# Model Calibration by Optimization Methods

Tao Luo, Deng Pan, Wenchao Zhang

South University of Science and Technology of China

Mar 25, 2014

Table of contents

# 1. INTRODUCTION

## 1.1. Model Calibration.

A derivative pricing model is said to be calibrated to a set of benchmark instruments if the value of these instruments, computed in the model, correspond to their market prices. *Model calibration*[Sto97] is the procedure of selecting model parameters in order to verify the calibration condition. The goal of the calibration routine is to find the set of model parameters that minimizes the difference between the model predictions and available market data.

Model calibration can be viewed as the *inverse problem* associated with the pricing of derivatives. In the theoretical situation where prices of call options are available for all strikes and maturities, the calibration problem can be explicitly solved using an inversion formula. In real situations, given a finite (and often sparse) set of derivative prices, model calibration is an ill-posed problem whose solution often requires a *regularization* method.

In a word, Model calibration is the task of deriving model parameters by matching quoted stock/option prices.

## 1.2. Optimization Method.

Since calibration can be view as the inverse problem, and of course we can solve it by some tranditional optimization methods, like Downhill simplex method, damped Gauss-Newton method, Quasi-Newton Mathod, Genetic algorithm. But here we are focus on the Simulated Annealing.

The calibration procedure can be interpreted as an optimization problem of the form

$$\begin{aligned} \min_{x\in\mathcal{X}} f(x) \quad &, \quad \mathcal{X}\subseteq\mathbb{R}^n \\ \boldsymbol{x} \;&=\; (x_1,x_2,...,x_n)' \end{aligned}$$

In a real situation, calibrating parameters always have some restrictions, such as $\sigma \geqslant 0$, so we always do the optimization problem as follows:

$$\begin{aligned} &\min_x f(x) \\ s.j.\quad g_i(x) \;&\leqslant\; 0, \quad \text{for } i=1,2,...,m \\ h_j(x) \;&=\; 0, \quad \text{for } j=1,2,...,p \\ l.b_k \leqslant x_k \;&\leqslant\; u.b_k, \text{for } k=1,2,...,n \end{aligned} \tag{1}$$

# 2. SIMULATED ANNEALING

## 2.1. What is Simulated Annealing.

*Simulated Annealing* (SA) is motivated by an analogy to annealing in solids. The idea of SA comes from a paper published by Metropolis etc al in 1953 [Met53]. The algorithm in this paper simulated the cooling of material in a heat bath. This is a process known as annealing.

If we heat a solid past melting point and then cool it, the structural properties of the solid depend on the rate of cooling. If the liquid is cooled slowly enough, large crystals will be formed. However, if the liquid is cooled quickly (quenched) the crystals will contain imperfections.

Metropolis's algorithm simulated the material as a system of particles. The algorithm simulates the cooling process by gradually lowering the temperature of the system until it converges to a steady, frozen state.

In 1982, Kirkpatrick et al[Kir] took the idea of the Metropolis algorithm and applied it to optimisation problems. The idea is to use simulated annealing to search for feasible solutions and converge to an optimal solution.

*We can view this method as a combinition of Monte-Carlo method and hill climbing, and actually, it decreses the mount of calculation of* Monte-*Carlo method and increases the stability of hill climbing.*

### 2.2. Simulated Annealing versus Hill Climbing.

Hill climbing suffers from problems in getting stuck at local minima (or maxima). We could try to overcome these problems by trying various techniques. We could try a hill climbing algorithm using different starting points. We could increase the size of the neighbourhood so that we consider more of the search space at each move. Unfortunately, neither of these have proved satisfactory in practice when using a simple hill climbing algorithm.

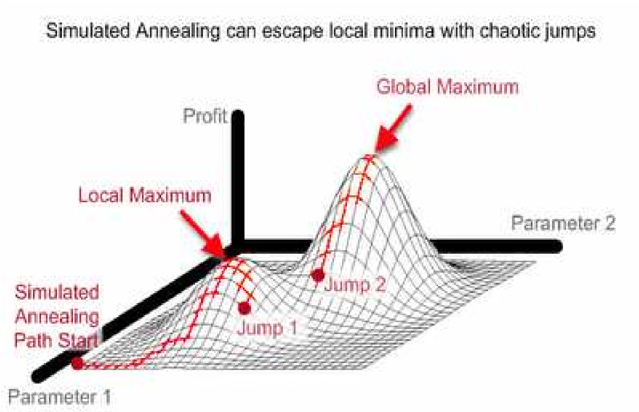


**Figure 1.** How to "climb mountain" for SA

Simulated annealing solves this problem by allowing worse moves (lesser quality) to be taken some of the time. That is, it allows some uphill steps so that it can escape from local minima. Unlike hill climbing, simulated annealing chooses a random move from the neighbourhood.[K.A95] We can see the relationship of hill climbing and simulated annealing as follows:

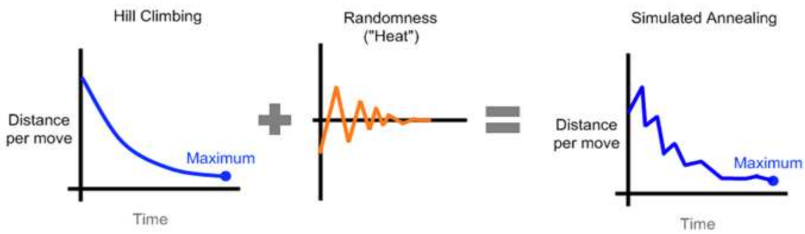


**Figure 2.** Hill Climbing and Simulated Annealing

If the move is better than its current position then simulated annealing will always take it. If the move is worse (i.e. lesser quality) then it will be accepted based on some probability.

### 2.3. Acceptance Criteria.

The law of thermodynamics state that at temperature, $T$, the probability of an increase in energy of magnitude, $\delta E$, is given by

$$P(\delta E) = \exp(-\delta E / kT) \tag{2}$$

Where $k = 1.3806488(13) \times 10^{23} J/K$ is a constant known as *Boltzmann's constant*.

The simulation in the Metropolis algorithm calculates the new energy of the system. If the energy has decreased then the system moves to this state. If the energy has increased then the new state is accepted using the probability returned by the above formula.

A certain number of iterations are carried out at each temperature and then the temperature is decreased. This is repeated until the system freezes into a steady state.

This equation is directly used in simulated annealing, although it is usual to drop the Boltzmann constant as this was only introduced into the equation to cope with different materials. Therefore, the probability of accepting a worse state is given by the equation

$$P = \exp\left(-c/T\right) > r \tag{3}$$

Where $c$ is the change in the evaluation function, $T$ is the current temperature and $r$ is a random number between 0 and 1.

The probability of accepting a worse move is a function of both the temperature of the system and of the change in the cost function. It can be appreciated that as the temperature of the system decreases the probability of accepting a worse move is decreased. This is the same as gradually moving to a frozen state in physical annealing.

Also note, that if the temperature is zero then only better moves will be accepted which effectively makes simulated annealing act like hill climbing.

There are some neccessary conditions in SA, which are the basic assumption to do this procedure: The initial condition is that all states must communicate, which means that the starting point should not affect results. The second condition is that the probability matrix is irreducible, i.e. $Q_{ij} > 0$ for all possible states $i$ and $j$, and what's more, $Q_{ij}$ is not finite nilponent.

### 2.4. Implementation of Simulated Annealing.

There are five step for SA procedure,

I. Select starting temperature and initial parameter values
II. Randomly select a new point in the neighborhood of the original
III. Compare the two points using the Metropolis criterion
IV. Repeat steps 2 and 3 until system reaches equilibrium state...
V. Decrease temperature and repeat the above steps, stop when system reaches frozen state

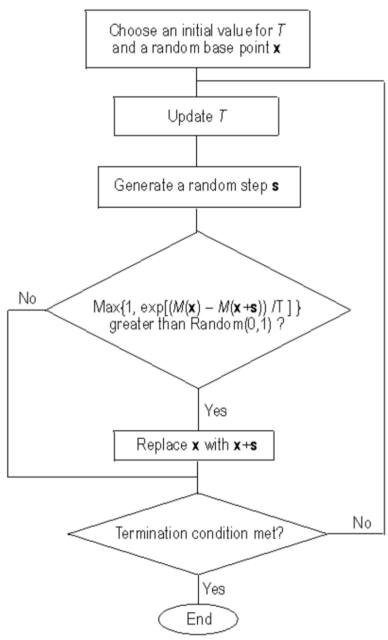


**Figure 3.** Flow chart of SA

One of the parameters to the algorithm is the schedule. This is the cooling schedule.

This algorithm assumes that the annealing process will continue until the temperature reaches zero. Some implementations keep decreasing the temperature until some other condition is met. For example, no change in the best state for a certain period of time.

The way this algorithm is presented may hide another aspect of the algorithm that is shown more directly in some other presentations.

That is, a particular phase of the search normally continues at a certain temperature until some sort of equilibrium is reached. This might be a certain number of iterations or it could be until there has been no change in state for a certain number of iterations.

#### 2.4.1. About temperatures.

The cooling schedule of a simulated annealing algorithm consists of four components.

- Starting Temperature
- Final Temperature
- Temperature Decrement
- Iterations at each temperature

**Starting Temperature.**

The starting temperature must be hot enough to allow a move to almost any neighbourhood state. If this is not done then the ending solution will be the same (or very close) to the starting solution. Alternatively, we will simply implement a hill climbing algorithm.

However, if the temperature starts at too high a value then the search can move to any neighbour and thus transform the search (at least in the early stages) into a random search. Effectively, the search will be random until the temperature is cool enough to start acting as a simulated annealing algorithm.

The problem is finding the correct starting temperature. At present, there is no known method for finding a suitable starting temperature for a whole range of problems.

If we know the maximum distance (cost function difference) between one neighbour and another then we can use this information to calculate a starting temperature.

Another method, suggested in Rayward-Smith[Ray96], is to start with a very high temperature and cool it rapidly until about 60% of worst solutions are being accepted. This forms the real starting temperature and it can now be cooled more slowly.

A similar idea, suggested in Dowsland[K.A95], is to rapidly heat the system until a certain proportion of worse solutions are accepted and then slow cooling can start. This can be seen to be similar to how physical annealing works in that the material is heated until it is liquid and then cooling begins (i.e. once the material is a liquid it is pointless carrying on heating it).

**Final Temperature.**

As for Final temperature, it is usual to let the temperature decrease until it reaches zero. However, this can make the algorithm run for a lot longer, especially when a geometric cooling schedule is being used.

In practise, it is not necessary to let the temperature reach zero because as it approaches zero the chances of accepting a worse move are almost the same as the temperature being equal to zero.

**Temperature Decrement.**

One way to decrement the temperature is a simple linear method.

An alternative is a geometric decrement

$$t = t\alpha$$

where $\alpha < 1$.

Experience has shown that $\alpha$ should be between 0.8 and 0.99, with better results being found in the higher end of the range. Of course, the higher the value of $\alpha$, the longer it will take to decrement the temperature to the stopping criterion.

**Iterations at each Temperature.**

The final decision we have to make is how many iterations we make at each temperature. A constant number of iterations at each temperature is an obvious scheme.

Another method, first suggested by Lundy[Lun86] is to only do one iteration at each temperature, but to decrease the temperature very slowly. The formula they use is

$$t = t/(1+\beta t)$$

where $\beta$ is a suitably small value.

### 2.5. Summary.

Using simulated annealing it has been proved that it is possible to converge to the best solution. The problem is, it may take more time than an exhaustive search. So although it may not be practical to find the best solution using simulated annealing, simulated annealing does have this important property which is being used as the basis for future research.

Many of the techniques we have looked at are not only applicable to simulated annealing. For example, the performance improvements with regards to the cost function can obviously be used in other evolutionary and meta-heuristic algorithms.

## 3. Sequential Quadratic Programming

### 3.1. Introduction.

*Sequential quadratic programming* (SQP) is an iterative method for nonlinear optimization. SQP methods are used on problems for which the objective function and the constraints are twice continuously differentiable.[Cen]

SQP methods solve a sequence of optimization subproblems, each of which optimizes a quadratic model of the objective subject to a linearization of the constraints. If the problem is unconstrained, then the method reduces to Newton's method for finding a point where the gradient of the objective vanishes. If the problem has only equality constraints, then the method is equivalent to applying Newton's method to the first-order optimality conditions, or Karush–Kuhn–Tucker conditions, of the problem.

Based on the work of Biggs et al[M.C75], the method allows you to closely mimic Newton's method for constrained optimization just as is done for unconstrained optimization. At each major iteration, an approximation is made of the Hessian of the Lagrangian function using a quasi-Newton updating method. This is then used to generate a QP subproblem whose solution is used to form a search direction for a line search procedure. An overview of SQP is found in Fletcher[R.87] et al. The general method is following.

### 3.2. Algorithm.

The principal idea of SQP is the formulation of a QP (*Quadratic Programming*) subproblem based on a quadratic approximation of the Lagrangian function:

$$L(x,\lambda) = f(x) + \sum_{i=1}^{m} \lambda_i \cdot g_i(x) \tag{4}$$

We obtain the QP subproblem by linearizing the nonlinear constraints.

$$\begin{aligned} &\min_{d\in\mathbb{R}^n} \frac{1}{2}d^T H_k d + \nabla f(x_k)^T d \\ &\nabla g_i(x_k)^T d + g_i(x_k) = 0 \quad , \quad i = 1, ..., m_e \\ &\nabla g_i(x_k)^T d + g_i(x_k) \leqslant 0 \quad , \quad i = m_e + 1, ..., m \end{aligned} \tag{5}$$

This subproblem can be solved using any QP algorithm. The solution is used to form a new iterate.

$$x_{k+1} = x_k + \alpha_k d_k \tag{6}$$

The step length parameter $\alpha_k$ is determined by an appropriate line search procedure so that a sufficient decrease in a merit function is obtained.

A nonlinearly constrained problem can often be solved in fewer iterations than an unconstrained problem using SQP. One of the reasons for this is that, because of limits on the feasible area, the optimizer can make informed decisions regarding directions of search and step length.

### 3.3. SQP Implementation.

The SQP implementation consists of three main stages, which are discussed briefly in the following subsections:

1. Updating the Hessian Matrix
2. Quadratic Programming Solution
3. Line Search and Merit Function

#### 3.3.1. Updating the Hessian Matrix.

At each major iteration a positive definite quasi-Newton approximation of the Hessian of the Lagrangian function, $H$, is an estimate of the Lagrange multipliers.

$$H_{k+1} = H_k + \frac{q_k q_k^T}{q_k^T s_k} - \frac{H_k s_k s_k^T H_k^T}{s_k^T H_k s_k} \tag{7}$$

where

$$\begin{aligned} s_k &= x_{k+1} - x_k \\ q_k &= \left(\nabla f(x_{k+1}) + \sum_{i=1}^{m} \lambda_i \cdot \nabla g_i(x_{k+1})\right) - \left(\nabla f(x_k) + \sum_{i=1}^{m} \lambda_i \cdot \nabla g_i(x_k)\right) \end{aligned}$$

#### 3.3.2. Quadratic Programming Solution.

At each major iteration of the SQP method, a QP problem of the following form is solved, where $A_i$ refers to the `i`th row of the $m$-by-$n$ matrix $A$.

$$\begin{aligned} \min_{d\in\mathbb{R}^n} q(d) &= \frac{1}{2}d^T \mathrm{Hd} + c^T d, \\ A_i d &= b_i, i = 1, ..., m_e \\ A_i d &\leqslant b_i, i = m_e + 1, ..., m \end{aligned} \tag{8}$$

#### 3.3.3. Line Search and Merit Function.

The solution to the QP subproblem produces a vector $d_k$, which is used to form a new iterate equation 6, and what's more, The step length parameter $\alpha_k = \alpha$ is determined in order to produce a sufficient decrease in a merit function of the following form is used in this implementation.

$$\Psi(x) = f(x) + \sum_{i=1}^{m_e} r_i \cdot g_i(x) + \sum_{i=m_e+1}^{m} r_i \cdot \max[0, g_i(x)] \tag{9}$$

## 4. Examples by using SA and SQP

Since Matlab has such optimization methods like SA and SQP, we can directly call such function to realize our question, the most important thing left is to write a unresolve function and the constraint conditions.

Here for our task 1, we have such problem as follows:

$$\begin{array}{rrcl} & f\colon \mathbb{R}^4 & \rightarrow & \mathbb{R} \\ & f(x) & = & x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4, \\ s.j. & g_i(x) & \leqslant & 0, i = 1, 2, 3 \end{array} \tag{10}$$

where $g_i$ is given by

$$\begin{aligned} g_1(x) &= x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8 \\ g_2(x) &= x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 - x_1 - x_4 - 10 \\ g_3(x) &= 2x_1^2 + x_2^2 + x_3^2 + 2x_1 - x_2 - x_4 - 5 \end{aligned}$$

For these two methods, we have following results:

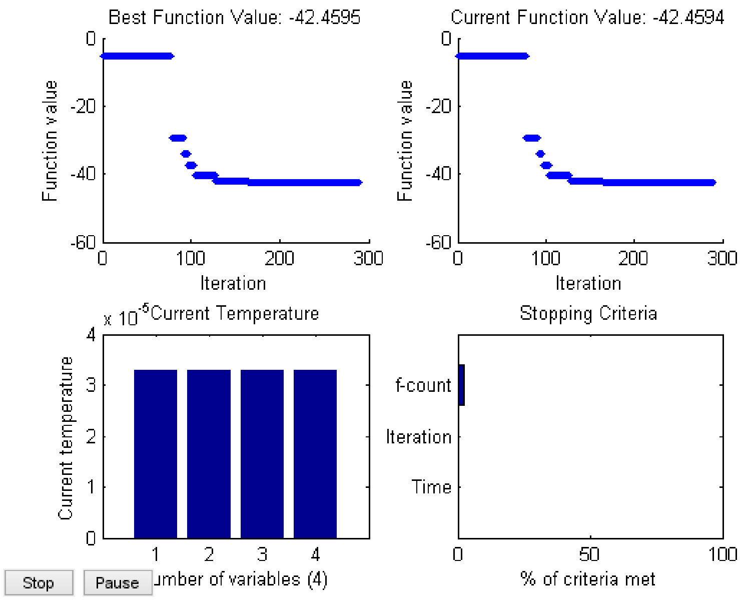


**Figure 4.** Output of SA for task1

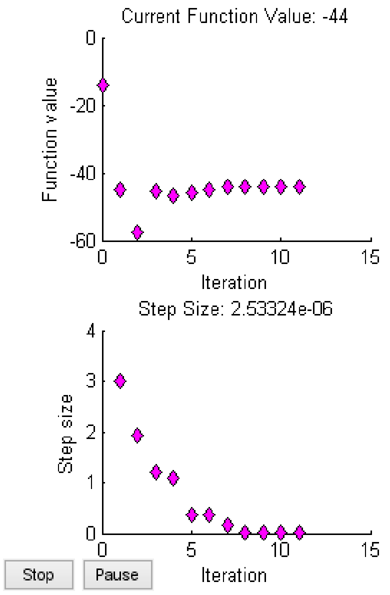


**Figure 5.** Output of SQP for task 1

As for task 2, we have such problem as follows:

$$\begin{array}{rrcl} & f\colon \mathbb{R}^2 & \rightarrow & \mathbb{R} \\ & f(x) & = & \exp(x_1)(4x_1^2+2x_2^2+4x_1x_2+2x_2+1), \\ s.j. & g_i(x) & \leqslant & 0, i=1,2 \end{array} \tag{11}$$

where $g_i$ is given by

$$\begin{aligned} g_1(x) &= -x_1x_2-10 \\ g_2(x) &= x_1x_2-x_1-x_2+1.5 \end{aligned}$$

For these two methods, we have following results:

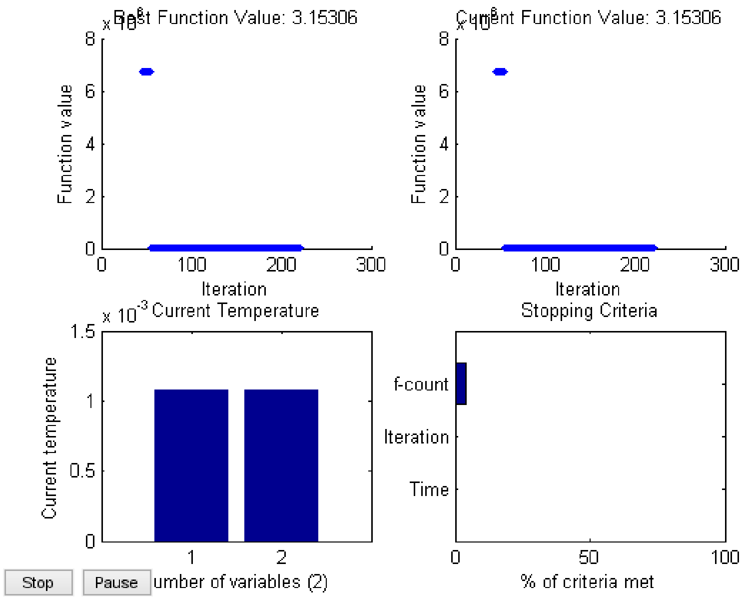


**Figure 6.** Output of SA for task 2

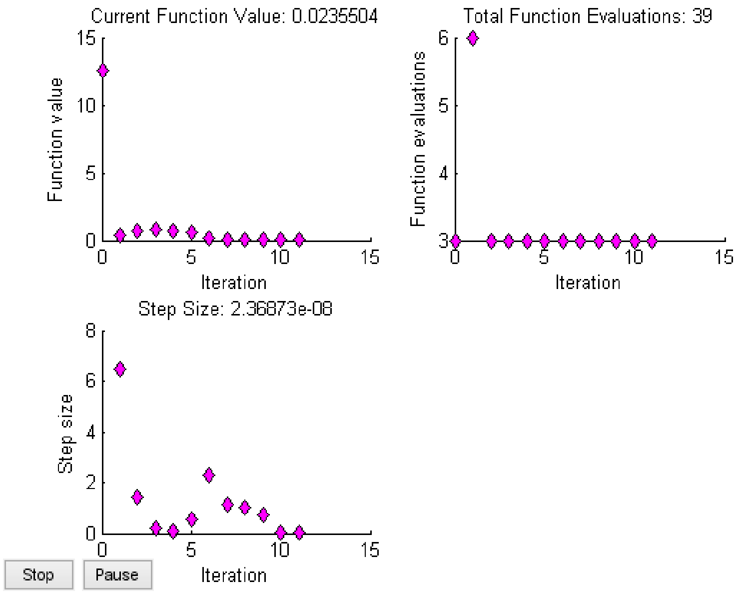


**Figure 7.** Output of SQP for task 2

### 4.1. Conclusions.

It seems that both SA and SQP have abilities to get a stable correct solution of non-linear constraint problem. In some way, SA may using more time for computing, since usually SA start at a very huge number (temperature). And for these two task, both of them have smooth curve and constraint conditions, so SA seems not better than SQP.

The most important things is that SA is not dependent on the initial state, but maybe not always true for SQP, although SQP is a update version to solve nonlinear systems. (SQP will converge to a local minimal/maximal point sometime).

## 5. Derive the shift and variance

### 5.1. Asset dynamics and parameter estimation.

Under the assumption of B-S model the movement of asset price in the market follows :

$$\log(\Delta S_i) = \left(\mu - \frac{1}{2}\sigma^2\right)\Delta t_i + \sigma\sqrt{\Delta t_i}\boldsymbol{\xi}_i$$

where $S$ is the stock price, $\mu$ and $\sigma^2$ is the drift and volatility of this stock, $\boldsymbol{\xi}_i$ is iid standard normal distribution.

If the price strickly follows this assumption, moreover $\Delta t_i$'s are usually the intervals with same span. We can conclude $\log(\Delta S) \sim N\big(\big(\mu - \frac{1}{2}\sigma^2\big)\Delta t, \sigma^2\Delta t\big)$. by analyzing the daily price, we can estimate $\sigma^2$ and $\mu$.

### 5.2. Data characteristics.

We use google's stock price of past three years to estimate its volatility and drift. those historical price can be downloaded from finance.yahoo.com. we first take the logrithm of the price the make the trend more like a straight line. Figure 1. is the raw price Figure 2. show the log price.

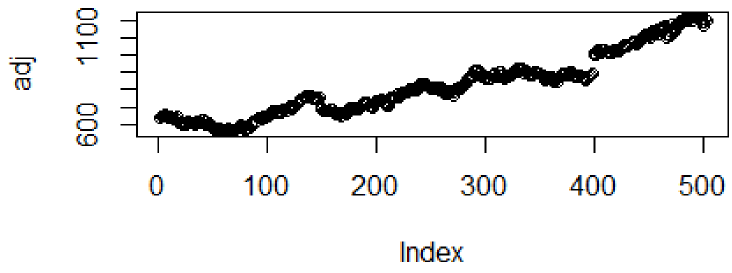


**Figure 8.** Google stock prices

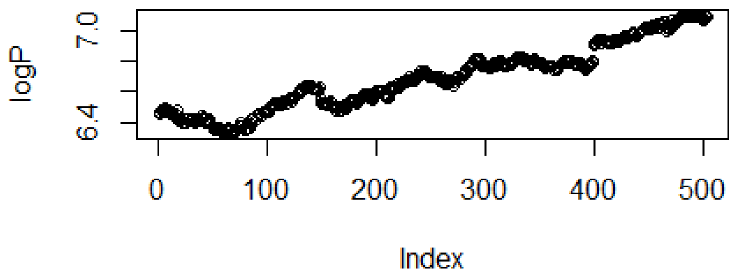


**Figure 9.** Google Log-prices

Note that the price trend is like broken straight lines, this is becasuse there turns to be some periodical financial reports of this corporation, which may reveal some unexpected states of company. Also we plot the histogram to have a look at the distribution of $\log(\Delta S)$, almostly comply with normal distribution:

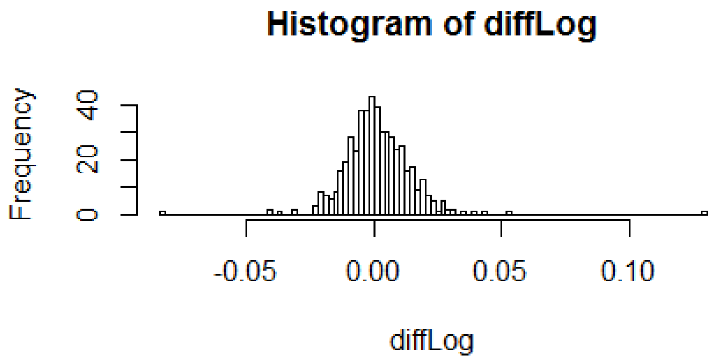


**Figure 10.** Histogram of difference of log-prices

### 5.3. Calculating $\sigma^2$ and $\mu$.

We use $R$ to accomplish our statistic work, the code is as follows:

```
setwd("f:/")
priceTab = read.csv("Goog3Y.csv")
priceTab = priceTab[order(nrow(priceTab):1),];
adj = priceTab[,7];
logP = log(adj);
diffLog = diff(logP);
sigmaSq = var(diffLog);
mu = mean(diffLog) + 0.5*sigmaSq;
```

Both of two parameters turn out to be very small, and since the trend and variance is so apparent, we omit taking the hypothesis test. Unfortunatly, we can't tell the confidencen intervals of these two parameters because their estimators' distribution can't be calculated easily.

| | |
|---|---|
| mu | 0.00136680563478634 |
| sigmaSq | 0.000185877033783368 |

**Figure 11.** Output from R

### 5.4. Some observations and comments.

1. historical price indicates the volatility is not constant, high volatility is usually followed by high volatility, so is low volatility. this phemomenon is called volatility cluster. people create GARCH model to explain this autocorrelated volatility;
2. price may have some great jumps, which can't be explained by this diffusion model. and that's why jump-diffussion models are studied after Black-Scholes.

## 6. A REPORT ON ZYXH

We now focus on a stock named ZYXH, while this analysis procedure might suit for many other different cases.

### 6.1. Fundamental Analysis.

#### 6.1.1. Economical analysis.

Since GDP of China always be stable, another aspects may account more.

Though Central Bank and Four major commercial banks of China are cooperating in depressing so called Yu'ebao, that actually in my opinion stablized the financial world because this industry used to be totally controled by the former, while Yu'ebao made the interest rate market became volatile.

No matter whether the traditional banks or Yu'ebao can win the game, here must be an overall revolution in financial market both fundamentally and structurely. Thus one cannot ignore this potential challenge in the future to any extent if he invests stocks.

Another essential economical affair occurred recently is related to Central Bank too. The exchange rate, i.e. the value of China Yuan in global market is down rapidly after a continuous increase in past few years which makes ex-rate escalate to like 6: 1. The latest quote of China yuan to US dollar is about 6.2128. This can be a very serious problem for export oriented firms.

### 6.2. Industry Analysis.

#### 6.2.1. Outlook of industry.

For ZYXH, it belongs to Industry of Medicine.

Not exaggerating, Medicine is one of the few strong industries in such down-turn stock market. By an article published 27th March, Medicine index rises 4.18%, and the Medical Service board makes the most contribution which rises 26.92%.

There are several reasons of policy of course. The 18th National People's Congress released several revolution blueprints over Medicine (Health care reform) to alleviate conflicts between hospitals and patients.[1]

Since the Health Care Reform could have a long way to go, we can predict that the benefit to Medicine industry will be constant in the foreseeable futrue.

### 6.2.2. Competitions inside industry.

估值比较

| 排名 | 代码 | 简称 | PEG | 市盈率 | | | | | 市销率 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 12A | TTM | 13E | 14E | 15E | 12A | TTM | 13E | 14E | 15E |
| >55 | 600645 | 中源协和 | -- | 641.84 | -2,565.56 | -- | -- | -- | 19.42 | 25.28 | -- | -- | -- |
| 行业平均 | | | -2.78 | 28.69 | 38.42 | 85.84 | 66.85 | 66.85 | 2.19 | 2.90 | 6.15 | 5.05 | 4.19 |
| 行业中值 | | | 1.68 | 51.20 | 45.34 | 43.73 | 33.86 | 24.91 | 4.44 | 6.18 | 6.56 | 5.19 | 4.16 |
| 1 | 600285 | 羚锐制药 | 0.69 | 79.27 | 32.75 | 30.89 | 36.23 | 24.69 | 3.85 | 5.08 | 4.67 | 3.59 | 2.87 |
| 2 | 002422 | 科伦药业 | 0.80 | 17.72 | 16.47 | 15.60 | 12.70 | 10.10 | 4.57 | 2.98 | 2.78 | 2.23 | 1.82 |
| 3 | 002001 | 新 和 成 | 0.84 | 16.48 | 20.02 | 14.35 | 11.00 | 8.65 | 3.76 | 3.70 | 3.51 | 2.86 | 2.36 |
| 4 | 600351 | 亚宝药业 | 0.96 | 46.65 | 33.11 | 36.05 | 26.10 | 19.16 | 2.92 | 3.40 | 3.25 | 2.65 | 2.16 |
| 5 | 000538 | 云南白药 | 0.98 | 35.97 | 25.66 | 25.18 | 22.28 | 17.86 | 3.45 | 3.71 | 3.52 | 2.99 | 2.58 |

杜邦分析比较

| 排名 | 代码 | 简称 | ROE(%) | | | | 净利率(%) | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | 3年平均 | 10A | 11A | 12A | 3年平均 | 10A | 11A | 12A |
| >147 | 600645 | 中源协和 | 10.28 | 6.32 | 15.02 | 9.49 | 5.75 | 3.12 | 9.49 | 4.63 |
| 行业平均 | | | 14.65 | 17.42 | 14.28 | 12.24 | 17.91 | 18.35 | 18.88 | 16.50 |
| 行业中值 | | | 12.14 | 13.49 | 10.90 | 9.30 | 12.46 | 13.23 | 12.94 | 10.98 |
| 1 | 002653 | 海思科 | 44.70 | 67.48 | 40.68 | 25.94 | 53.16 | 52.57 | 51.67 | 55.24 |
| 2 | 300298 | 三诺生物 | 41.47 | 60.05 | 49.11 | 15.24 | 38.10 | 34.24 | 42.09 | 37.99 |
| 3 | 300361 | 奥赛康 | 36.72 | 34.13 | 37.24 | 38.80 | 11.65 | 11.46 | 11.62 | 11.86 |
| 4 | 300326 | 凯利泰 | 32.79 | 46.36 | 39.53 | 12.48 | 53.76 | 60.66 | 46.01 | 54.60 |
| 5 | 300314 | 戴维医疗 | 26.78 | 35.00 | 32.86 | 12.48 | 25.57 | 21.72 | 26.10 | 28.88 |

**Figure 12.** Competitions inside industry

From the above tables we can clearly see that ZYXH perform really bad compared with its competitors. Especially the P/E ratio is tens of times over other competiters, which means this stock is nothing but bubble.

In this sense, by no means I will buy this stock. But wait, for speculators, this is somewhat a better security than other relatively stable stocks.

Speaking of which, this industry is seriously dependent on patents. And technology improvements can save a struggling company from bankrupcy and to be outstanding overnight. Reversely, if one firm can't go on in researching then it may soon be eliminated by the market.

### 6.3. Fundamental Analysis.

Below is the financial report of ZYXH over seasons.

| 每股指标 | 13-09-30 | 13-06-30 | 13-03-31 | 12-12-31 | 12-09-30 | 12-06-30 | 12-03-31 | 11-12-31 | 11-09-30 |
|---|---|---|---|---|---|---|---|---|---|
| 基本每股收益(元) | 0.0113 | 0.0154 | -0.0230 | -0.0140 | 0.0129 | 0.0314 | 0.0120 | 0.0206 | 0.0196 |
| 每股净资产(元) | 0.4545 | 0.4428 | 0.4300 | 0.4508 | 0.4730 | 0.5240 | 0.4920 | 0.4800 | 0.4590 |
| 每股公积金(元) | 0.0004 | -- | -- | -- | 0.0010 | 0.0648 | 0.0648 | 0.0648 | 0.0648 |
| 每股未分配利润(元) | -0.6171 | -0.6285 | -0.6439 | -0.6204 | -0.5995 | -0.6124 | -0.6438 | -0.6562 | -0.6768 |
| 每股经营现金流(元) | 0.1276 | 0.0582 | -0.0090 | -0.1425 | 0.1311 | 0.2081 | 0.0200 | 0.1325 | -0.0017 |
| **成长能力指标** | **13-09-30** | **13-06-30** | **13-03-31** | **12-12-31** | **12-09-30** | **12-06-30** | **12-03-31** | **11-12-31** | **11-09-30** |
| 营业收入(元) | 9664万 | 9553万 | 7044万 | 9034万 | 8021万 | 8022万 | 4957万 | 5214万 | 6136万 |
| 毛利润(元) | 6723万 | 6648万 | 4653万 | 5949万 | 5361万 | 5544万 | 3544万 | 3914万 | 4431万 |
| 归属净利润(元) | 368万 | 502万 | -764万 | -454万 | 419万 | 1022万 | 403万 | 669万 | 638万 |
| 扣非净利润(元) | 167万 | 51.7万 | -1255万 | -956万 | 181万 | 469万 | 369万 | 90.1万 | 570万 |
| 营业收入同比增长(%) | 20.49 | 19.08 | 42.10 | 73.25 | 30.73 | 21.20 | -26.06 | -16.61 | -4.61 |
| 归属净利润同比增长(%) | -12.08 | -50.87 | -289.33 | -167.84 | -34.43 | 58.50 | 3.42 | 0.12 | -15.68 |
| 扣非净利润同比增长(%) | -7.57 | -88.95 | -439.70 | -1,160.41 | -68.23 | -24.91 | -4.73 | -82.91 | -4.68 |
| 营业收入环比增长(%) | 1.16 | 35.61 | -22.02 | 12.63 | -0.01 | 61.82 | -4.93 | -15.02 | -7.30 |
| 归属净利润环比增长(%) | -26.70 | -- | -- | -208.48 | -59.04 | 153.36 | -39.74 | 4.86 | -1.00 |
| 扣非净利润环比增长(%) | 223.48 | -- | -- | -627.76 | -61.35 | 26.80 | 309.92 | -84.19 | -8.63 |
| **盈利能力指标** | **13-09-30** | **13-06-30** | **13-03-31** | **12-12-31** | **12-09-30** | **12-06-30** | **12-03-31** | **11-12-31** | **11-09-30** |
| 摊薄净资产收益率(%) | 2.81 | 2.55 | -2.75 | -2.73 | 2.66 | 5.23 | 3.19 | 2.47 | 5.42 |
| 摊薄总资产收益率(%) | 0.64 | 0.54 | -0.62 | -0.62 | 0.64 | 1.31 | 0.84 | 0.78 | 1.77 |
| 毛利率(%) | 69.57 | 69.59 | 66.05 | 65.85 | 66.84 | 69.10 | 71.48 | 75.07 | 72.22 |
| 净利率(%) | 3.81 | 5.26 | -10.84 | -5.03 | 5.22 | 12.74 | 8.14 | 12.84 | 10.40 |

**Figure 13.** Financial report of ZYXH over seasons

1. For some more information, please visit: http://www.dwz.cn/fA93j.

As we can see, though didn't deficit recently, nearly zero earnings been made per share, far less than what other con-industrial firms made. And it is that no signal of improvement that makes the situation even worse.

A serious problem seems suffering a currency shortage.

百分比报表

| 指标 2013-09-30 | 金额(元) | 占比 |
|---|---|---|
| 总资产 | 13.4亿 | 100% |
| 流动资产 | 7.63亿 | 57.12% |
| 货币资金 | 6.15亿 | 46.05% |
| 应收账款 | 5900万 | 4.42% |
| 存货 | 2728万 | 2.04% |
| 预付账款 | 1543万 | 1.15% |
| 非流动资产 | 5.73亿 | 42.88% |
| 固定资产 | 1.99亿 | 14.88% |
| 无形资产 | 9181万 | 6.87% |
| 长期待摊费用 | 1.31亿 | 9.77% |
| 金融资产 | -- | -- |
| 总负债金额 | 10.3亿 | 100% |
| 流动负债 | 10.3亿 | 99.72% |
| 非流动负债 | 287万 | 0.28% |

**Figure 14.** Percentage report

Notice that current liabilities is larger than current assets. Which means

$$\text{Current Ratio} = \frac{\text{CurrentAssets}}{\text{CurrentLiabilities}} < 1$$

Regularly, this ratio should be approximately 2 if a firm perform well. The lower the ratio, the more serious the liability problem the firm has. This is a intensive signal tell us that 'this is a trash security'.

Also, the Net Working Capital is actually negative, bad situation.

Let's talk about Inventory Turnover

$$\text{Inventory Turnover} = \frac{\text{Annual sales}}{\text{Inventory}} = 2.16$$

Intuitively, this ratio must be close to 1. Again this seems not good. And consequently the large inventory unsold could result in a consecutive liability problems mentioned above.

Needless to say, the Net Profit Margin is also of tragedy. Now let's finally talk about Price-to-Book Ratio:

First

$$\text{Book Value} = \frac{\text{Equity}}{\text{No. Shares}} = (3.1 \times 10^8) / (34929 \times 10^4) = 0.8875$$

Then

$$P/B = \frac{\text{stock price}}{\text{book valuew}} = 25.54/0.8875 = 28.777$$

Now with this obvious result one can claim that ZYXH is nothing but a bubble.

## 7. SUMMARY

We have studied mainly two ways to calibrate our stock model by using optimization methods: Simulated Annealing and Sequential Quadratic Programming. We have done several experiments for these two methods and even rewrite code for SA. Furthermore, it's not just for more theoretic problem like task 1 and 2, we also did some macroscopic things for real stock market.

## LIST OF FIGURES



## BIBLIOGRAPHY

**[Cen]** Mathwork Documentation Center. Constrained nonlinear optimization algorithms. *Http://www.mathworks.cn/cn/help/optim/ug/constrained-nonlinear-optimization-algorithms.html*.

**[K.A95]** Dowsland K.A. Simulated annealing. in modern heuristic techniques for combinatorial problems. *McGraw-Hill*, , 1995.

**[Kir]** Vecchi M.P. Kirkpatrick S, Gelatt C.D. Optimization by simulated annealing. *Science*, 220(4598).

**[Lun86]** Mees A. Lundy M. Convergence of an annealing algorithm. *Math. Prog.*, 34:111–124, 1986.

**[M.C75]** Biggs M.C. "constrained minimization using recursive quadratic programming," towards global optimization. *North-Holland*, :341–349, 1975.

**[Met53]** Rosenbluth M.N., Teller A.H., Teller E Metropolis N., Rosenbluth A.W. Equation of state calculation by fast computing machines. *E.J. of Chem. Phys.*, 21:1087–1091, 1953.

**[R.87]** Fletcher R. Practical methods of optimization. *John Wiley and Sons*, , 1987.

**[Ray96]** Reeves C.R., Smith G.D. Rayward-Smith V.J., Osman I.H. Smodern heuristic search methods. *John Wiley & Sons*, , 1996.

**[Sto97]** R. Storn. Differential evolution, a simple and efficient heuristic strategy for global optimization over continuous spaces. *Journal for Global Optimization*, 11(Dordrecht):341–359, 1997.