LongV$_\sigma$ega.com

Financial Engineering Tool

Replication Strategy and Algorithm

E-Mail: feedback@longvega.com

Contents

Abstract

This is the "paper work" behind the Financial Engineering tool which is available on www.longvega.com. This is not a scientific paper. The reason why I wrote it is simply because I needed to construct the replication algorithm which is used by the tool and I needed to write things down clearly. I am aware of the fact that this paper is sometimes trivial. Some simple mathematical framework needed to be used to describe the payoffs accurately. Math in this work is used to describe payoffs and not as a toolbox for the solution of the problems.

The Financial Engineering tool automatically replicates and prices a given continuous piecewise linear payoff function. So far the tool can only handle payoffs on a stock, where the payoff is denominated in the same currency as the stock. Cross currency payoffs with a guaranteed exchange rate (quantos) or flexible exchange rates might be implemented in the future. The following questions needed to be answered, before starting to implement the replication algorithm:

- Can any continuous piecewise linear payoff function be replicated by a portfolio of options?
- If so, how many replicating portfolios do exist for a specified payoff function?
- Which instruments are included in this portfolio?

The result of this paper is an algorithm, which describes the replication procedure of any general continuous piecewise linear payoff on a stock which has discrete stock prices. Let me emphasize again, that I don't build an accurate model for a market with discrete stock prices from the theoretical point of view. I rather do it heuristically from a practical point of view. Furthermore I will state heuristically, that we can approximate any piecewise linear payoff function with a countable infinite number of portfolios, which only include calls. This statement assumes that the considered payoff function is not trivial, e.g. not constant zero. No other instruments will be needed for the approximation, so the calls are our basic tool.

1 Basics

1.0.1 Payoff Functions

We will define a simple payoff function in a real financial market. A payoff function in this work describes the payoff of a single derivative, or a portfolio of derivatives at maturity (in case of a portfolio, all derivatives in the portfolio must have the same maturity). We will look at a derivative or a portfolio of derivatives which are not path-dependent and whose underlying is a stock S which is denominated in the same currency as the payoff. This portfolio can for example be a replicated structured equity product or an option-position like a Straddle or Strangle. Furthermore we assume that the portfolio, which belongs to the payoff function, does not have any payoffs during the tenor of the options. This restriction has a significant impact on our replication strategy: we won't use the underlying as a replication component, since stocks can pay dividends during a predefined timeframe.
This restriction allows us to implement a general tool which can replicate payoffs, where S is an underlying which pays dividends or doesn't. This assumption is often met by structured equity products, where the dividend "is used" to finance the structure. The assumption above does not mean, that we can't price payoffs of options, where the underlying is a stock which pays dividends. It just says, that our portfolio doesn't pay any dividends, since it consists of options. The Financial Engineering tool is able to price all structured equity products which meet the requirements above.

The prices of stocks are always ≥ 0. Since this paper is focused on the deduction of a practical algorithm, we will look at real stock prices in a non-continuous world. Let D be the set of possible payoffs of a position in derivatives, with the derivatives having the same maturity T. We define without loss of generality $D := \{k * 0.01; k \in \mathbb{Z}\}$, $D^+ := \{k * 0.01; k \in \mathbb{N}\}$, $\mathbb{N}$ being the set of positive integers, 0 included. We are looking at a stock, which has the price $S(t)$ at time t, with $S(t) \in D^+$, for instance $S(t) = 34.56\$ = 3456 * 0.01\$$. Stock prices can't be negative, so D^+ is an appropriate set for stock prices in the real financial market. A payoff can be negative, for example if you have a short position in a call.

Definition 1. A payoff function in this paper is a mapping $f : D^+ \rightarrow D$. We claim that $f(x)$ is defined $\forall x \in D^+$, e.g. if there is no payoff for some $x \in D^+$, the payoff function will be zero.

An example of a payoff function is $f(S(T)) = max(S(T) - K, 0), K \in D^+$ which is the payoff of a plain vanilla call. Payoff functions can be represented in charts. Figure (1.1) shows a classical Strangle position.

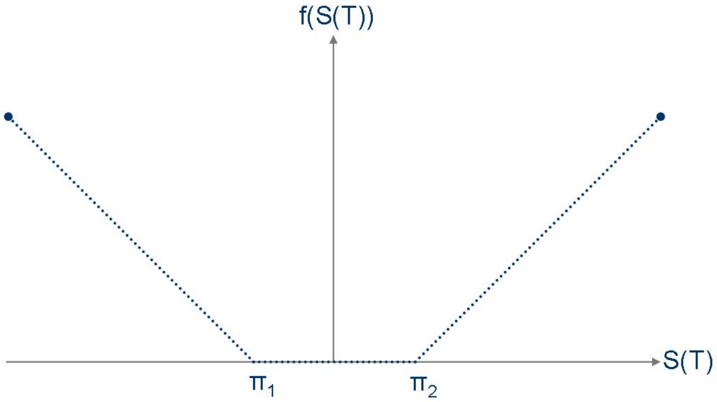

Figure 1.1: Payoff of a Strangle

1.0.2 Approximations

We will use a small set of tools to replicate continuous piecewise linear payoff functions. The put-call-parity tells us, how to construct a put with a stock, a call and a zerobond, so we don't need the put in our consideration. We need the stock and the bond. Do we really? Indeed we can approximate both instruments by other instruments. We won't describe the basics of option theory but start in "the middle of the battle area".
Let us assume a strike which is very close to zero in the Black-Scholes world. The equations below (see [3], page 192) show the price for a plain vanilla call and a digital call, paying L, if the option is in the money at maturity.

$$C(t,K,S) = Se^{-d(T-t)}N(d_1) - K^{-r(T-t)}N(d_2) \tag{1.1}$$

$$C(t,K,S,L)^{dig} = Le^{-r(T-t)}N(d_2) \tag{1.2}$$

$$d_1 = \frac{ln(\frac{S}{K}) + (r-d+\frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}, d_2 = d_1 - \sigma\sqrt{T-t}$$

We see, that the Black Scholes formula only accepts strikes greater than zero, otherwise we would devide by zero in (S/K). This doesn't limit our possible payoffs, since lognormal stock prices take only values greater than zero anyways, if the stock isn't worthless from the beginning. What happens, if the strike is very close to zero (eg. approaching zero from the right)?

From the numerical point of view we see, that the second part of equation (1.1) will become zero ($N(d_2)$) approaches 1 since $ln(S/K)$ approaches infinity), and the first part will approach $Se^{-d(T-t)}$(because $N(d_1)$ approaches 1 as well). Let us look at the call in an intuitive way. It can be seen as a portfolio with a long position in an "asset or nothing call" and a short position in a "cash or nothing" call with a digital payoff of K if the stock price is above strike K at maturity (see Crack, page 142 [2]). If K is close to zero, the price of the short position in the digital call will be close to zero as well. Why paying for a call which pays almost nothing if the stock is in the money and zero otherwise? The price of the asset or nothing call will approach $Se^{-d(T-t)}$ since it is equivalent to the right to receive the stock at maturity if the stock price is above a number close to zero. How much would you have to pay for a dividend paying stock

today at time t, to hold 1 unit of it at T? Obviously not $S(t)$, since you can use the dividend during the timeframe to buy more stock. You would pay $Se^{-d(T-t)}$ at time t. If you would like to own one unit of stock at maturity, you can buy $Se^{-d(T-t)}$ units of stock now and reinvest the dividend continuously in new stock shares or you can buy a call with a strike close to zero now, which is worth $Se^{-d(T-t)}$ as well and you will receive one unit of the stock at maturity. Obviously the call with a strike close to zero has the same price as the stock in case of a dividend yield of zero, which is $S(t)$ at time t. We just approximated the dividend paying stock in our put-call parity by a call with a strike close to zero.
A similar discussion will replace the bond with a digital call with a strike close to zero and a digital payoff, which has the same value as the nominal of the bond. You can see in Crack (page 142 [2]) that $N(d_2)$ in equation (1.2) is the risk-neutral probability of the stock finishing in the money. The risk-neutral probability that the stock finishes above zero is one for a lognormal distributed stock price (assuming a bond as a numeraire). The value of a digital call with a digital payoff of K approaches $Ke^{-r(T-t)}$ as the strike approaches zero. Note that the strike and digital payoff are now different. This price is equivalent to the price of a zero bond with a nominal of K at time t.

Until now we have reduced our universe of replication tools to plain vanilla and digital calls. Note that we are approximating and not replacing the other instruments. We can do it as close as we want in the continuous world. A digital call is not plain vanilla, but we can again approximate this call by a portfolio of plain vanilla calls (see [1], page 293). This means: In the theoretical world we can approximate all instruments with a portfolio of plain vanilla calls. If the Black-Scholes Formula would allow a strike of zero, than we wouldn't need approximations, but replace the bond and the stock by calls with a strike of zero.

What about our D^+ world? Continuous stock prices are nice. If we take a stock price close to zero, we will find infinitely many which are even more close to zero. Thus we can always make (S/X) big enough to let $ln(S/X)$ approach infinity and consequently $N(d_1)$ approach 1. This is not possible for discrete stock prices in D^+, although in most cases it is a very good approximation. The accuracy of the approximation breaks down, if the stock becomes almost worthless, as $ln(S/X)$ is not large enough to make $N(d_1)$ be close enough to one. Let's take a look at two examples.

Example 1 (Approximation of the stock)
We assume a stock price of 100$ and a strike of 0.01$, a dividend yield of 0.02 and a continuously compounded interest rate of 0.03. Furthermore we assume the call with the above attributes to mature in 1 year. This call would cost 98.01$, where $Se^{-d(T-t)}$ would result in 98.02$. If we take 1$ as the stock price, we would get 0.9802$ for $Se^{-d(T-t)}$, but only 0.9792$ for the option. If we take 0.001$ as the strike, the approximation would be better.

Example 2 (Approximation of the bond)
We assume the same data as above, a spot stock price of 100$ and a strike of 0.01$. A digital call with a digital payoff of 100$ with the above attributes would cost 97.04$, a bond with a nominal of 100$ would cost 97.04$ as well. And now the surprise: If we assume the stock to be at 0.10$, the value of this digital would still be 97.04. It is not

as sensitive to stock prices as the plain vanilla call and a very good approximation to the price of a zerobond.

The Financial Engineering tool uses the call with a strike close to zero as an approximation for the underlying to avoid a component, which pays dividends during the tenor. We rather "invest" the present value of the dividends in other components.

2 Static Replication

2.0.3 Replication of a Call

Is a call the smallest unit in the "universe of replication tools" or can you even replicate this small unit again? Yes you can. Although the following replications are trivial and artificial, we have to take them into consideration to cover all aspects of replication strategies.
Every call on $S(t) \in D^+$ can be statically replicated by an infinite number of portfolios. By infinite we mean countable infinite. The number of replicating portfolios is equal to the cardinal number of D^+. How comes? For simplicity we from now on will focus on the strike, thus omitting the time component and the stock in $C(t, K, S)$ and write $C(K)$ instead. Let's start simple. We can replicate a call with strike K by:

- bying $C(K + x), x \in D^+$
- bying a zerobond with a nominal of x
- selling $P(K + x)$, so a put with strike $K + x$
- bying $P(K)$ which is a put with strike K

This portfolio has the following payoff at maturity:

$$max\{S(T) - (K + x), 0\} + x - max\{(K + x) - S(T), 0\} + max\{K - S(T), 0\}$$

One can proof that this portfolio has the same payoff as $C(K)$. Since we can take any $x \in D^+$ to achieve a different replication, the number of replicating portfolios will have the same cardinality as the set D^+, which is countable infinite many. What if we use our approximations from the previous chapter? Than we obviously can approximate the instruments above with calls. We are approximating a call with a portfolio of calls. To summarize it: we see, a call is not the smallest unit in the universe of replication tools. Even a simple call can be approximated by an infinite number of portfolios, which include only calls. In the option replication world, there is no such a unit like an elektron in Physics. A call is rather like a fractal: if we zoom in, we see, that it is artificially constructed by other calls, and these calls are again constructed by other calls. The result of this discussion is: If we have a payoff function which can be replicated by a portfolio which includes at least one call (and if it is only one call), than this payoff function has an infinite number of replicating portfolios. A put can be replicated with a portfolio, which includes a call. The put can be approximated countable infinite many times.

Why do I write down this rather trivial and artificial considerations? Before programming the tool I needed to think about the number of possible replications of a special payoff. I realized, that there are infinite many. I wanted the tool to show more than just one and for a good reason not all possible replications. In an arbitrage free market the payoffs all have the same price, so why bothering about different portfolios, is one not enough? Let's take a closer look at a protective put, protected at 100% of the spot price of the stock, see chart (2.1). According to the put-call parity this position can be replicated by a stock and a put or a zerobond and a call. If an investment bank sells such a structure (a Capital Protected Note), it needs to hedge the position to get rid of the risk. Hedging the first position might be cheaper than hedging the second position. There might be higher costs associated with hedging the stock than hedging the zero bond. This is the reason, why the tool shows different replication strategies.

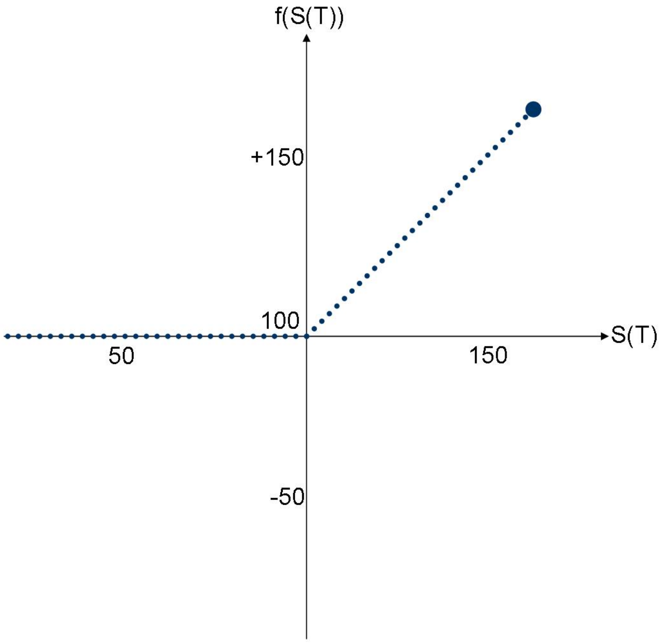

Figure 2.1: Payoff of a Protective Put

2.0.4 Replication of Continuous Piecewise Linear Payoff Functions

Non linear payoff functions are rare in practical Financial Engineering. The most common payoff functions are piecewise linear functions, such as butterflies, condors, covered calls and others. These payoffs of course have convexities during the tenor, but not at maturity(although piecewise linear payoffs can be convex as well). There are non-linear payoffs, like payoffs of power options. In contrast our restriction to non-path dependent payoffs is really a restriction, since many structured products use path-dependent payoffs with knock-out or best-off features. This is something that might be added to the tool in the future, although some more work needs to be done to accomplish that task.

In this section, we will derive an algorithm, which will produce "non-artificial"' replicating portfolios for continuous piecewise linear functions. This algorithm is used in the Financial Engineering Tool which is available on `www.longvega.com`. What do we mean with a non-artificial replicating portfolio? Since there is an infinite number of portfolios for all payoffs, we need to select some of them, which make sence from a practical point of view. Let's take a look at an example, the strangle payoff function of figure(2.2) which I show again for convenience and because I invested some time and love to draw it.

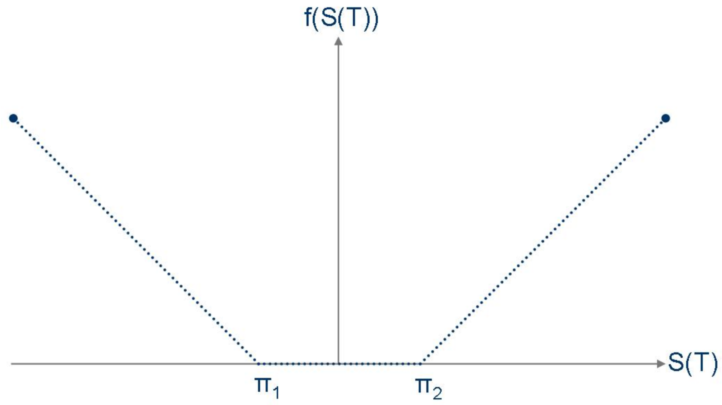

Figure 2.2: Payoff of a Strangle

This payoff function can be replicated by:

- bying $C(\pi_2)$, which is a call with strike π_2
- bying $P(\pi_1)$, which is a put with strike π_1

Is there any other replication? We see that this payoff has a call in it, so according to our result in the last chapter there are infinite many different replications, but another replication would be:

- bying a bond with a nominal of π_1
- selling a plain vanilla call with a strike close to 0
- bying a plain vanilla call with strike π_1
- bying $C(\pi_2)$

We should focus on replicating portfolios which include options, whose strikes are at the "kinks" in the payoff function. The Financial Engineering tool is in fact a "Kink Engineering tool". Usually the kinks are designed with a purpose. If for example an investor would like to have a leverage from a special strike, the payoff function would have a kink at this strike. The replication portfolio will include a long position in a call with this strike. If we would replicate with some random strikes, like the call with strike $K+x$ in the previous section, the intention behind the position wouldn't be clear.

As we see, a general, accurate algorithm for all replications for continuous piecewise linear payoffs would be very useful. Let's tackle this challenge now. I won't define a continuous piecewise linear function formally here since it is a simple function but written down it looks more complicated than necessary. Let π_i be "our kinks" for

$i = 0, ...k$, so exactly $(k+1)$ kinks. We will assume a kink at 0, although this is not really a kink but just the starting point of the payoff. In the following algorithm we will take long or short positions in different fractions of calls respective puts. If the fraction is positive, it means that we're long the instrument. If the fraction is negative, we're short. So if we say "buy -1" call, we actually short the call.

To point it out: the following algorithm will produce $(k+1)$ or less different replicating portfolios for a continuous piecewise linear payoff with $k+1$ kinks. The algorithm automatically terminates because we assumed a finite number of kinks.

2.0.5 Algorithm

Set $\lambda_i \quad := \frac{f(\pi_{i+1})-f(\pi_i)}{\pi_{i+1}-\pi_i}$ for $i = 0, ..., k-1$
Set $\lambda_k \quad := \frac{f(\pi_k+0.01)-f(\pi_k)}{0.01}$
Set $s_{f(\pi_i)} := sign(f(\pi_i)) \in \{-1, 1\}$
Set $s_{\lambda_i} \quad := sign(\lambda_i) \in \{-1, 1\}$

STEP 0(INITIALIZATION)

START **Set** $i = 0; i_r = 0; i_l = 0$

STEP 1 (LEVELING)

If	$f(\pi_i) = 0$	Go to Step 2
Else		Buy $s_{f(\pi_i)}$ fractions of a zerobond with a nominal of $\lvert f(\pi_i)\rvert$. Go to Step 2
EndIf		

STEP 2(REPLICATION TO THE RIGHT)

If	$i_r = i$	buy λ_i calls with strike π_{i_r}
ElseIf	$i_r \leq k$	buy $(\lambda_{i_r} - \lambda_{i_{r-1}})$ calls with strike π_{i_r}
Else		Go to Step 3
EndIf		
		Set $i_r = i_r + 1$, go to Step 2

STEP 3(REPLICATION TO THE LEFT)

If	$i_l = 0$	Go to Step 4
ElseIf	$i_l = i$	buy $-\lambda_{i_{l-1}}$ puts with strike π_{i_l}
Else		buy $(\lambda_{i_l} - \lambda_{i_{l-1}})$ puts with strike π_{i_l}
EndIf		
		Set $i_l = i_l - 1$, go to Step 3

STEP 4

If	$i < k$	Save portfolio, set $i = i + 1, i_r = i, i_l = i$, Go to Step 1
Else		STOP
EndIf		

Example

Suppose we would like to replicate the Bear Spread in the figure below.

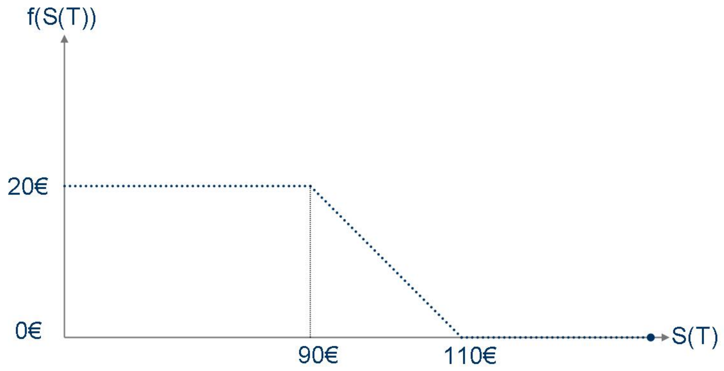

Figure 2.3: Payoff of a Bear Spread

This payoff has the following kinks, respective values:

$\pi_0 = 0$	$f(\pi_0) = 20$	$s_{f(\pi_0)} = 1$
$\pi_1 = 90$	$f(\pi_1) = 20$	$s_{f(\pi_1)} = 1$
$\pi_2 = 110$	$f(\pi_2) = 0$	$s_{f(\pi_2)} = 1$

We than calculate:

$\lambda_0 = 0$	$s_{\lambda_0} = 1$
$\lambda_1 = -1$	$s_{\lambda_1} = -1$
$\lambda_2 = 0$	$s_{\lambda_2} = 1$

We proceed with the algorithm:

STEP 0(INITIALIZATION)

- $i = 0; i_r = 0; i_l = 0$

STEP 1 (LEVELING)

- $f(\pi_0) = 20$, so not equal 0, so buy $s_{f(\pi_0)} = 1$ zerobond with a nominal of $f(\pi_0) = 20$.
- Go to step 2.

STEP 2(REPLICATION TO THE RIGHT)

- $i_r = 0$ and $i = 0$, so we buy $\lambda_0 = 0$ calls with a strike of $\pi_0 = 0$. So we don't buy anything at all. We set $i_r = 1$ and go again to step 2.
- This time $i_r = 1$ and i is still 0. So we are getting into the ElseIf block. Since $k = 2$ and $i_r = 1$ we buy $(\lambda_1 - \lambda_0) = -1$ calls with a strike of $\pi_1 = 90$. So we short a call with a strike at 90. We set $i_r = 2$ and go back again.

- Since $k = 2$ and $i_r = 2$ we buy $(\lambda_2 - \lambda_1) = 0 - (-1) = 1$ call with a strike of $\pi_2 = 110$.
- according to the algorithm we than have to go to Step 3.

STEP 3(REPLICATION TO THE LEFT)

- i_l is still 0, so we go to step 4.

STEP 4

- $i = 0$ which is smaller than $k = 2$, so we set $i = 1$,$i_r = 1$,$i_l = 1$
- our first portfolio is finished, we then go to step 1

We summarize our first Portfolio:

- 1 zerobond with a nominal of 20
- -1 call with a strike of $= 90$
- 1 call with a strike of 110

We proceed with the algorithm:

STEP 1 (LEVELING)

- $f(\pi_1) = 20$, so not equal 0, so buy $s_{f(\pi_1)} = 1$ zerobond with a nominal of $f(\pi_1) = 20$.
- Go to Step 2.

STEP 2(REPLICATION TO THE RIGHT)

- $i_r = 1$ and $i = 1$, so we buy $\lambda_1 = -1$ calls with a strike of $\pi_1 = 90$. We set $i_r = 2$ and go again to step 2.
- This time $i_r = 2$ and i is still 1. So we are getting into the ElseIf block. Since $k = 2$ and $i_r = 2$ we buy $(\lambda_2 - \lambda_1) = 1$ calls with a strike of $\pi_2 = 110$.
- according to the algorithm we than have to go to Step 3.

STEP 3(REPLICATION TO THE LEFT)

- $i_l = 1$ and $i = 1$, so we are getting into the ElseIf block.
- we buy $-\lambda_0 = 0$ puts with a strike of $\pi_1 = 90$, so we don't buy anything at all.
- we set $i_l = 0$ and go back to step 3
- $i_l = 0$ and the If statement tells us to go to step 4

STEP 4

- $i = 1$ which is smaller than $k = 2$, so we set $i = 2$,$i_r = 2$,$i_l = 2$
- our second portfolio is finished, we then go to step 1

We summarize our second Portfolio:

- 1 zerobond with a nominal of 20
- -1 call with a strike of $= 90$
- 1 call with a strike of 110

We proceed with the algorithm:

STEP 1 (LEVELING)

- $f(\pi_2) = 0$, so we don't need to level and go to step 2

STEP 2(REPLICATION TO THE RIGHT)

- $i_r = 2$ and $i = 2$, so we buy $\lambda_2 = 0$ calls with a strike of $\pi_2 = 110$, we don't buy anything.
- We set $i_r = 3$ and go again to step 2.
- according to the algorithm we get into the Else block and than have to go to step 3.

STEP 3(REPLICATION TO THE LEFT)

- $i_l = 2$ and $i = 2$, so we buy $-\lambda_1 = -(-1) = 1$ put with strike $\pi_2 = 110$.
- we set $i_l = 1$ and go to step 3
- this time we end up in the else block and buy $(\lambda_1 - \lambda_0) = -1 - 0 = -1$ puts with strike $\pi_1 = 90$
- we set $i_l = 0$ and go to step 3
- we have to leave that block and go to step 4

STEP 4

- $i = 2$ which is not smaller than $k = 2$, we have to go to the Else block
- STOP tells us to leave the whole algorithm

We summarize our third Portfolio:

- 1 put with strike 110.
- -1 put with strike 90

We're done. We can see, that the first and second portfolio are equal. This is, because the payoff at the kinks at $\pi_0 = 0$ and $\pi_1 = 90$ has the same level. This is the reason, why I wrote before, that the algorithm results in $k+1$ or less *different* portfolios. The next page shows a program flowchart for the algorithm.

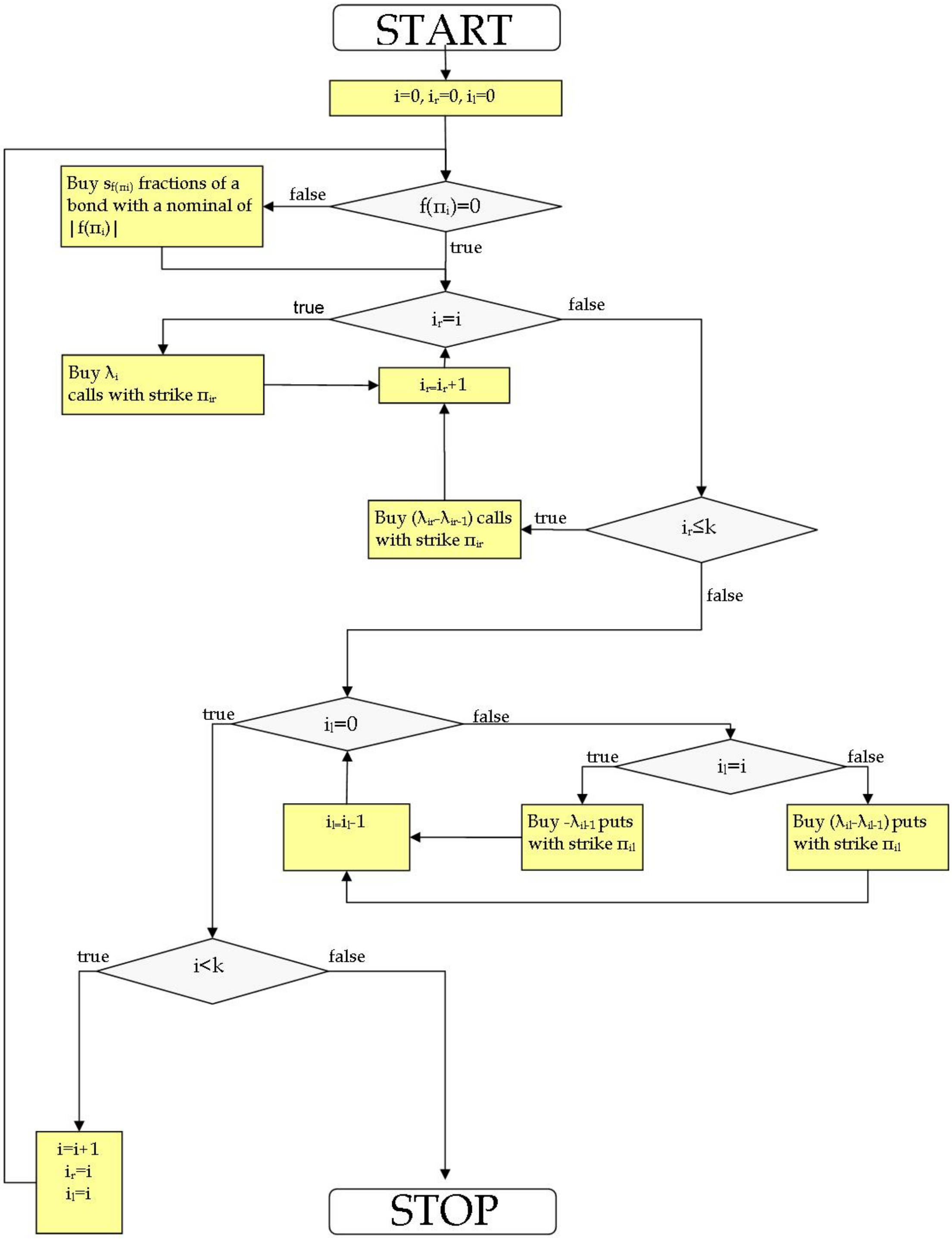

Figure 2.4: Program Flowchart

Bibliography

[1] Salih. N Neftci, *Principles of Financial Engineering*, Elsevier Academic Press, 2004

[2] Timothy Falcon Crack, *Basic Black-Scholes: Option Pricing and Trading*, 2004

[3] Paul Wilmott, *Quantitative Finance*, John Wiley Sons Ltd.,2001