Paramter Estimation In IGW

By Huasheng Liao

The Goal:

The goal is to find the parameters, $\vec{a}$ ($\vec{a}$ is a vector) in a given model or multiple models such that the given models' values f($\vec{x}$, $\vec{a}$) fit best some data points

To achieve the goal:

Given a model with unknown parameters, $\vec{a}$ =(a_1,a_2) to be estimsted:

Model: $\gamma(\boldsymbol{a}_1,\boldsymbol{a}_2)=\dfrac{\boldsymbol{a}_1\boldsymbol{x}}{\boldsymbol{a}_2+\boldsymbol{x}}$ = f($\vec{x}$, $\vec{a}$) (or complex numerical models)

And observations: γ_i at x_i, i=1,2,…,m

i	1	2	3	4	5	6	7
x	0.038	0.194	0.425	0.626	1.253	2.500	3.740
γ	0.050	0.127	0.094	0.2122	0.2729	0.2665	0.3317

One can have m residual functions:

$\boldsymbol{r}_i(\boldsymbol{a}_1,\boldsymbol{a}_2)=\gamma_i-\gamma(\boldsymbol{a}_1,\boldsymbol{a}_2,\boldsymbol{x}_i)$, i=1,2,…,m

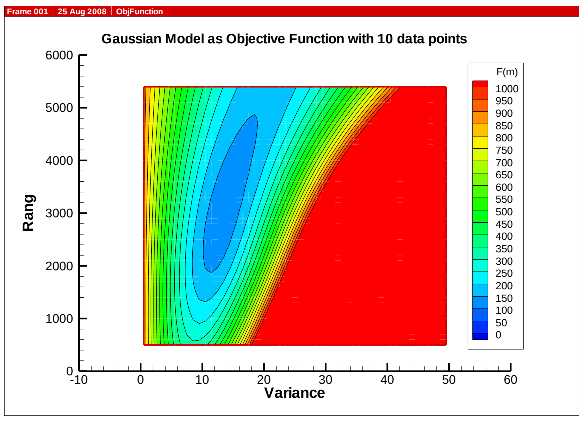
Note that, $\boldsymbol{r}_i(\boldsymbol{a}_1,\boldsymbol{a}_2)$ is function of unknown parameters, $\vec{a}$ =(a_1,a_2) to be estimsted.

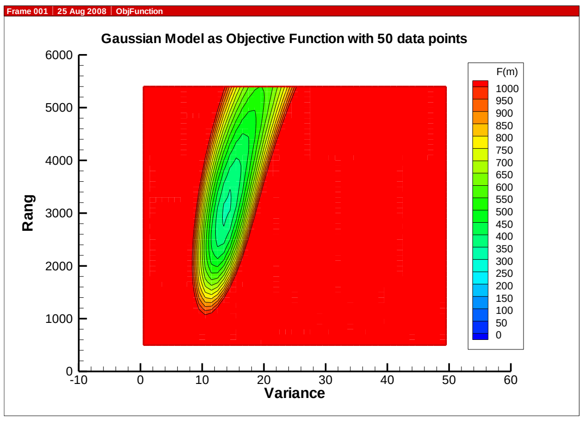
Best fitting means that those residuals at data points tend to minimum or zero: **Least Square Minimization (LSM) or Weighted Least Squares Minimization (WLSM)** is one of the good method to be used in parameters estimation.

LSM : the unknown values of the parameters in the models are estimated by finding the numerical values for the parameter estimates that minimize **the sum of the squared deviations** between the observed responses and the functional portion of the model.

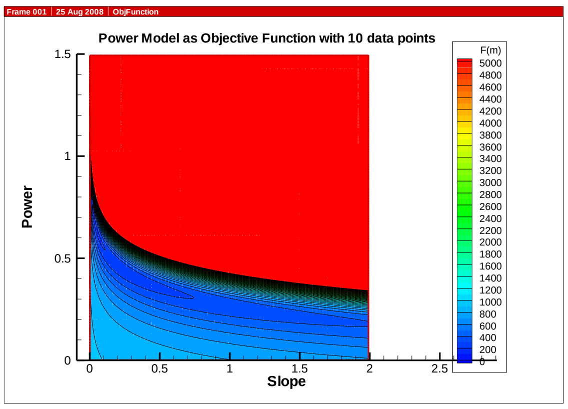
Sum of squares residuals: $\boldsymbol{S}=\sum_{i=1}^{m}\boldsymbol{r}_i^2(\boldsymbol{a}_1,\boldsymbol{a}_2)=\boldsymbol{S}(\boldsymbol{a}_1,\boldsymbol{a}_2)$

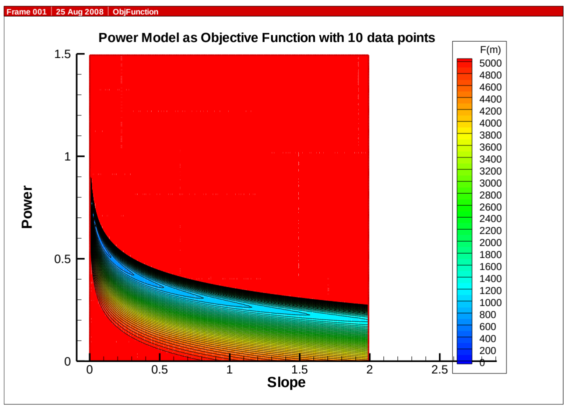
What does S look like?



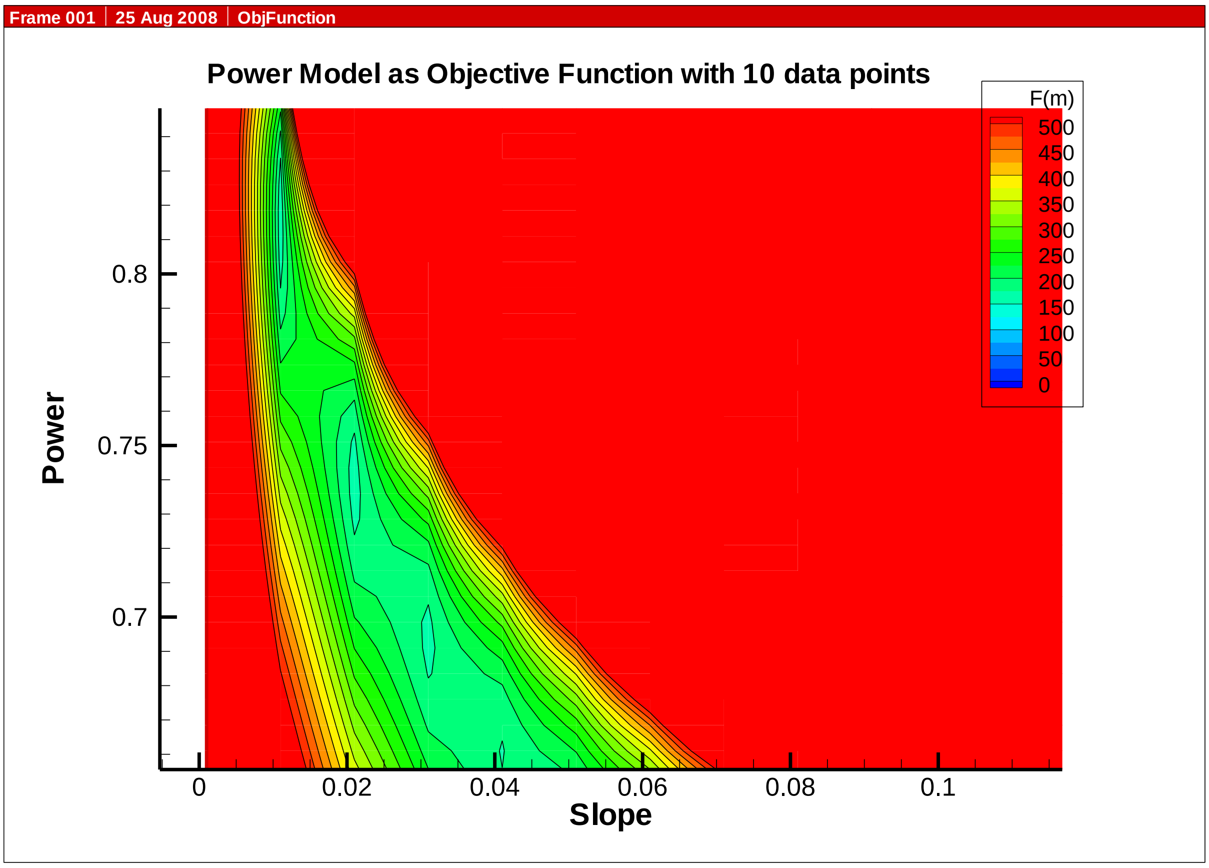


Gaussian model





Power Model



Local traps

Multiple minima

Multiple minima can occur in a variety of circumstances some of which are:

- A parameter is raised to a power of two or more. For example, when fitting data to a Lorentzian curve

$$f(x_i, a) = \frac{\alpha}{1 + (\frac{\gamma - xi}{a})^2}$$

where α is the height, γ is the position and a is the half-width at half height, there are two solutions for the half-width, and which give the same optimal value for the objective function.

- Two parameters can be interchanged without changing the value of the model. A simple example is when the model contains the product of two parameters, since αβ will give the same value as βα.
- A parameter is in a trigonometric function, such as sina , which has identical values at a'+2nπ. See Levenberg-Marquardt algorithm for an example.

Not all multiple minima have equal values of the objective function. False minima, also known as local minima, occur when the objective function value is greater than its value at the so-called global minimum. For example, the model

$$\boldsymbol{f}(\boldsymbol{x}_i,\boldsymbol{a}) = (1-3\boldsymbol{a}+\boldsymbol{a}^3)\boldsymbol{x}_i$$

has a local minimum at a= 1 and a global minimum at a= -3

To be certain that the minimum found is the global minimum, the refinement should be started with widely differing initial values of the parameters. When the same minimum is found regardless of starting point, it is likely to be the global minimum.

When multiple minima exist there is an important consequence: the objective function will have a maximum value somewhere between two minima. The normal equations matrix is not positive definite at a maximum in the objective function, as the gradient is zero and no unique direction of descent exists. Refinement from a point (a set of parameter values) close to a maximum will be ill-conditioned and should be avoided as a starting point. For example, when fitting a Lorentzian the normal equations matrix is not positive definite when the half-width of the band is zero.[7]

The goal turn out that we have solve a problem of finding a minimum of a function S:

$$\begin{cases} \min\ \ \mathrm{S}(\boldsymbol{a}_1,\boldsymbol{a}_2) = \sum_{\boldsymbol{i}=1}^{\boldsymbol{m}} \boldsymbol{r}_i^2(\boldsymbol{a}_1,\boldsymbol{a}_2) \\ \boldsymbol{s.j.to} \\ \boldsymbol{Constrains}\ 1 \\ \boldsymbol{constrains}\ 2 \\ \dots \\ \boldsymbol{constrain}\ \mathrm{n_c} \end{cases}$$

this is a nonlinear optimization problem and data-dependent

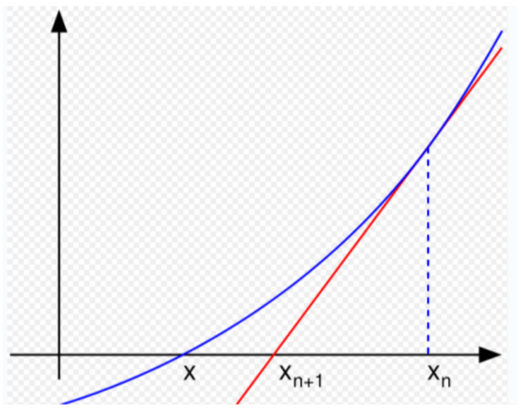
Method used in IGW

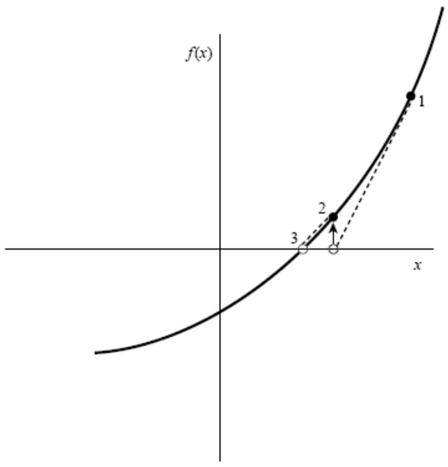
The **Gauss–Newton algorithm** is a method used to solve non-linear least squares problems. Unlike Newton's method has the advantage that second derivatives, which can be challenging to compute, are not required.

===

General Newton Method: Newton method , also know as the Newton-Raphson method, is perhaps the best know mehod for finding successively better approximations to the zeros (or roots) of a real-valued function. Newton’s method can often converge remarkably quickly, especially if the iteration begins “sufficiently near “ the desired root.

The idea of the method is as follows: one starts with an initail guess which is reasonably close to the true root, then the function is approximated by its tangent line, which can be computed using the tools of calculus, and one computes the x-intercept of this tangent line, which is easily done with elementary algebra. This x-intercept will typically be a better approximation to the function’s root than the original guess, and the method can be iterated.





An illustration of one iteration of Newton’s method (the function f is shown in blue and the tangent line is in red). We see that x_{n+1} is a better approximation than x_n for the root x of the function f.

Given a nonlinear function f(x) defined on the interval [a,b], is tangent line at point x0 can be written as

$$f_{linear}(x) = f(x_0) + \Delta x f'(x_0) \qquad (1)$$

Eq(1) can also be derived using Taylor seriers expansion by neglecting the second derivative terms and higher orders terms.

Finding the x-intercept of Eq(1) becomes to solve the following linear equation

$$f_{linear}(x) = f(x_0) + \Delta x f'(x_0) = 0 \qquad (2)$$

This gives

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \qquad (3)$$

Then we can derive the formula for a better approximation, x_2 , by referring to the diagram above:

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} \qquad (4)$$

The final iterative formula can be written as

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - f'(x_k)^{-1} f(x_k) \qquad (5)$$

Newton Method for Optimization

The necessary condition for the function $S = \sum_{i=1}^{m} r_i^2(a_1, a_2)$ to have an extremum is that the partial derivatives vanish i.e.

$$\sum_{i=1}^{m} r_i \frac{\partial r_i}{\partial a_j} = 0 \ \ (j = 1,2,\cdots,n) \qquad (6)$$

or, equivalently,

$$J^T(\vec{a})\vec{r}(\vec{a}) = 0 \qquad (1)$$

Where $J_{ij} = \frac{\partial r_i}{\partial a_j}$ is Jacobian matrix of residual vector $\vec{r} = (r_1, r_2, \cdots, r_m)$.

Equation (1) is usually a system of non-linear equations that numerically, can be solved by Netown-Rasphson's method. ➔ how ?

Denoting nonlinear system eq(6) as

$$\frac{\partial S}{\partial a_j} = F_j(\vec{a}) = 0 \quad (j = 1, 2, \cdots, n)$$

Where F$_j$ is nonlinear funcion of vector $\vec{a}$. Referring to eq(5),

$$F_j(\vec{a} + \Delta\vec{a}) = F_j(\vec{a}) + \sum_{i=1}^{n} \frac{\partial F_j}{\partial a_i} \Delta\vec{a} + O(\Delta\vec{a}^2) \quad (j = 1, 2, \cdots, n)$$

The matrix of partial derivatives appearing in above equation is the Jacobian matrix J:

$$J_{ij} = \frac{\partial F_i}{\partial a_j}$$

In matrix notation,

$$F(\vec{a}^{(k+1)}) = F(\vec{a}^{(k)}) + J_F((\vec{a}^{(k)})(\vec{a}^{(k+1)} - \vec{a}^{(k)}) + O(\Delta\vec{a}^2)$$

By neglecting terms of order $O(\Delta\vec{a}^2)$ and higher and by setting $F(\vec{a}^{(k+1)}) = 0$, one obtains the similar iterative formula (Rather than actually computing the inverse of this matrix, one can save time by solving the system of linear equations)

$$J_F((\vec{a}^{(k)})(\vec{a}^{(k+1)} - \vec{a}^{(k)}) = -F(\vec{a}^{(k)})$$

Recalling that $F_j(\vec{a}) = \frac{\partial S}{\partial a_j}$ and $J_{F,ij} = \frac{\partial F_i(\vec{a})}{\partial a_j} = \frac{\partial}{\partial a_j}(\frac{\partial S(\vec{a})}{\partial a_i}) = \frac{\partial^2 S(\vec{a})}{\partial a_i \partial a_j}$, from above equation, gives

$$H(\vec{a}^{(k+1)} - \vec{a}^{(k)}) = -\nabla S$$

Where $\vec{g} = \nabla S$ is gradient, H is Hessian matrix of S, or

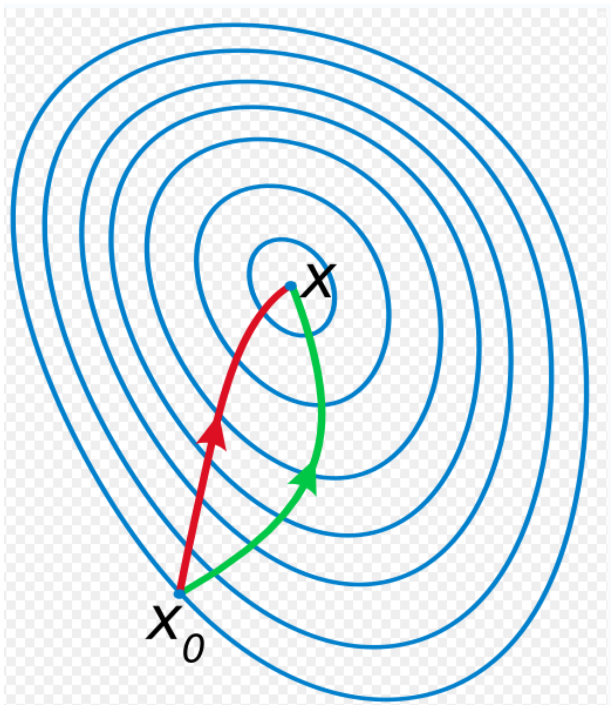
$$\vec{\boldsymbol{a}}^{(\boldsymbol{k}+1)} = \vec{\boldsymbol{a}}^{(\boldsymbol{k})} - \boldsymbol{H}^{-1}\boldsymbol{g}$$

This is the recurrence relation for Newton's method for minimizing a function $\boldsymbol{S}$ of parameters, where **g** denotes the gradient vector of *S* and **H** denotes the Hessian matrix of *S*.

As can be seen , Newton's method needs to calculate the Hessian, which is always an expensive operation, so it is very hard to apply to those cases with difficulties in the second derivatives calculations. For these cases an approximation for Hessian is used instead, which leads to various modified Newton methods such as Gauss-Newton, quasi-Newton and others.

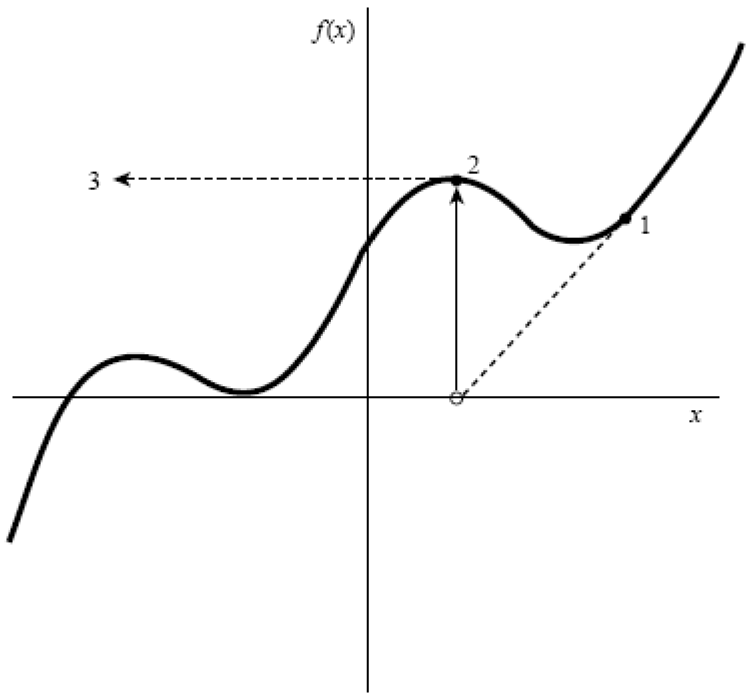
Note 1:

Newton's method is different from Gradient descent method: $\vec{\boldsymbol{a}}^{(\boldsymbol{k}+1)} = \vec{\boldsymbol{a}}^{(\boldsymbol{k})} - \alpha^{(\boldsymbol{k})}\boldsymbol{g}$, which is based on the observation that if the real-valued function S($\vec{\boldsymbol{a}}^{(\boldsymbol{k})}$) is defined and differentiable in a neighborhood of point $\vec{\boldsymbol{a}}^{(\boldsymbol{k})}$, then S decreases fastest if one goes from $\vec{\boldsymbol{a}}^{(\boldsymbol{k})}$ in the direction of the negative gradient of S at $\vec{\boldsymbol{a}}^{(\boldsymbol{k})}$, that is, $\boldsymbol{S}(\vec{\boldsymbol{a}}^{(\boldsymbol{k}+1)}) \le \boldsymbol{S}(\vec{\boldsymbol{a}}^{(\boldsymbol{k})}) \le \boldsymbol{S}(\vec{\boldsymbol{a}}^{(\boldsymbol{k}-1)}) \le \cdots$, where $\alpha^{(\boldsymbol{k})}$ >0, a small enough number.

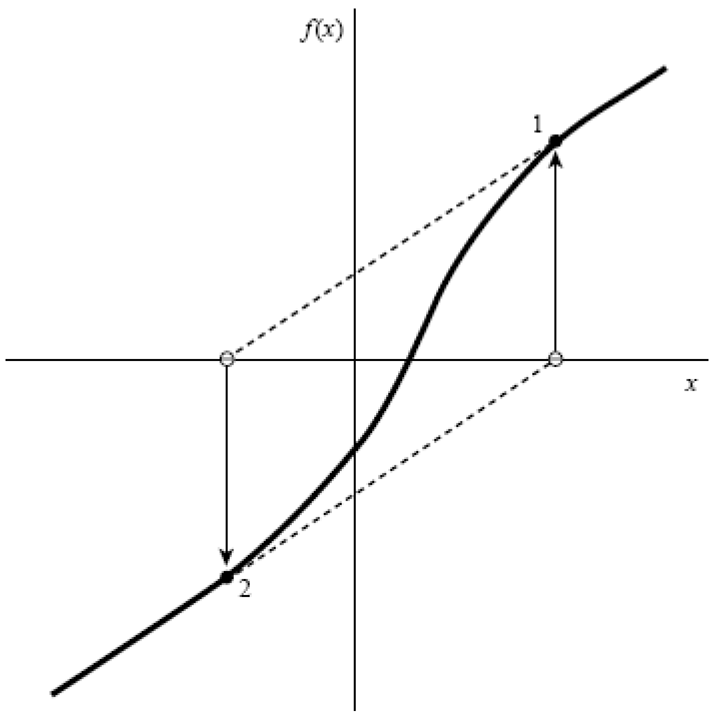


A comparison of gradient descent (green) and Newton’s mehod(red) for minimizing a function (with small step sizes). Newton’s method uses curvature information to take a more direct route.

Note 2:



Zero derivative



same derivative at different point

Gauss–Newton algorithm : Derivation course I

Since $\boldsymbol{S} = \sum_{i=1}^{m} \boldsymbol{r}_i^2(\boldsymbol{a}_1, \boldsymbol{a}_2)$, the gradient is given by

$$\boldsymbol{g}_j = 2\sum_{i=1}^{m} \boldsymbol{r}_i \frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{a}_j}$$

Elements of the Hessian are calculated by differentiating the gradient elements, g_j, with respect to $\vec{\boldsymbol{a}}$:

$$\boldsymbol{H}_{jk} = 2\sum_{i=1}^{m} \left(\frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{a}_j}\frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{a}_k} + \boldsymbol{r}_i \frac{\partial^2 \boldsymbol{r}_i}{\partial \boldsymbol{a}_j \partial \boldsymbol{a}_k}\right)$$

The Gauss–Newton method is obtained by ignoring the second-order derivative terms (the second term in this expression). That is, the Hessian is approximated by

$$\boldsymbol{H}_{jk} \approx 2\sum_{i=1}^{m} \frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{a}_j}\frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{a}_k} = 2\sum_{i=1}^{m} \boldsymbol{J}_{ij}\boldsymbol{J}_{ik}$$

The gradient and the approximate Hessian can be written in matrix notation as

$$\vec{\boldsymbol{g}} = 2\boldsymbol{J}^T\vec{\boldsymbol{r}}$$
$$\boldsymbol{H} \approx 2\boldsymbol{J}^T\boldsymbol{J}$$

🡸== at least J_{ij} need to be evaluated ➔ sensitivities Mtx

These expressions are substituted into the recurrence relation above (Eq(2) to obtain the operational equations, the resulting linear system with unknown vector, $\Delta\vec{\boldsymbol{a}}^{(\boldsymbol{k})}$

$$\boldsymbol{J}^T(\vec{\boldsymbol{a}}^{(\boldsymbol{k})})\boldsymbol{J}(\vec{\boldsymbol{a}}^{(\boldsymbol{k})})\Delta\vec{\boldsymbol{a}}^{(\boldsymbol{k})} = -\boldsymbol{J}(\vec{\boldsymbol{a}}^{(\boldsymbol{k})})\vec{\boldsymbol{r}}(\vec{\boldsymbol{a}}^{(\boldsymbol{k})})$$

Once unknown vector, $\Delta\vec{\boldsymbol{a}}^{(\boldsymbol{k})}$ has been solved, the following

$$\vec{\boldsymbol{a}}^{(\boldsymbol{k}+1)} = \vec{\boldsymbol{a}}^{(\boldsymbol{k})} - \alpha\Delta\vec{\boldsymbol{a}}^{(\boldsymbol{k})}$$

is taken as the new approximation of the optimum. The relaxation factor α is a parameter of the method.

Gauss–Newton algorithm : Derivation cousre II

Consider a set of m data points, (x1,y1),(x2,y2), …, (xm,ym), and a model function (a curve) y=f(x,a), that in addition to the variable x also dependes on n parameters, a=(a1,a2,…,an), with m>=n. It is desired to find the vector a of parameters such that the model (curve) fits best the given data in the least squares sense, that is, the sum of squares

$$\boldsymbol{S} = \sum_{\boldsymbol{i}=1}^{\boldsymbol{m}} \boldsymbol{r}_{\boldsymbol{i}}^2(\boldsymbol{a}_1, \boldsymbol{a}_2)$$

Is minimized, where the residuals (errors) ri are given by

$$\boldsymbol{r}_{\boldsymbol{i}}(\boldsymbol{a}_1, \boldsymbol{a}_2) = \boldsymbol{y}_{\boldsymbol{i}} - \boldsymbol{f}(\boldsymbol{a}, \boldsymbol{x}_{\boldsymbol{i}}), \quad \text{i=1,2,…,m}$$

The minimum valuesof S occurs when the gradient is zero. Since the model contains n parameters there are n gradient equations:

$$\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{a}_{\boldsymbol{j}}} = 2\sum_{\boldsymbol{i}=1}^{\boldsymbol{m}} \boldsymbol{r}_{\boldsymbol{i}} \frac{\partial \boldsymbol{r}_{\boldsymbol{i}}}{\partial \boldsymbol{a}_{\boldsymbol{j}}} = 0 \quad (\boldsymbol{j} = 1, 2, \cdots, \boldsymbol{n})$$

In a non-linear system, the derivatives $\frac{\partial \boldsymbol{r}_{\boldsymbol{i}}}{\partial \boldsymbol{a}_{\boldsymbol{j}}}$ are functions of both the independent variable

and the parameters, so these gradient equations do not have a closed solution. Instead, initial values must be chosen for the parameters. Then, the parameters are refined interatively, that is, the values are obtained by successive approximation,

$$\boldsymbol{a}_{\boldsymbol{j}} \approx \boldsymbol{a}_{\boldsymbol{j}}^{\boldsymbol{k}+1} = \boldsymbol{a}_{\boldsymbol{j}}^{\boldsymbol{k}} + \Delta \boldsymbol{a}_{\boldsymbol{j}}$$

Here, k is an iteration number and the vector of increments, $\Delta\boldsymbol{a}$ is known as the shift vector. At each iteration **the model is linearized** by approximation to a first-order Taylor series expansion about $\boldsymbol{a}_{\boldsymbol{j}}^{\boldsymbol{k}}$

$$\begin{aligned}\boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{a}) &= \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{a}^k) + \sum_{j}^{n} \frac{\partial \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{a}^k)}{\partial \boldsymbol{a}_j}(\boldsymbol{a}_j - \boldsymbol{a}_j^k) + \frac{1}{2}\sum_{j}^{n}\sum_{l}^{n} \frac{\partial^2 \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{a}^k)}{\partial \boldsymbol{a}_j \partial \boldsymbol{a}_l}(\boldsymbol{a}_j - \boldsymbol{a}_j^k)(\boldsymbol{a}_l - \boldsymbol{a}_l^k) + \cdots \\ &= \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{a}^k) + \sum_{j}^{n} \boldsymbol{J}_{ij}\Delta \boldsymbol{a}_j + \frac{1}{2}\sum_{j}^{n}\sum_{l}^{n} \boldsymbol{H}_{jl(i)}\Delta \boldsymbol{a}_j \Delta \boldsymbol{a}_l + \cdots\end{aligned}$$

Or approximately

$$f(x_i,a) \approx f(x_i,a^k)+\sum_j^n J_{ij}\Delta a_j$$

The jacobian, J is a function of constants, the independent variables and the parameters, so it changes from iteration to the next. Thus, **in term of the linearized model**, $\frac{\partial r_i}{\partial a_j}=J_{ij}$ and the residuals are given by

$$r_i = y_i - f(a,x_i) = y_i - f(x_i,a^k) - \sum_j^n J_{ij}\Delta a$$

Substituting these expressions into the gradient equations, they become

$$-2\sum_{i=1}^m r_i \frac{\partial r_i}{\partial a_j} = -2\sum_{i=1}^m \frac{\partial r_i}{\partial a_j}[y_i - f(x_i,a^k) - \sum_l^n J_{il}\Delta a] = 0 \ \ (j=1,2,\cdots,n)$$

Or

$$-2\sum_{i=1}^m J_{ij}[y_i - f(x_i,a^k) - \sum_l^n J_{il}\Delta a] = 0 \ \ (j=1,2,\cdots,n)$$

Which, on rearrangement, become n simultaneous linear equations, the normal equations

$$\sum_{i=1}^m\sum_l^n J_{ij}J_{il}\Delta a_l = \sum_{i=1}^m J_{ij}[y_i - f(x_i,a^k) \ \ (j=1,2,\cdots,n)$$

The normal equations are written in matrix notation as

$$J^T(\vec{a}^{(k)})J(\vec{a}^{(k)})\Delta\vec{a}^{(k)} = J^T(\vec{a}^{(k)})\vec{r}(\vec{a}^{(k)})$$

When the observations are not equally reliable, a weighted sum of squares may be minimized,

$$S = \sum_{i=1}^m W_{ii} r_i^2(a_1,a_2)$$

Each element of the diagonal weight matrix, W should , ideally, be equal to the reciprocal of the variance of the measurement. This implies that the observations are uncorrelated. If the observations are correlated, the expression

$$S = \sum_{i=1}^m\sum_j^m r_i W_{ij} r_j$$

Applies. In this case the weight matrix should ideally be equal to the inverse if the variance-covariance matrix of the observations. The normal equations are then

$$J^T(\vec{a}^{(k)})WJ(\vec{a}^{(k)})\Delta\vec{a}^{(k)} = J^T(\vec{a}^{(k)})W\vec{r}(\vec{a}^{(k)})$$

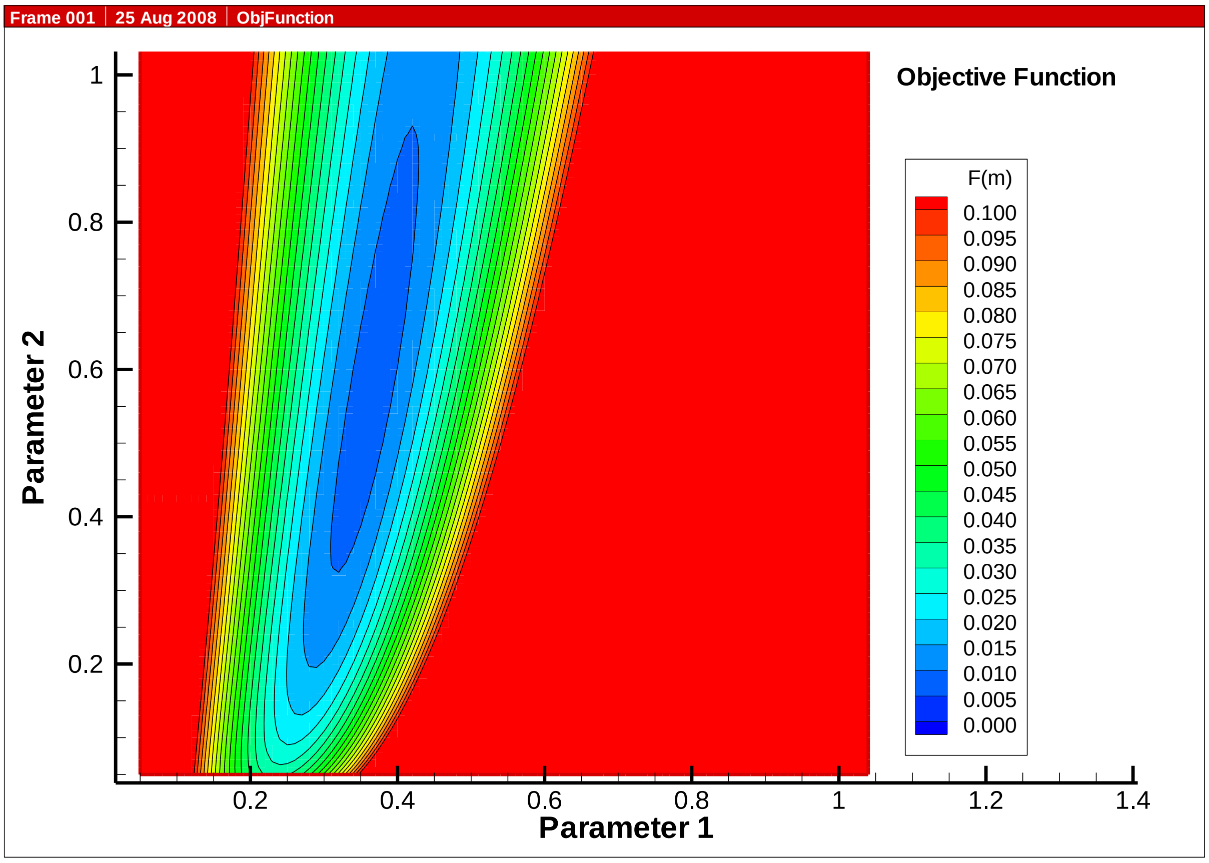
These equations form the basis for the Gauss-Newton algorithm for a non-linear least squares problem.

Example

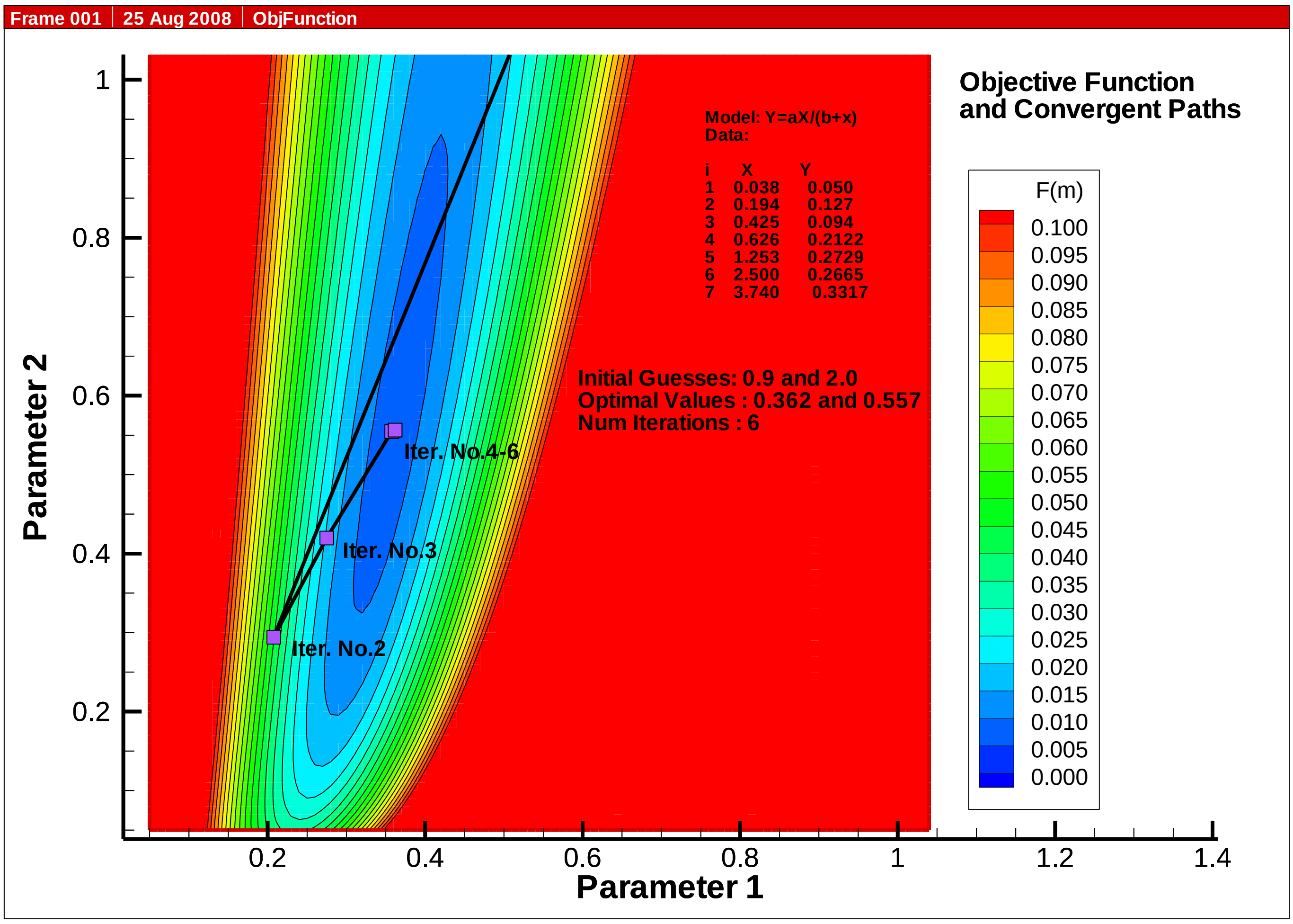
Model: $\gamma(a_1, a_2) = \dfrac{a_1 x}{a_2 + x}$

And observations:

i	1	2	3	4	5	6	7
x	0.038	0.194	0.425	0.626	1.253	2.500	3.740
γ	0.050	0.127	0.094	0.2122	0.2729	0.2665	0.3317



Objective Function within $0<a_1<1.05$ and $0<a_2<1.05$



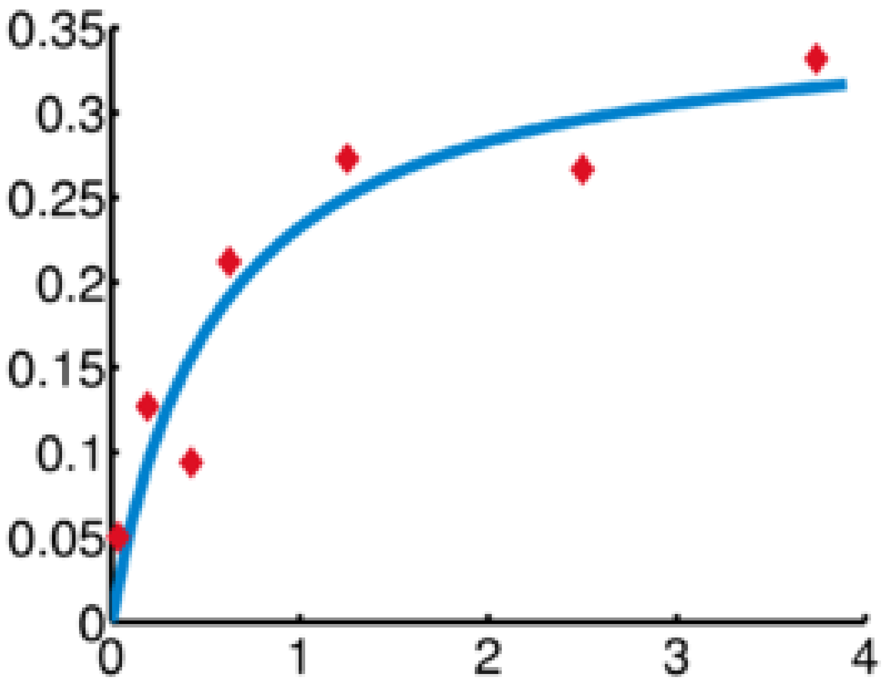
Model: $Y=aX/(b+x)$

Data:

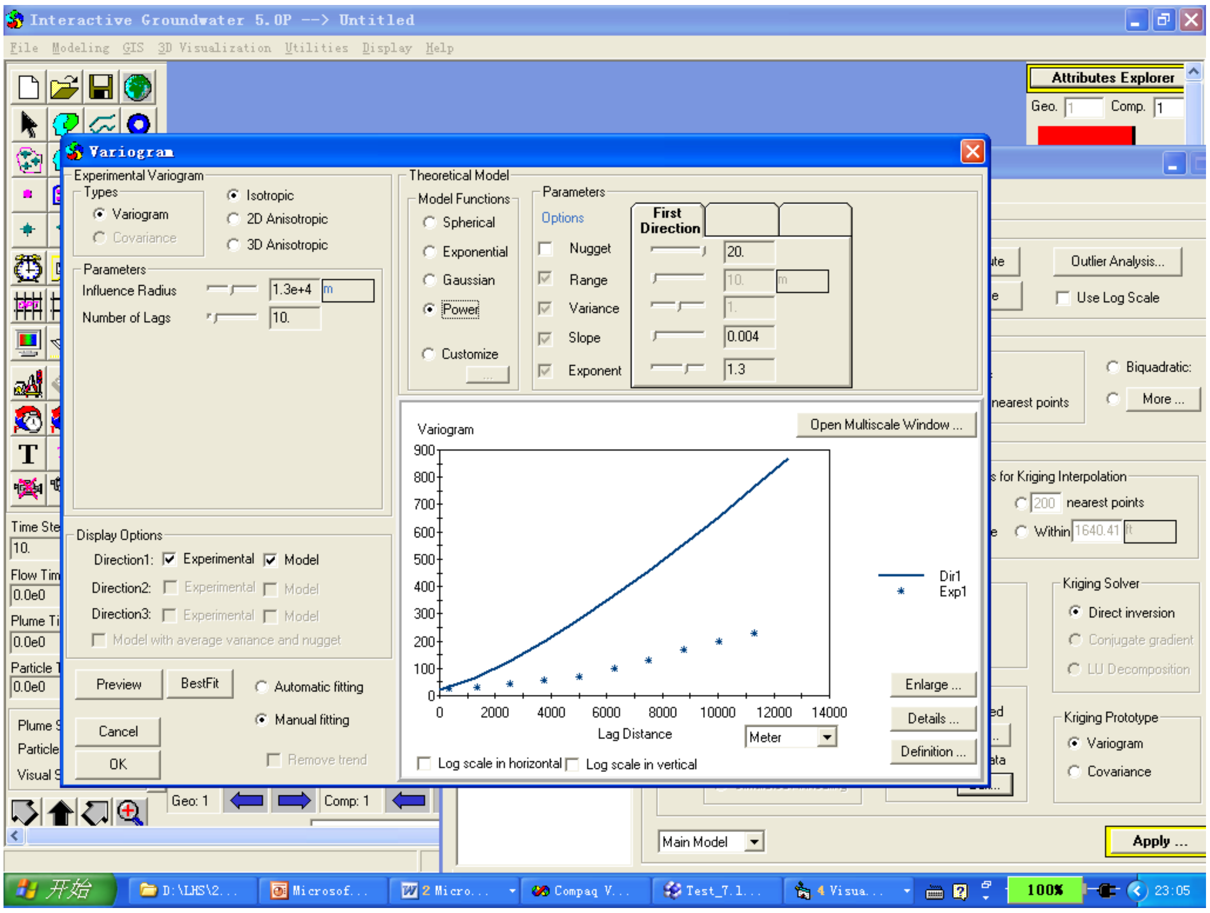
i	X	Y
1	0.038	0.050
2	0.194	0.127
3	0.425	0.094
4	0.626	0.2122
5	1.253	0.2729
6	2.500	0.2665
7	3.740	0.3317

Initial Guesses: 0.9 and 2.0
Optimal Values : 0.362 and 0.557
Num Iterations : 6

Convergent Paths starting with initial guesses a_1=0.9 and a_2=2.0
Optimal paramters are a_1=0.362 and a_2=0.557



1 Automatic Variogram Fitting


Interactive Groundwater 5.0P --> Untitled

Variogram

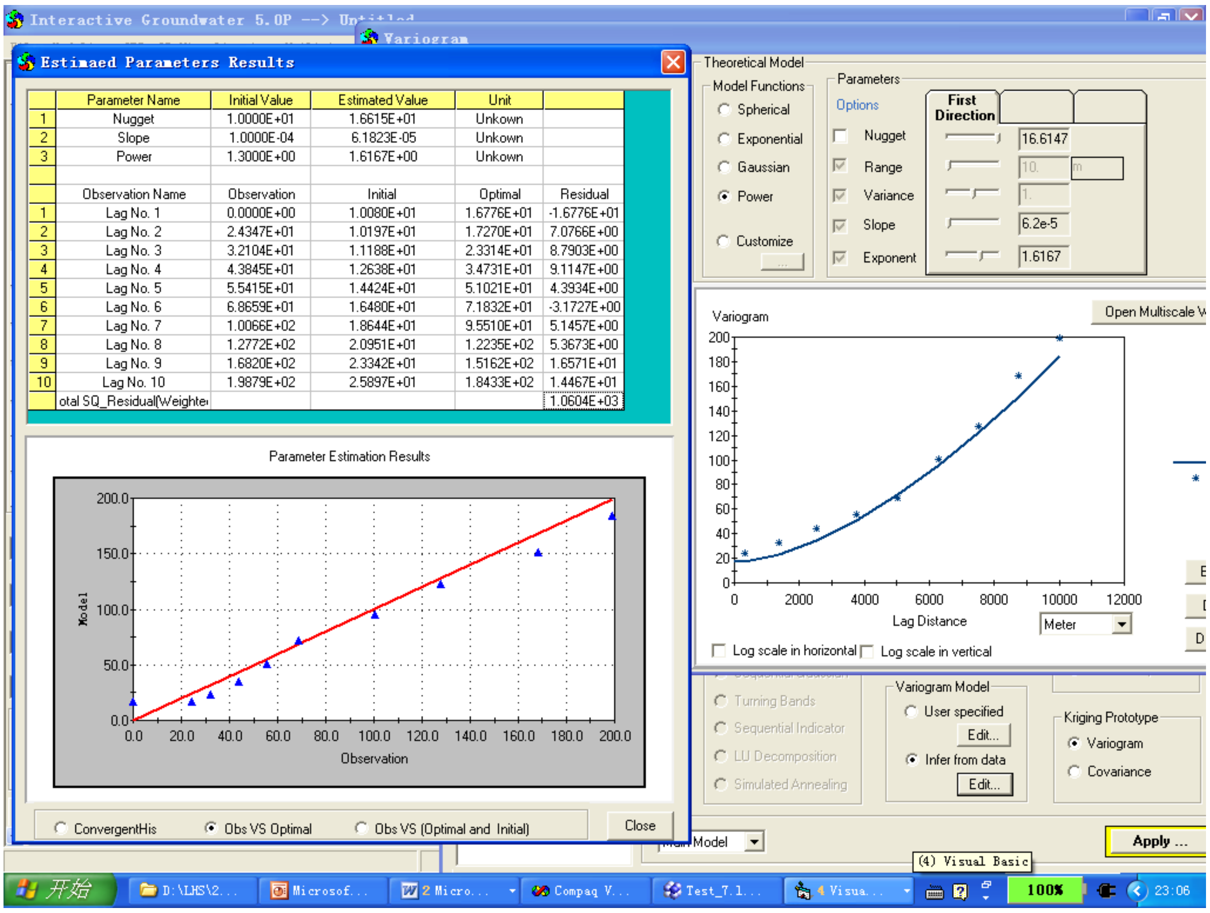
Experimental Variogram
Types: [✓ Variogram] Covariance
[✓ Isotropic] 2D Anisotropic 3D Anisotropic
Parameters
Influence Radius 1.3e+4 m
Number of Lags 10.

Theoretical Model
Model Functions: Spherical Exponential Gaussian [✓ Power] Customize
Parameters — First Direction
Nugget 20.
[✓ Range] 10. m
[✓ Variance] 1.
[✓ Slope] 0.004
[✓ Exponent] 1.3

Display Options
Direction1: [✓ Experimental] [✓ Model]
Direction2: Experimental Model
Direction3: Experimental Model
Model with average variance and nugget

Preview BestFit Cancel OK
Automatic fitting [✓ Manual fitting]
Remove trend

Button "BestFit"



	Parameter Name	Initial Value	Estimated Value	Unit	
1	Nugget	1.0000E+01	1.6615E+01	Unkown	
2	Slope	1.0000E-04	6.1823E-05	Unkown	
3	Power	1.3000E+00	1.6167E+00	Unkown	
	Observation Name	Observation	Initial	Optimal	Residual
1	Lag No. 1	0.0000E+00	1.0080E+01	1.6776E+01	-1.6776E+01
2	Lag No. 2	2.4347E+01	1.0197E+01	1.7270E+01	7.0766E+00
3	Lag No. 3	3.2104E+01	1.1188E+01	2.3314E+01	8.7903E+00
4	Lag No. 4	4.3845E+01	1.2638E+01	3.4731E+01	9.1147E+00
5	Lag No. 5	5.5415E+01	1.4424E+01	5.1021E+01	4.3934E+00
6	Lag No. 6	6.8659E+01	1.6480E+01	7.1832E+01	-3.1727E+00
7	Lag No. 7	1.0066E+02	1.8644E+01	9.5510E+01	5.1457E+00
8	Lag No. 8	1.2772E+02	2.0951E+01	1.2235E+02	5.3673E+00
9	Lag No. 9	1.6820E+02	2.3342E+01	1.5162E+02	1.6571E+01
10	Lag No. 10	1.9879E+02	2.5897E+01	1.8433E+02	1.4467E+01
	otal SQ_Residual(Weighte				1.0604E+03

Model Functions: [✓ Power]

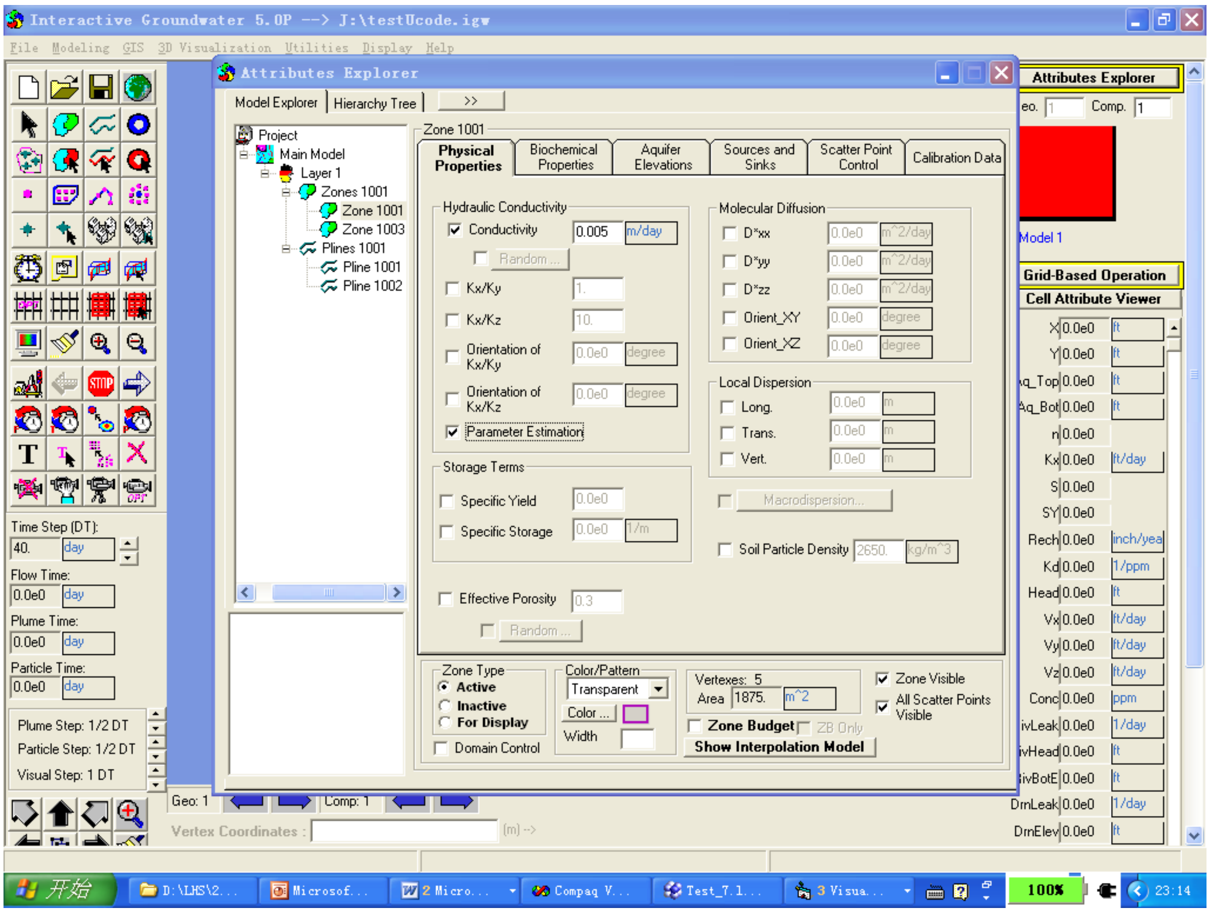
Parameters	First Direction
Nugget	16.6147
Range	10.
Variance	1.
Slope	6.2e-5
Exponent	1.6167

Variogram Model: [✓ Infer from data]
Kriging Prototype: [✓ Variogram]
[✓ Obs VS Optimal]

Parameters Estimation window

2 Parameter Estimation by using IGW

(1) Seleting parameter zones: Check those polygon-based parameters to be estimated when create your conceptual model (only conductivity and recharge are available right now)


Interactive Groundwater 5.0P --> J:\testUcode.igw

Attributes Explorer

Model Explorer | Hierarchy Tree

Project
- Main Model
  - Layer 1
    - Zones 1001
      - Zone 1001
      - Zone 1003
    - Plines 1001
      - Pline 1001
      - Pline 1002

Zone 1001

Physical Properties | Biochemical Properties | Aquifer Elevations | Sources and Sinks | Scatter Point Control | Calibration Data

Hydraulic Conductivity
- [✓] Conductivity 0.005 m/day
- [ ] Random ...
- [ ] Kx/Ky 1.
- [ ] Kx/Kz 10.
- [ ] Orientation of Kx/Ky 0.0e0 degree
- [ ] Orientation of Kx/Kz 0.0e0 degree
- [✓] Parameter Estimation

Storage Terms
- [ ] Specific Yield 0.0e0
- [ ] Specific Storage 0.0e0 1/m

- [ ] Effective Porosity 0.3
- [ ] Random ...

Molecular Diffusion
- [ ] D*xx 0.0e0 m^2/day
- [ ] D*yy 0.0e0 m^2/day
- [ ] D*zz 0.0e0 m^2/day
- [ ] Orient_XY 0.0e0 degree
- [ ] Orient_XZ 0.0e0 degree

Local Dispersion
- [ ] Long. 0.0e0 m
- [ ] Trans. 0.0e0 m
- [ ] Vert. 0.0e0 m

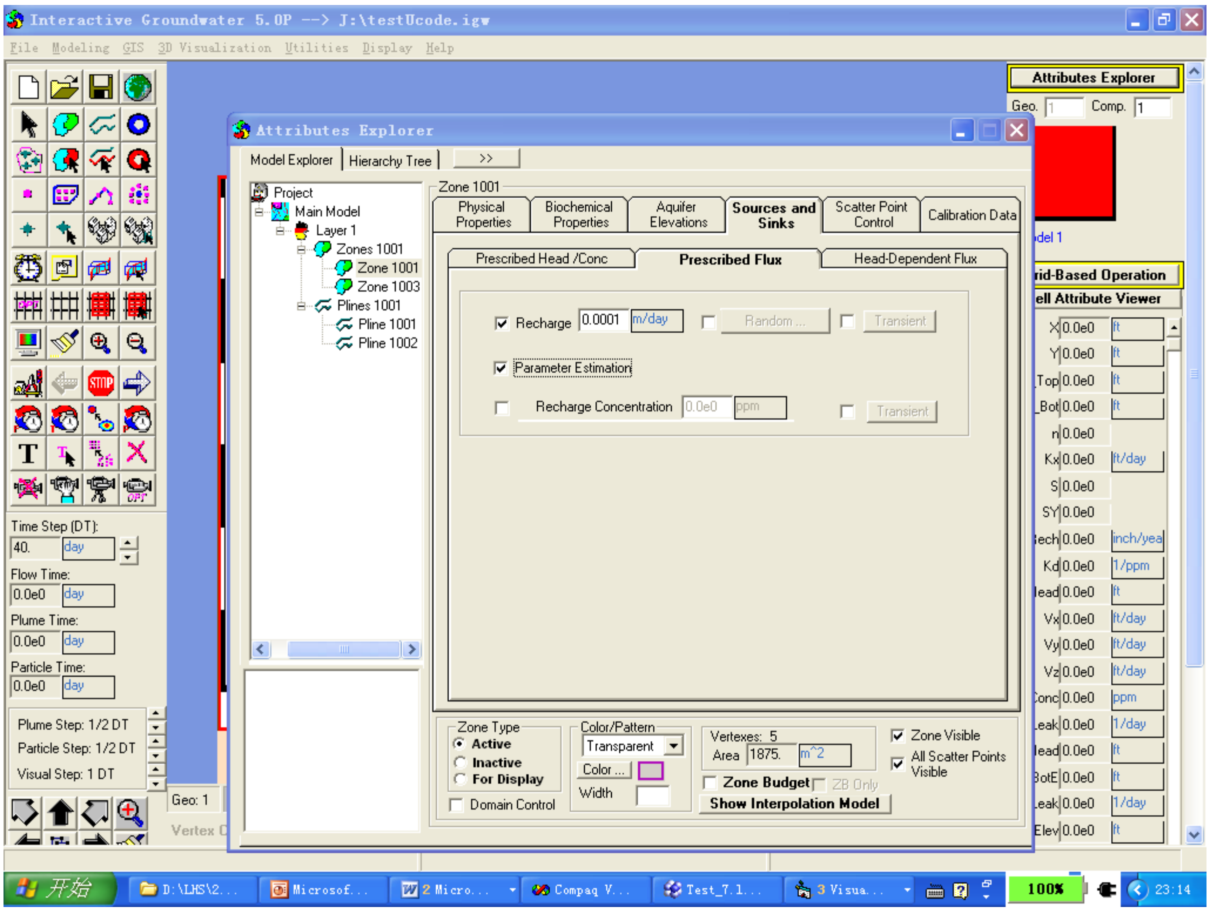
- [ ] Macrodispersion...
- [ ] Soil Particle Density 2650. kg/m^3

Zone Type: [✓ Active] Inactive For Display
- [ ] Domain Control

Color/Pattern: Transparent; Color ...; Width

Vertexes: 5; Area 1875. m^2
- [ ] Zone Budget [ ] ZB Only
- Show Interpolation Model
- [✓] Zone Visible
- [✓] All Scatter Points Visible

Time Step (DT): 40. day; Flow Time: 0.0e0 day; Plume Time: 0.0e0 day; Particle Time: 0.0e0 day; Plume Step: 1/2 DT; Particle Step: 1/2 DT; Visual Step: 1 DT


Interactive Groundwater 5.0P --> J:\testUcode.igw

Attributes Explorer

Zone 1001

Physical Properties | Biochemical Properties | Aquifer Elevations | Sources and Sinks | Scatter Point Control | Calibration Data

Prescribed Head /Conc | Prescribed Flux | Head-Dependent Flux

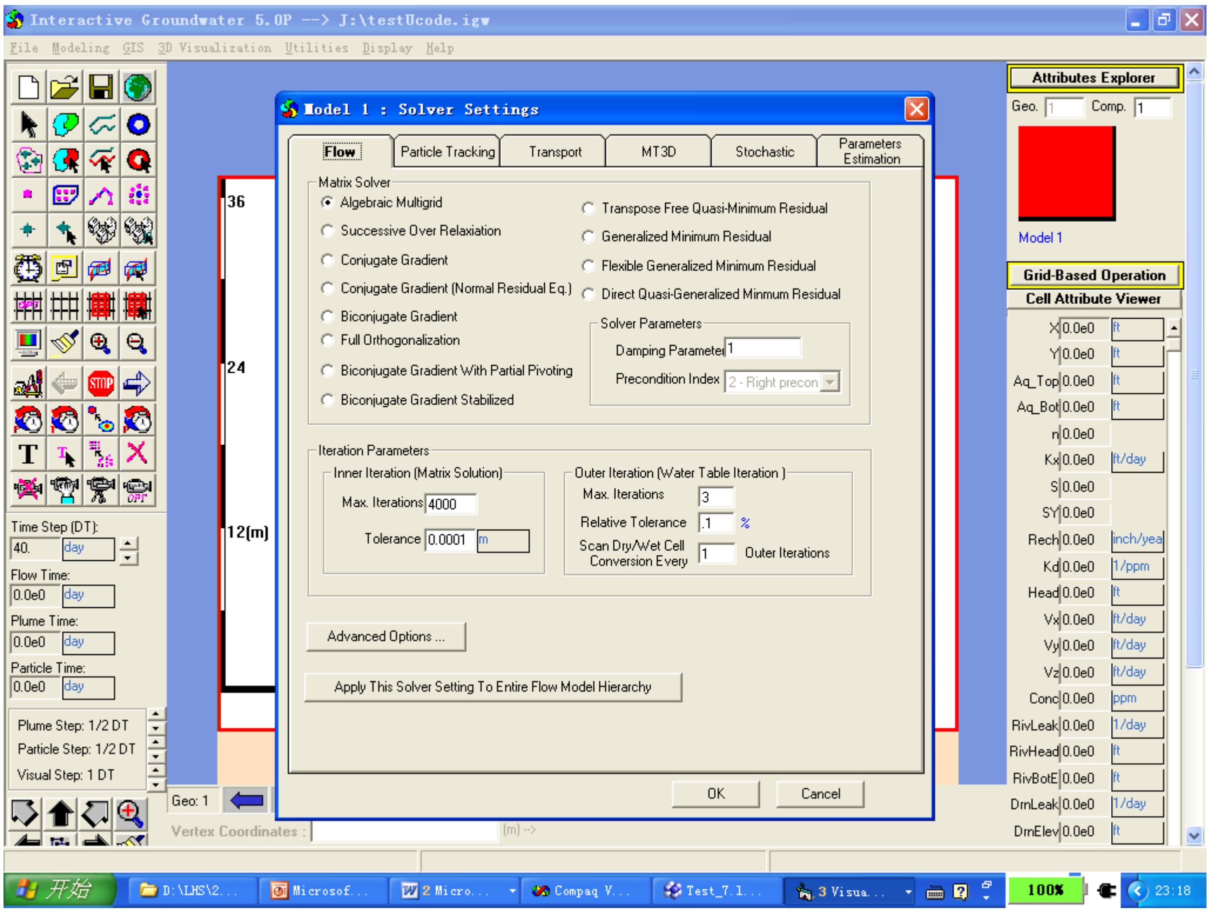
- [✓] Recharge 0.0001 m/day [ ] Random ... [ ] Transient
- [✓] Parameter Estimation
- [ ] Recharge Concentration 0.0e0 ppm [ ] Transient

Zone Type: [✓ Active] Inactive For Display
- [ ] Domain Control

Color/Pattern: Transparent; Color ...; Width

Vertexes: 5; Area 1875. m^2
- [ ] Zone Budget [ ] ZB Only
- Show Interpolation Model
- [✓] Zone Visible
- [✓] All Scatter Points Visible

(2) Select Solver: hit lead to



Model 1 : Solver Settings

Flow | Particle Tracking | Transport | MT3D | Stochastic | Parameters Estimation

Matrix Solver
- [✓ Algebraic Multigrid]
- Successive Over Relaxation
- Conjugate Gradient
- Conjugate Gradient (Normal Residual Eq.)
- Biconjugate Gradient
- Full Orthogonalization
- Biconjugate Gradient With Partial Pivoting
- Biconjugate Gradient Stabilized
- Transpose Free Quasi-Minimum Residual
- Generalized Minimum Residual
- Flexible Generalized Minimum Residual
- Direct Quasi-Generalized Minmum Residual

Solver Parameters
Damping Parameter 1
Precondition Index 2 - Right precon

Iteration Parameters
Inner Iteration (Matrix Solution)
Max. Iterations 4000
Tolerance 0.0001 m

Outer Iteration (Water Table Iteration)
Max. Iterations 3
Relative Tolerance .1 %
Scan Dry/Wet Cell Conversion Every 1 Outer Iterations

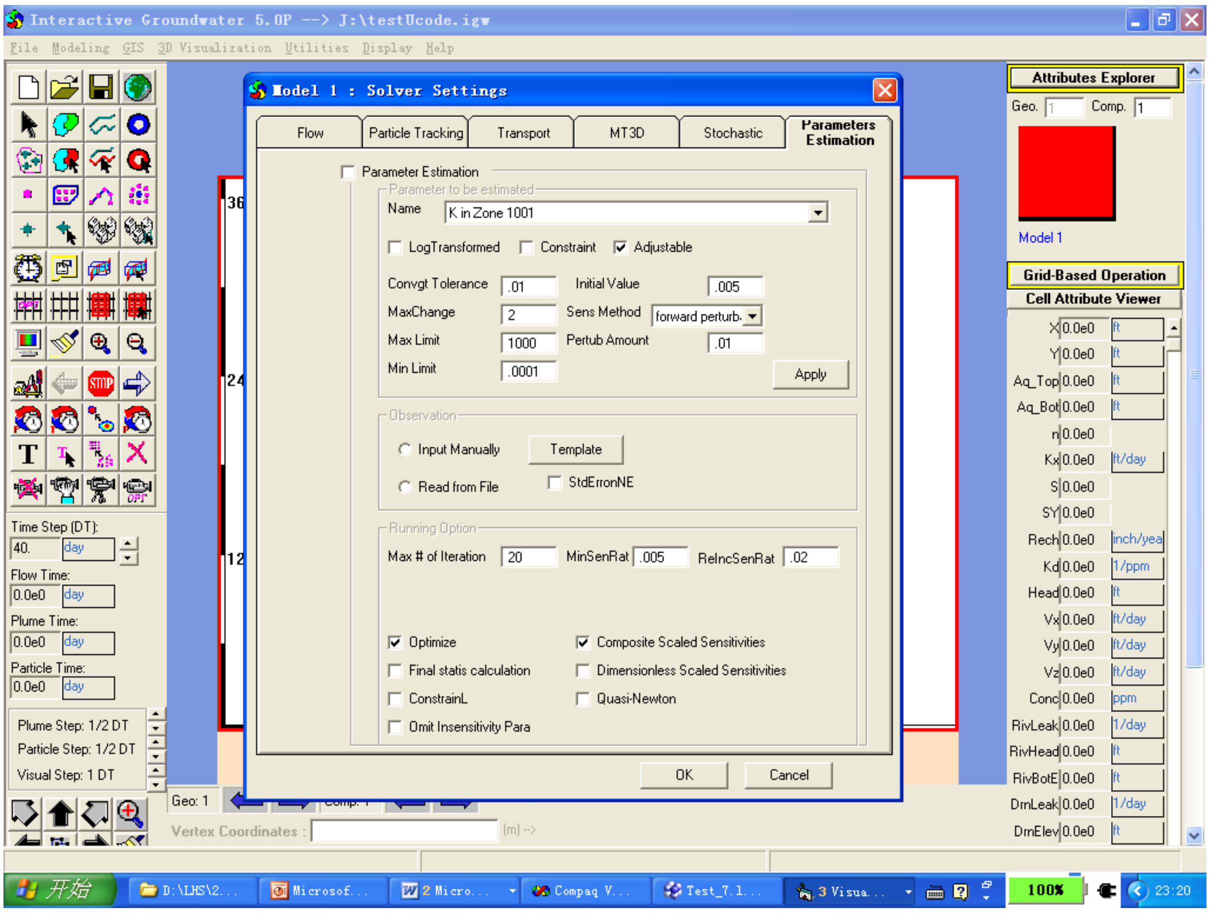
Advanced Options ...
Apply This Solver Setting To Entire Flow Model Hierarchy
OK Cancel

Time Step (DT): 40. day
Flow Time: 0.0e0 day
Plume Time: 0.0e0 day
Particle Time: 0.0e0 day
Plume Step: 1/2 DT
Particle Step: 1/2 DT
Visual Step: 1 DT

X	0.0e0	ft
Y	0.0e0	ft
Aq_Top	0.0e0	ft
Aq_Bot	0.0e0	ft
n	0.0e0	
Kx	0.0e0	ft/day
S	0.0e0	
SY	0.0e0	
Rech	0.0e0	inch/yea
Kd	0.0e0	1/ppm
Head	0.0e0	ft
Vx	0.0e0	ft/day
Vy	0.0e0	ft/day
Vz	0.0e0	ft/day
Conc	0.0e0	ppm
RivLeak	0.0e0	1/day
RivHead	0.0e0	ft
RivBotE	0.0e0	ft
DrnLeak	0.0e0	1/day
DrnElev	0.0e0	ft

New tab in "Solver" toobar: Parameter Estimation

(3) Check "parameter estimation" check box



Parameter Estimation

Parameter to be estimated

Name: K in Zone 1001

LogTransformed ☐ Constraint ☐ Adjustable ✓

Convgt Tolerance	.01	Initial Value	.005
MaxChange	2	Sens Method	forward perturb.
Max Limit	1000	Pertub Amount	.01
Min Limit	.0001		

Observation

○ Input Manually

○ Read from File ☐ StdErronNE

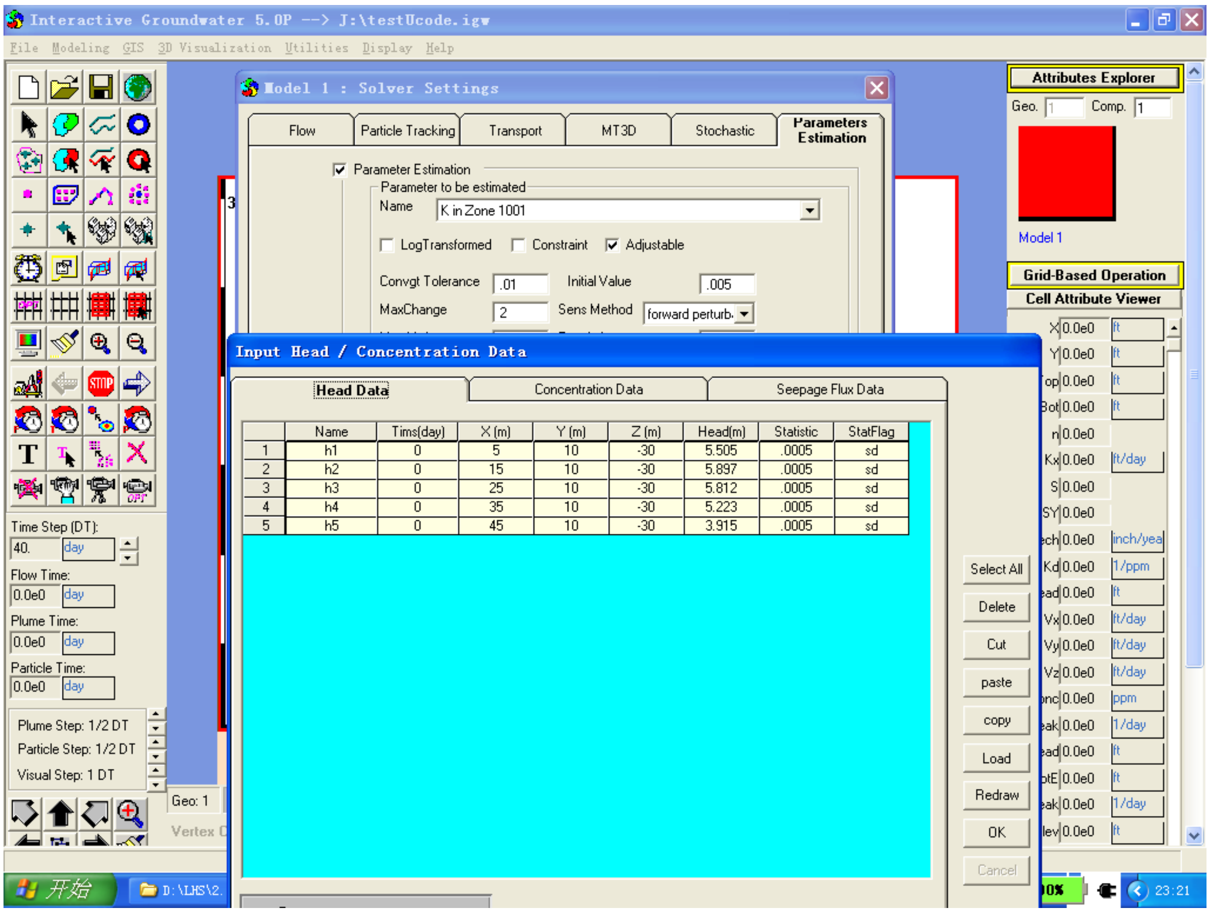
Running Option

Max # of Iteration: 20 MinSenRat: .005 RelncSenRat: .02

✓ Optimize	✓ Composite Scaled Sensitivities
☐ Final statis calculation	☐ Dimensionless Scaled Sensitivities
☐ ConstrainL	☐ Quasi-Newton
☐ Omit Insensitivity Para	

In the new tab: Newton method option, constrains, method to calculate senstivity matrix, …

(4) Observation input



[✓] Parameter Estimation

Parameter to be estimated

Name: K in Zone 1001

[ ] LogTransformed [ ] Constraint [✓] Adjustable

Convgt Tolerance: .01 Initial Value: .005

MaxChange: 2 Sens Method: forward perturb.

Time Step (DT): 40. day

Flow Time: 0.0e0 day

Plume Time: 0.0e0 day

Particle Time: 0.0e0 day

Plume Step: 1/2 DT

Particle Step: 1/2 DT

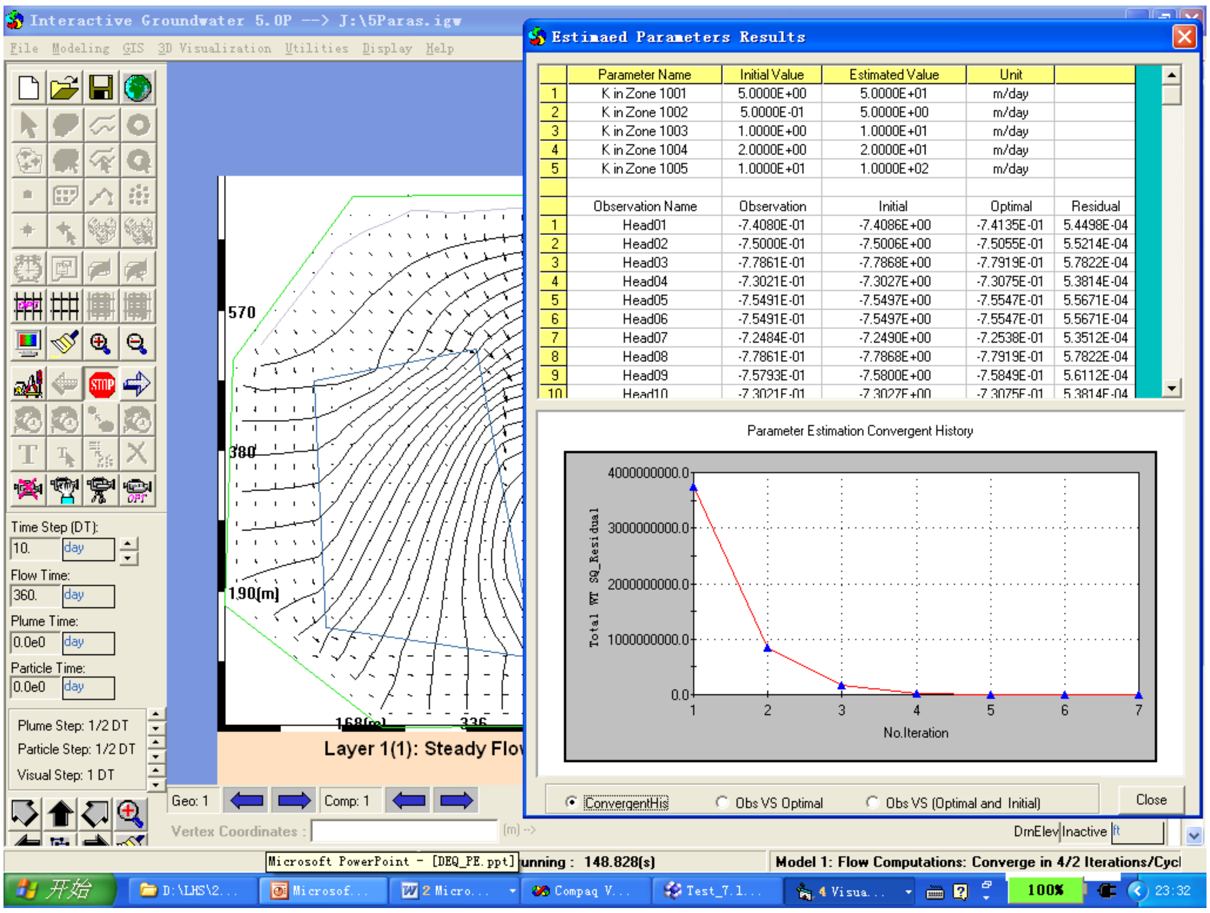
Visual Step: 1 DT

Geo: 1 Comp: 1

	Name	Tims(day)	X (m)	Y (m)	Z (m)	Head(m)	Statistic	StatFlag
1	h1	0	5	10	-30	5.505	.0005	sd
2	h2	0	15	10	-30	5.897	.0005	sd
3	h3	0	25	10	-30	5.812	.0005	sd
4	h4	0	35	10	-30	5.223	.0005	sd
5	h5	0	45	10	-30	3.915	.0005	sd

Observation input: file/ manually input/copy-paste

(5) Results --- parameter estimation results window

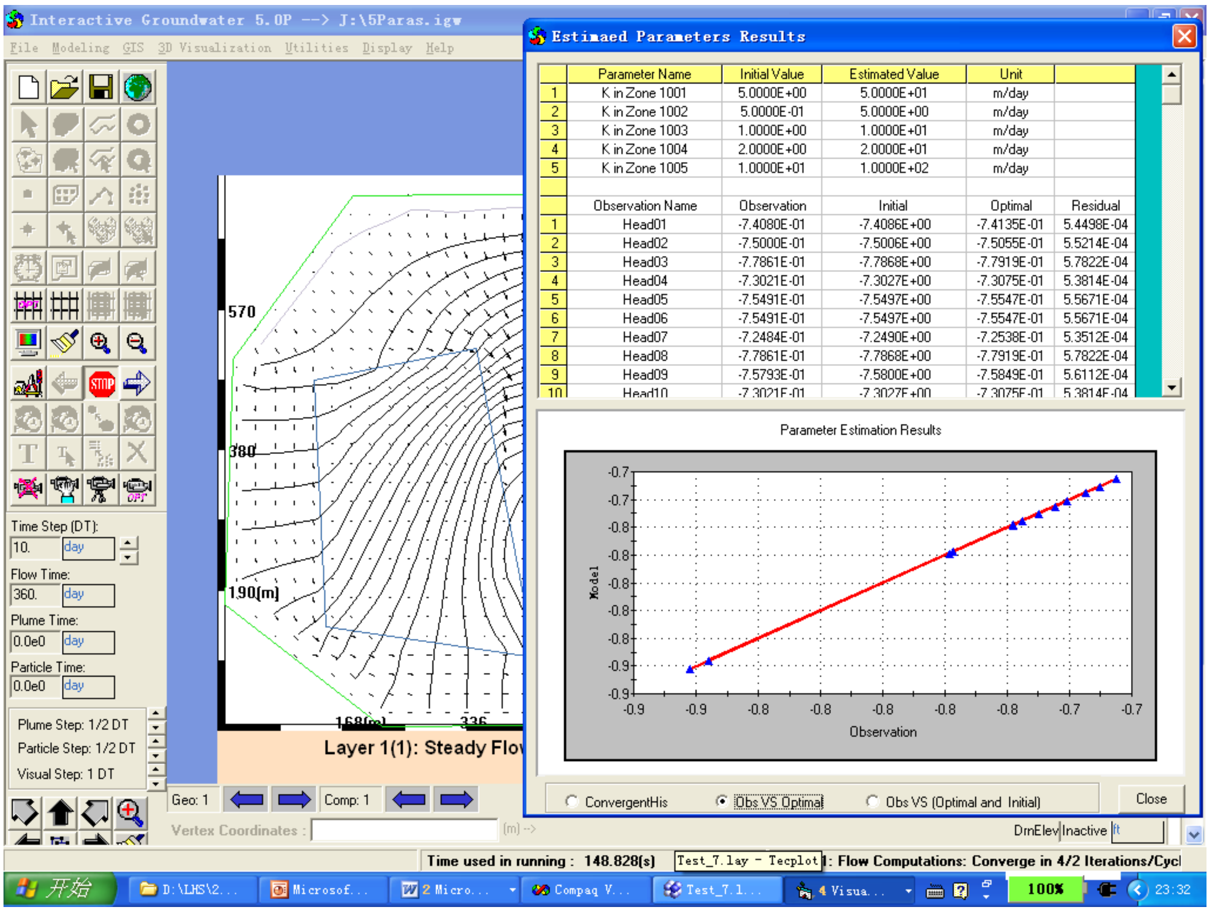


Estimaed Parameters Results

	Parameter Name	Initial Value	Estimated Value	Unit	
1	K in Zone 1001	5.0000E+00	5.0000E+01	m/day	
2	K in Zone 1002	5.0000E-01	5.0000E+00	m/day	
3	K in Zone 1003	1.0000E+00	1.0000E+01	m/day	
4	K in Zone 1004	2.0000E+00	2.0000E+01	m/day	
5	K in Zone 1005	1.0000E+01	1.0000E+02	m/day	
	Observation Name	Observation	Initial	Optimal	Residual
1	Head01	-7.4080E-01	-7.4086E+00	-7.4135E-01	5.4498E-04
2	Head02	-7.5000E-01	-7.5006E+00	-7.5055E-01	5.5214E-04
3	Head03	-7.7861E-01	-7.7868E+00	-7.7919E-01	5.7822E-04
4	Head04	-7.3021E-01	-7.3027E+00	-7.3075E-01	5.3814E-04
5	Head05	-7.5491E-01	-7.5497E+00	-7.5547E-01	5.5671E-04
6	Head06	-7.5491E-01	-7.5497E+00	-7.5547E-01	5.5671E-04
7	Head07	-7.2484E-01	-7.2490E+00	-7.2538E-01	5.3512E-04
8	Head08	-7.7861E-01	-7.7868E+00	-7.7919E-01	5.7822E-04
9	Head09	-7.5793E-01	-7.5800E+00	-7.5849E-01	5.6112E-04
10	Head10	-7.3021E-01	-7.3027E+00	-7.3075E-01	5.3814E-04

(● ConvergentHis) ○ Obs VS Optimal ○ Obs VS (Optimal and Initial)

Microsoft PowerPoint - [DEQ_PE.ppt] unning : 148.828(s) Model 1: Flow Computations: Converge in 4/2 Iterations/Cycl

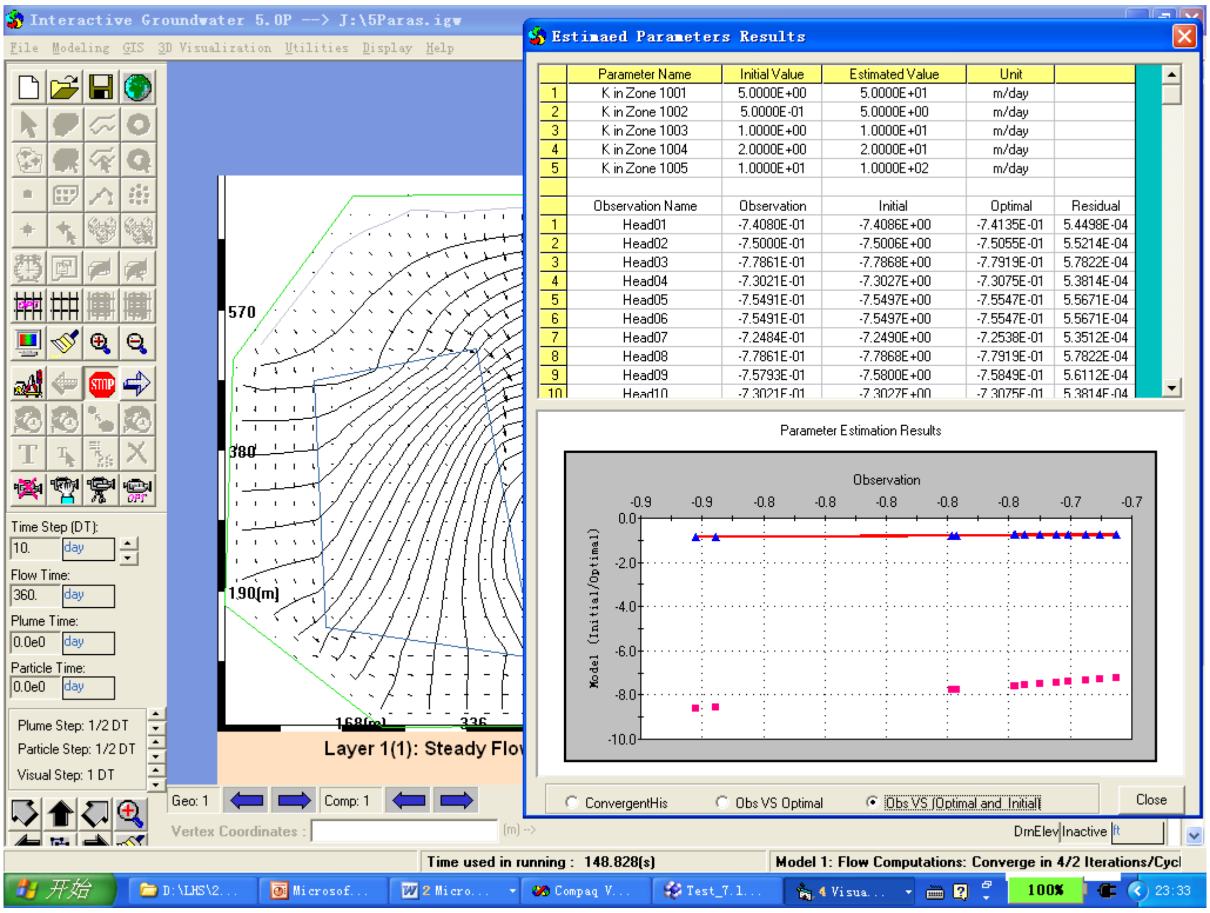


Estimaed Parameters Results

	Parameter Name	Initial Value	Estimated Value	Unit	
1	K in Zone 1001	5.0000E+00	5.0000E+01	m/day	
2	K in Zone 1002	5.0000E-01	5.0000E+00	m/day	
3	K in Zone 1003	1.0000E+00	1.0000E+01	m/day	
4	K in Zone 1004	2.0000E+00	2.0000E+01	m/day	
5	K in Zone 1005	1.0000E+01	1.0000E+02	m/day	
	Observation Name	Observation	Initial	Optimal	Residual
1	Head01	-7.4080E-01	-7.4086E+00	-7.4135E-01	5.4498E-04
2	Head02	-7.5000E-01	-7.5006E+00	-7.5055E-01	5.5214E-04
3	Head03	-7.7861E-01	-7.7868E+00	-7.7919E-01	5.7822E-04
4	Head04	-7.3021E-01	-7.3027E+00	-7.3075E-01	5.3814E-04
5	Head05	-7.5491E-01	-7.5497E+00	-7.5547E-01	5.5671E-04
6	Head06	-7.5491E-01	-7.5497E+00	-7.5547E-01	5.5671E-04
7	Head07	-7.2484E-01	-7.2490E+00	-7.2538E-01	5.3512E-04
8	Head08	-7.7861E-01	-7.7868E+00	-7.7919E-01	5.7822E-04
9	Head09	-7.5793E-01	-7.5800E+00	-7.5849E-01	5.6112E-04
10	Head10	-7.3021E-01	-7.3027E+00	-7.3075E-01	5.3814E-04

○ ConvergentHis (● Obs VS Optimal) ○ Obs VS (Optimal and Initial)

Time used in running : 148.828(s) Test_7.lay - Tecplot]: Flow Computations: Converge in 4/2 Iterations/Cycl


Estimaed Parameters Results

	Parameter Name	Initial Value	Estimated Value	Unit	
1	K in Zone 1001	5.0000E+00	5.0000E+01	m/day	
2	K in Zone 1002	5.0000E-01	5.0000E+00	m/day	
3	K in Zone 1003	1.0000E+00	1.0000E+01	m/day	
4	K in Zone 1004	2.0000E+00	2.0000E+01	m/day	
5	K in Zone 1005	1.0000E+01	1.0000E+02	m/day	
	Observation Name	Observation	Initial	Optimal	Residual
1	Head01	-7.4080E-01	-7.4086E+00	-7.4135E-01	5.4498E-04
2	Head02	-7.5000E-01	-7.5006E+00	-7.5055E-01	5.5214E-04
3	Head03	-7.7861E-01	-7.7868E+00	-7.7919E-01	5.7822E-04
4	Head04	-7.3021E-01	-7.3027E+00	-7.3075E-01	5.3814E-04
5	Head05	-7.5491E-01	-7.5497E+00	-7.5547E-01	5.5671E-04
6	Head06	-7.5491E-01	-7.5497E+00	-7.5547E-01	5.5671E-04
7	Head07	-7.2484E-01	-7.2490E+00	-7.2538E-01	5.3512E-04
8	Head08	-7.7861E-01	-7.7868E+00	-7.7919E-01	5.7822E-04
9	Head09	-7.5793E-01	-7.5800E+00	-7.5849E-01	5.6112E-04
10	Head10	-7.3021E-01	-7.3027E+00	-7.3075E-01	5.3814E-04

Table: Parameters estimated/residuals

Charts:

Iteration history

Observations VS model values

Initial guesses VS Optimal values

Demonstration:

1 GIS data variogram model

2 5 K parameters estimated by using IGW flow model

3 5 K + 1 recharge parameters estimated by using IGW flow model