A Comparative Study on Fitting of Gielis Curves by Classical versus Generalized Simulated Annealing Methods

SK Mishra
Dept. of Economics
NEHU, Shillong (India)

Introduction: The Gielis superformula

$$r(\theta) = f(\theta).\left[\left|\tfrac{1}{a}\cos(\tfrac{m}{4}\theta)\right|^{n_2} + \left|\tfrac{1}{b}\sin(\tfrac{m}{4}\theta)\right|^{n_3}\right]^{(-n_1^{-1})} = f(\theta).g(\theta) \ ; \ a,\ b,\ m > 0 \qquad \dots (1)$$

describes almost any closed curve in terms of the deformed circle (or ellipse), $g(\theta)$, and another function, $f(\theta)$, and their parameters (Gielis, 2003; Gielis and Gerats, 2004). The function $f(\theta)$ may be considered as a modifier of the Gielis function, $g(\theta)$.

Estimation of Gielis Parameters: For a scientific purpose, Gielis parameters need to be estimated from empirical data. Presently, we are concerned with the possibilities of the same. Let the n true points be $[z_i = (x_i, y_i);\ i = 1, 2, ..., n]$, of which the corresponding observed values are $z' = (x_i', y_i')$, possibly with errors of measurement and displacement of origin by (c_x, c_y), unknown to us. Let $(\tilde{c}_x, \tilde{c}_y)$ be the approximate or assumed values of (c_x, c_y). Let us denote by $\tilde{z}_i = (\tilde{x}_i, \tilde{y}_i) = (x_i' - \tilde{c}_x, y_i' - \tilde{c}_y)$. From these values we obtain $\tilde{r}_i = \sqrt{(\tilde{x}_i^{\ 2} + \tilde{y}_i^{\ 2})}$. We also obtain $\tilde{\theta}_i = \tan^{-1}(\tilde{y}_i / \tilde{x}_i)$. On the other hand, we obtain $\hat{r}_i = g(\tilde{\theta}_i, \tilde{a}, \tilde{b}, \tilde{m}, \tilde{n}_1, \tilde{n}_2, \tilde{n}_3).f(\tilde{\theta})$, where $g(.)$ is the Gielis super-formula defined in (4) and $f(\theta)$ is variously defined. The wavy bar on the arguments of $g(.)$ and $f(.)$ indicates that all parameters have taken on some assumed values, which may not be the correct values. The deviation of assumed values of parameters from their true values gives rise to $d_i = abs(\tilde{r}_i - \hat{r}_i)$ and consequently $S^2 = \sum_{i=1}^{n} d_i^2 \geq 0.$ Only if the assumed values of parameters are the true values, S^2 can be zero, but smaller it is, closer are the assumed values of the parameters from their true values (assuming empirical uniqueness of the parameters to a given set of data). Thus we have to find the values of Gielis parameters in $g(.)$ and $f(.)$ such that S^2 is minimum.

Minimization of S^2 poses formidable problems due to two reasons. First, the Gielis parameters are possibly not unique to data suggesting that minima (local as well as global) are located in the valleys. The three parameters, n_1, n_2 and n_3 of the deformed circle, $g(\theta)$, interact with each other even if we assume that the parameters of the modifying function, $f(\theta)$, do not influence them. However, that is not the case. As a matter of fact, all of them interact with each other. A large number of experiments carried out by the author make the basis of this view. Secondly, the parameters span a highly nonlinear surface of S^2, which has innumerably many local minima (Mishra, 2006).

- Working Paper (June, 2006): Dept. of Economics, NEHU, Shillong (India) - 793022

As it is well known, most of the nonlinear optimization procedures that were developed in the 1960's or before are extremely prone to be caught in the local optima if the surface to be optimized is substantially irregular, ridged and multi-modal. In the due course, researchers in the field of operations research turned to learning from nature and imitating the process in which natural processes attain a minimum. Understanding the process of adaptation of living beings to their environment for a survival led to development of the 'genetic algorithm' (Holland, 1975) and the optimization method based on adaptation (Goldberg, 1989; Wright, 1991). This method mimics the process of survival of the fittest. A few other methods such as the Particle Swarm methods (Eberhart and Kennedy, 1995; see Parsopoulos and Vrahatis, 2002) were motivated by the behaviour of the living beings. On the other side, researchers learned from physics – the process of annealing in metallurgy (Kirkpatrick et al., 1983) and the method of 'simulated annealing' was developed. This method was improved (Tsallis and Stariolo, 1995) by the replacement of the Gaussian (Boltzmann-Gibbs) visiting distribution (used by CSA) and the Cauchy-Lorentz visiting distribution (used by the Fast Simulated Annealing – FSA – method) by the Tsallis visiting distribution, making the earlier methods (CSA and FSA) as special cases.

The Simulation Experiments: We have experimented with nine different models. All these models are instances of a deformed circle, $g(.)$, modified by different modifier functions, $f(.)$. Three typical instances of $g(.)$ have been chosen. The parameters of $g(.)$ are given in table A.1. Three typical modifier functions are chosen, as given below. The chosen values of n_4 and n_5 are also given in table-A.1.

$f_1(\theta) = r = [n_4(3\cos(t) - \cos(3t))^2 + n_5(3\sin(t) - \sin(3t))^2]^{0.5}$: (Nephroid) ... (2)

$f_2(\theta) = \mathrm{r} = \mathrm{n}_4 + \mathrm{n}_5 \cos(\mathrm{t})$: (Limaçon) ... (3)

$f_3(\theta) = \mathrm{r} = \mathrm{n}_4 - \mathrm{n}_5 \cos(\mathrm{t}) + \mathrm{abs}(\cos(\mathrm{t}))^3$ ...(4)

In all the three modifier functions, n_4 and n_5 are parameters and $0 \le t \le 2\pi$.

In case of each model, hundred uniformly distributed random points have been generated with the parameters specified in the relevant $g(.)$ and $f(.)$. The Classical simulated annealing (CSA of Kirkpatrick et al., 1983) and the Generalized simulated annealing (GSA of Tsallis and Stariolo, 1995) methods of optimization have been repetitively applied to estimate the parameters. The CSA method requires the bounds (the lower and the upper limits; LL and UL) on the parameters to be specified. For all the nine models we have used the identical set of bounds, specified in table-A.1. However, GSA does not require such bounds. The jointly estimated parameters of $g(.)$ and $f(.)$ by both methods are presented in table-A.1. Their graphs are presented in Fig.A.1 and Fig.A.2 $M_{ij}(i, j = 1, 2, ..., 3)$. The red points are those generated by the true parameters, the blue ones are generated by using the CSA-estimated parameters and the green ones are those generated by the GSA-estimated parameters. For each model, the estimated (blue as well as green) points are superimposed on the generated (red) points to facilitate a visual assessment of the quality of fit, which is quantitatively represented by the value of S^2. ***For GSA we have used the program developed by Mundim (1996) as the central subprogram under our main program.***

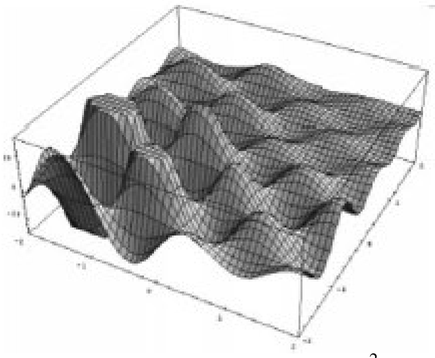
The Findings: The CSA method of optimization performs very well in fitting the Gielis curves to data. It performs better than the GSA. The GSA has almost always been caught in the sub-optimal traps as indicated by the relatively large values of S^2 and bad fit presented in the figures. In general, increase in the number of iterations to obtain GSA estimates improves performance greatly. When max number of iterations (NStopMax) was 1000, we get inferior estimates (see rows G in table A.1 in the appendix and associated plots in Fig-A.1). With NstopMax =10000 the estimates are much better and often close to CSA estimates (see rows H in table A.1 in the appendix and associated plots in Fig. A.2). Save in an isolated case of Model-21, CSA estimates give better fit (smaller S^2) than the GSA estimates even when NStopMax is 10000.

Generalized Simulated Annealing program converges much faster. It does not require limits on the decision variables either. The initial guesses of the decision variables may simply be generated randomly, lying between -0.5 to +0.5 or so. However, its disadvantage is that it does not ensure a better result vis-à-vis the Classical Simulated Annealing method. Could we increase NStopMax to 100,000 for the GSA? Possibly, it could do better than CSA. But, in that case it is as slow as the CSA.

Observations: There could be several possible reasons for a relatively poor performance of the GSA vis-à-vis the CSA. First, there could be some bugs in the codes written by the present author (which integrates Mundim's program as a procedure or subprogram for function minimization) due to which Mundim's program fails to work properly. Secondly, it might be so that Mundim's program itself has some bug or limitation. Thirdly, granted that codes are all right, we have specified the parameters wrongly. Lastly, GSA algorithm has some serious limitations.

In order to investigate into the possible reasons mentioned above, we have tested Mundim's program as it is. His program needs a subroutine FUNCT(X, F) to be provided by the user. We have used the following subroutine (Levy function No. 5). In the **gsa.in** file (that gives inputs to the program) we have specified qA = 1.5, qT=1.5, qV=1.5, NStopMax = 1000, To = 1.0E-00 and NDimension = 2.

```
      SUBROUTINE FUNCT(X,F)
      PARAMETER (MaxDim=500)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION X(MaxDim)
C     Levy # 5 (Levy et al. 1981) -----------------------------------
      f1=0.0d+00
      f2=0.0d+00
      do i=1,5
      f1=f1+(i*dcos((i-1)*x(1)+i))
      f2=f2+(i*dcos((i+1)*x(2)+i))
      enddo
      f3=(x(1)+1.42513d+00)**2
      f4=(x(2)+0.80032d+00)**2
      f=(f1*f2) + (f3+f4)
      RETURN
      END
```



Plot of Levy # 5 within the cube $[-2, 2]^2$

With $x_i \in [-10,10];\ i = 1,2$ this function has about 760 local minima and one global minimum with function value $f\,(x^{**}) = -176.1375$, at $x^{**} = (-1.3068, -1.4248)$ as mentioned by Parsopoulos and Vrahatis, 2002. The large number of local minimizers makes it difficult for any method to locate the global minimum. That is why this function is used as a test problem. Mundim's program gives $f\,(x^*) = -176.137392$ for the decision variables $x^* = (1.30707675, 1.42458832)$ run with the initial (arbitrary) value $x^0 = (2, 3)$.

However, when Mundim's program is run with $x^0 = (4, 4)$ we get $f\,(x^*\ ) = -103.608526$ at $x^* = (4.9679732, 4.85541621)$, which is clearly a local minimum. When run with $x^0 = (-1, 5)$, Mundim's program gives $f\,(x^*\ ) = -144.524898$ at $x^* = (-1.3063705, 4.85553323)$, which, again, a local minimum. Thus, it indicates that, perhaps, Mundim's program works well only if x^0 (the initial guess) is close to the optimal x, that is, x^{**}. *It appears that t**his is a serious limitation of Mundim's program.***

We test Mundim's program with Levy function No. 8. This function has three decision variables; $x = (x_1, x_2, x_3)$. The subroutine FUNCT(X, F) is given below. In the **gsa.in** file (that provides inputs to the program) we have specified qA = 1.5, qT=1.5, qV=1.5, NStopMax = 1000, To = 1.0E-00 and NDimension = 3.

```
      SUBROUTINE FUNCT(X,F)
      PARAMETER (MaxDim=500)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION X(MaxDim),Y(MAXDIM)
C     LEVY # 8 FUNCTION ---------------------------------------------
      PI=4.D+00*DATAN(1.D+00)

      DO I=1,3
      Y(I)=1.D+00+(X(I)-1.D+00)/4.D+00
      END DO
      F1=DSIN(PI*Y(1))**2
      F3=(Y(3)-1.D+00)**2
      F2=0.D+00
      DO I=1,2
      F2=F2+((Y(I)-1.D+00)**2)*(1.D+00+10.D+00*(DSIN(PI*Y(I+1)))**2)
      ENDDO
      F=F1+F2+F3
      RETURN
      END
```

Levy's function No. 8 for $x_i \in [-10, 10]$; $i = 1, 2, 3$ has one global minimum at the point $x^{**} = (1, 1, 1)$ with function value $f(x^{**}) = 0$, and, approximately, 125 local minima (Parsopoulos and Vrahatis, 2002). When run with the initial guess $x^0 = (-1, 2, 1)$, Mundim's program gives us $f(x^*) = 1.86677499\text{E-}006$ and $x^* = (1.00100189, 1.00086019, 1.00426821)$, which is very close to the global optimal value. When run with initial guess x^0 = (2, -2, 3), it gives us the values : $f(x^*)$ = 2.28761655E-006 and x^* = (1.00141443, 1.00254639, 0.997106632), which is again very close to the global optimal value. Run with x^0 = (2, 2, 3), we get $f(x^*)$ = 2.29972367E-006 and x^* = (0.99966183, 1.00259893, 0.994633726), which is very close to the global minimum. However, when run with x^0 = (5, 5, 5), we get $f(x^*)$ = 2.88283762 and x^* = (4.60405911, 4.95069414, 4.95861021), which is a local optimum, far away from the global optimum.

These two examples clearly suggest that Mundim's codes have some problems in searching the global optimum with an arbitrary starting point. Since, in practice, it is difficult to guess correct initial values (more so when there are many decision variables) to run the program, Mundim's codes are of limited value. This may be the reason why GSA has almost always performed poorly vis-à-vis CSA in our case (of fitting the Gielis curves).

References

Eberhart R.C. and Kennedy J.: “A New Optimizer using Particle Swarm Theory”, *Proceedings Sixth Symposium on Micro Machine and Human Science*, pp. 39–43. IEEE Service Center, Piscataway, NJ, 1995.

Gielis, J. and Gerats, T.: “A Botanical Perspective on Modeling Plants and Plant Shapes” *International Conference on Computing Communications and Control Technologies*. Austin, TX, Aug. 14-17, 2004.

Gielis, J.: “A Generic Geometric Transformation that unifies a Wide Range of Natural and Abstract Shapes”, *American Journal of Botany,* 90(3): pp. 333–338, 2003.

Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, Reading, Mass, 1989.

Holland, J.: *Adaptation in Natural and Artificial Systems,* Univ. of Michigan Press, Ann Arbor, 1975.

Kirkpatrick, S., Gelatt, C.D. Jr., and Vecchi, M.P.: "Optimization by Simulated Annealing", *Science*, 220, 4598, 671-680, 1983.

Mishra, S.K.: "On Estimation of the Parameters of Gielis Superformula from Empirical Data" *Social Science Research Network* (SSRN): http://ssrn.com/abstract=905051, Working Paper Series, 2006.

Mishra, S.K.: "Experiments on Estimation of the Parameters of Gielis Super-Formula by Simulated Annealing Method of Optimization" *Social Science Research Network* (SSRN): http://ssrn.com/abstract=910800 , 2006.

Mundim, K.C.: “Generalized Simulated Annealing”, (provides GSA Fortran Program to download) available at www.unb.br/iq/**kleber**/GSA/artigo2/node2.html , 1996

Mundim, K.C. and Tsallis, C.: “Geometry Optimization and Conformational Analysis Through Generalized Simulated Annealing*”,Int. Journal of Quantum Chemistry*, 58,, pp. 373-381, 1996.

Parsopoulos, K.E. and Vrahatis, M.N., “Recent Approaches to Global Optimization Problems Through Particle Swarm Optimization”, *Natural Computing*, 1 (2-3), pp. 235-306, 2002.

Tsallis, C. and Stariolo, D.A.: “Generalized Simulated Annealing”, ArXive cond-mat/9501047 v1 12 Jan, 1995.

Wright, A.H.: “Genetic Algorithms for Real Parameter Optimization”, in GJE Rawlings (ed) *Foundations of Genetic Algorithms*, Morgan Kauffmann Publishers, San Mateo, CA, pp. 205-218, 1991.

Appendix

Table-A.1. True & Estimated Gielis Parameters (Modified Curves) with Limits on them

	c_x	c_y	a	b	n_1	n_2	n_3	m	n_4	n_5	S^2	
M	0	0	1	1	0.6	2	3	3	3	2	0	T
#	-15	-15	0	0	-80	-80	-80	0	-10	-10	LL	
11	15	15	50	50	80	80	80	80	10	10	UL	
	0.010	-0.082	0.763	27.987	3.859	2.677	-0.089	6.022	5.898	3.918	0.435	C
	0.177	-0.397	4.297	3.451	2.724	9.665	-0.204	2.983	5.096	3.703	20.884	G
	-1.589	-0.150	1.596	1.019	0.577	0.610	3.639	-1.987	2.681	1.256	1.566	H
M	0	0	1	1	2	2	6	5	3	2	0	T
#	-15	-15	0	0	-80	-80	-80	0	-10	-10	LL	
12	15	15	50	50	80	80	80	80	10	10	UL	
	-0.005	0.088	0.568	0.831	9.809	3.550	-0.290	10.024	4.714	3.155	0.952	C
	0.353	-0.554	8.315	-0.503	3.222	4.710	-0.356	-2.516	3.675	2.605	16.633	G
	2.115	0.642	-0.050	-0.048	0.632	-1.245	-0.380	-1.072	0.769	0.242	8.535	H
M	0	0	1	1	8	4	-4	6	3	2	0	T
#	-15	-15	0	0	-80	-80	-80	0	-10	-10	LL	
13	15	15	50	50	80	80	80	80	10	10	UL	
	0.000	0.000	0.208	2.830	20.109	9.940	-10.062	6.000	8.484	5.652	0.001	C
	1.066	0.002	0.858	-0.304	3.070	0.558	-10.754	2.013	1.286	1.560	50.009	G
	-0.003	-0.002	-2.660	-1.398	-0.282	0.675	0.314	-3.004	0.759	0.497	1.139	H
M	0	0	1	1	0.6	2	3	3	3	2	0	T
#	-15	-15	0	0	-80	-80	-80	0	-10	-10	LL	
21	15	15	50	50	80	80	80	80	10	10	UL	
	0.040	0.015	0.951	0.472	-63.643	-4.760	19.382	6.077	3.091	2.029	0.264	C
	1.157	-0.474	1.750	-0.593	0.300	1.039	0.207	-0.926	1.439	0.739	1.139	G
	3.223	0.069	-0.737	4.040	4.531	2.391	0.424	-4.100	3.807	-1.639	0.140	H
M	0	0	1	1	2	2	6	5	3	2	0	T
#	-15	-15	0	0	-80	-80	-80	0	-10	-10	LL	
22	15	15	50	50	80	80	80	80	10	10	UL	
	-0.063	0.049	0.410	2.496	13.073	3.808	-0.537	9.940	3.914	2.707	0.315	C
	0.358	0.022	26.318	19.861	2.312	-0.045	0.207	39.566	4.312	2.308	8.821	G
	0.081	-0.014	-1.512	-1.405	-1.209	0.682	1.206	5.024	3.604	2.286	1.078	H
M	0	0	1	1	8	4	-4	6	3	2	0	T
#	-15	-15	0	0	-80	-80	-80	0	-10	-10	LL	
23	15	15	50	50	80	80	80	80	10	10	UL	
	0.000	0.000	43.724	10.289	-58.308	42.069	29.224	6.000	9.624	6.415	0.005	C
	0.252	-0.062	23.221	19.907	2.266	-0.162	3.957	41.494	3.272	1.905	31.973	G
	-0.002	0.017	0.126	0.089	2.240	-1.557	-1.105	-2.987	1.354	0.901	0.158	H
M	0	0	1	1	0.6	2	3	3	3	2	0	T
#	-15	-15	0	0	-80	-80	-80	0	-10	-10	LL	
31	15	15	50	50	80	80	80	80	10	10	UL	
	-0.982	-0.039	0.949	5.796	15.318	4.294	1.637	5.959	3.381	1.294	0.504	C
	-3.514	-0.069	20.062	20.115	6.366	0.900	0.865	40.511	2.509	-0.483	3.033	G
	-3.512	-0.042	-1.234	1.252	3.029	3.388	2.509	-3.977	2.847	-0.528	0.550	H
M	0	0	1	1	2	2	6	5	3	2	0	T
#	-15	-15	0	0	-80	-80	-80	0	-10	-10	LL	
32	15	15	50	50	80	80	80	80	10	10	UL	
	1.217	0.027	13.680	50.000	50.370	11.819	10.236	8.052	7.876	2.254	2.371	C
	-3.858	-0.119	24.807	25.118	-4.337	-0.057	1.849	42.345	3.419	-1.203	9.177	G
	-1.377	-0.045	-1.577	2.403	1.546	1.275	1.174	4.011	2.343	0.668	3.740	H

Table continued to next page

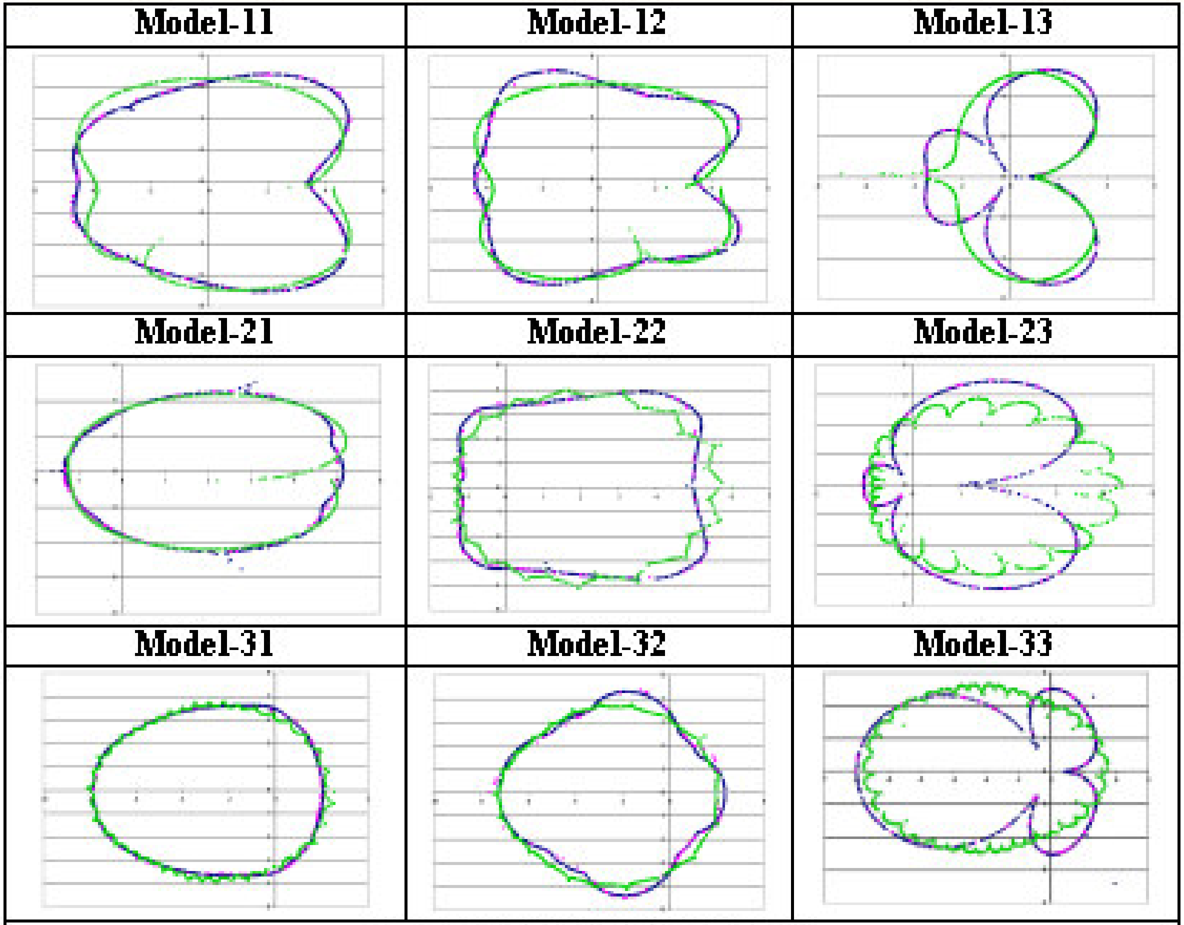
Table- A.1. as a continuation -- from the earlier page

	c_x	c_y	a	b	n_1	n_2	n_3	m	n_4	n_5	S^2	
M	0	0	1	1	8	4	-4	6	3	2	0	T
#	-15	-15	0	0	-80	-80	-80	0	-10	-10	LL	
33	15	15	50	50	80	80	80	80	10	10	UL	
	0.000	0.000	0.004	1.003	-78.492	-21.839	39.256	6.000	2.995	1.996	0.006	C
	-1.672	0.096	26.101	26.161	-1.528	0.158	0.061	39.434	2.059	0.271	25.234	G
	-0.064	-0.075	1.180	1.077	-0.313	1.346	1.038	3.017	1.840	1.163	3.884	H

M=Model; LL=Lower Limits; UL=Upper Limits; T=True Parameters; C and G are Estimated Parameters : C = SA (Classical SA) G =GSA (Generalized SA) with 1000 iterations; H =GSA (Generalized SA) with 10000 iterations

Figures.A.1. (Plots of Generated, CSA-estimated and GSA-estimated points)

[For GSA 1000 iterations are used]



Model-11 | Model-12 | Model-13

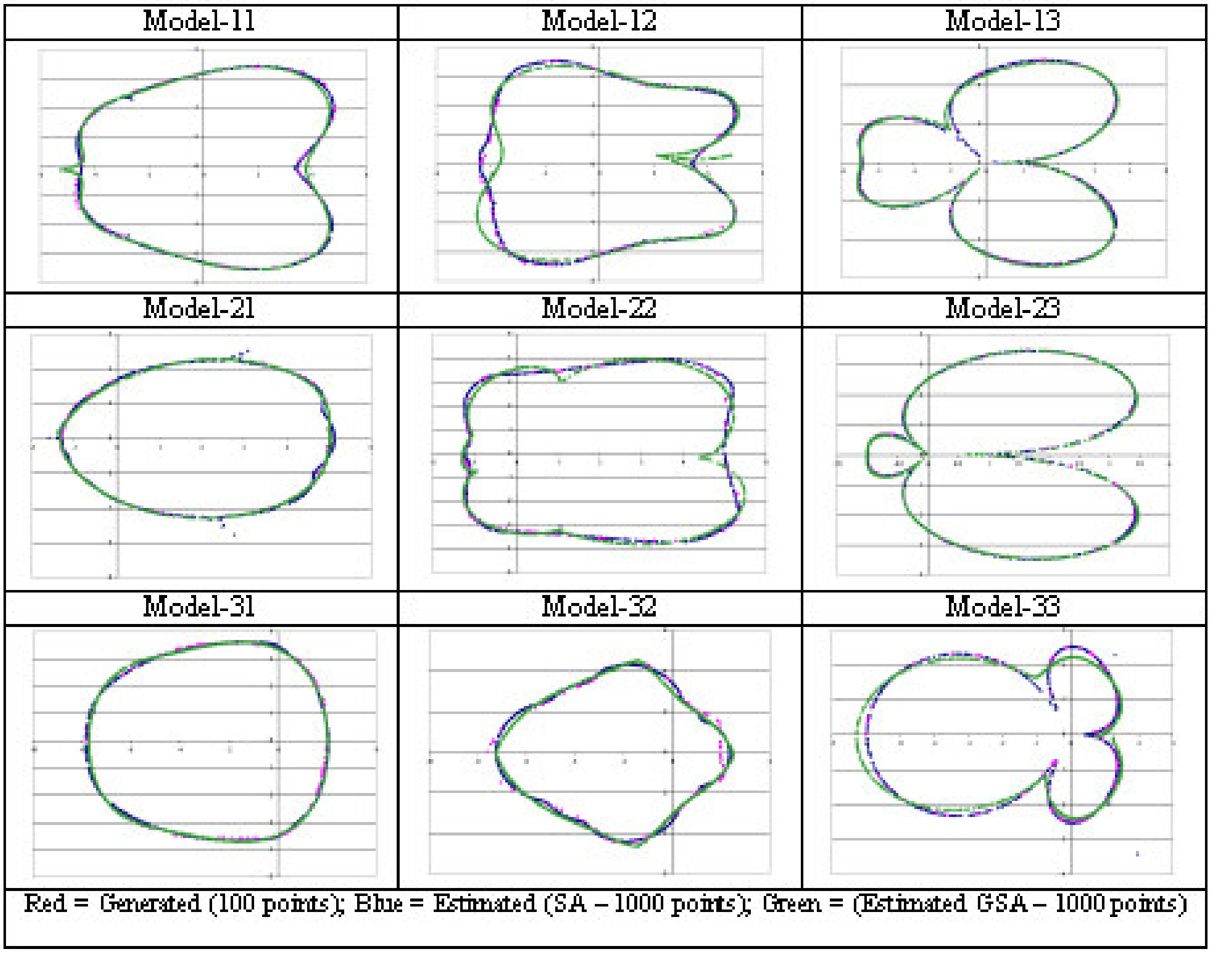
Model-21 | Model-22 | Model-23

Model-31 | Model-32 | Model-33

Red = Generated (100 points)

Blue = Estimated (SA 1000 points); Green = Estimated (GSA 1000 points)

Figures-A.2. (Plots of Generated, CSA-estimated and GSA-estimated points)
[For GSA 10,000 iterations are used]



Model-11

Model-12

Model-13

Model-21

Model-22

Model-23

Model-31

Model-32

Model-33

Red = Generated (100 points); Blue = Estimated (SA – 1000 points); Green = (Estimated GSA – 1000 points)

Author's Contact: mishrasknehu@hotmail.com Ph: 913642550102 (R), 913642723175 (O)