SYMBOLIC REGRESSION

SYMBOLIC REGRESSION

- **In symbolic regression, we seek**
  - **Functional form of a good fit**
  - **All numerical parameters**
  - **Size and shape of the mathematical expression**

- **The fitness measure is usually error**
  - **Sum, over *N* fitness cases, of absolute (or squared) error**

- **Also Called**
  - **System identification**
  - **Non-parametric regression**
  - **"Black Box" problem**
  - **Model building**
  - **Empirical discovery**
  - **Forecasting / Prediction**

DISCOVERY OF TRIGONOMETRIC IDENTITIES

FOR COS 2X

$$\mathbf{Cos\ 2x = Cos^2\ x\ Sin^2\ x}$$
$$\mathbf{=1 - 2\ Sin^2\ x}$$

• Begin by creating DISCRETE set of fitness cases from the target function, Cos 2x.

• Then, do symbolic regression on the data

• Including Cosine in function set permits trivial results, such as ...
Cos 2*x* = `(COS (+ X X))` = Cos (*x* + *x*),
and less interesting results, such as
Cos 2*x* = `(COS (- (- 1 1) (+ X X)))` = Cos (-2*x*)

• However, omitting Cosine precludes discovery ...

$\mathbf{Cos}\ 2x = 2\ \mathbf{Cos}^2 x - 1$

TABLEAU FOR DISCOVERY OF TRIGONOMETRIC IDENTITIES

Objective:	**Find a new, different, and unobvious mathematical expression, in symbolic form, that equals a given mathematical expression, in symbolic form, for all values of its independent variable.**
Terminal set:	X, **the constant** 1.0.
Function set:	+, -, *, %, SIN.
Fitness cases:	**The 20 pairs (x_i, y_i) where the x_i are random points in the interval [0, 2 π] radians and where the y_i are the values of the given mathematical expression (Cos 2x_i).**
Raw fitness:	**The sum, taken over the 20 fitness cases, of the absolute value of the difference between y_i and the value produced by the S-expression for x_i.**

Standardized fitness:	**Same as raw fitness for this problem.**
Hits:	**Number of points where S-expression comes within 0.01 of the desired value.**
Wrapper:	**None.**
Parameters:	**M = 500. G = 51.**
Success predicate:	**An S-expression scores 20 hits.**

DISCOVERY OF TRIGONOMETRIC IDENTITIES FOR COS 2X

Best-of-Generation 13 of one run:

```
(- (- 1 (* (SIN X) (SIN X))) (*
(SIN X) (SIN X)))
```

Equivalent to...

1 - 2 Sin^2 x

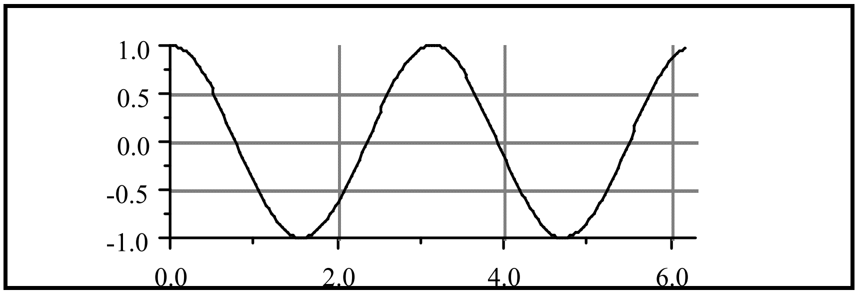
DISCOVERY OF TRIGONOMETRIC IDENTITIES FOR COS 2X

Best-of-Gen 30 of another run:

```
(SIN (- (- 2 (* X 2))
     (SIN (SIN (SIN (SIN (SIN
(SIN (* (SIN (SIN 1))

(SIN (SIN 1))

))))))))
```



DISCOVERY OF CONSTANT GENERATION

Sin [1] = 0.841

Sin [Sin [1]] = 0.746

[Sin [Sin[1]] * Sin [Sin [1]] = 0.528

[Sin [Sin [Sin[1]] * Sin [Sin [1]] = 0.504

[Sin [Sin [Sin [Sin[1]] * Sin [Sin [1]] = 0.483

[Sin [Sin [Sin [Sin [Sin[1]] * Sin [Sin [1]] = 0.464

DISCOVERY OF CONSTANT GENERATION

[Sin [Sin [Sin [Sin [Sin [Sin[1]] * Sin [Sin [1]] = 0.448

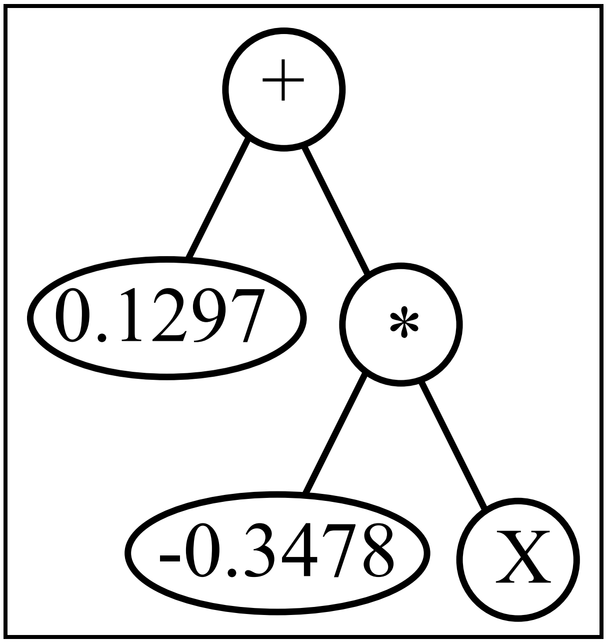
[Sin Sin [Sin [Sin [Sin [Sin [Sin[1]] * Sin [Sin [1]] = 0.433

Then
2 - Sin [Sin [Sin [Sin [Sin [Sin [Sin [Sin[1]] * Sin [Sin [1]]]]]]]] = 1.57

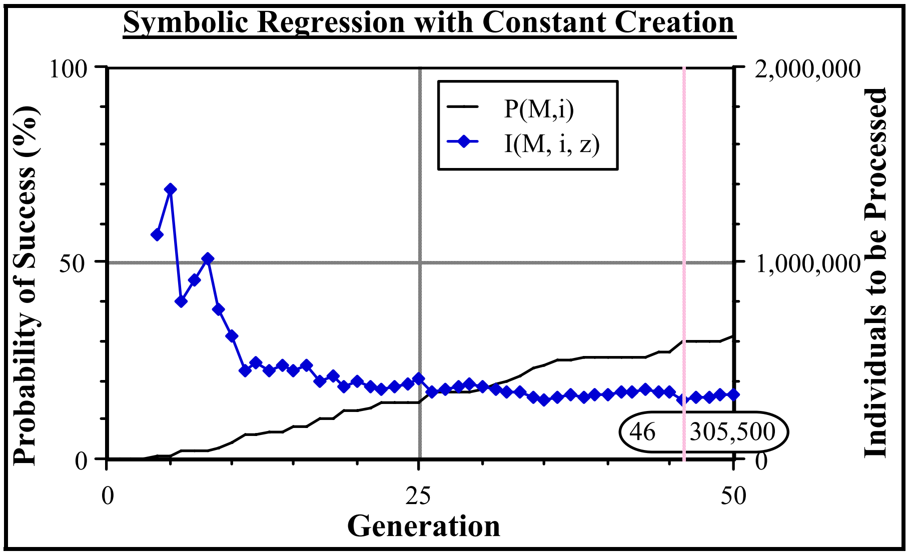
Thus,
Cos 2x = Sin (π /2 - 2x) is the identity.

"EPHEMERAL" RANDOM CONSTANTS

- **Terminal set T = {X, ℜ}**
- **Random constants ℜ range over specified range, such as -5.0 to +5.0**
- **In generation 0, each occurrence of ℜ is replaced by a SEPARATELY and INDEPENDENTLY chosen random constant**
- **The constant never change during the run**
- **The random constants combine with available functions to create new values**
- **In one-byte representation, 256-*N* constants**
- **Note: the choice of range matters!**



PERFORMANCE CURVES FOR THE SYMBOLIC REGRESSION PROBLEM WITH CONSTANT CREATION WITH 2.718X^2 + 3.1416X AS THE TARGET FUNCTION



A SECOND APPROACH TO RANDOM CONSTANTS

PERTURBABLE RANDOM CONSTANTS

- **The numerical parameter value is established by a single perturbable numerical value (coded by 30 bits in our system)**
- **The perturbable numerical values ARE CHANGED during the run (unlike the constant numerical terminals of the first approach).**
- **In the initial random generation, each perturbable numerical value is set, separately and independently, to a random value in a chosen range (e.g., between +5.0 and -5.0).**

PERTURBABLE RANDOM CONSTANTS — CONTINUED

- **In later generations, the perturbable numerical value may be perturbed by a relatively small amount determined probabilistically by a Gaussian probability distribution.**
  - **The existing to-be-perturbed value is the mean of the Gaussian distribution.**
  - **A relatively small preset parameter establishes the standard deviation of the Gaussian distribution. For example, the standard deviation of the Gaussian perturbation maybe 1.0**
- **This second approach has the advantage (over the first approach) of changing numerical parameter values by a relatively small amount and therefore searching the space of possible parameter values most thoroughly in the immediate neighbor of the value of the existing value (which is, because of Darwinian selection, is necessarily part of a relatively fit individual).**

PERTURBABLE RANDOM CONSTANTS — CONTINUED

- **These perturbations are implemented by a genetic operation for mutating the perturbable numerical values.**
- **It is also possible to perform a special crossover operation in which a copy of a perturbable numerical value is inserted in lieu of a chosen other perturbable numerical value.**
- **Our recent experience (albeit limited) is that this second approach, patterned after the Gaussian mutation operation used in evolution strategies and evolutionary programming, appears to work better than the first approach.**

A THIRD APPROACH TO RANDOM CONSTANTS

PERTURBABLE RANDOM CONSTANTS IN ARITHMETIC-PERFORMING SUBTREES

- **The third approach employs arithmetic-performing subtrees in conjunction with perturbable numerical values.**
- **This approach differs from the second approach in that a full subtree is used, instead of only a single perturbable numerical value.**
- **This approach is especially appropriate and advantageous when there are external global variables or when automatically defined functions, such as `ADF0`, and dummy variables (formal parameters), such as `ARG0`, are involved in establishing numerical parameter values for electrical components in circuits or establishing numerical parameter values for other functions.**

ECONOMETRIC EXCHANGE EQUATION

120 actual quarterly values (from 1959:1 to 1988:4) of:

• the annual rate for the United States Gross National Product in billions of 1982 dollars (called GNP82)

• the Gross National Product Deflator (normalized to 1.0 for 1982) (called GD),

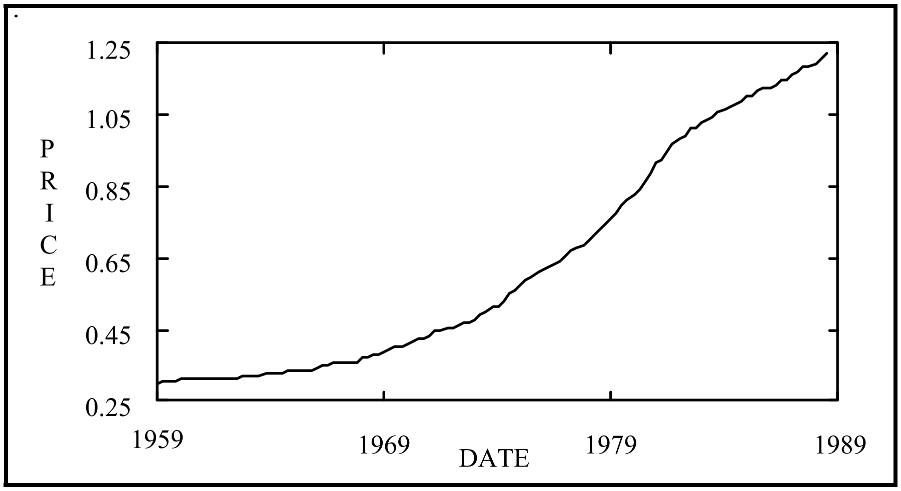
• the monthly values of the seasonally adjusted money stock M2 in billions of dollars, averaged for each quarter (called M2)

• the monthly interest rate yields of 3-month Treasury bills, averaged for each quarter (called FYGM3)

ECONOMETRIC EXCHANGE EQUATION

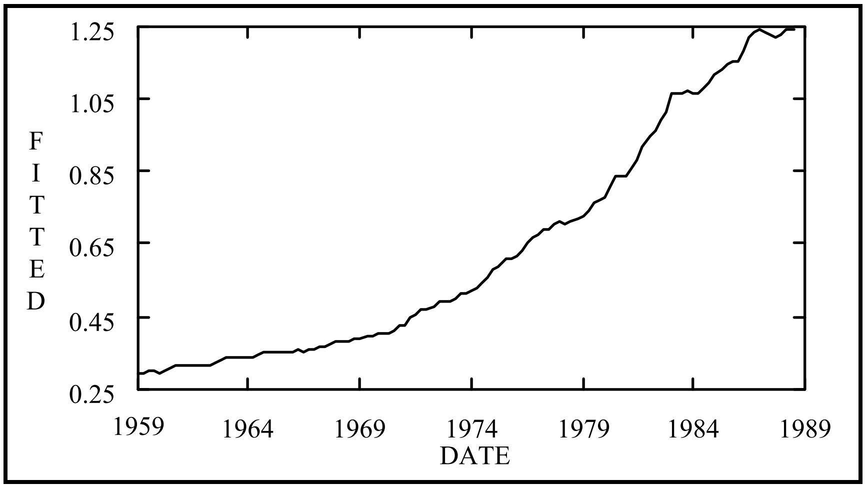
$$GD = \frac{(1.6527 * M2)}{GNP82}$$

GROSS NATIONAL PRODUCT DEFLATOR (GD) FROM 1959:1 TO 1988:4



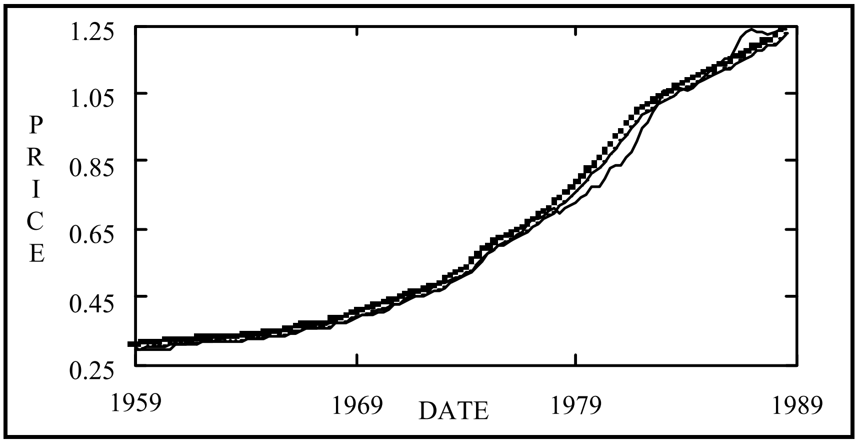
ECONOMETRIC EXCHANGE EQUATION

FITTED GD TIME SERIES



ECONOMETRIC EXCHANGE EQUATION

GROSS NATIONAL PRODUCT DEFLATOR GD OVERLAID WITH FITTED TIME SERIES



ECONOMETRIC EXCHANGE EQUATION

RESIDUALS BETWEEN GROSS NATIONAL PRODUCT DEFLATOR GD AND THE FITTED TIME SERIES

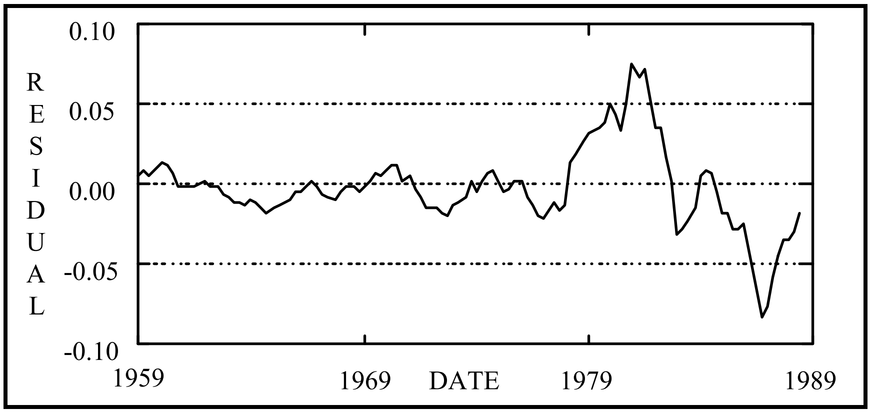


TABLEAU FOR EMPIRICAL DISCOVERY OF ECONOMETRIC EXCHANGE EQUATION

Objective:	**Find an econometric model for the price level, in symbolic form, that fits a given sample of 80 actual quarterly data points.**
Terminal set:	GNP82, FM2, FYGM3, ←, **where the ephemeral random floating-point constant ← ranges over the interval [–1.000, +1.000].**
Function set:	+, -, *, %, EXP, RLOG.
Fitness cases:	**The given sample of 80 quarterly data points.**
Raw fitness:	**The sum, taken over 80 quarters, of the squares of differences between the S-expression for the price level expressed in terms of the three independent variables and actual GD time series.**

Standardized fitness:	**Equals raw fitness for this problem.**
Hits:	**Number of fitness cases for which the S-expression comes within 1% of actual value of GD time series.**
Wrapper:	**None.**
Parameters:	***M* = 500. *G* = 51.**
Success predicate:	**An S-expression scores 80 hits.**

ECONOMETRIC EXCHANGE EQUATION

Best-of-run individual

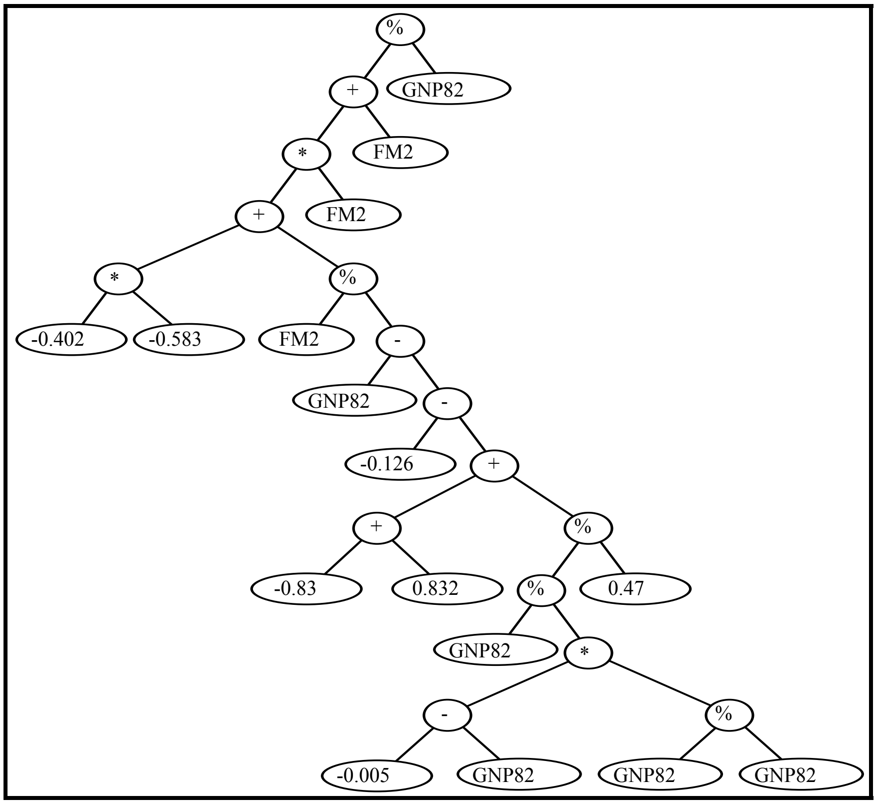
```
(% (+ (* (+ (* -0.402 -0.583)
(% FM2 (- GNP82 (- 0.126 (+ (+
-0.83 0.832) (% (% GNP82 (* (-
0.005 GNP82) (% GNP82
GNP82)))0.47))))))) FM2) FM2)
GNP82)
```

Sum of squared errors of 0.009272
Equivalent to...

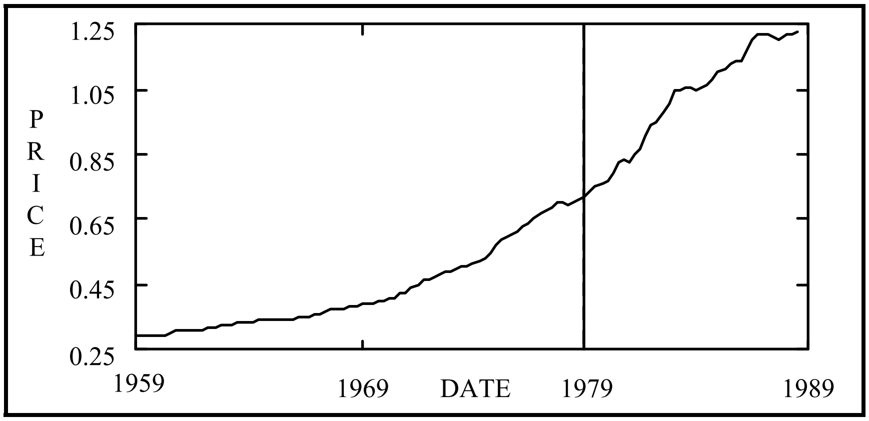
$$GD = \frac{(1.634 * M2)}{GNP82}$$

ECONOMETRIC EXCHANGE EQUATION

BEST-OF-RUN INDIVIDUAL FOR THE PRICE LEVEL USING THE FIRST TWO-THIRDS OF DATA



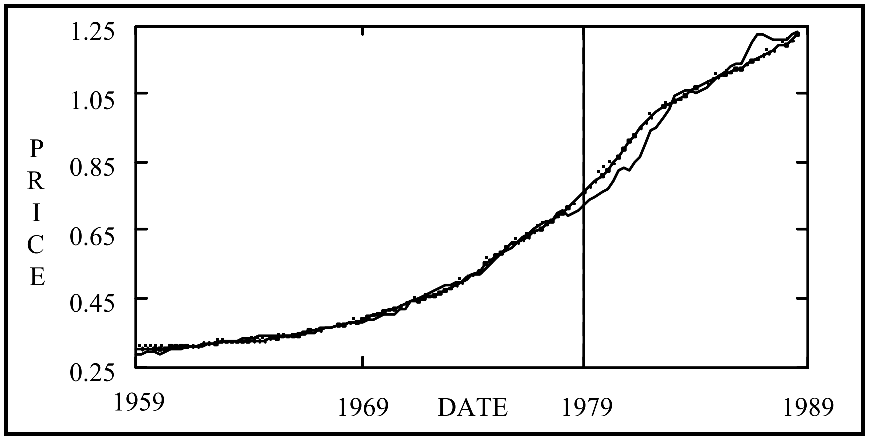
GRAPH OF BEST-OF-RUN INDIVIDUAL FOR THE PRICE LEVEL USING THE FIRST TWO-THIRDS OF THE DATA



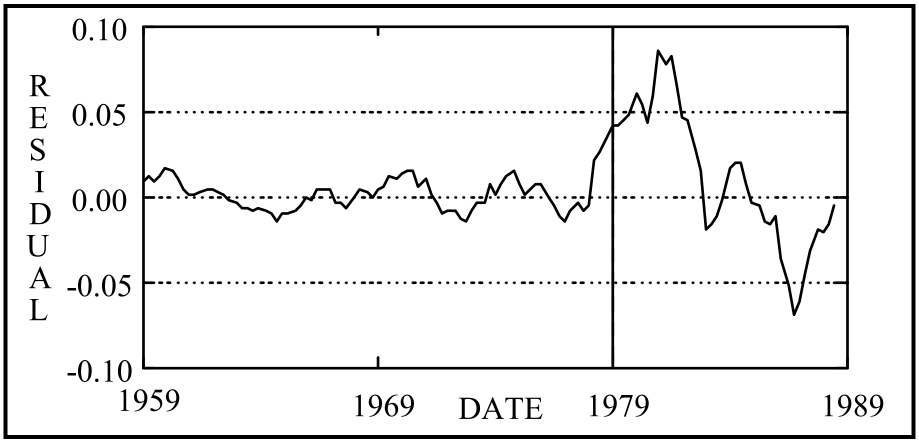
SQUARED ERRORS AND CORRELATIONS USING THE FIRST TWO-THIRDS OF THE DATA

Data range	**1- 120**	**1 - 80**	**81 - 120**
R^2	**0.993480**	**0.997949**	**0.990614**
Sum of squared errors	**0.075388**	**0.009272**	**0.066116**

GD WITH FITTED TIME SERIES USING THE FIRST TWO-THIRDS OF THE DATA



RESIDUALS



GRAPH OF BEST-OF-RUN INDIVIDUAL FOR THE PRICE LEVEL USING THE LAST TWO-THIRDS OF THE DATA

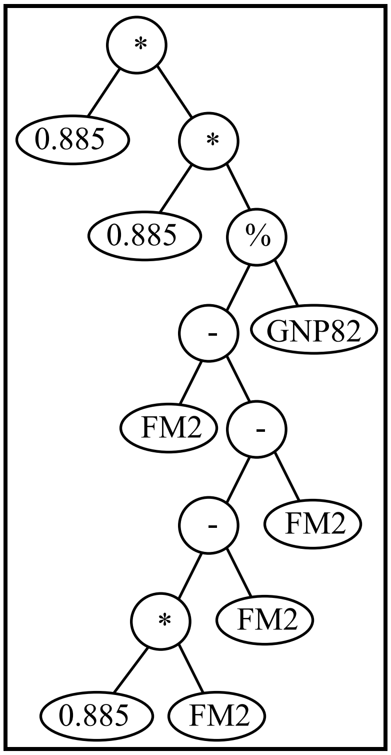
Best-of-run individual:

```
(* 0.885 (* 0.885 (% (- FM2 (-
(- (* 0.885 FM2) FM2) FM2))
GNP82)))
```

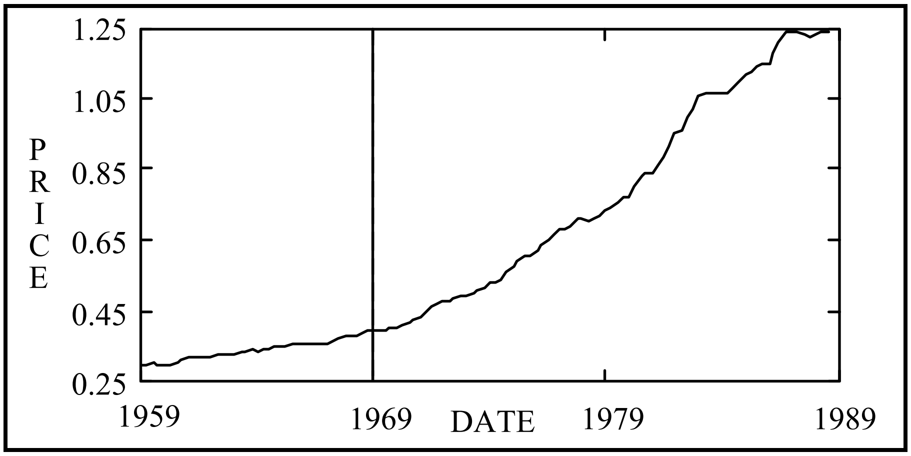
Equivalent to ...

$$GD = \frac{(1.6565 * M2)}{GNP82}$$

BEST-OF-RUN INDIVIDUAL FOR THE PRICE LEVEL USING THE LAST TWO-THIRDS OF THE DATA



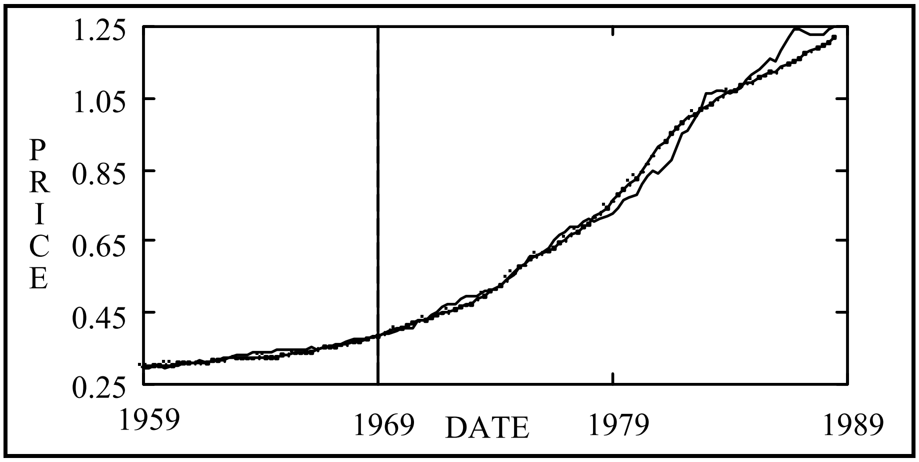
GRAPH OF BEST-OF-RUN INDIVIDUAL FOR THE PRICE LEVEL USING THE LAST TWO-THIRDS OF THE DATA



SQUARED ERRORS AND CORRELATIONS USING THE LAST TWO-THIRDS OF THE DATA

Data range	**1- 120**	**1 - 40**	**41 - 120**
R^2	**0.993130**	**0.999136**	**0.990262**
Sum of squared errors	**0.079473**	**0.003225**	**0.076247**

GD WITH FITTED TIME SERIES USING THE LAST TWO-THIRDS OF THE DATA



RESIDUALS

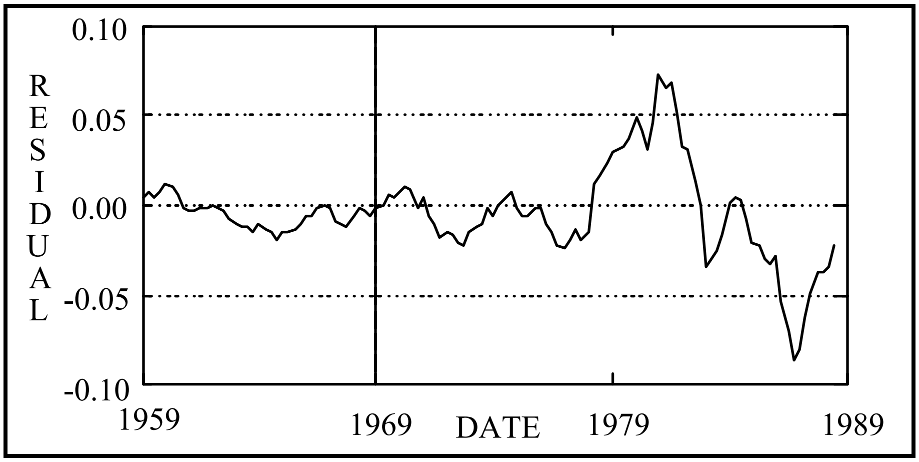


TABLEAU FOR EMPIRICAL DISCOVERY OF KEPLER'S THIRD LAW

Objective:	**Find a scientific law that fits a given sample of empirical data points.**
Terminal set:	`DIST.`
Function set:	`+, -, *, %, SRT, SIN, COS.`
Fitness cases:	**The given sample of nine data points for the nine planets.**
Raw fitness:	**The sum, over the nine fitness cases, of the absolute value of the differences between the value of the period *P* produced by the S-expression and the target value of *P* associated with that planet.**
Standardized fitness:	**Same as raw fitness for this problem.**
Hits:	**Number of fitness cases for which the value of the period *P* produced by the S-expression is within 1% of the target value of *P*.**

Wrapper:	**None.**
Population size:	**500.**
Termination:	**Maximum number of generations *G* = 51.** **Also terminate if an S-expression scores nine hits.**

KEPLER'S LAW - EMPIRICAL DISCOVERY

Gen 0 or early generation corresponds to 1608 ...

```
  (* DIST DIST)
```

KEPLER'S LAW - EMPIRICAL DISCOVERY

Later generations correspond to 1618 ...

```
(SRT (* DIST (* DIST DIST)))

(* DIST (SRT DIST))

(- (* DIST (SRT DIST)) (SIN
0.0))

(* DIST (+ (- DIST DIST) (+ (-
DIST DIST) (SRT DIST))))
```

SYMBOLIC INTEGRATION
COS X + 2X + 1

- **Given curve is Cos x + 2x + 1**
- **Begin by creating DISCRETE set of fitness cases from the curve-to-be-integrated (line 1)**
- **Evaluate the curve-to-be-integrated for each fitness case on line 2**
- **Numerically integrate, from 0 to x_i (line 3)**
- **Do symbolic regression on line 3 to get symbolic expression for the integral (line 4)**
- **Errors on line 5 should be small**
- **Integral in symbolic form is Sin x + x^2 + x**

1	*xi*	0.00	1.57	3.14	4.71	6.28
2	*y* =Cos *xi* + 2*xi* +1	2.00	4.14	6.28	10.42	14.57
3	$\boxed{{}^{xi}_{x=0}\ \text{Cosx+2x+1 dx}}$	0.00	4.82	13.01	26.13	45.76
4	Sin $x + x^2 + x$	0.00	5.04	13.01	25.92	45.76
5	Absolute error	0.00	0.21	1.78	0.21	0.00

TABLEAU FOR SYMBOLIC INTEGRATION

Objective:	**Find a function, in symbolic form, that is the integral of a curve presented either as a mathematical expression in symbolic form or as a given finite sample of points (x_i, y_i).**
Terminal set:	`X.`
Function set:	`+, -, *, %, SIN, COS, EXP, RLOG.`
Fitness cases:	**Sample of 50 data points (x_i, y_i).**
Raw fitness:	**The sum, taken over the 50 fitness cases, of the absolute value of the difference between the individual genetically produced function $f_j(x_i)$ at domain point x_i and the value of the numerical integral I(x_i).**
Standardize d fitness:	**Same as standardized fitness for this problem.**

Hits:	**Number of fitness cases coming within 0.01 of the target value I(x_i).**
Wrapper:	**None.**
Parameters:	***M* = 500. *G* = 51.**
Success predicate:	**An S-expression scores 50 hits.**

SYMBOLIC INTEGRATION

Best-of-run:

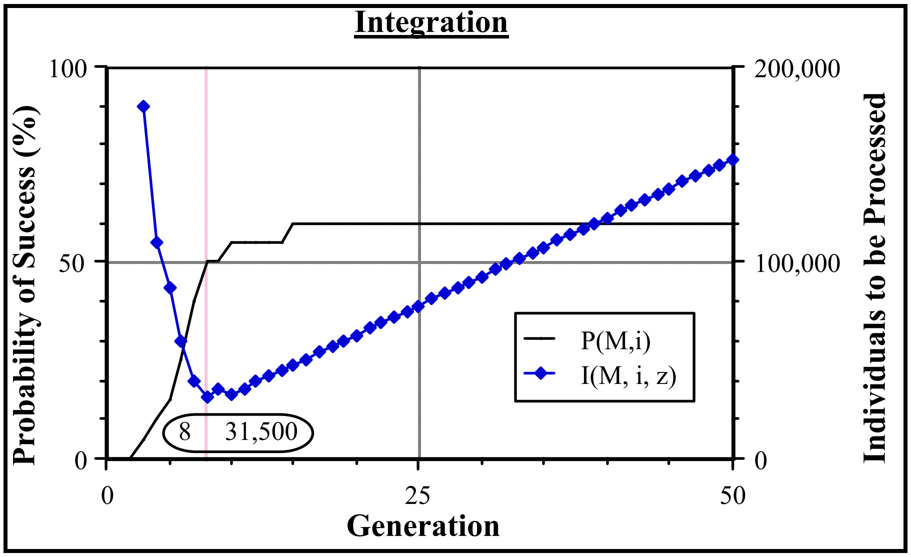
```
 (+ (+ (- (SIN X) (- X X)) X) (*
 X X))
```

Equivalent to ...

Sin x + x^2 + x

which is, in fact, the symbolic integral of Cos x + 2x + 1

PERFORMANCE CURVES FOR THE SYMBOLIC INTEGRATION PROBLEM



SYMBOLIC DIFFERENTIATION

The given curve is xe^x + Sin x + x,

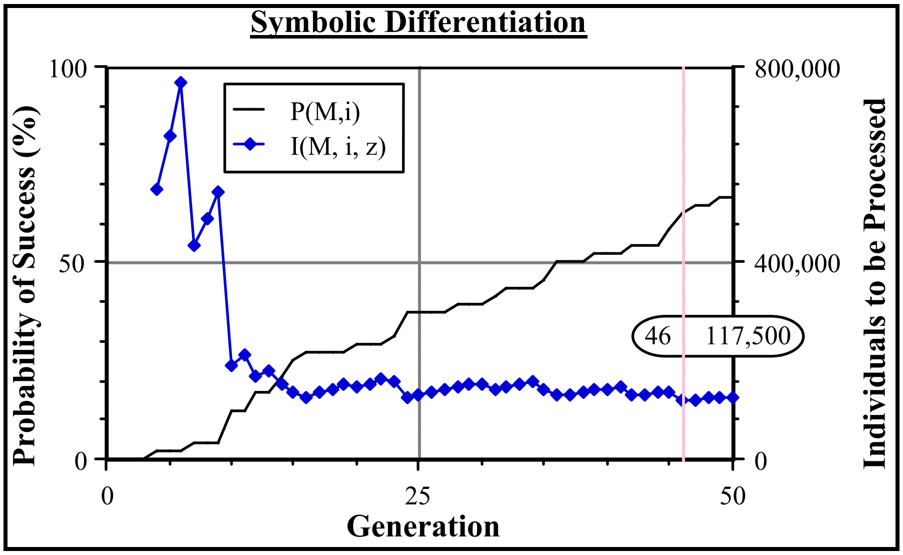
Its derivative in symbolic form is
$xe^x + e^x$ + Cos x + 1

Best-of-run – 199 hits – Standardized fitness of 2.52 (an average of 0.0126 per fitness case):

```
(+ (+ (+ (REXP X) (* (REXP X)
X)) (RCOS (% (* (* (% (- X X)
X) X) X) (+ (+ (REXP X) (* (+ X
X) X)) (* (+ (- X X) (REXP X))
(RLOG (REXP X)))))))) (RCOS X))
```

Equivalent to ...
$xe^x + e^x$ + Cos x + 1

PERFORMANCE CURVES FOR SYMBOLIC DIFFERENTIATION



EXAMPLE 1 – DIFFERENTIAL EQUATIONS

Differential equation

$$\frac{dy}{dx} + y \text{ Cos } x = 0$$

where $y_{initial}$ = 1.0 for $x_{initial}$ of 0.0.

Solution:

$$E^{-\text{SIN } X}.$$

TABLEAU FOR EXAMPLE 1 – DIFFERENTIAL EQUATIONS

Objective:	**Find a function, in symbolic form, which, when substituted into the given differential equation, satisfies the differential equation and its initial conditions.**
Terminal set:	x.
Function set:	+, -, *, %, SIN, COS, EXP, RLOG.
Fitness cases:	**Randomly selected sample of 200 values of the independent variable x_i in some interval of interest.**
Raw fitness:	**The sum, taken over the 200 fitness cases, of 75% of the absolute value of the value assumed by the genetically produced function $f_j(x_i)$ at domain point x_i plus 25% of 200 times of the absolute value of the difference between $f_j(x_{initial})$ and the given value $y_{initial}$.**

Standardized fitness:	**Same as raw fitness for this problem.**
Hits:	**Number of fitness cases for which the standardized fitness is less than 0.01.**
Wrapper:	**None.**
Parameters:	**M = 500. G = 51.**
Success predicate:	**An S-expression scores 198 or more hits.**

EXAMPLE 1 – DIFFERENTIAL EQUATIONS

Best-of-gen 0 – Raw fitness 58.09 – 3 hits:
$e^{1 - e^{x}}$

Best-of-gen 2 – Raw fitness 44.23 – 6 hits:
$e^{1 - e^{\mathrm{Sin}\ x}}$.

EXAMPLE 1 – DIFFERENTIAL EQUATIONS

Best-of-run – Raw fitness 0.057 – 199 hits

$e^{-\text{Sin x}}$

1	x_i	**0.0**	**0.25**	**0.50**	**0.75**	**1.0**
2	$y = e^{1 - e^{x}}$	**1.00**	**0.753**	**0.523**	**0.327**	**0.179**
3	**Cos** x_i	**1.00**	**0.969**	**0.876**	**0.732**	**0.540**
4	y *** Cos** x_i	**1.00**	**0.729**	**0.459**	**0.239**	**0.097**
5	$\frac{dy}{dx}$	**–0.989**	**–0.955**	**–0.851**	**–0.687**	**–0.592**
6	$\frac{dy}{dx} + y * \text{Cos } x$	**0.011**	–0.225	0.392	–0.447	**–0.495**

EXAMPLE 1 – DIFFERENTIAL EQUATIONS-GENERATION 2

1	x_i	**0.0**	**0.25**	**0.50**	**0.75**	**1.0**
2	$y = e^{1 - e^{\text{Sin}\, x}}$	**1.00**	**0.755**	**0.541**	**0.376**	**0.267**
3	**Cos** x_i	**1.00**	**0.969**	**0.878**	**0.732**	**0.540**
4	y *** Cos** x_i	**1.00**	**0.732**	**0.474**	**0.275**	**0.144**
5	$\frac{dy}{dx}$	**–0.979**	**–0.919**	**–0.758**	**–0.547**	**–0.437**
6	$\frac{dy}{dx} + y * \text{Cos}\, x$	**0.021**	–0.187	–0.283	–0.271	**–0.292**

EXAMPLE 1 – DIFFERENTIAL EQUATIONS-GENERATION 6

1	x_i	**0.0**	**0.25**	**0.50**	**0.75**	**1.0**
2	$y = e^{-\text{Sin } x}$	**1.0**	**0.781**	**0.619**	**0.506**	**0.431**
3	**Cos** x_i	**1.0**	**0.969**	**0.878**	**0.732**	**0.540**
4	y * **Cos** x_i	**1.0**	**0.757**	**0.543**	**0.370**	**0.233**
5	$\frac{dy}{dx}$	**–0.877**	**–0.762**	**–0.550**	**–0.376**	**–0.299**
6	$\frac{dy}{dx} + y * \text{Cos } x$	**0.123**	–0.005	–0.007	–0.006	**–0.067**

EXAMPLE 2 – DIFFERENTIAL EQUATIONS

$$\frac{dy}{dx} - 2y + 4x = 0$$

with an initial condition such that $y_{initial} = 4$ when $x_{initial} = 1$.

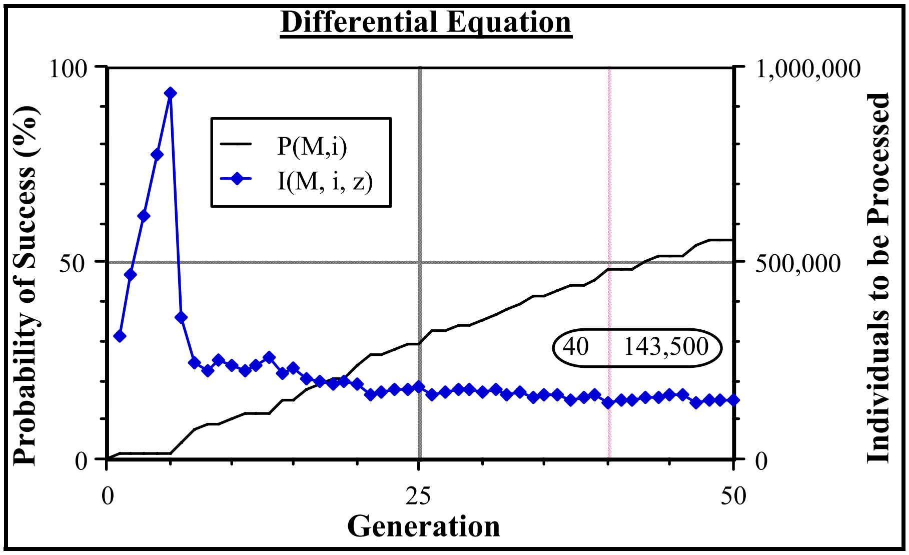
Best-of-run of generation 28:

```
(+ (* (EXP (- X 1)) (EXP (- X
1))) (+ (+ X X) 1))
```

Equivalent to ...

$$e^{-2}e^{2x} + 2x + 1$$

PERFORMANCE CURVES FOR EXAMPLE 2 – DIFFERENTIAL EQUATIONS



INTEGRAL EQUATIONS

Integral equation is

$$y(t) - 1 + 2 \int_{r=0}^{r=t} \mathrm{Cos}(t - r)\, y(r)\, dr = 0$$

Best-of-run solution:

$$y(t) = 1 - 2te^{-t}$$

INVERSE FUNCTIONS

Given the pairs (x_i, y_i) where $y_i = 2\ |x_i$.
{(9,6), (16,8), (25,10), (36,12), (2.25, 3.0), ...}

Interchange the pairs to create...
{(6,9), (8,16), (10,25), (12,36), (3.0, 2.25), ...}
so that now $y_i = \left(\frac{xi}{2}\right)^{2}$

Best-of-run solution for inverting the function $yi = 2\ |x_i$ **is**

$$y_i = \left(\frac{xi}{2}\right)^{2}$$

INVERSE FUNCTIONS

GUDERMANNIAN FUNCTION

$$2\ \mathrm{Tan}^{-1}(e^{x}) - \frac{\pi}{2}$$

INVERSE GUDERMANNIAN FUNCTION

$$\ln\ (\mathrm{Sec}\ x + \mathrm{Tan}\ x) = \ln\ \left(\frac{1}{\mathrm{Cos}\ x} + \frac{\mathrm{Sin}\ x}{\mathrm{Cos}\ x}\right)$$

T = {x, ←}

F = {+, -, *, %, SIN, COS, EXP, RLOG}

- **50 randomly chosen values of the independent variable over the range [–4.0, +4.0]**

INVERSE FUNCTIONS – GUDERMANNIAN FUNCTION

Best-of-generation 32 – Error averages less than 0.005 per fitness case:

```
(+ (- (% (RLOG (COS X)) (* (RLOG
0.48800004)
                            (* (+ (- X X)
(COS -0.8))
                               X)))
      (- (COS -0.8) (COS -0.8)))
   (* (COS (- (COS (COS (+ (RLOG X)
                            (RLOG (COS
X)))))
                (RLOG X)))
      (* (COS (- (COS -0.8) (RLOG X)))
         (* (- (% (RLOG (COS X))
                   (* (RLOG 0.48800004)
                      (* (+ (- X X) (COS -
0.8)) X)))
                (SIN X))
            (RLOG (COS (RLOG X)))))))
```

SEQUENCE INDUCTION

Sequence of integers:

**1, 15, 129, 547,
1593, 3711, 7465, 13539,
22737, 35983, 54321, 78915, 111049, 152127,
203673, 267331,
344865, 438159, 549217,680163, ...**

TABLEAU FOR SEQUENCE INDUCTION

Objective:	**Find a mathematical expression for a given finite sample of a sequence where the target sequence is $5j^4 + 4j^3 + 3j^2 + 2j + 1$.**
Terminal set:	**Sequence index J and ←, where the ephemeral random constant ← ranges over the integers 0, 1, 2, and 3.**
Function set:	+, -, *.
Fitness cases:	**First 20 elements of the sequence.**
Raw fitness:	**The sum, taken over the 20 fitness cases, of the absolute value of the difference between the value produced by the S-expression for sequence position J and the actual value of the target sequence for position J.**

Standardized fitness:	**Same as raw fitness for this problem.**
Hits:	**Number of fitness cases for which the value produced by the S-expression for sequence position J exactly matches the actual value of the target sequence for position J.**
Wrapper:	**None.**
Parameters:	***M* = 500. *G* = 51.**
Success predicate:	**An S-expression scores 20 hits.**

SEQUENCE INDUCTION

Best-of-generation 42:

```
(+ (+ (- (* (* 0 1) (- (* 3 J)
(+ (* 0 1) J))) 2) (* (* (* 2
J) (+ 1 J)) (* (+ J J) (- J
2)))) (- (- (+ 2 0) (* (* 1 J)
(- (- (- (+ (- (* 2 J) (+ 2 0))
(- J 3)) (- J 1)) (* (* 3 J) (+
J 1))) (- (- (+ J J) (* (- (-
(+ J (+ 0 J)) (- J 2)) (* (* 3
J) (+ J 1))) 3)) (* (- J 2) (-
2 J)))))) (* (- (+ 2 J) (* J
2)) (* (* J J) (- J 3))))))
```

Equivalent to ...

$5j^4 + 4j^3 + 3j^2 + 2j \underline{\pm 0}$

SEQUENCE INDUCTION

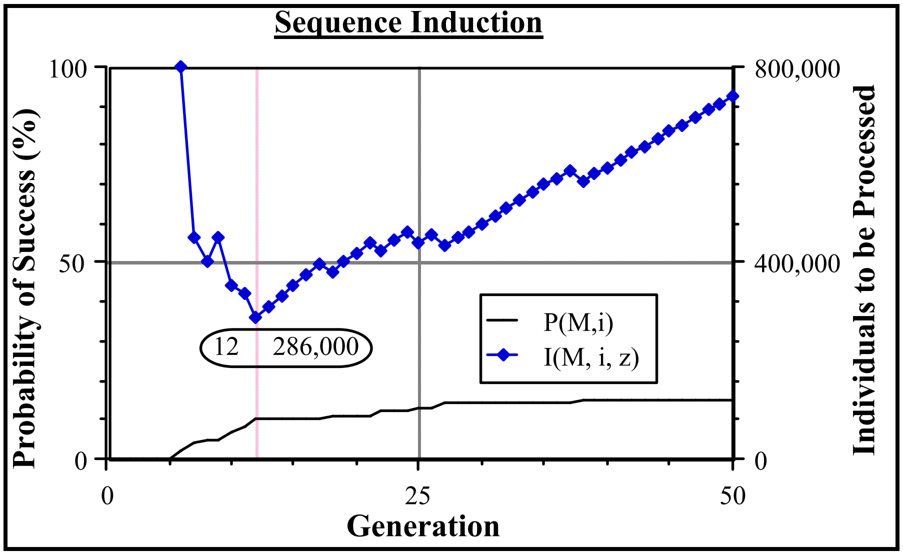
Best-of-run for generation 43:

```
(+ (+ (- (* (* 0 1) (- (* 3 J)
(+ (* 0 1) J))) 2) (* (* (* 2
J) (+ 1 J)) (* (+ J J) (- J
2)))) (- (- (+ 3 0) (* (* 1 J)
(- (- (- (+ (- (* 2 J) (+ 2 0))
(- J 3)) (- J 1)) (* (* 3 J) (+
J 1))) (- (- (+ J J) (* (- (-
(+ J (+ 0 J)) (- J 2)) (* (* 3
J) (+ J 1))) 3)) (* (- J 2) (-
2 J)))))) (* (- (+ 2 J) (* J
2)) (* (* J J) (- J 3))))))
```

Equivalent to ...

$5j^4 + 4j^3 + 3j^2 + 2j \underline{+1}$

PERFORMANCE CURVES FOR THE SEQUENCE INDUCTION PROBLEM WITH $5J^4 + 4J^3 + 3J^2 + 2J + 1$ AS THE TARGET FUNCTION



TARGET IMAGE FOR PROBLEM OF PROGRAMMATIC IMAGE COMPRESSION PRODUCED BY THE EXPRESSION $3X^2 + 2Y^2 - 0.85$

WRAPPER (OUTPUT INTERFACE) FOR PROGRAMMATIC IMAGE COMPRESSION

Wrapper maps return value of program into [–1.0, +1.0] and thence into the desired range of 128 integral color values from 0 to 127:

```
(* 64 (+ 1 (MAX -1.0 (MIN 1.0
S-EXPRESSION))))
```

TABLEAU FOR PROGRAMMATIC IMAGE COMPRESSION

Objective:	**Find a S-expression that returns the color value for each pixel in a two-dimensional image.**
Terminal set:	**X,Y, ← , where the ephemeral random floating-point constant ← ranges over the interval [–1.0, +1.0].**
Function set:	**+, -, *, %.**
Fitness cases:	**Two-dimensional array of 900 pixels.**
Raw fitness:	**The sum, taken over the 900 fitness cases, of the absolute value of the difference between the color value produced by the S-expression for position (X, Y) and the color value of the target image for position (X,Y).**
Standardized fitness:	**Same as raw fitness for this problem.**

Hits:	**Number of fitness cases for which value of the wrapperized S-expression comes within 6 color values (out of 128) of correct value.**
Wrapper:	**Converts arbitrary floating-point number into one of the 128 color values.**
Parameters:	***M* = 2,000 (with over-selection). *G* = 51.**
Success predicate:	**S-expression scores 900 hits.**

BEST-OF-GENERATION INDIVIDUAL FROM GENERATION 0 FOR PROGRAMMATIC IMAGE COMPRESSION

Best-of-generation 0 – Fitness of 260.9 (over 900 fitness cases):

```
(* (* (- (* (% Y X) X) (% (* Y
Y) (+ 0.0458984 -0.106705))) X)
X)
```

BEST-OF-GENERATION INDIVIDUAL FROM GENERATION 6 FOR PROGRAMMATIC IMAGE COMPRESSION

Best-of-generation 6 – Fitness of 18.93 (over 900 fitness cases):

```
(+ (+ (+ (* (+ (* X X) (* Y Y))
(% Y Y)) (+ (* Y Y) -0.8116))
(+ 0.0458984-0.106705)) (* (% X
0.51979) X))
```

Equivalent to the expression...

```
(+ (* 2.9239 X X) (* 2 Y Y) -
0.8724)
```