Re: Exact or Least−Squares Solution 5 Equations

Source: http://sci.tech−archive.net/Archive/sci.math.num−analysis/2007−10/msg00432.html

- *From*: spellucci@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx (Peter Spellucci)
- *Date*: Fri, 26 Oct 2007 12:48:53 +0000 (UTC)

In article <1193323844.403041.220490@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>,
monir <monirg@xxxxxxxxxxxx> writes:

On Oct 25, 5:40 am, spellu...@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
(Peter Spellucci) wrote:

In article <1193280435.133367.29...@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>,
monir <mon...@xxxxxxxxxxxx> writes:

>Hello;
>
>I've tried a number of techniques to solve 5 equations in 5 unknowns
>with no avail!! I hope someone might have either come across a
>similar problem or has the appropriate analytical tools and would be
>kind enough to share his/her expertise.
>
>Here's a brief discription of the problem.
>
>5 UNKNOWNS: B, C, D, F, G
>
>SQRT { (x2−F)^2 + (y2−G)^2 } = { a + B.t2 + C.t22 + D.t23 } Exp(m.t2)
>SQRT { (x3−F)^2 + (y3−G)^2 } = { a + B.t3 + C.t32 + D.t33 } Exp(m.t3)
>SQRT { (x4−F)^2 + (y4−G)^2 } = { a + B.t4 + C.t42 + D.t43 } Exp(m.t4)
>SQRT { (x5−F)^2 + (y5−G)^2 } = { a + B.t5 + C.t52 + D.t53 } Exp(m.t5)
>SQRT { (x6−F)^2 + (y6−G)^2 } = { a + B.t6 + C.t62 + D.t63 } Exp(m.t6)
>
>All other quantities in the above equations are known.
>
>Q1: What are the exact expressions for B, C, D, F, G ??
>
>Q2: In the event that such exact expressions are defficult/impossible
>to derive (I certaily hope not!), then what are the least−squares
>regression formulas ??
>
>Your expert help would be greatly appreciated.
>Monir
>

this system looks a little bit strange:
on the right hand side we have a linear part in the three unknowns.
if you can pick here a submatrix of the 5 by 3 matrix formed by
[t2 t22 t23]
....
[t6 t62 t63]
which has rank 3 you could solve the corresponding 3 equations for B,C,D
in terms of the sqrt... involving the unknowns F and G, insert in the other
two equations, with only F,G remaining as unknowns and then solving these
using e.g. Newtons method.
if this is impossible, then the system might have no solution at all and you
need
to use nonlinear least squares minimizer to get a "quasisolution"
for this I would work in two stages:
first squaring both sides, then building the difference LHS−RHS, squaring
again
and summing up you get a sum which is quartic in the 5 unknowns,
miminimize
this by an appropriate code (e.g. ELSUNC) , then going back to the
orignal equation, building RHS−LHS squaring and summing up, doing the
same with the solution from the first step as an initial guess.
this because the least squares solution of the original system will not be
identical to that of the squared system (the optimal sum of squares will not be
zero in most cases) .
a quartic in 5 unknowns can also be quite hard to minimize, but the problem
involving the square roots might be even harder.
hth
peter

Peter;
Thank you for your prompt and helpful suggestions.
How about the following approach:
1. start with an initial "good" guess of F and G
2. the 5 equations then become:
t2.B + t22.C + t23.D = h2 ....(1)
t3.B + t32.C + t33.D = h3 ....(2)
t4.B + t42.C + t43.D = h4 ....(3)
−−−−−−−−−−−
t5.B + t52.C + t53.D = h5 ....(4)
t6.B + t62.C + t63.D = h6 ....(5)
3a. solve the top 3 simultaneous linear equations for B, C, D
{it would be more practical here to apply the general solution
formula, x(i) = ..., i=1,3
x(i) ,i=1,3 refers to B, C, D. plse see item 5 below) }
3b. substitute the values of B, C, D into equations (4) and (5) and
solve for G:
(x6 − x5) = sqrt {k62 − (y6 − G)^2} − sqrt {k52 − (y5 −
G)^2} ....(6)
(possibly by a goal seek scenario if I can't derive G = ...);
and

F = x5 − sqrt {k52 − (y5 − G)^2} ....(7)
3c. use the new values of F and G and repeat steps 2, 3a, 3b above
until (hopefully!) a
reasonable convergence is achieved.

4a. an alternative to step 3. above would be to obtain the least
squares solution of the 3
variables B, C, D based on the coefficients of the 5 simultaneous
linear equations.
{similarly, I would prefer here to use the applicable general
regression formula
x(i) = ..., i=1,3 (plse see item 5 below}
4b. establish a convergence criterion for the solution, and repeat 4a
above by modifying the
initial values of F and G. Repeat until a target value is
achieved.

5. Since I'm dealing with max 5 equations, I would appreciate if
someone can (with no
much effort) provide:
− the analytical solution formula x(i) = ..., i=1, N for N
simultaneous linear
equations; and / or
− the regression expression x(i) =......., i=1, 3 based on the
coeffs of N simultaneous
linear equations
(unfortunately, my equation solver can handle max 3 unknowns in
only 4 linear equations.
Will keep searching!)

Thank you again for your help.
Monir

what you imagine to do is in short this:
you have a system of n equations in n unknowns x
T_i(x) = 0 i=1,..,n
now , because of the special strucuture of the system, and bbeing hindered by
extremely poor software, you want to do the following:
split x into [y,z];
y=[F,G]; z=[B,C,D];
split also T into R,S (of the corresponding dimensions) and now iterate:

given [y^0,z^0]
solve
R(y^1,z^0)=0 for y^1 ;
solve
S(y^1,z^1)=0 for z^1 ;
repeat until convergence is achieved.

o.k. this is long known as kind of block−Newton−Gauss−Seidel.

whether this converges and how fast depends strongly on the properties
of the
Jacobian of _ T (!) _ (the complete one, not the two small ones)
There are sufficient convergence criteria, but these
are quite restrictive.
also, you want to do the solving (which in this specific case involves solution
of two linear systems of equatuons only) using explicit formulae for the inverse,
obviously in order to circumvent the weakness of your software.
for the 2 by two system the formula is
(in MATLAB notation: ; begins a new row ; [...] denotes a matrix or vector)
inverse([a b ; c d]) = (1/(a*d−b*c)) [d −b ; −c a]
and this might be used.
for a three by three system also there exists such a formula, the formula for
the inverse via the adjoint:

inverse(A) = (1/det(A)) *adjoint(A) ;
the adjoint(A) has in position (j,i) the element det(A'(i,j))(−1)^{i+j}
where A'(i,j) is obtained from A by deleting row i and column j.
in your case this means 9 determinants of 2 by 2 matrices
and the denominator in the first formula is the determinant of the 2 by 2
matrix
but I would strongly discourage the use of these formulae because the numerical
evaluation can be subject to severe roundoff error.
since also your approach as a whole is questionable in respect to applicability
as well as efficiency, why not simply using a hand crafted damped
Newtons method say, with a intial guess
obtained by say setting F,G to zero and solving three equations for B,C,D
and then the other two for G,H?
(... I assume you work on a small dedicated processor such that the use of
well known standard software from netlib is excluded?)

hth
peter

.