Re: Fitting One Curve Between 2 Others

Source: http://sci.tech−archive.net/Archive/sci.math.num−analysis/2006−12/msg00178.html

---

- *From*: spellucci@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx (Peter Spellucci)
- *Date*: Fri, 15 Dec 2006 13:00:36 +0000 (UTC)

In article <1166137856.280182.244850@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>,
"Bret Cahill" <BretCahill@xxxxxxx> writes:

> One sinusoidal curve is a distance h above another. I need to fit a third discontinuous curve consisting of straight line segments between the first two curves that will minimize the difference in slopes between the line segments and the first two curves at the contact points.
>
> The amplitudes can be anything less than 3h and the frequencies of the first two curves some integer ratio of each other up to 3. There are no phase angle restrictions.
>
> I need to get pointed in the right direction.
>
> Bret Cahill

your question makes little sense if you don't restrict the number
of line segments.
(otherwise take the piecewise linear continuous interpolant of the upper
curve where it is concave and the piecewise linear continuous of the lower
function there this is convex, then you can make the error in the slopes
arbitrarily small, if the pieces are sufficiently short, without violating
the side conditions)

also the fact that these are sinusoidal functions
does little help with phase angles can be different, since you
know nothing then about the positions of the maxima and minima.

with amplitudes of
height 3h and a curve distance of h (i guess you mean

f1(x)−f2(x)>= h for all x ??

) the two curves could even overlap, hence also this must be better specified.
in principle you have a semiinfinite optimization problem, if you restrict
the number of line segments. if the positions of the links is open,
this will be a very hard problem, but solvable with some powerful code.

with a finite number of segments given, say N, you can position the
links (resp. jumps) at posiitons xi(i) i=1,....,N−1 .
you characterize each segment as
a(i)+b(i)*(x−xi(i)) for xi(i)<= x <= xi(i+1)
with slope b(i) and write down the error criterion immediately.
with some calculus you might be able to express directly the condition

f1(x)>= a(i)+b(i)*(x−xi(i)) >= f2(x) for xi(i) <= x <= xi(i+1)

as a finite set of conditions on a(i) and b(i), all depending on the positions
xi(i) and xi(i+1) of course.

then it remains to set up the error criterion:
minimize the maximum abs deviation of the slopes
the sum of abs deviations of the slopes
the sum of squares of deviations ...

depending on this you will get different (nonlinear) optimization problems
mabe smooth or unsmooth. dpending on these properties you then can decide
which of the many optimization methods you might use here
hth
peter
.