Trigonometric interpolation

From Wikipedia, the free encyclopedia

In mathematics, **trigonometric interpolation** is interpolation with trigonometric polynomials. Interpolation is the process of finding a function which goes through some given data points. For trigonometric interpolation, this function has to be a trigonometric polynomial, that is, a sum of sines and cosines of given periods. This form is especially suited for interpolation of periodic functions.

An important special case is when the given data points are equally spaced, in which case the solution is given by the discrete Fourier transform.

Formulation of the interpolation problem

A trigonometric polynomial of degree K has the form

$$p(x) = a_0 + \sum_{k=1}^{K} a_k \cos(kx) + \sum_{k=1}^{K} b_k \sin(kx). \qquad (1)$$

This expression contains $2K + 1$ coefficients, $a_0, a_1, \ldots a_K, b_1, \ldots, b_K$, and we wish to compute those coefficients so that the function passes through N points:

$$p(x_n) = y_n, \quad n = 0, \ldots, N - 1.$$

Since the trigonometric polynomial is periodic with period 2π, the N points can be distributed and ordered in one period as

$$0 \leq x_0 < x_1 < x_2 < \ldots < x_{N-1} < 2\pi.$$

(Note that we do *not* in general require these points to be equally spaced.) The interpolation problem is now to find coefficients such that the trigonometric polynomial p satisfies the interpolation conditions.

Formulation in the complex plane

The problem becomes more natural if we formulate it in the complex plane. We can rewrite the formula for a trigonometric polynomial as $p(x) = \sum_{k=-K}^{K} c_k e^{ikx}$, where i is the imaginary unit. If we set $z = e^{ix}$, then this becomes

$$q(z) = \sum_{k=-K}^{K} c_k z^k,$$

with

$$q(e^{ix}) \triangleq p(x).$$

This reduces the problem of trigonometric interpolation to that of polynomial interpolation on the unit circle. Existence and uniqueness for trigonometric interpolation now follows immediately from the corresponding results for polynomial interpolation.

For more information on formulation of trigonometric interpolating polynomials in the complex plane see , p135 Interpolation using Fourier Polynomials (http://www.physics.arizona.edu/~restrepo/475A/Notes/sourcea.p df).

Solution of the problem

Under the above conditions, there exists a solution to the problem for *any* given set of data points $\{x_k, y_k\}$ as long as N, the number of data points, is not larger than the number of coefficients in the polynomial, i.e., $N \leq 2K+1$ (a solution may or may not exist if $N>2K+1$ depending upon the particular set of data points). Moreover, the interpolating polynomial is unique if and only if the number of adjustable coefficients is equal to the number of data points, i.e., $N = 2K + 1$. In the remainder of this article, we will assume this condition to hold true.

Odd number of points

If the number of points N is odd, say $N=2K+1$, applying the Lagrange formula for polynomial interpolation to the polynomial formulation in the complex plane yields that the solution can be written in the form

$$p(x) = \sum_{k=0}^{2K} y_k\, t_k(x), \tag{5}$$

where

$$t_k(x) = e^{-iKx+iKx_k} \prod_{m=0,m \neq k}^{2K} \frac{e^{ix} - e^{ix_m}}{e^{ix_k} - e^{ix_m}}.$$

The factor $e^{-iKx+iKx_k}$ in this formula compensates for the fact that the complex plane formulation contains also negative powers of e^{ix} and is therefore not a polynomial expression in e^{ix}. The correctness of this expression can easily be verified by observing that $t_k(x_k) = 1$ and that $t_k(x)$ is a linear combination of the right powers of e^{ix}. Upon using the identity

$$e^{iz_1} - e^{iz_2} = 2i \sin\left(\frac{z_1 - z_2}{2}\right) e^{i\frac{1}{2}z_1 + i\frac{1}{2}z_2}, \tag{2}$$

the coefficient $t_k(x)$ can be written in the form

$$t_k(x) = \prod_{m=0, m \neq k}^{2K} \frac{\sin \frac{1}{2}(x - x_m)}{\sin \frac{1}{2}(x_k - x_m)}. \tag{4}$$

Even number of points

If the number of points N is even, say $N=2K$, applying the Lagrange formula for polynomial interpolation to the polynomial formulation in the complex plane yields that the solution can be written in the form

$$p(x) = \sum_{k=0}^{2K-1} y_k\, t_k(x), \tag{6}$$

where

$$t_k(x) = e^{-iKx + iKx_k} \frac{e^{ix} - e^{i\alpha_k}}{e^{ix_k} - e^{i\alpha_k}} \prod_{m=0, m\neq k}^{2K-1} \frac{e^{ix} - e^{ix_m}}{e^{ix_k} - e^{ix_m}}. \tag{3}$$

Here, the constants α_k can be chosen freely. This is caused by the fact that the interpolating function (**1**) contains an odd number of unknown constants. A common choice is to require that the highest frequency is of the form a constant times $\cos(Kx)$, i.e. the $\sin(Kx)$ term vanishes, but in general the phase of the highest frequency can be chosen to be φ_K. To get an expression for α_k, we obtain by using (**2**) that (**3**) can we written in the form

$$t_k(x) = \frac{\cos\left(Kx - \frac{1}{2}\alpha_k + \frac{1}{2}\sum\limits_{m=0, m\neq k}^{2K-1} x_m\right) + \sum\limits_{m=-(K-1)}^{K-1} c_k e^{imx}}{2^N \sin(\frac{x_k - \alpha_k}{2}) \prod\limits_{m=0, m\neq k}^{2K-1} \sin(\frac{x_k - x_m}{2})}.$$

This yields

$$\alpha_k = \sum_{m=0, m\neq k}^{2K-1} x_m - 2\varphi_K$$

and

$$t_k(x) = \frac{\sin \frac{1}{2}(x - \alpha_k)}{\sin \frac{1}{2}(x_k - \alpha_k)} \prod_{m=0, m\neq k}^{2K-1} \frac{\sin \frac{1}{2}(x - x_m)}{\sin \frac{1}{2}(x_k - x_m)}.$$

Note that care must be taken in order to avoid infinities caused by zeros in the denominators.

Equidistant nodes

Further simplification of the problem is possible if nodes x_m are equidistant, i.e.

$$x_m = \frac{2\pi m}{N},$$

see Zygmund for more details.

Odd number of points

Further simplification by using (**4**) would be an obvious approach, but is obviously involved. A much simpler approach is to consider the Dirichlet kernel

$$D(x, N) = \frac{1}{N} + \frac{2}{N} \sum_{k=1}^{\frac{1}{2}(N-1)} \cos(kx) = \frac{\sin \frac{1}{2} N x}{N \sin \frac{1}{2} x},$$

where $N > 0$ is odd. It can easily be seen that $D(x, N)$ is a linear combination of the right powers of e^{ix} and satisfies

$$D(x_m, N) = \begin{cases} 0 \text{ for } m \neq 0 \\ 1 \text{ for } m = 0 \end{cases}.$$

Since these two properties uniquely define the coefficients $t_k(x)$ in (**5**), it follows that

$$t_k(x) = D(x - x_k, N) = \begin{cases} \dfrac{\sin \frac{1}{2} N(x - x_k)}{N \sin \frac{1}{2}(x - x_k)} \text{ for } x \neq x_k \\ \lim_{x \to 0} \dfrac{\sin \frac{1}{2} N x}{N \sin \frac{1}{2} x} = 1 \text{ for } x = x_k \end{cases}$$

$$= \frac{\operatorname{sinc} \frac{1}{2} N(x - x_k)}{\operatorname{sinc} \frac{1}{2}(x - x_k)}.$$

Here, the sinc-function prevents any singularities and is defined by

$$\operatorname{sinc} x = \frac{\sin x}{x}.$$

Even number of points

For N even, we define the Dirichlet kernel as

$$D(x, N) = \frac{1}{N} + \frac{1}{N} \cos \frac{1}{2} N x + \frac{2}{N} \sum_{k=1}^{\frac{1}{2} N - 1} \cos(kx) = \frac{\sin \frac{1}{2} N x}{N \tan \frac{1}{2} x}.$$

Again, it can easily be seen that $D(x, N)$ is a linear combination of the right powers of e^{ix}, does not contain the term $\sin \dfrac{1}{2} N x$ and satisfies

$$D(x_m, N) = \begin{cases} 0 \text{ for } m \neq 0 \\ 1 \text{ for } m = 0 \end{cases}.$$

Using these properties, it follows that the coefficients $t_k(x)$ in (**6**) are given by

$$t_k(x) = D(x - x_k, N) = \begin{cases} \dfrac{\sin \frac{1}{2} N(x-x_k)}{N \tan \frac{1}{2}(x-x_k)} \text{ for } x \neq x_k \\ \lim\limits_{x \to 0} \dfrac{\sin \frac{1}{2} Nx}{N \tan \frac{1}{2} x} = 1 \text{ for } x = x_k. \end{cases}$$

$$= \frac{\operatorname{sinc} \frac{1}{2} N(x - x_k)}{\operatorname{sinc} \frac{1}{2}(x - x_k)} \cos \frac{1}{2}(x - x_k)$$

Note that $t_k(x)$ does not contain the $\sin \frac{1}{2} Nx$ as well. Finally, note that the function $\sin \frac{1}{2} Nx$ vanishes at all the points x_m. Multiples of this term can, therefore, always be added, but it is commonly left out.

Implementation

A MATLAB implementation of the above can be found here (http://www.math.udel.edu/~braun/M428/Matlab/interpolation/triginterp.m) and is given by:

```matlab
function P = triginterp(xi,x,y)
% TRIGINTERP Trigonometric interpolation.
% Input:
%   xi  evaluation points for the interpolant (vector)
%   x   equispaced interpolation nodes (vector, length N)
%   y   interpolation values (vector, length N)
% Output:
%   P   values of the trigonometric interpolant (vector)
N = length(x);
% Adjust the spacing of the given independent variable.
h = 2/N;
scale = (x(2)-x(1)) / h;
x = x/scale;  xi = xi/scale;
% Evaluate interpolant.
P = zeros(size(xi));
for k = 1:N
  P = P + y(k)*trigcardinal(xi-x(k),N);
end

function tau = trigcardinal(x,N)
ws = warning('off','MATLAB:divideByZero' );
% Form is different for even and odd N.
if rem(N,2)==1    % odd
  tau = sin(N*pi*x/2) ./ (N*sin(pi*x/2));
else              % even
  tau = sin(N*pi*x/2) ./ (N*tan(pi*x/2));
end
warning(ws)
tau(x==0) = 1;       % fix value at x=0
```

Relation with the discrete Fourier transform

The special case in which the points x_n are equally spaced is especially important. In this case, we have

$$x_n = 2\pi \frac{n}{N}, \qquad 0 \leq n < N.$$

The transformation that maps the data points y_n to the coefficients a_k, b_k is obtained from the discrete Fourier transform (DFT) of order N.

$$Y_k = \sum_{n=0}^{N-1} y_n \, e^{-i2\pi \frac{nk}{N}}$$

$$y_n = p(x_n) = \frac{1}{N} \sum_{k=0}^{N-1} Y_k \, e^{i2\pi \frac{nk}{N}}$$

(Because of the way the problem was formulated above, we have restricted ourselves to odd numbers of points. This is not strictly necessary; for even numbers of points, one includes another cosine term corresponding to the Nyquist frequency.)

The case of the cosine-only interpolation for equally spaced points, corresponding to a trigonometric interpolation when the points have even symmetry, was treated by Alexis Clairaut in 1754. In this case the solution is equivalent to a discrete cosine transform. The sine-only expansion for equally spaced points, corresponding to odd symmetry, was solved by Joseph Louis Lagrange in 1762, for which the solution is a discrete sine transform. The full cosine and sine interpolating polynomial, which gives rise to the DFT, was solved by Carl Friedrich Gauss in unpublished work around 1805, at which point he also derived a fast Fourier transform algorithm to evaluate it rapidly. Clairaut, Lagrange, and Gauss were all concerned with studying the problem of inferring the orbit of planets, asteroids, etc., from a finite set of observation points; since the orbits are periodic, a trigonometric interpolation was a natural choice. See also Heideman *et al.* (1984).

Applications in numerical computing

Chebfun, a fully integrated software system written in MATLAB for computing with functions, uses trigonometric interpolation and Fourier expansions for computing with periodic functions. Many algorithms related to trigonometric interpolation are readily available in Chebfun; several examples are available here (http://www.chebfun.org/docs/guide/guide11.html).

References

- Kendall E. Atkinson, *An Introduction to Numerical Analysis* (2nd edition), Section 3.8. John Wiley & Sons, New York, 1988. ISBN 0-471-50023-2.
- M. T. Heideman, D. H. Johnson, and C. S. Burrus, "Gauss and the history of the fast Fourier transform (http://nctuocw.sjtu.edu.cn/new/course/fourier/supplement/heideman-johnson-etal1985.pdf)," *IEEE ASSP Magazine* **1** (4), 14–21 (1984).
- G.B. Wright, M. Javed, H. Montanelli, and L.N. Trefethen, "Extension of Chebfun to periodic functions (http://eprints.maths.ox.ac.uk/1876/1/NA_15_02.pdf)," *SIAM. J. Sci. Comput.*, 37 (2015), C554-C573
- A. Zygmund, *Trigonometric Series Volume II*, Chapter X, Cambridge University Press, 1988.

External links

- www.chebfun.org (http://www.chebfun.org)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Trigonometric_interpolation&oldid=788685026"