Re: Numerical integration at arbitrary x

Source: http://sci.tech−archive.net/Archive/sci.math.num−analysis/2004−12/0242.html

---

From: Fred Krogh (*fkrogh_at_mathalacarte.com*)
Date: 12/06/04

```
Date: Mon, 06 Dec 2004 14:34:56 -0800
```

Anonymous wrote:
> Peter Spellucci wrote:
>
>> In article <bl0sd.137237$5K2.133198@attbi_s03>,
>> Anonymous <nospam@noISP.com> writes:
>> >I'm faced with integrating a function numerically, given its value
>> at non−equally−spaced >points. That is,
>> >
>> >INTEGRAL(y(x) dx) between limits x=a and x=b, when I know only
>> y_i(x_i), i=1,...,N. The >x_i are approximately, but not exactly,
>> equally spaced between a and b.
>
>
>> A
>> the function values are exact to full precision?
>> 1)then you could use a (composed) gauss rule and obtain the function
>> values at the
>> unknown positions using a high order local interpolation. this gives
>> the error of the gauss−formula plus the integral of the interpolation
>> error, which
>> should be reasonably small. 2) you also can use local (piecewise) high
>> order polynomial interpolation (this does not require equidistant
>> nodes), e.g. a 9 point newton cotes formula and integrate this one
>> (using an appropriate gauss rule applied
>> to the polynomial)
>> B
>> the function values are a little bit noisy.?
>>
>> then interpolate by a cbic spline and
>> integrate this one. does also nt require equidistance. you use
>> simpsons rule
>> with the grid halved and applied tio the spline in order to get the
>> spline exaclty integrated.
>>
>> C
>> the function values have considerable noise?:
>> smooth the data by a smoothing spline and integrate this one (see
>> before).

>> you also could use a nonequidistant fft for smoothing , apply a window
>> then
>> transform back and integrate.
>
>
> These methods all seem to depend on local fitting of y_i with a simple
> curve for which either we know the integral analytically, or can
> calculate it numerically using one of the methods for equidistant points.
>
> A naive approach is to use a trapezoidal rule on the original points
> (which, I neglected to mention, are closely spaced), but this is not
> very accurate (about 1%). So I'm hoping there is an appropriate
> refinement of this naive technique that is more accurate.
>
You might want to try using Modified (Scaled) divided differences. See

@Incollection{Krogh:1974:CSI,
  author = "Fred T. Krogh",
   title = "Changing Stepsize in the Integration of Differential
    Equations Using Modified Divided Differences",
   editor = "Dale G. Bettis",
   booktitle = "Proceedings of the Conference on the Numerical Solution
             of Ordinary Differential Equations",
   number = 362,
   series = Lect−Notes−Math,
   type = "Part",
   pages = "22−−71",
   publisher = "Springer Verlag",
   address = "Berlin",
   year ="1974",
  bibdate = "Thurs Jan 23 1997",
}

or

@Book{Shampine:1975:CSO,
  Author = "L. F. Shampine and M. K. Gordon",
  Title = "Computer Solution of Ordinary Differential Equations",
  publisher = "W. H. Freeman",
  address = "San Francisco",
  pages = "318",
  year = "1975",
}

Both of these are for ODE's, but if you follow the derivation for the part of the algorithm for interpolation, it is not difficult to generalize for quadrature. An advantage is that by examination of the differences you can select an order that works well and get an error estimate. If your data is noisy it should show up in the error estimate. Note than you may need to break up the interval into subintervals, depending on the integration order you end up selecting.

Regards,
Fred