



WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
 Add links

[Create account](#) [Log in](#)

Article [Talk](#)

[Read](#) [Edit](#) [View history](#)

Tricubic interpolation

From Wikipedia, the free encyclopedia

In the [mathematical](#) subfield [numerical analysis](#), **tricubic interpolation** is a method for obtaining values at arbitrary points in [3D space](#) of a function defined on a [regular grid](#). The approach involves approximating the function locally by an expression of the form

$$f(x,y,z)=\sum_{i=0}^3\sum_{j=0}^3\sum_{k=0}^3a_{ijk}x^iy^jz^k.$$

This form has 64 coefficients *a_{ijk}*; requiring the function to have a given value or given [directional derivative](#) at a point places one linear constraint on the 64 coefficients.

The term *tricubic interpolation* is used in more than one context; some experiments measure both the value of a function and its spatial derivatives, and it is desirable to interpolate preserving the values and the measured derivatives at the grid points. Those provide 32 constraints on the coefficients, and another 32 constraints can be provided by requiring smoothness of higher derivatives.^[1]

In other contexts, we can obtain the 64 coefficients by considering a 3x3x3 grid of small cubes surrounding the cube inside which we evaluate the function, and fitting the function at the 64 points on the corners of this grid.

The [cubic interpolation](#) article indicates that the method is equivalent to a sequential application of one-dimensional cubic interpolators. Let CINT_{*x*}(*a*_{−1}, *a*₀, *a*₁, *a*₂) be the value of a monovariate cubic polynomial (e.g. constrained by values, *a*_{−1}, *a*₀, *a*₁, *a*₂ from consecutive grid points) evaluated at *x*. In many useful cases, these cubic polynomials have the form

CINT_{*x*}(*u*_{−1}, *u*₀, *u*₁, *u*₂) = **v**_{*x*} · (*u*_{−1}, *u*₀, *u*₁, *u*₂) for some vector **v**_{*x*} which is a function of *x* alone. The tricubic interpolator is equivalent to:

$$s(i,j,k) = \text{The value at grid point } (i,j,k)$$
$$t(i,j,z) = \text{CINT}_z(s(i,j,-1), s(i,j,0), s(i,j,1), s(i,j,2))$$
$$u(i,y,z) = \text{CINT}_y(t(i,-1,z), t(i,0,z), t(i,1,z), t(i,2,z))$$
$$f(x,y,z) = \text{CINT}_x(u(-1,y,z), u(0,y,z), u(1,y,z), u(2,y,z))$$
$$i,j,k \in \{-1,0,1,2\} \text{ and } x,y,z \in [0,1].$$

where

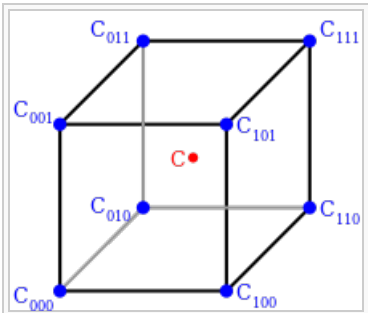
At first glance, it might seem more convenient to use the 21 calls to CINT described above instead of the 64 × 64 matrix described in Lekien and Marsden.^[2] However, a proper implementation using a sparse format for the matrix (that is fairly sparse) makes the latter more efficient. This aspect is even much more pronounced when interpolation is needed at several locations inside the same cube. In this case, the 64 × 64 matrix is used once to compute the interpolation coefficients for the entire cube. The coefficients are then stored and used for interpolation at any location inside the cube. In comparison, sequential use of one-dimensional integrators CINT_{*x*} performs extremely poorly for repeated interpolations because each computational step must be repeated for each new location.

See also [\[edit\]](#)

- Cubic interpolation
- Bicubic interpolation
- Trilinear interpolation

References [\[edit\]](#)

- ↑ Tricubic Interpolation in Three Dimensions (2005), by F. Lekien, J. Marsden, Journal of Numerical Methods and Engineering [↗](#)
- ↑ Tricubic Interpolation in Three Dimensions (2005), by F. Lekien, J. Marsden, Journal of Numerical Methods and Engineering [↗](#)



3D interpolation tries to assign a value at the red point *C* given values at the blue corner points.

External links [[edit](#)]

- [Java/C++ implementation of tricubic interpolation](#)
- [C++ implementation of tricubic interpolation](#)
- [Python implementation](#)

Categories: [Multivariate interpolation](#)

This page was last modified on 20 February 2014, at 04:06.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Mobile view](#)

