Local Evolutionary Search Enhancement by Random Memorizing

Hans-Michael Voigt and Jan Matti Lange
GFaI - Center for Applied Computer Science, Berlin, Germany
{voigt,jl}@gfai.de

Abstract— **For the calibration of laser induced plasma spectrometers robust and efficient local search methods are required. Therefore, several local optimizers from nonlinear optimization, random search and evolutionary computation are compared. It is shown that evolutionary algorithms are superior with respect to reliability and efficiency. To enhance the local search of an evolutionary algorithm a new method of random memorizing is introduced. This method is applied to one of the most simple evolutionary algorithm, the (1+1)-Evolution Strategy. It leads to a substantial gain in efficiency for a reliable local search. Finally, laser induced plasma spectroscopy and the calibration of a real example are scetched.**

Keywords— **Evolutionary Computation, Local Search, Robust Search, Efficient Search, Calibration, Laser Spectrometry.**

I. Introduction and Motivation

Laser induced plasma spectrometry is one of the latest developments in measurement technologies. It is very precise, can be applied to multi-element analysis without any preparation, and can be used especially in process measurement such as glass and steel processing.

Since laser spectrometry is very precise the calibration of such devices has to be also very precise. This is a difficult task.

The calibration accuracy depends on the time dependent modeling of the atomic light emission of a mircoplasma. The calibration model is usually nonlinear because of different physical effects like re-absorption and light scattering.

It equally well depends on the precise adaptation of the calibration model parameters to the calibration samples.

This problem is ill-posed. Regularization has to be done due to facts from laser physics. Furthermore, from experience, it is ill-conditioned with condition indices $\lambda_{max}/\lambda_{min} > 10^6$. It is also high dimensional. For the calibration of 60 elements 120 to 180 parameters have to be adapted.

For the calibration we designed a hierarchical adaptation procedure. At higher levels the search space will be constrained by evidences from plasma physics. Coarse search procedures are applied. At the lowest level a fine tuning of the model parameters is done.

It is outside the scope of this paper to describe the whole procedure. Here we will explain to some extent the lowest level of calibration parameter fine tuning.

For this task the goal was to find an efficient and robust local search algorithm which does not use analytic gradient information.

There was made an extensive exploration of the available algorithms. A comparison is given in section II.

For the fine tuning we apply the simplest evolutionary algorithm, the (1+1)-Evolution Strategy – but with a memory. This memory is accessed at random to find search direction based on former good solutions. By a beam search this memorized directions will be exploited.

In section III we present a conceptual algorithm for the enhancement of evolutionary algorithms with global step size adaptation by random memorizing. For an implementation with an (1+1)-Evolution Strategy results for difficult unimodal test functions are presented and compared.

Finally, in section IV the application to the calibration of laser spectrometers is described in more detail.

II. Searching for a Robust and Efficient Local Optimizer

The assessment of the problem solving capacity of different local optimizers[1] was done using the well known Rosenbrock function f_R (see Table IX) for $n = 2, ..., 100$ variables. The initial point was set to $x_i = 0, i = 1, ..., n$. The number of function evaluations was counted until the function value $f_{stop} \leq 10^{-9}$ was reached or a maximum number of generations had passed.

The results are summarized in the following tables where MEAN is the average number of function evaluations to reach $f_{stop} \leq 10^{-9}$, SD the corresponding standard deviation, SUCC the number of successful runs related to the total number of runs (succ/total), and BF gives the best function value reached for nonconverging runs.

A natural first step in looking for good local optimizers is to consult available program packages including procedures for mathematical programming or nonlinear optimization (e.g. [14]). Among the recommended procedures are the simplex method [10] and different variants of second order optimization methods based on

[1] The results of Tables I to V were kindly provided by Dirk Schlierkamp-Voosen.

0-7803-4869-9/98 $10.00©1998 IEEE

conjugate directions [12] or quasi Newton procedures [11].

The results of the application of these methods to the minimization of function f_R are given in Tables I to III.

TABLE I
Results for the simplex method of Nelder and Mead

n	MEAN	SD	SUCC	BF
2	1.690E+02	0.000E+00	0/1	4.384E-01
3	2.030E+02	0.000E+00	0/1	1.603E+00
10	3.382E+03	0.000E+00	1/1	
20	1.009E+05	0.000E+00	1/1	
30	1.705E+05	0.000E+00	0/1	1.535E+01
100	2.000E+05	0.000E+00	0/1	8.635E+01

As it can be seen at once non of these methods is reliable enough to locate the optimum with the desired accuracy for dimensions up to 100 variables. Similar observations have been made already for other functions in [3].

In a next step a random search technique [13] and the dynamic hill-climber [7] were analyzed. The results are given in Tables IV and V. The effort to reach the optimum with the random search technique for a very moderate dimension of $n = 10$ was much higher than with the conjugate direction method.

And even the dynamic hill-climber, though very reliable for dimensions up to $n = 30$, could not locate the optimum for $n = 100$.

Therefore we looked for alternatives which may be offered by evolutionary algorithms.

As has been shown [8] [9] the Breeder Genetic Algorithm with fuzzy gene pool recombination, the utilization of covariances and generalized elitist selection is a very reliable search method. The same is true for the $(1, \lambda)$-Evolution Strategy with covariance matrix adaptation [5]. Unfortunately, both methods require to solve complete eigenvalue problems. This operation is of order $\mathcal{O}(n^3)$ thus increasing the computational overhead considerably for high-dimensional problems.

TABLE II
Results for the conjugate direction method of Powell

n	MEAN	SD	SUCC	BF
2	4.350E+02	0.000E+00	1/1	
3	6.380E+02	0.000E+00	1/1	
10	5.975E+03	0.000E+00	1/1	
20	1.907E+04	0.000E+00	1/1	
30	4.278E+04	0.000E+00	1/1	
50	6.248E+07	0.000E+00	1/1	
60	2.098E+05	0.000E+00	0/1	4.766E-09
70	3.748E+05	0.000E+00	0/1	3.892E-09
80	3.569E+05	0.000E+00	0/1	4.551E-09
100	5.507E+05	0.000E+00	0/1	

TABLE III
Results for the Quasi-Newton variable metric method without derivatives of Stewart

n	MEAN	SD	SUCC	BF
2	7.350E+02	0.000E+00	0/1	2.553E-02
3	2.985E+03	0.000E+00	0/1	7.549E-01
10	4:820E+02	0.000E+00	0/1	8.713E+00
20	1.580E+02	0.000E+00	0/1	1.880E+01
30	2.170E+02	0.000E+00	0/1	2.870E+01
100	6.370E+02	0.000E+00	0/1	9.799E+01

TABLE IV
Results for the random search technique of Solis and Wets

n	MEAN	SD	SUCC
2	3.780E+03	0.000E+00	1/1
3	1.598E+04	0.000E+00	1/1
10	2.770E+05	0.000E+00	1/1

The best results came up with the $(1, \lambda)$-Evolution Strategy with the generating set adaptation [6]. The results are given in Table VI. This procedure was very robust with in general less computational effort compared to the other methods. But the effort was still too high for problems having more than 100 variables.

III. Adding Random Memorizing to Evolutionary Algorithms

What are the lessons to be learned from the previous experiments?

First of all, good local search techniques do have a memory – either a sequential one like the conjugate direction method [12], the quasi Newton method [11], the Evolution Strategies with generating set adaptation [6] or with covariance matrix adaptation [5]– or a parallel one like the Breeder Genetic Algorithm with fuzzy gene pool recombination and covariance utilization [9].

TABLE V
Results for the dynamic hillclimber of Yuret and de la Maza

n	MEAN	SD	SUCC	BF
2	3.910E+02	0.000E+00	30/30	
3	1.998E+03	0.000E+00	30/30	
10	5.866E+04	0.000E+00	30/30	
20	1.418E+05	0.000E+00	30/30	
30	2.345E+05	0.000E+00	30/30	
100	5.827E+03	0.000E+00	0/30	9.698E+00

TABLE VI

RESULTS FOR THE EVOLUTION STRATEGY WITH THE GENERATING SET ADAPTATION OF HANSEN, OSTERMEIER, AND GAWELCZYK

n	MEAN	SD	SUCC
2	5.043E+02	1.237E+02	5/5
3	1.025E+03	1.966E+02	5/5
10	1.027E+04	7.875E+02	5/5
20	4.398E+04	3.627E+03	5/5
30	1.129E+05	6.576E+02	5/5

Having a closer look at all methods which use a sequential memory reveals that it is used in general in a rather deterministic or derandomized way for the construction of new search directions.

On the other hand there are already preliminary investigations into direction mutations [2] without using a memory. But the results are unsatisfactory.

Therefore it was decided to analyze the impact of randomly accessing a sequential memory for the determination of new search directions.

A. Conceptual Local Evolutionary Algorithm with Random Memorizing

The basic idea is the following: Use any Evolutionary Algorithm (EA) which has a global step size control. Add a sequential memory to the Evolutionary Algorithm were previous best solutions in search space are stored up to a certain depth. Determine by means of the EA a better solution. Take a solution from memory at random and compute from the solution of the EA and the randomly memorized solution a search direction. Follow this search direction by increasing multiples of the global step size as long as better solutions will be found. Then go ahead with the EA, etc.

This procedure is formalized in Table VII.

Obviously, there are two parameters concerning the memory access. That is the memory depth d_M and the beam search factor b. Furthermore the type of the probability density distribution for the random access of the sequential memory is important. For simplicity we used a uniform distribution. Other distributions may be more appropriate.

B. The Enhanced $(1+1)$*-Evolution Strategy*

To verify the outlined algorithm we instantiated it with the most simple EA, the $(1+1)$-Evolution Strategy. A thorough theoretical analysis of the $(1+1)$-Evolution Strategy is given in [1] [4]. The parameters of the EA were set as recommended there. The memory depth was set rather high to $d_m = 2 \cdot n ... n^3$ and the beam search factor to $b = 2$.

TABLE VII

OUTLINE OF THE LOCAL EVOLUTIONARY SEARCH WITH RANDOM MEMORIZING (LESRM)

```
set generation counter $g := 0$;
set memory counter $c := 0$;
set beam search factor $b$;
set maximal number of generations $g_{max}$;
set memory depth $d_m$;
set initial values of variables $x_g$;
set initial global step size $d_g$;
set termination criterion $\epsilon$;
{ set initial memory content $m_i$,
   $i = 0, ..., d_m - 1$;
}
compute fitness $f := f(x_g)$;
do
    do
        $g := g + 1$;
        { apply the EA for this generation
            $f_g, x_g, d_g := EA(x_{g-1}, d_{g-1})$;
        }
    until $f_g < f_{g-1}$;
    update memory $m_{(c \bmod d_m)} := x_g$;
    { select a memory content $m_r$
       with $r \in \{0, ..., d_m\}, r \neq c \bmod d_m$
       randomly drawn with uniform
       probability $1/(d_m - 1)$;
    }
    { compute the direction from
       $s := (x_g - m_r)/||(x_g - m_r)||$;
    }
    set beam search counter $c_b = 0$;
    set actual beam search factor $b_a := 1$;
    set $x_{c_b} := x_g$;
    do
        $c_b := c_b + 1$;
        $b_a := b_a \cdot b$;
        { compute new beam point
            $x_{c_b} := x_{c_b - 1} + s \cdot b_a \cdot d_g$;
        }
    while $f(x_{c_b}) < f(x_{c_b - 1})$;
    update memory $m_{(c \bmod d_m)} := x_{c_b}$;
    $c := c + 1$;
    $x_g := x_{c_b}$;
until $(f(x_{c_b}) < \epsilon)$   or   $(g \geq g_{max})$;
```

A first experiment was made for Rosenbrock's function with $n = 20$ and the initial point $x_i = 0, i = 1, ..., n$ resulting in a real surprise. This experiment is shown in Figure 1. The $(1+1)$-Evolution Strategy with random memorizing is about four times faster then the $(1, \lambda)$-Evolution Strategy with generating set adaptation. To add a further surprise it is about twice as fast then the $(1, \lambda)$-Evolution Strategy with covariance matrix adaptation [5].

There is almost no computational overhead for the extraction of information from the memory.

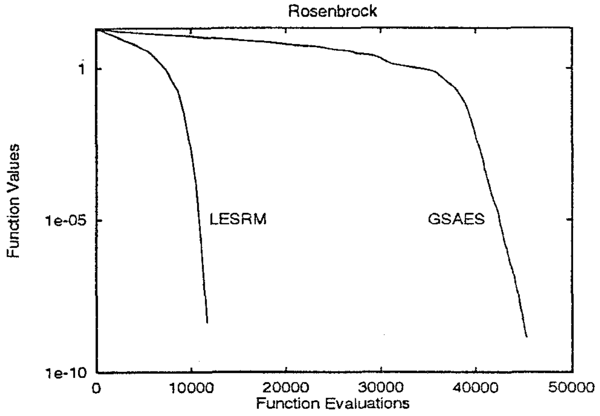

Fig. 1. Results for f_R for the $(1, \lambda)$-Evolution Strategy with generatingset adaptation (GSAES) and the (1+1)-Evolution Strategy with random memorizing (LESRM), $n = 20$

We then computed the minimum of f_R for dimensions $n = 5, ..., 200$ and $\epsilon = 10^{-10}$ as shown in Table VIII. For this function this method was superior to all other considered approaches with respect to efficiency and robustness.

TABLE VIII

Results for the $(1+1)$-Evolution Strategy with Local Memorizing for Rosenbrock's Function

n	MEAN	SD	SUCC
5	1.643533e+03	1.600629e+02	30/30
10	4.631000e+03	4.114435e+02	30/30
20	1.449407e+04	9.077249e+02	30/30
30	2.887270e+04	2.795075e+03	30/30
40	5.057945e+04	1.719543e+03	30/30
100	2.948448e+05	3.209187e+03	30/30
200	1.107836e+06	9.806003e+04	30/30

Finally, we made a benchmarking with the functions given in Table IX. f_R is the Rosenbrock function, f_S the sphere function, f_E is a hyper ellipsoid with condition index 10^6 [6], and f_P is a parabolic ridge [1]. The functions f_E and f_P were transformed for each run to a randomly generated basis system [6]. The initial points were set in the corresponding basis system to $x_i = 0$ for f_R and f_P and $x_i = 1$ for f_S and f_E, $i = 1, ..., n$. The runs were terminated at $f_{stop} < \epsilon = 10^{-10}$ for f_R, f_S and f_E. For f_P it was set to $f_{stop} < \epsilon = -10^5$.

TABLE IX

Test Functions for the $(1+1)$-ES with Random Memorizing

$$f_R(x) = \sum_{i=1}^{n-1} \left(100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right)$$

$$f_S(x) = \sum_{i=1}^{n} x_i^2$$

$$f_E(x) = \sum_{i=1}^{n} \left(1000^{(i-1)/(n-1)} x_i\right)^2$$

$$f_P(x) = -x_1 + \sum_{i=2}^{n} x_i^2$$

For dimensions $n = 5, 10, 20, 40, 100$ we made 30 runs each. The average number of function evaluation to reach the minimum with the given precision together with the standard deviation is depicted in Figure 2.

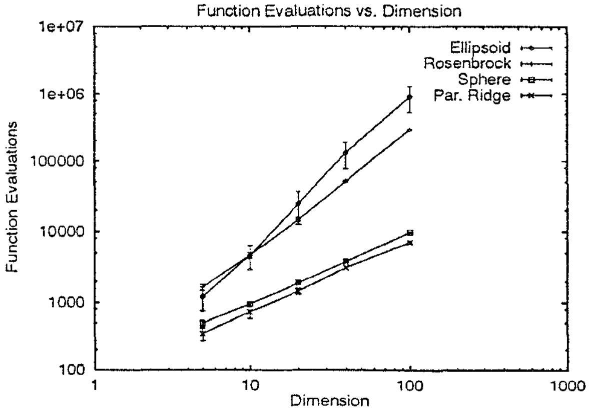

Fig. 2. Results for the minimization of the functions given in Table IX, the parameter settings are explained in the text, average values are shown together with the standard deviations

It is notable that all runs for all functions converged with the given precision. Comparing the results with data from [6] and [5] the local evolutionary search algorithm with random memorizing needs about half the number of function evaluations than the $(1, \lambda)$-Evolution Strategy with covariance matrix adaptation and only a quarter of the number of function evaluations of the $(1, \lambda)$-Evolution Strategy with generating set adaptation.

IV. Calibration of a Laser Induced Atomic Emission Plasma Spectrometer

Laser induced atomic emission plasma spectrometry is the latest development in a new generation of innovative measuring technologies. Such devices are employed in various industrial applications for analytical control of raw materials, products and processes.

The advanced measuring method is based on the time-resolved spectral analysis of a laser-induced micro-plasma. It allows a reliable qualitative and quantitative analysis of solid materials. The light source is a Q-switched Nd-YAG laser with a pulse-width of 5 ns. A lens and a mirror focus the radiation on the sample. Due to the high power density of the laserbeam atoms and molecules are emitted from the sample and a plasma is built. The excited atoms and ions in the plasma emit spectral lines that are characteristic for the respective element. A parabolic mirror collects the emitted light and focuses it on an optical fiber which transports the light to the entrance slit of an echelle spectrograph. The parabolic mirror system and the exciting laser radiation are on the same optical axis. This arrangement guarantees a high constancy of the signal. The echelle grating produces a great number of spectral orders with high spectral resolution. An additional prism separates the different spectra into the vertical direction. Figuring the stapled spectral orders on the focal plane of the spectrograph generates a two-dimensional picture with high resolution. The spectral range is 180 - 750 nm with a resolution of a few pm.

The measurement of spectra in two dimensions allows the simultaneous analysis of all relevant spectral lines. 60 elements can be determined simultaneously.

An image intensifier and a camera are installed on the focal plane of the spectrograph. The camera comprises a CCD with 1024 x 1024 pixels. Defined time delays relative to the laser pulse and time gates for measurements can be adjusted by controlling the intensifier through electronic signals. The gain of the intensifier, the exposure time and the time delay are optimized to get a high signal-to-noise ratio of the spectral lines. The detection system measures the intensities of the element-characteristic spectral lines and transmits the data to a computer.

During measurement the system control synchronizes the intensified camera system and the laser. A fast-pulse-generator with a time resolution of 5 ns controls the delay and integration time. A 16 bit AD-unit guarantees a high dynamic range for intensity data. Setup of the measurement and data analysis is managed by a PC.

Programmable line and pixelbinning reduces the time for data analysis. The data readout of the CCD array summarizes all unnecessary information to one line, which is not evaluated (hardware binning). All other data are transmitted to a memory and analyzed by a computer. The computer summarizes areas in these lines without significance (software binning). This data is rejected, too. In this way it is possible to get the whole analytic estimation in less than 3 seconds.

The measurement device works in wavelength ranges higher 190 nm without protective gas. For wavelength areas smaller 190 nm N_2 is used to suppress interfering absorption of air. Due to the small diameter of the laserbeam and to the fact that no protective gas is necessary, a sample preparation is not required.

For the calibration of such devices it is assumed that there is a relation between the intensity I_{ik} of emitted light at certain spectral lines and the concentration c_{ik} for element i in calibration sample k, i.e.

$$c_{ik} = \frac{f_i(I_{ik}, p_i)}{\sum_{j=1}^{n} f_j(I_{jk}, p_j)} \quad (1)$$

The normalization has to be done to cope with concentrations. The vectors p_i are the calibration model parameters which has to be adapted such that a calibration error measure based on the known calibration sample concentrations $\hat{c}_{ik}$ is minimized.

Because of the normalization term all parameters are mutually dependent. This makes the problem difficult. Furthermore, it cannot be assumed that the model functions are continuous differentiable.

Figure 3 shows calibration results for real data from five samples and five elements with a rather small calibration error.

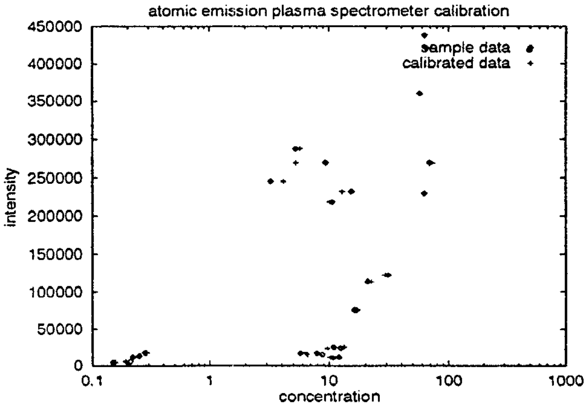

Fig. 3. Intensities vs. known calibration sample concentrations ⋄ and estimated concentrations + based on the calibration model adaptation

V. Conclusions

Evolutionary Algorithms can be upgraded by a sequential memory which is accessed randomly to generate promising new search directions. The computational overhead for exploiting the memory is neglectible. With a beam search along these new directions it has been shown that even for the simplest (1 + 1)-Evolution Strategy a robust and efficient algorithm can be designed which needs less function evaluations than more sophisticated approaches.

At present this approach is used for the fine tuning of calibration models for laser induced plasma spectrometers.

One prerequisite that this approach will work is that the used Evolutionary Algorithm must be able to generate better solutions. It should not converge prematurely. For the considered (1 + 1)-Evolution Strategy this need not necessarily happen. It cannot work for points where the level set is such that increasing or decreasing the global step size leaves the success probability for finding a better solution at a constant very low value. It is necessary that the EA creeps, even with the least progress, towards the optimum.

Future work will be dedicated to the theoretical analysis of the local evolutionary search enhancement by random memorizing.

Acknowledgments

The many helpful discussions with Heinz Mühlenbein and Dirk Schlierkamp-Voosen from the German National Center for Information Technologies (GMD) are gratefully acknowledged. We would like to thank especially Dirk Schlierkamp-Voosen for doing the comparative experiments given in Tables I to V with the different hill-climbers.

References

[1] Ingo Rechenberg, *Evolutionsstrategie '94*, Frommann–Holzboog, 1994.

[2] Adam Ghozeil and David B. Fogel, *A Preliminary Investigation into Directed Mutations in Evolutionary Algorithms*, In: H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel (Eds.). *Parallel Problem Solving from Nature – PPSN IV, 329 - 335.* Lecture Notes in Computer Science 1141. Springer, 1996

[3] Hans-Paul Schwefel, *Evolution and Optimum Seeking*, Wiley, 1995.

[4] Hans-Georg Beyer, *Towards a Theory of Evolution Strategies: Some Asymptotical Results from the* $(1 \overset{+}{,} \lambda)$ *- Theory*, Evolutionary Computation 1 (2) 165 - 188, 1993.

[5] Nikolaus Hansen and Andreas Ostermeier, *Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation* In: *Proceedings of the 1996 IEEE Int. Conf. on Evolutionary Computation*, 312 - 317. IEEE Press, 1996

[6] Nikolaus Hansen, Andreas Ostermeier, and Andreas Gawelczyk, *On the Adaptation of Arbitrary Normal Mutation Distributions in Evolutions Strategies: The Generating Set Adaptation* In: L.J. Eshelman (Ed.). *Proceedings of the Sixth Int. Conf. on Genetic Algorithms*, 57 - 64, Morgan Kaufmann, 1995

[7] Deniz Yuret and Michael de la Maza, *Dynamic Hillclimbing: Overcoming the Limitations of Optimization Techniques.* In: *The Second Turkish Symposium on Artificial Intelligence and Neural Networks*, 208 - 212, 1993

[8] Hans-Michael Voigt, Heinz Mühlenbein, and Dragan Cvetcovic, *Fuzzy Recombination for the Breeder Genetic Algorithm.* In: L.J. Eshelman (Ed.). *Proceedings of the Sixth Int. Conf. on Genetic Algorithms*, 104 - 113, Morgan Kaufmann, 1995

[9] Hans-Michael Voigt and Heinz Mühlenbein, *Gene Pool Recombination and the Utilization of Covariances for the Breeder Genetic Algorithm* In: *Proceedings of the 1995 IEEE Int. Conf. on Evolutionary Computation*, 172 - 177. IEEE Press, 1995

[10] J.A. Nelder and R. Mead, *A Simplex Method for Function Minimization*, Comp. J. 7, 308 - 313, 1965

[11] G.W. Stewart, *A Modification of Davidon's Minimization Method to Accept Difference Approximations of Derivatives* JACM 14, 72 - 83, 1967

[12] M.J.D. Powell, *An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives* Comp. J. 7, 155 - 162, 1964

[13] Francisco J. Solis and Roger J.-B. Wets, *Minimization by Random Search Techniques* Operations Research, 19 - 30, 1981

[14] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, *Numerical Recipes in C, 2nd Edition* Cambridge University Press, 1992