Testing MagicPlot 2.5 Application with NIST Datasets

Alexander Levantovsky, Magicplot Systems, LLC
Saint Petersburg, Russia
May 22, 2013

Introduction

American National Institute of Standards and Technology (NIST) has published a number of datasets for testing implementations of various math algorithms. MagicPlot 2.5 successfully passed those tests.

The reference datasets can be found on NIST page:
http://www.itl.nist.gov/div898/strd/general/dataarchive.html

1. Nonlinear Regression Test

Algorithm

MagicPlot uses its own implementation of widely known **Levenberg-Marquardt** nonlinear fitting algorithm optimized for multi-core processors. Fit function partial derivatives are computed numerically using central difference formula.

MagicPlot preferences:

- Iterations stop tolerance: 10^{-12}
- Maximal number of iterations: 10000
- 4-core processor was used

Results

The results are shown in the table below. MagicPlot results agree with certified values in first 5…11 digits. We also compared standard deviation of parameters.

From NIST tests description:

> *"Except in cases where the certified value is essentially zero (for example, as occurs for the three Lanczos problems), a good nonlinear least squares procedure should be able to duplicate the certified results to at least 4 or 5 digits".*

Some starting values for certain datasets (BoxBOD, MGH17, both start 1) result in numeric over/underflow in exp(); an error message was shown in such cases. Note that the certified values were achieved with 128-bit precision computing while MagicPlot and other modern software use 64-bit precision (`double` data type) which leads to overflow or underflow with these parameters.

One of the datasets (MGH10) needed extremely many iterations (25147).

Note that the number of iterations may vary for different implementations of the same algorithm, depending on internally used constants.

Test name, start number	Number of coincident decimal digits				Number of iterations	Formula, y(x)=...
	Values	Values Std. Dev.	Residual	Residual Std. Dev.		
Bennett5 Start 1	7	6	11	10	1316	b1 * (b2+x)^(-1/b3)
Bennett5 Start 2	7	6	11	10	1094	b1 * (b2+x)^(-1/b3)
BoxBOD Start 1	Failed: exp() overflow					b1*(1-exp(-b2*x))
BoxBOD Start 2	7	7	10	11	12	b1*(1-exp(-b2*x))
Chwirut1 Start 1	8	8	11	10	12	exp(-b1*x)/(b2+b3*x)
Chwirut1 Start 2	8	8	11	10	7	exp(-b1*x)/(b2+b3*x)
Chwirut2 Start 1	9	9	11	10	14	exp(-b1*x)/(b2+b3*x)
Chwirut2 Start 2	8	9	11	10	7	exp(-b1*x)/(b2+b3*x)
DanWood Start 1	9	9	11	11	7	b1*x^b2
DanWood Start 2	10	9	11	11	6	b1*x^b2
ENSO Start 1	5	6	11	12	38	b1 + b2*cos(2*pi*x/12) + b3*sin(2*pi*x/12) + b5*cos(2*pi*x/b4) + b6*sin(2*pi*x/b4) + b8*cos(2*pi*x/b7) + b9*sin(2*pi*x/b7)
ENSO Start 2	5	6	11	12	28	b1 + b2*cos(2*pi*x/12) + b3*sin(2*pi*x/12) + b5*cos(2*pi*x/b4) + b6*sin(2*pi*x/b4) + b8*cos(2*pi*x/b7) + b9*sin(2*pi*x/b7)
Eckerle4 Start 1	9	6	10	11	4973	(b1/b2) * exp(-0.5*((x-b3)/b2)^2)
Eckerle4 Start 2	9	6	10	11	9	(b1/b2) * exp(-0.5*((x-b3)/b2)^2)
Gauss1 Start 1	8	8	11	11	6	b1*exp(-b2*x) + b3*exp(-(x-b4)^2 / b5^2) + b6*exp(-(x-b7)^2 / b8^2)
Gauss1 Start 2	8	8	11	11	7	b1*exp(-b2*x) + b3*exp(-(x-b4)^2 / b5^2) + b6*exp(-(x-b7)^2 / b8^2)
Gauss2 Start 1	10	8	10	10	7	b1*exp(-b2*x) + b3*exp(-(x-b4)^2 / b5^2) + b6*exp(-(x-b7)^2 / b8^2)
Gauss2 Start 2	10	8	10	10	7	b1*exp(-b2*x) + b3*exp(-(x-b4)^2 / b5^2) + b6*exp(-(x-b7)^2 / b8^2)
Gauss3 Start 1	10	8	10	10	8	b1*exp(-b2*x) + b3*exp(-(x-b4)^2 / b5^2) + b6*exp(-(x-b7)^2 / b8^2)
Gauss3 Start 2	10	8	10	10	8	b1*exp(-b2*x) + b3*exp(-(x-b4)^2 / b5^2) + b6*exp(-(x-b7)^2 / b8^2)
Hahn1 Start 1	7	8	10	11	12	(b1+b2*x+b3*x^2+b4*x^3) / (1+b5*x+b6*x^2+b7*x^3)
Hahn1 Start 2	8	9	10	11	12	(b1+b2*x+b3*x^2+b4*x^3) / (1+b5*x+b6*x^2+b7*x^3)
Kirby2 Start 1	7	8	11	10	10	(b1 + b2*x + b3*x^2) / (1 + b4*x + b5*x^2)
Kirby2 Start 2	8	8	11	10	9	(b1 + b2*x + b3*x^2) / (1 + b4*x + b5*x^2)
Lanczos1 Start 1	10	3	2	3	324	b1*exp(-b2*x) + b3*exp(-b4*x) + b5*exp(-b6*x)
Lanczos1 Start 2	10	2	2	2	365	b1*exp(-b2*x) + b3*exp(-b4*x) + b5*exp(-b6*x)
Lanczos2 Start 1	8	7	9	10	325	b1*exp(-b2*x) + b3*exp(-b4*x) + b5*exp(-b6*x)
Lanczos2 Start 2	8	7	9	10	349	b1*exp(-b2*x) + b3*exp(-b4*x) + b5*exp(-b6*x)
Lanczos3 Start 1	7	7	10	11	342	b1*exp(-b2*x) + b3*exp(-b4*x) + b5*exp(-b6*x)
Lanczos3 Start 2	8	7	10	11	372	b1*exp(-b2*x) + b3*exp(-b4*x) + b5*exp(-b6*x)
MGH09 Start 1	6	6	11	11	338	b1*(x^2+x*b2) / (x^2+x*b3+b4)
MGH09 Start 2	6	6	11	11	31	b1*(x^2+x*b2) / (x^2+x*b3+b4)
MGH10 Start 1	10	7	11	11	25147	b1 * exp(b2/(x+b3))

MGH10 Start 2	10	7	11	11	516	b1 * exp(b2/(x+b3))
MGH17 Start 1	Failed: exp() overflow					b1 + b2*exp(-x*b4) + b3*exp(-x*b5)
MGH17 Start 2	9	8	11	11	41	b1 + b2*exp(-x*b4) + b3*exp(-x*b5)
Misra1a Start 1	11	9	10	10	50	b1*(1-exp(-b2*x))
Misra1a Start 2	10	10	10	10	7	b1*(1-exp(-b2*x))
Misra1b Start 1	9	8	11	11	31	b1 * (1-(1+b2*x/2)^(-2))
Misra1b Start 2	11	10	11	11	7	b1 * (1-(1+b2*x/2)^(-2))
Misra1c Start 1	10	10	11	11	7	b1 * (1-(1+2*b2*x)^(-.5))
Misra1c Start 2	10	9	11	11	7	b1 * (1-(1+2*b2*x)^(-.5))
Misra1d Start 1	11	10	11	11	8	b1*b2*x*((1+b2*x)^(-1))
Misra1d Start 2	11	10	11	11	7	b1*b2*x*((1+b2*x)^(-1))
Rat42 Start 1	9	9	11	10	13	b1 / (1+exp(b2-b3*x))
Rat42 Start 2	9	9	11	10	7	b1 / (1+exp(b2-b3*x))
Rat43 Start 1	8	8	11	11	24	b1 / ((1+exp(b2-b3*x))^(1/b4))
Rat43 Start 2	7	8	11	11	10	b1 / ((1+exp(b2-b3*x))^(1/b4))
Roszman1 Start 1	7	8	12	11	7	b1 - b2*x - atan(b3/(x-b4))/pi
Roszman1 Start 2	9	9	12	11	6	b1 - b2*x - atan(b3/(x-b4))/pi
Thurber Start 1	6	5	11	10	34	(b1 + b2*x + b3*x^2 + b4*x^3) / (1 + b5*x + b6*x^2 + b7*x^3)
Thurber Start 2	6	5	11	10	30	(b1 + b2*x + b3*x^2 + b4*x^3) / (1 + b5*x + b6*x^2 + b7*x^3)

2. Descriptive Statistics Test

We have also tested MagicPlot descriptive statistics tool with NIST dataset. MagicPlot results are shown in full precision, without trailing zeros.

Dataset name	Certified Mean	Certified Std.Dev.	MagicPlot Mean	MagicPlot Std. Dev.
PiDigits	4.53480000000000	2.86733906028871	4.5348	2.867339060288708
Lottery	518.958715596330	291.699727470969	518.9587155963303	291.6997274709691
Lew	-177.435000000000	277.332168044316	-177.435	277.3321680443161
Mavro	2.00185600000000	0.000429123454003053	2.001856	4.291234540030854E-4
Michelso	299.852400000000	0.0790105478190518	299.8524	0.07901054781905066
NumAcc1	10000002	1	1.0000002E7	1.0
NumAcc2	1.2	0.1	1.2	0.09999999999999998
NumAcc3	1000000.2	0.1	1000000.2000000001	0.1000000000349246
NumAcc4	10000000.2	0.1	1.0000000200000001E7	0.10000000055879354