_n \in \mathbb{C}[x]} \|f - p_n\|_\infty \Rightarrow \exists y_0 > y_1 > \ldots > y_{n+1} \in [a, b]: \\ &(f - p_n^*)(y_i) = (-1)^i \|f - p_n^*\|_\infty \text{ or } (-1)^{i+1} \|f - p_n^*\|_\infty, \\ &i = 0, \ldots, n+1. \end{aligned}$$

Here $\mathbb{C}[x]$ denotes the vector space of polynomials with complex coefficients in the variable x. The Remes algorithm is an iterative procedure, and the polynomial $p_n^*(x)$ is only obtained as the limit.

Fourier Series

Let us return to a discrete approximation problem. Our interest is now in data exhibiting some periodic behavior, such as the description of rotation-invariant geometric figures or the sampling of a sound waveform. A suitable set of orthogonal basis functions is the set

$$1, \cos(t), \cos(2t), \ldots, \cos(nt), \sin(t), \sin(2t), \ldots, \sin(nt) \tag{13}$$

as long as the datapoints t_i with $i = 1, \ldots, m$ are evenly spaced on an interval of length 2π, because then for any two basis functions $b_j(t)$ and $b_k(t)$ from Eq. (13), we have

$$\sum_{i=1}^{m} b_j(t_i) b_k(t_i) = 0, \quad j \ne k.$$

For simplicity, we assume that the real data $f_1, \ldots, f_m$ are given on $[0, 2\pi]$ at

$$t_1 = 0, t_2 = \frac{2\pi}{m}, t_3 = \frac{4\pi}{m}, \ldots, t_m = \frac{2(m-1)\pi}{m}.$$

Let $m \ge 2k + 1$ and consider the approximation

$$\frac{\lambda_0}{2} + \sum_{j=1}^{k} \lambda_{2j} \cos(jt) + \sum_{j=1}^{k} \lambda_{2j-1} \sin(jt).$$

The values

$$\lambda_{2j} = \frac{2}{m} \sum_{i=1}^{m} f_i \cos(jt_i), \quad j = 0, \ldots, k$$

$$\lambda_{2j-1} = \frac{2}{m} \sum_{i=1}^{m} f_i \sin(jt_i), \quad j = 1, \ldots, k$$

minimize the ℓ_2-norm

$$\sum_{i=1}^{m}\left(\frac{\lambda_0}{2}+\sum_{j=1}^{k}\lambda_2\cos(jt_i)+\sum_{j=1}^{k}\lambda_{2j-1}\sin(jt_i)-f_i\right)^2.$$

If we form for $j=1,\ldots,k$ a single complex quantity $\Lambda_j=\lambda_{2j}+i\lambda_{2j-1}$, these summations can be computed using a discrete Fourier transform that maps the data f_i at the points t_i to the Λ_j.

If the datapoints are evenly distributed on an interval $[a,\ b]$ instead of $[0,\ 2\pi]$, then the problem may be transformed to the interval $[0,\ 2\pi]$ by the linear transformation

$$t\rightarrow\frac{2\pi}{b-a}(t-a).$$

MULTIVARIATE LEAST-SQUARES PROBLEMS

We focus on linear discrete multivariate least-squares problems:

- Information on multivariate generalizations of orthogonal polynomials can be found in Refs. 8 and 9.
- A practical reference for discrete least-squares models using multivariate basis functions is Ref 10.

BIBLIOGRAPHY

1. M. Gasca, and T. Sauer, Polynomial interpolation in several variables, *Adv. in Comput. Math.*, **12** (4): 377–410, 2000.
2. T. Sauer and Y. Xu, On multivariate Lagrange interpolation-*Math. Comp.*, **64** (211): 1147–1170, 1995.
3. C. deBoor, On the Sauer-Xu formula for the error in multivariate polynomial interpolation, *Math. Comp.*, **65** (215): 1231–1234, 1996.
4. Ming-Jun Lai, Multivariate splines for data fitting and approximation, in M. Neamtu and L. Schumaker, (eds.), *Approximation Theory XII*, Brentwood NY: Nashboro Press, 2008.
5. M.-J. Lai, and L. Schumaker, *Spline functions on triangulations*, Cambridge, UK: Cambridge University Press, 2007.
6. A. Cuyt, How well can the concept of Padé approximant be generalized to the multivariate case? *J. Comput. Appl. Math.*, **105**: 25–50, 1999.
7. S. Becuwe, A. Cuyt, and B. Verdonk, Multivariate rational interpolation of scattered data, in I. Lirkov, S. Margenov, J. Wasniewski, and P. Yalamov, (eds.), *LNCS 2907*, New York: Springer, 2004.
8. I. Dumitriu, A. Edelman, and G. Shuman, MOPS: Multivariate orthogonal polynomials symbolically. Technical report, 2008.
9. A. Cuyt, B. Benouahmane, and B. Verdonk, Spherical orthogonal polynomials and symbolic-numeric Gaussian cubature formulas, in M.et al. Bubak, (ed.), *LNCS 3037*, Berlin, Germany: Springer-Verlag, 2004.
10. G. Fasshauser, *Meshfree Approximation Methods with Matlab*, Hackensack, NJ: World Scientific, 2007.

ANNIE CUYT
University of Antwerp,
Antwerpen, Belgium