**Quasi-Monte Carlo Method in cash flow testing simulation**

Table of Content

[Abstract 2](#_Toc271014043)

[The introduction of survival model. 3](#_Toc271014044)

[1. The survival model 3](#_Toc271014045)

[2. Getting the survival function by using the China life table. 4](#_Toc271014046)

[The Quasi-Monte Carlo method. 6](#_Toc271014047)

[1. The Quasi-Monte Carlo method and the Halton sequences. 6](#_Toc271014048)

[2. The inverse transformation method 7](#_Toc271014049)

[The universal product model and the cash flow testing with the QMC 8](#_Toc271014050)

[1. The universal product model 8](#_Toc271014051)

[2. Using of the Quasi-Monte Carlo. 9](#_Toc271014052)

[3. The change of the cash flow pattern via using the stochastic model 9](#_Toc271014053)

[The simulation of the interest rate via the QMC method 11](#_Toc271014054)

[1. The Vasicek model for the interest rate 11](#_Toc271014055)

[2. Definition of the parameter 12](#_Toc271014056)

[3. Simulation with the quasi-Monte Carlo method 13](#_Toc271014057)

[Conclusion and Improvements 15](#_Toc271014058)

[1. Conclusions 15](#_Toc271014059)

[2. Improvements 15](#_Toc271014060)

[Reference: 16](#_Toc271014061)

[Attachment 17](#_Toc271014062)

Abstract

What actuary call cash flow testing is a large scale simulation pitting a company’s current policy obligation against the change of some basic assumptions. In most situation, actuary conduct the cash flow testing under the interest rate, which always s follow a statistical process. In this paper, I conduct the cash flow testing under the mortality rate, which is also a very important factor in the insurance company’s cash flow.

Firstly, I introduce the function of the survival model and got the parameter of the model by using the Chinese life table.

Secondly, I build a model of a ten years universal product (an insurance product based on an investment account and provide some life benefit.

Thirdly, I use the Inverse transformation method and Halton sequence in the Monte Carlo method to simulate the mortality in different scenarios to see the cash flow situation in those situations.

The meaning of my paper is as follows:

1. Introduce the stochastic method instead of deterministic method to conduct the cash flow testing. In China, the insurance company conducted the cash flow testing assuming a deterministic change of the assumptions. For example, assuming the interest rate increase 10% or decrease 10%. Here we introduce a meaningful and practical method of the stochastic simulation via the Monte Carlo method under the mortality rate.
2. Using the low discrepancy method instead of the pseudo-random method to increase the calculation speed of the simulation. We can see that the quasi-Monte Carlo method is more efficiency in the cash flow testing.

3）Change the pattern of the profit margin

The introduction of survival model.

1. The survival model

We will assume for now that T is a continuous random variable with probability density function (p.d.f.) f(t) and cumulative distribution function(c.d.f.)

giving the probability that the event has occurred by duration t.

It will often be convenient to work with the complement of the c.d.f, the survival function

which gives the probability of being alive at duration t, or more generally,the probability that the event of interest has not occurred by duration t.

Here the S(t) is what we usually called the survival model.

We also introduce the hazard function, which is as follows:

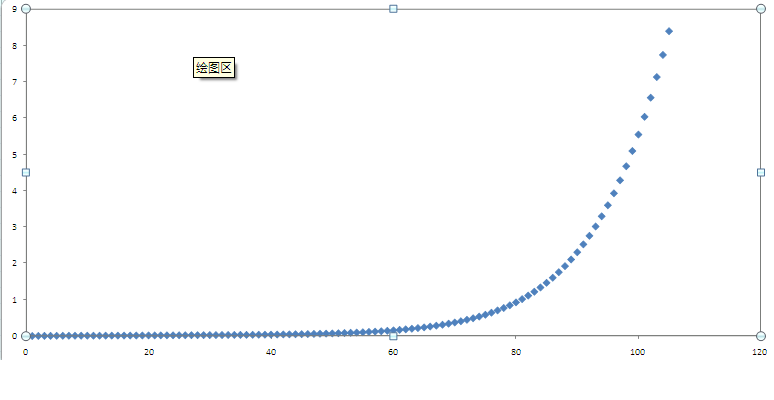
from the above formula we can got the following formula:

Hence

By using this formula we can start the next part.

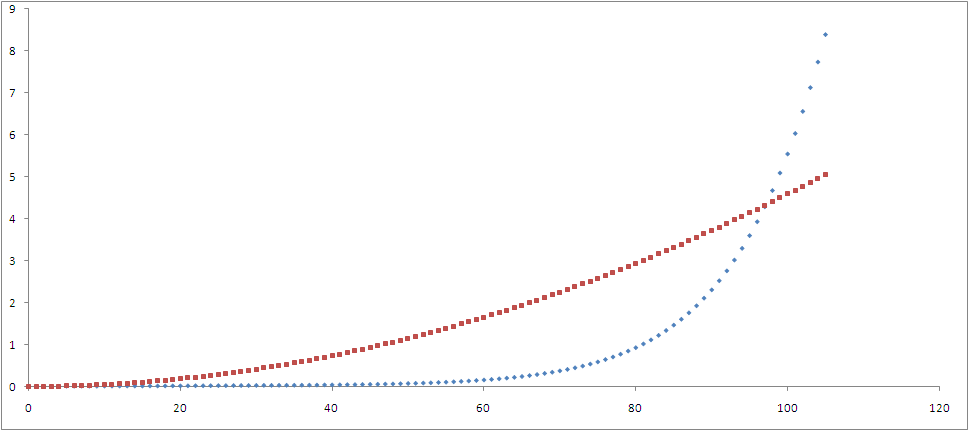
1. Getting the survival function by using the China life table.

I put China life table as the attachment. By using the China life table we can get the following chart



The x axes is the time ti and the Y axes is the lnS(ti). From the above chart, we can see that the figure is likely quadratic according to the time t, so we may assume that S(t)=

By using the regression method we can get the A is equal to 0.00046 and the comparison of the original data and the regression data are s follows:



Here the red data are the approximation data and the blue data is the original data.

I also listed the analysis results as follows:

|  |  |
| --- | --- |
| Regression statistics | |
| Multiple R | 0.8337377 |
| R Square | 0.6951186 |
| Adjusted R Square | 0.6921871 |
| Standard variance | 1.018089 |
| Numbers of observation | 106 |

We can see that the standard variance and the R square is not very big, so the result is feasible.

The Quasi-Monte Carlo method.

1. The Quasi-Monte Carlo method and the Halton sequences.

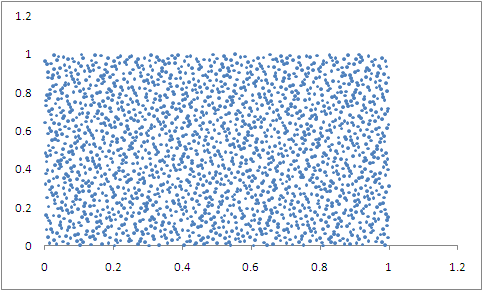
Quasi-Monte Carlo simulation is often described as the deterministic version of the ,onte Carlo simulation. It uses the uniformly distributed modulo 1 sequences (other names are low-discrepancy sequences, quasi-Monte Carlo sequences) to simulate the problem, instead of pseudorandom numbers used by the Monte Carlo.

The Halton sequence, is a well- known low discrepancy sequence. For an base b, if

then the

The Halton sequences is u.d. mod 1 if its base are relatively prime. Usually, these numbers are chosen simply as prime numbers.

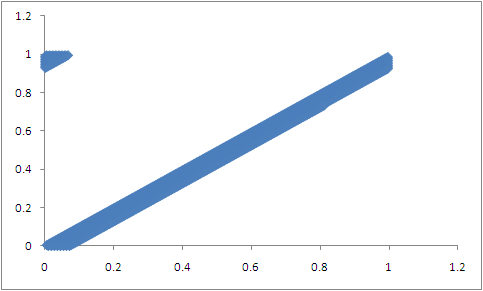
The following figure is the Halton sequences under the base 2 and 3



In my paper, I used the Halton sequence to generate the random data.

We can see that the results are fairly good, the data lies randomly in the interval (0,1)

However the results are not always good, you can see from the following chart, the data are generated via the base of 227 and 229. The data are not distributed randomly.



In this paper I use the Halton sequences generated by the base 3, the following is the chart which shows that the results are fairly good(totally 2000numbers)

1. The inverse transformation method

Through the above discussion, we can get the function of survival model and can generate the random value via the Halton sequences. Here I introduce the inverse transformation method to generate the time of death. The method of inverse transformation is as flows:

Let U1, U2,…,Un be an random sample of size n , ie, iid random variables, from the uniform distribution U(0,1). Let F be a distribution function and Then X1,X2,…,Xn is a random sample of size n form the distribution F. In detail, we can let

The universal product model and the cash flow testing with the QMC

1. The universal product model

Universal Life is a type of permanent life insurance based on a cash value. That is, the policy is established with the insurer where premium payments above the cost of insurance are credited to the cash value. The cash value is credited each month with interest, and the policy is debited each month by a cost of insurance (COI) charge, and any other policy charges and fees which are drawn from the cash value if no premium payment is made that month.

I built a model of an simple universal life product, the description of this product is as follows:

Age: From 20 to 60

Gender: Male

Premium: 10000

Policy term: 10 Years

Number of the insured: 2000

In each year, the Profit formula is as follows:

**Premium+ Investment Income – Acquisition cost – Maintenance Cost-Regulation Fee- Commission-Maturity Benefit – Death Benefit-Reserve Increase.**

I calculated the cash flow in each year and the Profit margin is the present value of the cash flow divided the total Premium obtained.

Some basic assumptions of the expense and the investment rate are as follows:

|  |  |  |
| --- | --- | --- |
| **Expense Allowance** |  |  |
| Acq Exp | Per Policy | 10.00 |
|  | % FYC | 0.0% |
| Main Exp | Per Policy | 20.00 |
|  | % Rsv | 0.50% |
| Comm |  | 4.200% |
| CIRC fees |  | 0.27% |
| Initial Fees |  | 5.0% |
| AV Management Fee |  | 0.0% |

|  |  |
| --- | --- |
| **Investment Assumption** | |
| year | interest rate |
| 1 | 4.30% |
| 2 | 4.45% |
| 3 | 4.60% |
| 4 | 4.75% |
| 5 | 4.75% |
| 6+ | 4.75% |

1. Using of the Quasi-Monte Carlo.

By using the Halton sequences, I Generated 2000 random numbers to simulate the 2000 policy holders. With the inverse transformation method, we can generate the death time of these policyholders to get the cash flow under the stochastic situations

I calculate the results of the deterministic model and the stochastic model under the 30,40,50,60 ages. The results are as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Age | 30 | 40 | 50 | 60 |
| Profit with simulation | 2.02% | 2.03% | 2.06% | 2.10% |
| Profit in deterministic | 2.12% | 2.11% | 2.08% | 1.99% |
| Related difference | -4.72% | -3.79% | -0.96% | 5.53% |

From the above table, we can see that the results of two methods are similar. Under the stochastic situation we may see the profit result in some extreme scenarios which is helpful for the insurance company to conduct risk management and do the stress testing.

1. The change of the cash flow pattern via using the stochastic model

In the deterministic model, we use the Chinese life table to get the mortality rate of each year. Hence the death benefit is smooth in each year, we can see it clearly in the following example. Let me assume that the mortality rate is equal to 0.01 in each year and the death benefit is 1000. So the death benefit inforce which is used in the profit calculation is as follows:

|  |  |  |
| --- | --- | --- |
| Year | Mortality Rate | Death Benefit |
| 1 | 0.01 | 10 |
| 2 | 0.01 | 10 |
| 3 | 0.01 | 10 |
| 4 | 0.01 | 10 |
| 5 | 0.01 | 10 |
| 6 | 0.01 | 10 |
| 7 | 0.01 | 10 |
| 8 | 0.01 | 10 |
| 9 | 0.01 | 10 |
| 10 | 0.01 | 10 |

However, in our stochastic model, we treat every policyholder independently. So the totally death benefit is only occur in a specific year, which give us a different cash flow pattern. Let see the following example. Assume the random number showed that the death occurred in the fourth year, the death benefit used for the cash flow is:

|  |  |  |
| --- | --- | --- |
| Year | Mortality Rate | Death Benefit |
| 1 | 0 | 0 |
| 2 | 0 | 0 |
| 3 | 0 | 0 |
| 4 | 1 | 1000 |
| 5 | 0 | 0 |
| 6 | 0 | 0 |
| 7 | 0 | 0 |
| 8 | 0 | 0 |
| 9 | 0 | 0 |
| 10 | 0 | 0 |

We can see only the fourth year, we will have the total death benefit and other years the benefit equals to zero. It give us a non smooth and discrete model which is more likely to the actually situation.

The simulation of the interest rate via the QMC method

In our universal product model, another important assumption is the interest rate which is used to get the present value of the future cash flow. What we used now is the constant 2.5%, **however the interest rate is not a constant and will be an stochastic process**. What the actuary call the scenario test is that change at least two assumptions and to see the influence of the profit.

1. The Vasicek model for the interest rate

Here we introduce the Vasicek interest rate model

dr_t = a(b-r_t)\, dt + \sigma \, dW_t

where *Wt* is a [Wiener process](http://en.wikipedia.org/wiki/Wiener_process) modeling the random market risk factor, in that it models the continuous inflow of randomness into the system. The [standard deviation](http://en.wikipedia.org/wiki/Standard_deviation) parameter, σ, determines the [volatility](http://en.wikipedia.org/wiki/Volatility_(finance)) of the interest rate and in a way characterizes the amplitude of the instantaneous randomness inflow. The typical parameters *b*,*a* and σ, together with the initial condition *r*0, completely characterize the dynamics, and can be quickly characterized as follows, assuming *a* to be non-negative:

* *b*: "long term mean level". All future trajectories of *r* will evolve around a mean level b in the long run;
* *a*: "speed of reversion". *a* characterizes the velocity at which such trajectories will regroup around *b* in time;
* σ: "instantaneous volatility", measures instant by instant the amplitude of randomness entering the system. Higher σ implies more randomness

From solving the stochastic differential equation, we can get that

 r(t) = r(0) e^{-a t} +  b \left(1- e^{-a t}\right) + \sigma e^{-a t}\int_0^t e^{a s}\,dW_s.\,\!

Using similar techniques as applied to the [Ornstein–Uhlenbeck](http://en.wikipedia.org/wiki/Ornstein%E2%80%93Uhlenbeck_process) stochastic process this has mean

E[*rt*] = *r*0*e* − *at* + *b*(1 − *e* − *at*)

and variance

\mathrm{Var}[r_t] = \frac{\sigma^2}{2 a}(1 - e^{-2at}).

1. Definition of the parameter

I get the historical interest rate in China via the website, the data is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Date | T in Year | Rate | change |
| 1998-7-1 | 0.000 | 4.77 |  |
| 1998-12-7 | 0.436 | 3.78 | -2.273 |
| 1999-6-10 | 0.942 | 2.25 | -3.019 |
| 2002-2-21 | 3.647 | 1.98 | -0.100 |
| 2004-10-29 | 6.334 | 2.25 | 0.100 |
| 2006-8-19 | 8.140 | 2.52 | 0.150 |
| 2007-3-18 | 8.718 | 2.79 | 0.467 |
| 2007-5-19 | 8.888 | 3.06 | 1.590 |
| 2007-7-21 | 9.060 | 3.33 | 1.564 |
| 2007-8-22 | 9.148 | 3.6 | 3.080 |
| 2007-9-15 | 9.214 | 3.87 | 4.106 |
| 2007-12-21 | 9.479 | 4.14 | 1.016 |
| 2008-10-9 | 10.282 | 3.87 | -0.336 |
| 2008-10-30 | 10.340 | 3.6 | -4.693 |
| 2008-11-27 | 10.416 | 2.52 | -14.079 |
| 2008-12-23 | 10.488 | 2.25 | -3.790 |

Here the change is the

a(b-rt) represents the expected change in the interest rate at t (drift factor)

I take the change as Y and the rt as x, can get the a,b from the formula Y=a\*b-ax

Form the regression model, we can get a=0.4974, b=2.3356

The result of the regression data are as follows:

|  |  |
| --- | --- |
| Regression analysis | |
| Multiple R | 0.123545 |
| R Square | 0.015263 |
| Adjusted R Square | -0.06049 |
| Standard error | 3.382722 |
| observation | 15 |

From the above interest rate data, we can also find the standard variance is equal to 0.8248

1. Simulation with the quasi-Monte Carlo method

From the formula:

dr_t = a(b-r_t)\, dt + \sigma \, dW_t

We can get

Zi

Here Zi follows the normal distribution.

Now I assume the R0 is 2.5% and is equal to 1 year, then simulate the future ten years interest rate use the quasi-Monte Carlo method.

The algorithm is as follows:

1. Let R0 equal to 2.5%
2. Generate the zi from the normal distribution.
3. Calculate the Ri with the formula Zi
4. Redo this for 1000 group
5. Calculate the expectation of the interest rate in each path

Form the simulation, I got the results as follows, the process can be seen in the model.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Time | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Rate | 2.417 | 2.376 | 2.355 | 2.344 | 2.340 | 2.337 | 2.335 | 2.334 | 2.333 | 2.334 |

With the interest rate, I redo the cash flow testing, and get the results as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Age | 30 | 40 | 50 | 60 |
| Profit with simulation rate | 2.03% | 2.04% | 2.07% | 2.11% |
| Profit with constant rate | 2.02% | 2.03% | 2.06% | 2.10% |

We can see that the under the simulation rate, the profit is relatively higher than the constant situation.

Conclusion and Improvements

1. Conclusions

* The survival model follows the function
* The Halton sequences give a good random number generator.
* The stochastic model can change the profit pattern more actually.
* The formula for the interest rate is

Zi

* The result of the stochastic model and the deterministic model are similar, however the stochastic under the quasi-Monte Carlo method can reflect the reality more clearly and can reflect the bad situation which is helpful to do the stress test.

1. Improvements

* Can use other Monte Carlo method to generate the random number and compared the results
* Can do more test under different samples numbers such as using 3000,4000, etc instead of 2000 to see the influence of the sample numbers.
* Can use another method to get the survival model function rather than the regressing method.
* Can test the difference of the speed between the deterministic and stochastic model.

Reference:

[1] Michael G. Hilgers, 2000, Quasi-Monte Carlo method in cash flow testing simulation, Proceedings of the 2000 winter simulation conference

[2] Bratley, P, B. L. Fox, 1992. Implementation and tests of low-discrepancy sequences. ACM Transactions on modeling and computer Simulation.

[3] Halton,J.H. 1960. On the efficiency of certain quasi- random sequences of points in evaluating multi-dimensional integrals. Numerical Mathematics 2:84-90

Attachment

**China life table**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Life | | | Annuity(yearly） | | |
|  | CL1 | CL2 | CL3 | CL4 | CL5 | CL6 |
| Age | Male | Female | Unisex | Male | Female | Unisex |
| 0 | 3.037 | 2.765 | 2.909 | 2.733 | 2.489 | 2.618 |
| 1 | 2.157 | 1.859 | 2.016 | 1.941 | 1.673 | 1.814 |
| 2 | 1.611 | 1.314 | 1.470 | 1.450 | 1.183 | 1.323 |
| 3 | 1.250 | 0.966 | 1.114 | 1.125 | 0.869 | 1.003 |
| 4 | 1.000 | 0.734 | 0.872 | 0.900 | 0.661 | 0.785 |
| 5 | 0.821 | 0.573 | 0.702 | 0.739 | 0.516 | 0.632 |
| 6 | 0.690 | 0.458 | 0.579 | 0.621 | 0.412 | 0.521 |
| 7 | 0.593 | 0.375 | 0.489 | 0.534 | 0.338 | 0.440 |
| 8 | 0.520 | 0.315 | 0.421 | 0.468 | 0.284 | 0.379 |
| 9 | 0.468 | 0.274 | 0.374 | 0.421 | 0.247 | 0.337 |
| 10 | 0.437 | 0.249 | 0.346 | 0.393 | 0.224 | 0.311 |
| 11 | 0.432 | 0.240 | 0.339 | 0.389 | 0.216 | 0.305 |
| 12 | 0.458 | 0.248 | 0.356 | 0.412 | 0.223 | 0.320 |
| 13 | 0.516 | 0.269 | 0.396 | 0.464 | 0.242 | 0.356 |
| 14 | 0.603 | 0.302 | 0.457 | 0.543 | 0.272 | 0.411 |
| 15 | 0.706 | 0.341 | 0.529 | 0.635 | 0.307 | 0.476 |
| 16 | 0.812 | 0.382 | 0.602 | 0.731 | 0.344 | 0.542 |
| 17 | 0.907 | 0.421 | 0.670 | 0.816 | 0.379 | 0.603 |
| 18 | 0.981 | 0.454 | 0.724 | 0.883 | 0.409 | 0.652 |
| 19 | 1.028 | 0.481 | 0.762 | 0.925 | 0.433 | 0.686 |
| 20 | 1.049 | 0.500 | 0.778 | 0.944 | 0.450 | 0.700 |
| 21 | 1.048 | 0.511 | 0.784 | 0.943 | 0.460 | 0.706 |
| 22 | 1.030 | 0.517 | 0.780 | 0.927 | 0.465 | 0.702 |
| 23 | 1.003 | 0.519 | 0.767 | 0.903 | 0.467 | 0.690 |
| 24 | 0.972 | 0.519 | 0.752 | 0.875 | 0.467 | 0.677 |
| 25 | 0.945 | 0.519 | 0.738 | 0.851 | 0.467 | 0.664 |
| 26 | 0.925 | 0.520 | 0.728 | 0.833 | 0.468 | 0.655 |
| 27 | 0.915 | 0.525 | 0.727 | 0.824 | 0.473 | 0.654 |
| 28 | 0.918 | 0.533 | 0.730 | 0.826 | 0.480 | 0.657 |
| 29 | 0.933 | 0.546 | 0.743 | 0.840 | 0.491 | 0.669 |
| 30 | 0.963 | 0.566 | 0.773 | 0.867 | 0.509 | 0.696 |
| 31 | 1.007 | 0.592 | 0.809 | 0.906 | 0.533 | 0.728 |
| 32 | 1.064 | 0.625 | 0.855 | 0.958 | 0.563 | 0.770 |
| 33 | 1.136 | 0.666 | 0.910 | 1.022 | 0.599 | 0.819 |
| 34 | 1.222 | 0.714 | 0.976 | 1.100 | 0.643 | 0.878 |
| 35 | 1.321 | 0.772 | 1.057 | 1.189 | 0.695 | 0.951 |
| 36 | 1.436 | 0.838 | 1.146 | 1.292 | 0.754 | 1.031 |
| 37 | 1.565 | 0.914 | 1.249 | 1.409 | 0.823 | 1.124 |
| 38 | 1.710 | 1.001 | 1.366 | 1.539 | 0.901 | 1.229 |
| 39 | 1.872 | 1.098 | 1.497 | 1.685 | 0.988 | 1.347 |
| 40 | 2.051 | 1.208 | 1.650 | 1.846 | 1.087 | 1.485 |
| 41 | 2.250 | 1.331 | 1.812 | 2.025 | 1.198 | 1.631 |
| 42 | 2.470 | 1.468 | 1.993 | 2.223 | 1.321 | 1.794 |
| 43 | 2.713 | 1.620 | 2.193 | 2.442 | 1.458 | 1.974 |
| 44 | 2.981 | 1.790 | 2.409 | 2.683 | 1.611 | 2.168 |
| 45 | 3.276 | 1.979 | 2.658 | 2.948 | 1.781 | 2.392 |
| 46 | 3.601 | 2.188 | 2.933 | 3.241 | 1.969 | 2.640 |
| 47 | 3.958 | 2.420 | 3.231 | 3.562 | 2.178 | 2.908 |
| 48 | 4.352 | 2.677 | 3.558 | 3.917 | 2.409 | 3.202 |
| 49 | 4.784 | 2.962 | 3.925 | 4.306 | 2.666 | 3.533 |
| 50 | 5.260 | 3.277 | 4.322 | 4.734 | 2.949 | 3.890 |
| 51 | 5.783 | 3.627 | 4.770 | 5.205 | 3.264 | 4.293 |
| 52 | 6.358 | 4.014 | 5.263 | 5.722 | 3.613 | 4.737 |
| 53 | 6.991 | 4.442 | 5.790 | 6.292 | 3.998 | 5.211 |
| 54 | 7.686 | 4.916 | 6.367 | 6.917 | 4.424 | 5.730 |
| 55 | 8.449 | 5.440 | 7.005 | 7.604 | 4.896 | 6.305 |
| 56 | 9.288 | 6.020 | 7.735 | 8.359 | 5.418 | 6.962 |
| 57 | 10.210 | 6.661 | 8.524 | 9.189 | 5.995 | 7.672 |
| 58 | 11.222 | 7.370 | 9.386 | 10.100 | 6.633 | 8.447 |
| 59 | 12.333 | 8.154 | 10.349 | 11.100 | 7.339 | 9.314 |
| 60 | 13.553 | 9.022 | 11.378 | 12.198 | 8.120 | 10.240 |
| 61 | 14.892 | 9.980 | 12.508 | 13.403 | 8.982 | 11.257 |
| 62 | 16.361 | 11.039 | 13.779 | 14.725 | 9.935 | 12.401 |
| 63 | 17.972 | 12.209 | 15.167 | 16.175 | 10.988 | 13.650 |
| 64 | 19.740 | 13.502 | 16.672 | 17.766 | 12.152 | 15.005 |
| 65 | 21.677 | 14.929 | 18.275 | 19.509 | 13.436 | 16.448 |
| 66 | 23.800 | 16.505 | 20.107 | 21.420 | 14.855 | 18.096 |
| 67 | 26.125 | 18.244 | 22.111 | 23.513 | 16.420 | 19.900 |
| 68 | 28.671 | 20.162 | 24.315 | 25.804 | 18.146 | 21.884 |
| 69 | 31.457 | 22.278 | 26.701 | 28.311 | 20.050 | 24.031 |
| 70 | 34.504 | 24.610 | 29.296 | 31.054 | 22.149 | 26.366 |
| 71 | 37.835 | 27.180 | 32.152 | 34.052 | 24.462 | 28.937 |
| 72 | 41.474 | 30.009 | 35.305 | 37.327 | 27.008 | 31.775 |
| 73 | 45.446 | 33.123 | 38.746 | 40.901 | 29.811 | 34.871 |
| 74 | 49.779 | 36.549 | 42.465 | 44.801 | 32.894 | 38.219 |
| 75 | 54.501 | 40.313 | 46.582 | 49.051 | 36.282 | 41.924 |
| 76 | 59.644 | 44.447 | 51.078 | 53.680 | 40.002 | 45.970 |
| 77 | 65.238 | 48.984 | 55.926 | 58.714 | 44.086 | 50.333 |
| 78 | 71.317 | 53.958 | 61.236 | 64.185 | 48.562 | 55.112 |
| 79 | 77.916 | 59.405 | 66.958 | 70.124 | 53.465 | 60.262 |
| 80 | 85.069 | 65.364 | 73.092 | 76.562 | 58.828 | 65.783 |
| 81 | 92.813 | 71.876 | 79.823 | 83.532 | 64.688 | 71.841 |
| 82 | 101.184 | 78.981 | 87.192 | 91.066 | 71.083 | 78.473 |
| 83 | 110.218 | 86.722 | 95.102 | 99.196 | 78.050 | 85.592 |
| 84 | 119.951 | 95.145 | 103.653 | 107.956 | 85.631 | 93.288 |
| 85 | 130.418 | 104.291 | 112.976 | 117.376 | 93.862 | 101.678 |
| 86 | 141.651 | 114.207 | 123.047 | 127.486 | 102.786 | 110.742 |
| 87 | 153.681 | 124.933 | 133.927 | 138.313 | 112.440 | 120.534 |
| 88 | 166.534 | 136.511 | 145.631 | 149.881 | 122.860 | 131.068 |
| 89 | 180.233 | 148.980 | 158.079 | 162.210 | 134.082 | 142.271 |
| 90 | 194.795 | 162.374 | 171.599 | 175.316 | 146.137 | 154.439 |
| 91 | 210.233 | 176.721 | 185.702 | 189.210 | 159.049 | 167.132 |
| 92 | 226.550 | 192.046 | 200.967 | 203.895 | 172.841 | 180.870 |
| 93 | 243.742 | 208.364 | 217.252 | 219.368 | 187.528 | 195.527 |
| 94 | 261.797 | 225.680 | 234.450 | 235.617 | 203.112 | 211.005 |
| 95 | 280.694 | 243.992 | 253.233 | 252.625 | 219.593 | 227.910 |
| 96 | 300.399 | 263.285 | 272.344 | 270.359 | 236.957 | 245.110 |
| 97 | 320.871 | 283.531 | 292.664 | 288.784 | 255.178 | 263.398 |
| 98 | 342.055 | 304.690 | 314.651 | 307.850 | 274.221 | 283.186 |
| 99 | 363.889 | 326.708 | 336.441 | 327.500 | 294.037 | 302.797 |
| 100 | 386.299 | 349.518 | 358.080 | 347.669 | 314.566 | 322.272 |
| 101 | 409.200 | 373.037 | 381.455 | 368.280 | 335.733 | 343.310 |
| 102 | 432.503 | 397.173 | 405.397 | 389.253 | 357.456 | 364.857 |
| 103 | 456.108 | 421.820 | 429.801 | 410.497 | 379.638 | 386.621 |
| 104 | 479.911 | 446.863 | 454.556 | 431.920 | 402.177 | 409.100 |
| 105 | 1,000.000 | 1,000.000 | 1,000.000 | 1,000.000 | 1,000.000 | 1,000.000 |