Bootstrapping of Commodity Forward Prices

Jake C. Fowler

January 2023

THIS DOCUMENT IS CURRENTLY WORK IN PROGRESS

1 Introduction

This document presents an algorithm for the boostrapping commodity forwards, futures and swap contracts.

It is applicable for linear commodity contracts contingent on either a continuous delivery of physical commodity, financial payoff based on the average of an index throughout the contract delivery period.

The rest of this document will refer to just forward prices and curves, even though the same methodology can be applied to futures and swap instruments as well.

Also the document refers to the *delivery period* of the contracts, in place of *delivery or fixing period* where *fixing period* is the set of time periods from which the average price of an index is used to calculate the payoff of a swap contract.

2 No-Arbitrage Forward Price Condition

This section describes the no-arbitrage relationship between the set of prices of forward contracts for a consecutive delivery period and price of a single lower granularity contract which spans the whole delivery period of these. It is this relationship which forms the basis of the boostrapping algorithm.

Denote:

- F as a forward price of the *big* contract. The notation does not contain the concept of the time when this is observed, but it can just be assumed that the current valuation date is earlier than t_s the start of the delivery period.
- f_i as the forward price for the ith *small* contract a higher granularity period than F has, with $i = 1 \ldots n$
- D_i as the discount factor from the settlement date of contract f_i to the valuation date. We assume that the delivery period of f_i is short enough for the whole contract to be settled on a single date.

- w_i is the weighting of the delivery period of contract i, relative to the delivery period that forward price F corresponds to.

To show the no-arbitrage relationship, construct a portfolio which is long one unit of the *big* contract and short w_i units of each *small* contract. This portfolio is essentially flat, does not require and upfront cash, and so should have zero PV. If the PV were positive then a long positive would present an arbitrage, as would a negative PV presents an arbigrate from taking a short position in the portfolio. Expressing the PV as the discounted cash flow from take the portfolio to settlement:

$$\sum_{i=1}^{n}(F - f_i)w_i D_i = 0 \tag{1}$$

Which can be rearranged to:

$$F = \frac{\sum_{i=1}^{n} f_i w_i D_i}{\sum_{i=1}^{n} w_i D_i} \tag{2}$$

This is the no-arbigrate relationship between the *big* contract and it's associated *small* contracts, and can be seen as a weighted average.

3 Bootstrapping Algorithm

3.1 Linear System

Such cases are referred to in this rest of this paper as *partially overlapping contracts* and an examples seen in real market data when a contract for a one week delivery partially overlaps with a month.

3.2 Introduction of Nullspace Component to Solution

The previous section has shown an example where the least squares solution does not give one we desire, due to a deviation from the prices of the input contracts where partially overlapping contracts are present. To rectify this, we can note that in the problematic scenario, and in the typical case of running the bootstrapping algorithm, the rank of matrix $\mathbf{A}$ will be less than it's number of columns n, the number of boostrapped prices. In such cases there will be infinite solutions, and a general form of these solutions found using the fact that any vector in the nullspace of $\mathbf{A}$ can be added to any particular solution.

If the solution which is closest to $\mathbf{0}$ is not the good, then we need to define a new vector $\mathbf{x}^{target}$, the solution which should be closest to will be choosen. In the case of the problematic example where all input contracts have the same price, then $\mathbf{x}^{target}$ with all elements equal to this price seems like the obvious choice. See the section below for a discussion on choosing $\mathbf{x}^{target}$.

Defined $\mathbf{K}$ as a matrix with columns containing the orthonormal basis of the nullspace of $\mathbf{A}$, and error vector as $\mathbf{e} = \mathbf{x}^{target} - \mathbf{x}$. We want to find a linear combination of the columns of $\mathbf{K}$ which can be added to $\mathbf{x}$ to move it closest to $\mathbf{x}^{target}$. This corresponds to find the least-squares solution of $\mathbf{c}$ to the following system:

$$\mathbf{Kc} = \mathbf{e} \tag{3}$$

The least-squares solution can be found be solving the Normal Equations:

$$\mathbf{K}^T\mathbf{Kc} = \mathbf{K}^T\mathbf{e} \tag{4}$$

As $\mathbf{K}$ is orthonormal $\mathbf{K}^T\mathbf{K} = \mathbf{I}$, hence $\mathbf{c}$ is calculated as:

$$\mathbf{c} = \mathbf{K}^T\mathbf{e} \tag{5}$$

Or, substituting for $\mathbf{e}$ and multiplying out the $\mathbf{K}^T$ term:

$$\mathbf{c} = \mathbf{K}^T\mathbf{x}^{target} - \mathbf{K}^T\mathbf{x} \tag{6}$$

$\mathbf{x}$ is in the Row Space of $\mathbf{A}$ (see Strang (1998)), hence orthogonal to any vector in the nullspace of $\mathbf{A}$, so $\mathbf{K}^T\mathbf{x} = \mathbf{0}$ and the expression for $\mathbf{c}$ can be simplified:

$$\mathbf{c} = \mathbf{K}^T\mathbf{x}^{target} \tag{7}$$

Using this result, our nullspace adjusted boostrapped prices can be written as

$$\mathbf{x}^* = \mathbf{x} + \mathbf{Kc} \tag{8}$$

Substituting for $\mathbf{x}$ and $\mathbf{c}$:

$$\mathbf{x}^* = \mathbf{A}^+\mathbf{b} + \mathbf{KK}^T\mathbf{x}^{target} \tag{9}$$

3.3 Choosing The Target Vector

In the case where all input forward prices are the same, choosing $\mathbf{x}^{target}$ is easy, but what about the more realistic case when all prices are different? This is somewhat arbitrary, but intuition says that each element of $\mathbf{x}^{target}$ should be a function of the prices of all input contracts which span the delivery period each element of $\mathbf{x}^{target}$ represents the bootstrapped price of. Some experimentation has shown that taking the price of contract with minimum length of delivery period yields good results. In the case where there is more than one contract has the same minimum length, which in practice would be unusual, then the price of the contract which is delivers earliest should be used.

4 Future Work

More research into choice of the target vector can be done, including an analysis of both the intuitive and mathematical reasoning behind the logic for calculating $\mathbf{x}^{target}$ presented in this paper, as well as alternative strategies. The results of varying $\mathbf{x}^{target}$ should be analysed with real market data to assess usage in practice.

The whole approach to bootstrapping in this paper should also be compared to algorithms which perform both boostrapping and interpolation, such as Benth et al. (2007). In the case of partially overalapping contracts, the inclusion of the nullspace component and target vector presented in this paper is used to impose a shape on the resulting boostrapped prices, but one could argue that using a spline is more suitable due to the smooth structure of a spline giving a more desirable shape. A piecewise flat curve could then be averaged back from the result of the spline, and the use of a bootstapper to create the input to another spline algorithm is no longer needed. Two downside of using a combined bootstrapping and interpolation algorithm can be seen. The first is that introduces additional complication in the case where only a piecewise flat curve is required from the bootstrapping. The second is that a higher order polynomial is required to ensure that the number of of unknowns (polynomial coefficients) is at least as large as the number of constraints in the linear system solved in the spline calculation. This is because the number of piecewise polynomials being solved for can vary when the input contracts are allowed to overlap. The higher the order of the piecewise polynomials, the more tendency it has to oscillate. The author has observed undesirable oscillations when using a 4th order polynomial, as used in Benth et al. (2007), when interpolating natural gas and power forward curves. In practice it is unlikely that a polynomial should change it's slope more than once when interpolating commodity forward curve data.

References

F. E. Benth, S. Koekebakker, and F. Ollmar. Extracting and applying smooth forward curves from average-based commodity contracts with seasonal variation. *The Journal of Derivatives*, 15:52–66, 2007.

Gilbert Strang. *Introduction to Linear Algebra: 2nd Edition.* Wellesley-Cambridge Press, 1998.