

etrm: Energy Trading and Risk Management in R

by Anders D. Sleire

Abstract This paper introduces [etrm](#), an R package with tools for trading and financial risk management in energy markets. Contracts for electric power and natural gas differ from most other commodities due to the fact that physical delivery takes place over a time interval, and not at a specific point in time. There is typically strong seasonality, limited storage and transmission capacity and strong correlation between price and required volume. Such characteristics need to be taken into account when pricing contracts and managing financial risk related to energy procurement. Tools for these task are usually bundled into proprietary Energy Trading Risk Management (ETRM) systems delivered by specialized IT vendors. The [etrm](#) package offers a transparent solution for building a forward price curve for energy commodities which is consistent with methods widely used in the industry. The user's fundamental market view may be combined with contract price quotes to form a forward curve that replicate current market prices, as described in [Ollmar \(2003\)](#) and [Benth et al. \(2007b\)](#). [etrm](#) also provides implementations of five portfolio insurance trading strategies for energy price risk management. The forward market curve and the energy price hedging strategies are core elements in an ETRM system, which to the best of the author's knowledge has not been previously available in the R ecosystem.

Introduction

The purpose of this paper is to introduce the R package [etrm](#) and its tools for energy trading and financial risk management. Substantial fluctuations in energy prices represent a significant risk for market players, in particular for large consumers, producers and utility companies, see [Benini et al. \(2002\)](#). The price dynamics is complex due to strong weather dependency and physical constraints related to storage, distribution, and the introduction of new technology. See for example [Nicolosi \(2010\)](#) for an analysis of renewable energy production and the negative prices following extreme events in the German power market. Derivatives securities, such as futures contracts, are often used to hedge against the commodity price risk. Specialized Energy Trading and Risk Management (ETRM) systems provide the necessary tools to handle key activities such as position management, valuation and risk reporting. Several proprietary alternatives exist. The annual *Energy Risk's Software Survey* in [Farrington \(2020\)](#) gives an overview of major providers along with rankings based on industry polls. There, ETRM systems are divided into the operational categories *derivatives software*, *physical trading and operations software* and *front- and middle-office functionality*. Historically, many system providers within this domain have bundled modules into large monolithic architectures serving a variety of purposes, including accounting and regulatory reporting. During the last decade, a general trend within system development has moved towards splitting software into smaller stand-alone components. The [etrm](#) package solely focus on financial trading, and may be viewed as a module for front- and middle-office functionality for energy derivatives. The package currently offers transparent tools for two main ETRM activities 1) construction of forward market curves and 2) implementation of trading strategies for price risk management.

After the liberalization of electricity and gas markets started in the 1990s, a rich research literature emerged. Topics studied include pricing and hedging in the forward market and modelling of spot price processes, see for example [Bessembinder and Lemmon \(2002\)](#), [Janczura et al. \(2013\)](#) and [Benth et al. \(2007a\)](#). Alternative methods for pricing options in power markets can be found in [Burger et al. \(2004\)](#) and [Benth and Schmeck \(2014\)](#). Textbooks such as [Eydeland and Wolyniec \(2002\)](#), [Benth et al. \(2008\)](#) and [Kirschen and Strbac \(2018\)](#) may be used to gain a more detailed overview of market structure, available instruments, methods for risk management and the related markets for fuel, freight and weather products. In this paper, we will cover some of the theory regarding forward curve modelling from [Ollmar \(2003\)](#) and [Benth et al. \(2007b\)](#). The theoretical framework for price risk management is gathered from the *portfolio insurance* literature, see [Leland \(1980\)](#), [Perold and Sharpe \(1988\)](#), [Leland and Rubinstein \(1976\)](#). This will be presented in further detail below.

We would like to note that there are some tools available outside the domain of proprietary ETRM software providers. Two examples are the MathWorks case studies [Sundar \(2021\)](#) and [Deoras \(2021\)](#), focusing on risk assessment for gas-fired power plants and electricity load and price forecasting using MATLAB. These topics are however somewhat ad-hoc, and the supplied code cannot be easily incorporated into a generic ETRM system for general use. Similarly within the R ecosystem, there are related tools in the Rmetrics suite of packages, such as [fOptions](#) and [fPortfolio](#), but they are not

directly applicable. Due to the unique properties of energy markets, standard methods for generating forward curves in interest rate markets cannot be used either. To fill this gap and provide practitioners and researchers with tools dedicated to energy price risk management, we have created [etrm](#). The package is available on CRAN, and may be installed and loaded into the R environment by running the following commands:

```
if(!require(etrm)==TRUE) {install.packages("etrm")}
library(etrm)
```

The rest of the article is organized as follows. First, we give a brief introduction to energy market forward curves and the maximum smoothness forward curve (MSFC) model. We describe the [etrm](#) implementation and provide examples of use. Second, a short treatment of energy price risk management with futures contracts is provided, followed by a presentation of five portfolio insurance models. The [etrm](#) implementation is described and illustrated with practical examples for energy portfolios with both short and long market exposure. The third part provide an overview of the [etrm](#) package structure, available functions and included data sets. Finally, the last section summarizes the paper and provide some suggestions for future work.

Energy market forward price curves

The standardised forwards for electricity and gas are contracts for flow delivery. The underlying commodity is not received at a fixed point in time, but over a time interval. In mature markets, participants can trade a variety of products, both over-the-counter (OTC) and on exchanges such as [Nasdaq Commodities](#), [European Energy Exchange](#) and the [Intercontinental Exchange](#). Liquidity is often best in the so called front-products, and there is normally higher activity in contracts for next week, month, quarter and year compared to similar products further ahead in time. Shorter period contracts may not even be available on a longer horizon, and seasonal price variation is thus not directly observable in prices far ahead in time. Transacted volume and prices also inhibit quite pronounced seasonality, during the year, week and within a specific day. For this reason, forward contracts are divided into categories based on a *load pattern*. In the *base load* contracts, volume is delivered at a constant rate during the contract period, while *peak load* products are linked to high volume hours, such as Monday to Friday from 8 am to 8 pm. Other, more exotic load patterns exist, but they are less common. Further details can be found in [Eydeland and Wolyniec \(2002\)](#) and [Kirschen and Strbac \(2018\)](#).

The aim of the energy market forward price curve calculation is to create a compact representation of the forward market, at a given point in time. The curve must be able to price the quoted instruments correctly, while accounting for typical energy market characteristics such as seasonality and (possibly overlapping) contracts for flow delivery. The curve is an essential decision making tool with many uses, such as pricing non-standard supply agreements, investment decisions and risk management.

The topic of forward curve fitting has been studied for decades in interest rate markets, see for example [McCulloch \(1971\)](#) and [Anderson et al. \(1996\)](#). These techniques cannot be applied directly to commodities with flow delivery and strong seasonality in prices. There are several alternative approaches to calculating a forward price curve for energy commodities. In [Fleten and Lemming \(2003\)](#), market data is combined with forecasts generated by a bottom-up model constrained by the bid/ask spread in order to meet the no-arbitrage condition. [Borak and Weron \(2008\)](#) propose a semiparametric factor model for the forward curve dynamics in electricity markets, while [Hildmann et al. \(2012\)](#) develop a calculation method by combining parametric estimation and prediction of futures prices under constraints.

In [etrm](#), we have opted for a method that combines a seasonal function with the maximum smoothness-approach from interest rate markets, see [Adams and Van Deventer \(1994\)](#). Hence, we follow the approach in [Ollmar \(2003\)](#) and [Benth et al. \(2007b\)](#). *Base load* contracts are used to calculate a curve with daily granularity. This method has several benefits. It produces a continuous curve which has a closed form solution, it is fast to calculate, flexible and used by many practitioners in the industry. The next section provide a brief overview of the method, followed by a description of the [etrm](#) implementation with examples.

Maximum smoothness forward curve model

Consider a market at time t with m forward contracts available for trading. Let the list

$$S_t = \{(\tau_1^s, \tau_1^e), (\tau_2^s, \tau_2^e), \dots, (\tau_m^s, \tau_m^e)\}$$

contain the start and end dates for each of these contracts. The time distance between τ_i^s and τ_i^e for a contract i in $1, \dots, m$ cover standardized periods such as *week*, *month*, *quarter* and *year*. Some of these settlement intervals might overlap, and in order to handle this we create a new list of dates $\{t_0, t_1, \dots, t_n\}$ to identify each separate sub period, see Figure 1. The new list is made by sorting all dates in S_t in ascending order and removing duplicates.

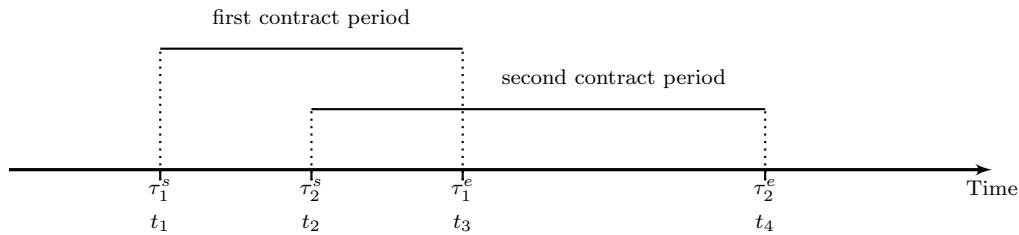


Figure 1: Illustration of two overlapping contracts with start (τ^s) and end (τ^e) dates. Due to the overlap, the total delivery period is split into sub intervals identified with $\{t_1, t_2, t_3, t_4\}$.

The forward price at time t for one unit of energy delivered at a constant rate $(\tau^e - \tau^s)^{-1}$ over the time interval (τ^s, τ^e) is denoted by $F(t, \tau^s, \tau^e)$, where $t \leq \tau^s < \tau^e$. A forward contract for a flow delivery may be thought of as the average of hypothetical single-delivery contracts. At time t , each of these would have a unique price $f(t, u)$ for the delivery at u with an infinitesimal delivery period. This leads to $F(t, \tau^s, \tau^e)$ being the weighted average

$$F(t, \tau^s, \tau^e) = \int_{\tau^s}^{\tau^e} w(u, \tau^s, \tau^e) f(t, u) du \quad (1)$$

where $w(u, \tau^s, \tau^e) = \frac{\hat{w}(u)}{\int_{\tau^s}^{\tau^e} \hat{w}(v) dv}$ is a weight function accounting for the rate of interest r and the time value of money. If the contract in question is settled at the end of the delivery period (forward contract), the weight function is given by $w(u, \tau^s, \tau^e) = 1/(\tau^e - \tau^s)$. If the contract is settled continuously over the delivery period (futures contract), $w(u, \tau^s, \tau^e) = \frac{re^{-ru}}{e^{-r\tau^s} - e^{-r\tau^e}}$. In the following we construct a forward market price curve for the entire horizon using a simplified notation $f(u)$ for the function describing the forward curve at time t . In order to model the strong seasonality in energy markets, the forward curve function is decomposed into two elements:

$$f(u) = \Lambda(u) + \epsilon(u) \quad u \in [t_0, t_n] \quad (2)$$

Following [Ollmar \(2003\)](#) we calculate $f(u)$ by combining a prior function $\Lambda(u)$ which contain our subjective views on the future prices with an adjustment function $\epsilon(u)$ to ensure match with the observed closing prices for the m contracts. The prior could be generated with a simple sinusoidal function or from a fundamental model more capable of describing the seasonality and calendar effects observed in energy markets. Should the prior be excluded, the seasonal price patterns will not be visible in the far end of the curve, where only yearly or seasonal contracts are available. Smoothing is calculated on the adjustment function, we aim to minimize the total curvature of $\Lambda(u)$ while preserving the information from the prior. Smoothness is defined as the integral of the second-order derivative of the function, and the smoothest possible curve over $[t_0, t_n]$ is achieved by minimising

$$\int_{t_0}^{t_n} [\epsilon''(u)]^2 du$$

under five constraints presented below. [Lim and Xiao \(2002\)](#) show the smoothest possible curve is found when the n sub periods are modelled by fourth-degree polynomials. We write $\epsilon(u)$ as a spline

$$\epsilon(u) = \begin{cases} a_1 u^4 + b_1 u^3 + c_1 u^2 + d_1 u + e_1, & u \in [t_0, t_1], \\ a_2 u^4 + b_2 u^3 + c_2 u^2 + d_2 u + e_2, & u \in [t_1, t_2], \\ \vdots & \vdots \\ a_n u^4 + b_n u^3 + c_n u^2 + d_n u + e_n, & u \in [t_{n-1}, t_n]. \end{cases}$$

In order to construct the forward curve function we need to identify the parameters of $\epsilon(u)$

$$\mathbf{x}^\top = [a_1 \ b_1 \ c_1 \ d_1 \ e_1 \quad a_2 \ b_2 \ c_2 \ d_2 \ e_2 \dots a_n \ b_n \ c_n \ d_n \ e_n]$$

by solving the quadratic optimisation problem

$$\min_x \int_{\tau^s}^{\tau^e} [\epsilon''(u; x)]^2 du \quad (3)$$

subject to

$$(a_{j+1} - a_j)u_j^4 + (b_{j+1} - b_j)u_j^3 + (c_{j+1} - c_j)u_j^2 + (d_{j+1} - d_j)u_j + e_{j+1} - e_j = 0 \quad (a)$$

$$4(a_{j+1} - a_j)u_j^3 + 3(b_{j+1} - b_j)u_j^2 + 2(c_{j+1} - c_j)u_j + (d_{j+1} - d_j) = 0 \quad (b)$$

$$12(a_{j+1} - a_j)u_j^2 + 6(b_{j+1} - b_j)u_j + 2(c_{j+1} - c_j) = 0 \quad (c)$$

$$\epsilon'(u_n; x) = 0 \quad (d)$$

$$\frac{1}{\tau_i^e - \tau_i^s} \int_{\tau_i^s}^{\tau_i^e} (\Lambda(u) + \epsilon(u)) du = F_i^c \quad (e)$$

for spline knot $j = 1, \dots, n-1$ and contract $i = 1, \dots, m$. The constraint in (a) ensures the adjustment function is continuous in the knots, while (b) and (c) imposes this restriction also for the first and second order differentials. The (d) constraint require the adjustment function to be horizontal at time T , and finally (e) also require the average value of the forward price function $f(u)$ over the delivery period for contract i to match the quoted closing price F_i^c . Here, we could take the interest rate effect from r into account and set the present value of the average of the forward price function equal to present value of the forward contract. Instead of doing that, we will follow [Benth et al. \(2008\)](#) and [Ollmar \(2003\)](#) and assume $r = 0$ such that the weight function in 1 is approximated with $w(u, \tau^s, \tau^e) \approx \frac{1}{\tau^e - \tau^s}$. Like [Benth et al. \(2008\)](#) we will argue that both the prior and the smoothing will outweigh a marginal interest rate effect. This minimisation problem can be expressed as

$$\min_{\mathbf{x}} \quad \mathbf{x}^\top \mathbf{H} \mathbf{x},$$

where \mathbf{x} is a $(5n \times 1)$ vector and

$$\mathbf{H} = \begin{bmatrix} h_1 & & 0 \\ & \ddots & \\ 0 & & h_n \end{bmatrix}, \mathbf{h}_j = \begin{bmatrix} \frac{144}{5} \Delta_j^5 & 18 \Delta_j^4 & 8 \Delta_j^3 & 0 & 0 \\ 18 \Delta_j^4 & 12 \Delta_j^3 & 6 \Delta_j^2 & 0 & 0 \\ 8 \Delta_j^3 & 6 \Delta_j^2 & 4 \Delta_j & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The block diagonal matrix \mathbf{H} has dimensions $(5n \times 5n)$ and $\Delta_j = t_{j+1}^l - t_j^l$. As the constraints in (3) are all linear in x and may be expressed on the form $\mathbf{A} \mathbf{x} = \mathbf{B}$, the problem may be rephrased as an unconstrained minimisation problem via the Lagrange multiplier method:

$$\min_{\mathbf{x}, \lambda} \quad \mathbf{x}^\top \mathbf{H} \mathbf{x} + \lambda^\top (\mathbf{A} \mathbf{x} - \mathbf{B})$$

where \mathbf{A} is a $(3n + m - 2 \times 5n)$ matrix and \mathbf{B} is a $(3n + m - 2 \times 1)$ vector. The spline parameter vector and Lagrange multipliers are identified by solving

$$\begin{bmatrix} 2\mathbf{H} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{B} \end{bmatrix} \quad (4)$$

where the left matrix has dimension of $(8n + m - 2) \times (8n + m - 2)$ and both vectors are of dimension $(8n + m - 2)$.

The "MSFC" class with examples

The forward curve calculation in [etrm](#) is implemented in the S4 "MSFC" class. By supplying all required arguments to the constructor function `msfc()`, the user may create an object that contains the calculation results, input arguments and further calculation details. In addition to the arguments from the list of contracts, the user may also provide a prior to the calculation. By default the prior is set to zero, but the user can input any vector expressing a belief regarding the market to be combined with the observed prices. An overview of the `msfc()` arguments can be found in Table 1.

Argument	Description	Default value
<code>tdate</code>	Trading date	<code>none</code>
<code>include</code>	Logical vector for contract selection	<code>none</code>
<code>contract</code>	Character vector with contract names	<code>none</code>
<code>sdate</code>	Date vector with contract start dates	<code>none</code>
<code>edate</code>	Date vector with contract end dates	<code>none</code>
<code>f</code>	Numeric vector with contract prices	<code>none</code>
<code>prior</code>	Numeric prior curve vector	<code>0</code>

Table 1: Arguments for the `msfc()` constructor function for forward curve calculation.

The "MSFC" class properties and available methods are summarised in Figure 2, and a brief description is provided in the following. The "Name" slot describe the type of forward curve model used by storing the character value "MSFC", while "TradeDate" keeps the trade date used in the calculation. A data frame containing details for selected contracts along with the calculated forward price based on the curve can be found in "BenchSheet". A count of the $(n - 1)$ number of polynomials used in the spline is available as a scalar value in "Polynomials", and the prior curve vector in "PriorFunc". The main calculation result is stored in a data frame which contains daily values for all selected contracts along with the calculated forward curve. The data frame span the date range from "TradeDate" to the end date of the contract furthest ahead in time, and can be found in "Results". The interested user may also extract additional information regarding the spline itself. Coefficients for all polynomials can be found in the "SplineCoef" list and the knotpoints separating them in the numeric vector "KnotPoints". Further details regarding the calculation of the daily forward curve values are available in the "CalcDat" data frame. This table is essentially an extended version of "Results", where numeric time vectors and the spline coefficients have been added.

MSFC
Name : "character" TradeDate : "date" BenchSheet : "data.frame" Polynomials : "numeric" PriorFunc : "numeric" Results : "data.frame" SplineCoef : "list" KnotPoints : "numeric" CalcDat : "data.frame"
<code>plot()</code> <code>summary()</code> <code>show()</code>

Figure 2: Attributes and methods of the "MSFC" class.

The "MSFC" class has the generic methods `plot()`, `summary()` and `show()`. The `plot()` method may be used to create a chart of the calculated curve and underlying contracts from the "Results" data frame. All plot methods in [etrm](#) are based on [ggplot2](#), see [Wickham \(2011a\)](#). The `summary()` method returns a list with three elements; a description string, a sample of the prior vector, and the bench sheet. Finally, the `show()` method returns the "Results" data frame.

We proceed with a practical example using two of the embedded [etrm](#) data sets to represent information available to a European power market participant. All market-related inputs to the `msfc()` constructor (trade date and contract properties) are required arguments. These are collected from a synthetic data set for the trading date 2013-05-13, and can be found in "powfutures130513" presented in Table 2. Contracts covering long time spans are excluded with `include = FALSE` if futures of shorter duration are available for the same time interval in order to preserve the seasonality available in market prices. We use a seasonal prior with high energy prices during the winter season, followed by

a drop toward the lower summer levels. It also take into account some well known calendar effects, such as weekends. The prior is simple, and merely used for illustrative purposes. It can be found in the data set "powpriors130513" included in the package. A calculation excluding the prior is also added for comparison.

Include	Contract	Start	End	Closing
TRUE	W21-13	2013-05-20	2013-05-26	33.65
TRUE	W22-13	2013-05-27	2013-06-02	35.77
TRUE	W23-13	2013-06-03	2013-06-09	36.58
TRUE	W24-13	2013-06-10	2013-06-16	35.93
TRUE	W25-13	2013-06-17	2013-06-23	33.14
TRUE	W26-13	2013-06-24	2013-06-30	34.16
FALSE	MJUN-13	2013-06-01	2013-06-30	35.35
TRUE	MJUL-13	2013-07-01	2013-07-31	33.14
TRUE	MAUG-13	2013-08-01	2013-08-31	35.72
TRUE	MSEP-13	2013-09-01	2013-09-30	38.41
TRUE	MOCT-13	2013-10-01	2013-10-31	38.81
TRUE	MNOV-13	2013-11-01	2013-11-30	40.94
FALSE	Q3-13	2013-07-01	2013-09-30	35.72
TRUE	Q4-13	2013-10-01	2013-12-31	40.53
TRUE	Q1-14	2014-01-01	2014-03-31	42.40
TRUE	Q2-14	2014-04-01	2014-06-30	33.39
TRUE	Q3-14	2014-07-01	2014-09-30	31.78
TRUE	Q4-14	2014-10-01	2014-12-31	38.25
TRUE	Q1-15	2015-01-01	2015-03-31	40.73
TRUE	Q2-15	2015-04-01	2015-06-30	32.64
TRUE	Q3-15	2015-07-01	2015-09-30	30.87
TRUE	Q4-15	2015-10-01	2015-12-31	37.22
FALSE	CAL-14	2014-01-01	2014-12-31	36.43
FALSE	CAL-15	2015-01-01	2015-12-31	35.12
TRUE	CAL-16	2016-01-01	2016-12-31	34.10
FALSE	CAL-17	2017-01-01	2017-12-31	35.22
FALSE	CAL-18	2018-01-01	2018-12-31	36.36

Table 2: Closing prices for futures contracts used in the forward curve calculation for 2013-05-13. Contracts are selected for the calculations with the include vector. Prices for these contracts can be found as horizontal lines in Figure 3

As shown in Figure 3, the shorter contracts close in time to the trading date clearly reflect a seasonal pattern. This is typical in power markets, where weather and calendar effects have strong influence on transacted volume and price formation. On a longer horizon however, this information is not observable in market prices, as the quoted contracts cover longer time spans. This is where price data may be supplemented with prior knowledge in order to create a representation of the market consistent with both the underlying fundamentals and the listed contracts. The following code will create the "MSFC" objects and plot calculation results:

```
library(etrm)
library(gridExtra)
data(powfutures130513)
data(powpriors130513)

# instance of MSFC class with prior
fwd.fut.wpri <- msfc(tdate = as.Date("2013-05-13"),
  include = powfutures130513$Include,
  contract = powfutures130513$Contract,
  sdate = powfutures130513$Start,
  edate = powfutures130513$End,
  f = powfutures130513$Closing,
  prior = powpriors130513$mod.prior)

# instance of MSFC class without prior
fwd.fut.npri <- msfc(tdate = as.Date("2013-05-13"),
  include = powfutures130513$Include,
  contract = powfutures130513$Contract,
```

```

sdate = powfutures130513$Start,
edate = powfutures130513$End,
f = powfutures130513$Closing,
prior = 0)

# the generic plot() method
pw <- plot(fwd.fut.wpri, ylab = "EUR/MWh", legend = "")
pn <- plot(fwd.fut.npri, ylab = "EUR/MWh", legend = "")

# combine plots
gridExtra::grid.arrange(pw, pn)

```

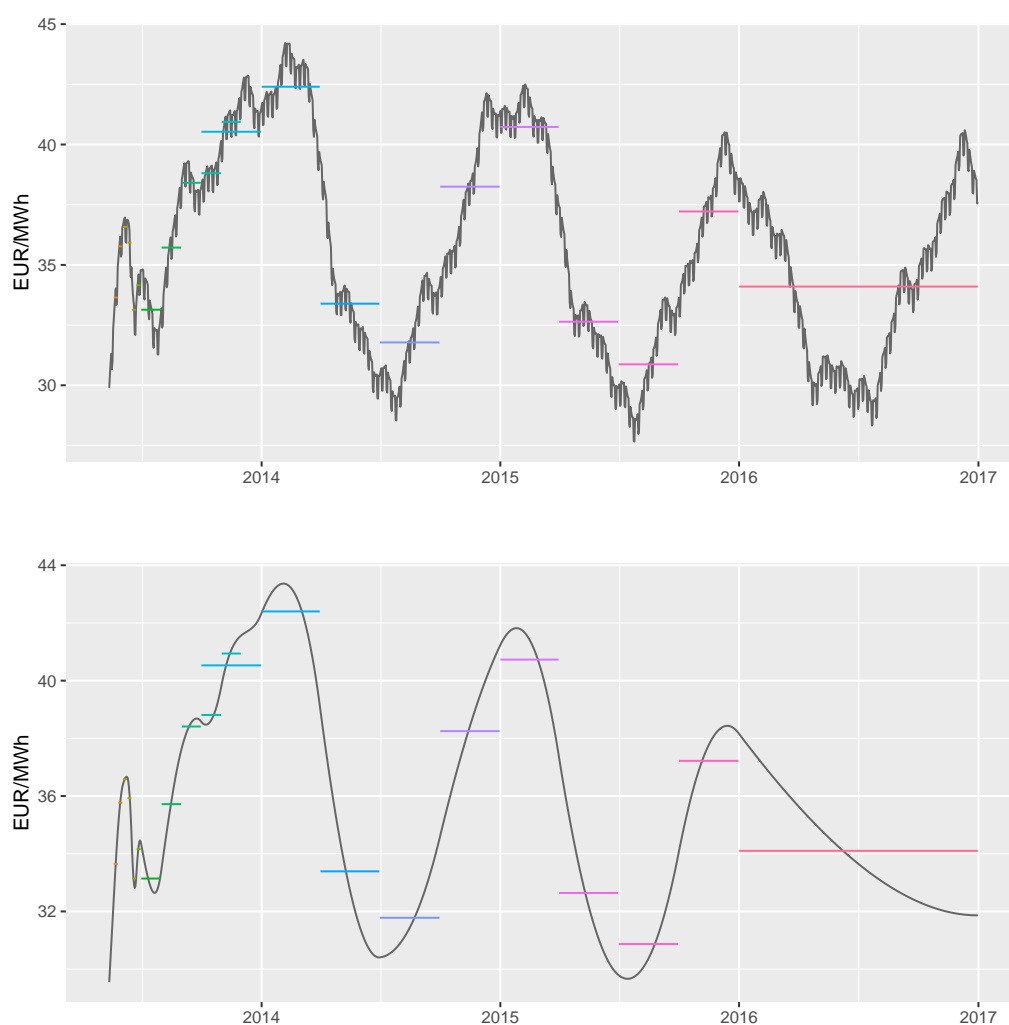


Figure 3: Two alternative forward curve calculations based on the same contract selection. In the top panel, a prior function is included in the calculation. This curve shows price variation on the weekly level, with lower prices during weekends. The prior also ensures that yearly seasonality is visible in the far end of the curve. The bottom plot is based solely on market prices, which does not reflect seasonality on such long horizon.

The computed prices may be verified via the `summary()` method, which also return a sample of the prior and information regarding the spline calculation:


```
> summary(fwd.fut.wpri)
$Description
[1] "MSFC of length 1329 built with 41 polynomials at trade date 2013-05-13"

$PriorFunc
[1] 30.10842 30.16396 30.19572 30.16144 29.06268 28.93272
```

```
$BenchSheet
      Include Contract      From      To Price  Comp
1    TRUE   W21-13 2013-05-20 2013-05-26 33.65 33.65
2    TRUE   W22-13 2013-05-27 2013-06-02 35.77 35.77
3    TRUE   W23-13 2013-06-03 2013-06-09 36.58 36.58
4    TRUE   W24-13 2013-06-10 2013-06-16 35.93 35.93
5    TRUE   W25-13 2013-06-17 2013-06-23 33.14 33.14
6    TRUE   W26-13 2013-06-24 2013-06-30 34.16 34.16
8    TRUE  MJUL-13 2013-07-01 2013-07-31 33.14 33.14
9    TRUE  MAUG-13 2013-08-01 2013-08-31 35.72 35.72
10   TRUE  MSEP-13 2013-09-01 2013-09-30 38.41 38.41
11   TRUE  MOCT-13 2013-10-01 2013-10-31 38.81 38.81
12   TRUE  MNOV-13 2013-11-01 2013-11-30 40.94 40.94
14   TRUE   Q4-13 2013-10-01 2013-12-31 40.53 40.53
15   TRUE   Q1-14 2014-01-01 2014-03-31 42.40 42.40
16   TRUE   Q2-14 2014-04-01 2014-06-30 33.39 33.39
17   TRUE   Q3-14 2014-07-01 2014-09-30 31.78 31.78
18   TRUE   Q4-14 2014-10-01 2014-12-31 38.25 38.25
19   TRUE   Q1-15 2015-01-01 2015-03-31 40.73 40.73
20   TRUE   Q2-15 2015-04-01 2015-06-30 32.64 32.64
21   TRUE   Q3-15 2015-07-01 2015-09-30 30.87 30.87
22   TRUE   Q4-15 2015-10-01 2015-12-31 37.22 37.22
25   TRUE   CAL-16 2016-01-01 2016-12-31 34.10 34.10
```

The forward curve values can be extracted along with daily prices for the contracts used in the calculation with the `show()` method:

```
> head(show(fwd.fut.wpri), 20)[1:5]
      Date      MSFC W21-13 W22-13 W23-13
1 2013-05-13 29.89373    NA     NA     NA
2 2013-05-14 30.40235    NA     NA     NA
3 2013-05-15 30.88704    NA     NA     NA
4 2013-05-16 31.30634    NA     NA     NA
5 2013-05-17 30.66200    NA     NA     NA
6 2013-05-18 30.98687    NA     NA     NA
7 2013-05-19 32.33591    NA     NA     NA
8 2013-05-20 32.74655 33.65    NA     NA
9 2013-05-21 33.19772 33.65    NA     NA
10 2013-05-22 33.63844 33.65    NA     NA
11 2013-05-23 34.02161 33.65    NA     NA
12 2013-05-24 33.34168 33.65    NA     NA
13 2013-05-25 33.62327 33.65    NA     NA
14 2013-05-26 34.91272 33.65    NA     NA
15 2013-05-27 35.24208    NA 35.77    NA
16 2013-05-28 35.59669    NA 35.77    NA
17 2013-05-29 35.92499    NA 35.77    NA
18 2013-05-30 36.17633    NA 35.77    NA
19 2013-05-31 35.34194    NA 35.77    NA
```


20 2013-06-01 35.44437 NA 35.77 NA

We have excluded columns from the data frame for the sake of presentation. Further details regarding the calculation such as spline coefficients and knot points can be found in the slots:

```
> slotNames(fwd.fut.wpri)
[1] "Name"          "TradeDate"     "BenchSheet"
[4] "Polynomials"   "PriorFunc"     "Results"
[7] "SplineCoef"    "KnotPoints"    "CalcDat"
```

See for example the numeric vector with knot points, measured in years from the trading date:

```
> fwd.fut.npri@KnotPoints
[1] 0.00000000 0.01917808 0.03561644 0.03835616 0.05479452 0.05753425 0.07397260
[8] 0.07671233 0.09315068 0.09589041 0.11232877 0.11506849 0.13150685 0.13424658
[15] 0.21643836 0.21917808 0.30136986 0.30410959 0.38356164 0.38630137 0.46849315
[22] 0.47123288 0.55068493 0.63561644 0.63835616 0.88219178 0.88493151 1.13150685
[29] 1.13424658 1.38356164 1.38630137 1.63561644 1.63835616 1.88219178 1.88493151
[36] 2.13150685 2.13424658 2.38356164 2.38630137 2.63561644 2.63835616 3.63835616
```

The coefficients for the first polynomial in the adjustment function spline can be found with

```
> fwd.fut.npri@SplineCoef[[1]]
[1] -355585.14451 10911.10580 -78.47028 151.90713 29.54903
```

The most elaborate presentation of the curve calculation is available in `fwd.fut.npri@CalcDat`. This slot contains a data frame with all calculation details for each of the daily values returned by `msfc()`. It is not included here due to space requirements.

Energy price risk management

Energy market participants may be exposed to number of risk factors such as volume, profile and basis risk, counter party defaults, foreign exchange and market liquidity, to name a few. See for example [Eydeland and Wolyniec \(2002\)](#) and [Kirschen and Strbac \(2018\)](#) for a comprehensive treatment of the topic. The main focus in [etrm](#) is on the market price risk of the energy commodity. Consider the price risk associated with the constant base load volume q to be delivered over a future time interval (τ^s, τ^e) . The risk can be mitigated by taking positions in the futures market during a trading period (t_0, T) , which ends before the actual delivery of the energy takes place at $T < \tau^s$.

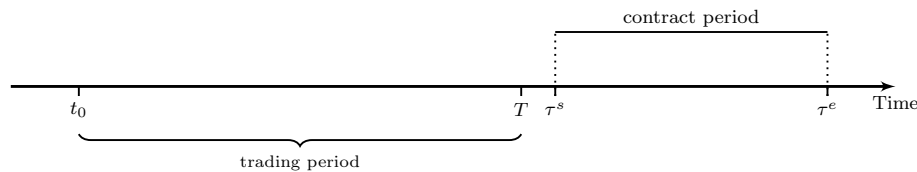


Figure 4: Trading and settlement periods for energy forward contract.

The price risk may be reduced by constructing a portfolio, consisting of the physical energy market exposure and derivatives contracts. The *portfolio price* per energy unit p_t is calculated as the weighted average of the value of the transacted volumes and the open volume evaluated mark-to-market

$$p_t = \frac{1}{q} \left(f_0 h_0 q + \sum_{i=1}^t f_i (h_i - h_{i-1}) q + f_t (1 - h_t) q \right) = f_0 h_0 + \sum_{i=1}^t f_i (h_i - h_{i-1}) + f_t (1 - h_t) \quad (5)$$

where $h_t \in (0, 1)$ is the hedge rate and f_t the futures price at time t . In the simplest possible scenario, the risk can be managed by locking the entire volume in the forward market. This removes the price risk and the portfolio owner knows up front what to pay or receive when delivery of the energy takes place. On the downside, one might regret locking if the market develops in a favourable way.

Portfolio insurance strategies

In the portfolio insurance approach, dynamic hedging strategies that allow buying and selling the hedging instrument are used to protect the portfolio, while seeking to benefit from advantageous market developments. Historically, the theory of portfolio insurance has focused on protection against downside risk in financial investment portfolios, see [Leland and Rubinstein \(1976\)](#), [Perold and Sharpe \(1988\)](#) and [Leland \(1980\)](#). Here, we apply the same principles to manage commodity price risk in the forward market. A consumer following a dynamic hedging program may control price risk by locking a share of future volume in the futures market, and increase (decrease) the share if the price increase (falls). A seller can implement similar strategies to maximise value of the energy portfolio. The size of the initial hedged share and how it is adjusted affects both the protective properties of the hedging scheme as well as its ability to exploit opportunities in the market. Trading activity needs

to be carefully managed and harmonized with overall objectives and risk preferences. A variety of portfolio insurance strategies offer different approaches to this task. The allocation strategies presented below all aim to control p_t and prevent breach of a pre-specified cap (or floor) price, p^* , under the hedge rate restriction $h_t \in (0, 1)$.

The **Constant proportion portfolio insurance (CPPI)** strategy was introduced by Perold (1986) and Black and Jones (1987) for management of investment portfolios with capital guarantees. When applied to an energy portfolio, it aims to insure the portfolio by protecting a target price, a cap (floor) value for the portfolio price. Prior to start of hedging, p^* is set equal to the highest (lowest) acceptable portfolio price. This target price must be set higher than the first day's market price f_0 to implement cap protection, or lower for a floor protection model. The difference between the target price and the current portfolio price is termed the cushion. The key idea of CPPI is that the proportion of the volume exposed to the market should be calculated as a constant multiple m of the cushion. The multiple is given by $m = \mu^{-1}$, where $\mu > 0$ is a risk factor set to handle the maximum daily price change to be handled by the hedging model, which again affects strategy gearing. The unexposed proportion, the hedge rate, is

$$h_t = \begin{cases} 0 & \text{if } c_t > \mu \\ 1 - mc_t & \text{if } \mu \geq c_t \geq 0 \\ 1 & \text{if } c_t < 0 \end{cases} \quad (6)$$

where $c_t = p^* - p_{t-1}$ for a short hedger, and $c_t = p_{t-1} - p^*$ for the long hedger.

In the **Dynamic proportion portfolio insurance (DPPI)** strategy, a decision rule similar to CPPI is applied, but the multiple m_t is allowed to vary. Changing market conditions may require re-evaluation of the risk factor in the multiple. Methods such as Value-at-Risk or Expected Shortfall, or even simple heuristics can be used for this purpose. For a further treatment of DPPI type strategies, the reader is referred to Lee et al. (2008) and Chen et al. (2008). In **etrm** we also allow adjustments in the target price p_t^* , to catch opportunities to lower the capped value, or to increase the floor value. The hedge rate is determined similarly to CPPI, with

$$p_t^* = \begin{cases} \min(\lambda p_{t-1}, p_{t-1}^*) & \text{short hedger} \\ \max(\lambda p_{t-1}, p_{t-1}^*) & \text{long hedger} \end{cases} \quad (7)$$

where $\lambda = \frac{p_o^*}{p_o}$ for a short hedger, and $\lambda = \frac{p_o}{p_o^*}$ for the long hedger.

Option based portfolio insurance (OBPI) was first introduced in Leland and Rubinstein (1976) as a means of providing insurance for investment portfolios. By combining an investment in a risky asset with a put option on the asset, the portfolio value is prevented from falling below the option strike price, K . A similar approach can be taken for the energy portfolio. As we are using futures contracts to manage the energy price risk, the Black-76 formula introduced in Black (1976) is used to approximate the contingent claim premium. The price at time t of the European call and put options with exercise date T and strike price K , on a futures contract with delivery start $\tau^s \geq T$ is given by

$$C(f_t, t, K, \sigma, r) = e^{-r(T-t)} [f_t N(d_1) - KN(d_2)] \quad (8)$$

$$P(f_t, t, K, \sigma, r) = e^{-r(T-t)} [KN(-d_2) - f_t N(-d_1)] \quad (9)$$

where f_t is the futures price and N is the cumulative distribution function of $N(0, 1)$, where

$$d_1 = \frac{\ln(f_t/K) + (\sigma^2/2)(T-t)}{\sigma\sqrt{(T-t)}} \quad (10)$$

$$d_2 = d_1 - \sigma\sqrt{(T-t)} \quad (11)$$

and r is the risk free rate of interest, σ the volatility of the underlying futures price and $(T-t)$ is the time to exercise date. The sensitivity in the option premiums with respect to changes in the underlying futures price is given by the call and put option deltas:

$$\frac{\partial C}{\partial f} = e^{-r(T-t)} N(d_1) \quad (12)$$

$$\frac{\partial P}{\partial f} = e^{-r(T-t)} N(-d_1) \quad (13)$$

These are used to synthesise the option, by setting portfolio hedge rate $h_t \in (0, 1)$ with the call (buyer) and put (seller) option deltas. By implementing this *delta hedging* scheme, a cap (floor) for the portfolio price is set at the option strike price K , adjusted for the option premium/ replication costs. For a more

detailed presentation of the underlying theory, the reader is referred to Bjork (2009).

Step hedge portfolio insurance (SHPI) is a simple and mechanical benchmark strategy that builds hedging positions gradually by transacting identical volumes each day through the trading period (t_0, T) , reaching a full hedge prior to the start of the settlement period. The hedge rate for a buyer at time t is given by

$$h_t = \begin{cases} \frac{t}{T-t_0+1} & \text{if } p_t < p^* \\ 1 & \text{if } p_t \geq p^* \end{cases} \quad (14)$$

The hedges for a seller is entered mechanically in a similar manner as long as $p_t > p^*$. In the event $p_t \leq p^*$ a full hedge $h_t = 1$ is implemented. By distributing the transacted volumes evenly across the trading period while monitoring the target, the strategy portfolio price will either be locked in at the target price, or end up equal to the average forward market price over (t_0, T) .

Finally, the **Stop loss portfolio insurance (SLPI)** is another simple benchmark, where no hedge positions are entered unless the target level is reached. For a buyer, this may be expressed as

$$h_t = \begin{cases} 0 & \text{if } p_t < p^* \\ 1 & \text{if } p_t \geq p^* \end{cases} \quad (15)$$

For a seller, the logic is reversed with $h_t = 0$ for $p_t > p^*$ and $h_t = 1$ for $p_t \leq p^*$. In the event that the target level is reached, the portfolio is kept fully hedged until start of settlement. If this does not occur, the portfolio follows the forward market, leaving an option to lock in the price at contract expiration.

The strategies presented above all have strengths and weaknesses. CPPI, SHPI and SLPI are simple and intuitive, but can be vulnerable to so-called *lock in*, the inability to improve portfolio price once the target level has been reached. This is also the case for DPPI. The OBPI does not suffer from this trait, but it relies on more assumptions regarding model parameters. In some scenarios, it will also generate more trading activity (costs), for example if the market fluctuate around the option strike price, K . Finally, as the strategies must be implemented in discrete time, they will all be exposed to gap risk.

The strategy classes with examples

The portfolio insurance strategies in **etrm** are implemented as S4 classes. Since they share many characteristics, they inherit most of their properties from a parent class, "GenericStrat". In fact, the implementation of the simple benchmark strategies SLPI and SHPI do not require any additional properties to be added to the parent. The remaining strategy classes have some additional model specific features, in accordance with the descriptions in previous section. This modular design offers flexibility, and new strategies for price risk management can easily be added to the package.

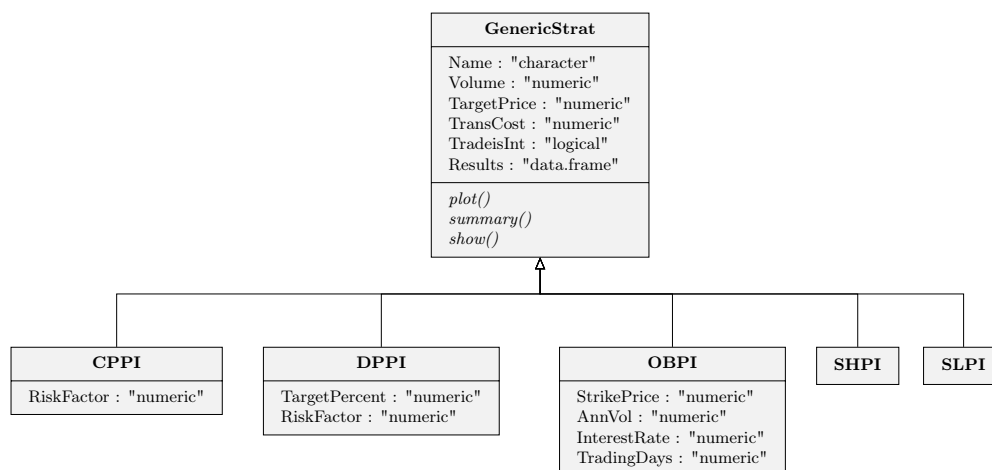


Figure 5: Attributes and methods for portfolio insurance strategy classes in the **etrm** package.

Figure 5 provide an overview of the class hierarchy, and a brief description is given in the following. In "GenericStrat", the "Name" property is used to store a strategy identifier. Allowed character values are "CPPI", "DPPI", "OBPI", "SHPI" and "SLPI". The volume to be managed and the corresponding price cap (floor) can be found in "Volume" and "TargetPrice", respectively. If a transaction cost has been included in the calculation of the portfolio price, this is to be found in "TransCost". One may also set a restriction on transactions by requiring that the smallest volume available for trading is

equal to 1 unit. This lot size limitation is stored as TRUE/FALSE in "TradeisInt". The main output from a strategy calculation can be found in "Results". This data frame keeps daily values for market prices, transactions, exposed volume, open volume, hedge rate, target price and portfolio price.

The generic methods `plot()`, `summary()` and `show()` are implemented in "GenericStrat" and inherited by the strategy classes. The `plot()` method returns a chart based on "Results", with daily values for portfolio, market and target prices and portfolio hedge rate. The `summary()` method returns a list with five elements; a description string, portfolio volume, target price, calculated churn rate and a data frame with summary statistics for the trading period. Finally, the `show()` method returns the "Results" data frame.

Argument	Description	Default value
q	Numeric volume	none
tdate	Date vector with trading days	none
f	Numeric price vector	none
tcost	Numeric transaction cost	0
int	Logical lot size integer restriction	TRUE

Table 3: Arguments shared by the portfolio insurance strategy functions.

The strategy constructor functions `cpqi()`, `dppi()`, `obpi()`, `shpi()` and `slpi()` share five of the arguments, see Table 3. Each strategy require some additional arguments to implement the models presented in previous section. All of these inputs are of "numeric" data type. They are summarised in Table 4.

Function	Argument	Description	Default value
<code>cpqi()</code>	tper	Target price factor	none
	rper	Risk factor percentage	none
<code>dppi()</code>	tper	Target price factor	none
	rper	Risk factor percentage	none
<code>obpi()</code>	k	Option strike price	$k = f_0$
	vol	Annualized volatility	none
	r	Interest rate	0
	tdays	Trading days per year	250
	daysleft	Days left to expiry	none
<code>shpi()</code>	tper	Target price factor	none
	daysleft	Days left to expiry	none
<code>slpi()</code>	tper	Target price factor	none

Table 4: Model specific arguments for the portfolio insurance strategy functions.

To illustrate further, we proceed with an example. Consider a European consumer of electricity procuring 30 MW to be delivered in 2006. The CAL-06 baseload power future from the synthetic `etrm` "powcal" data set is used as hedging instrument. Trading is started 500 days prior to the contract expiry, approximately a horizon of 2 years. For the "OBPI" strategy presented below, the target price is calculated as an expected price cap given by the option premium-adjusted strike price selected for the delta hedging scheme within a standard Black-76 option pricing framework. The default `obpi()` strike price is set at-the-money, in this case at 26.82 EUR/MWh. The expected target price illustrated with the horizontal dotted line in Figure 6 is calculated to be 29.84 EUR/MWh. The "OBPI" delta hedging scheme dictate an initial hedge rate of 57 percent, and the consumer enters a 17 MW position in CAL-06 on the first day of trading.

As time progresses and the market price changes, the `obpi()` function adjust the required hedge rate in order to replicate the call option on the CAL-06 contract. Hedge rate is gradually built up as the market increase from the second quarter of 2004, followed by a reduction after the sharp price drop starting late in the same year. Eventually, the volume is fully hedged due to the strong upwards price trend in 2005. The CAL-06 contract closes at 37.81 EUR/MWh on the expiry date, while the consumer has a hedge of 30 MW and a portfolio price of 29.29 EUR/MWh. The calculated price of the option to be synthesized (and the delta hedges) will depend on the Black-76 model parameters. In this example the risk free rate of interest is set to $r = 0$ and annualized volatility σ is assumed to be 20 percent. The following code will implement the strategy and create the plot in Figure 6:

```
# data frame with final 500 trading days for CAL-06 contract
dat06 <- tail(na.omit(powcal[, c(1,2)]), 500)

# instance of the OBPI class
cal06_obpi_b <- obpi(q = 30,
  tdate = dat06$Date,
  f = dat06$`CAL-06`,
  k = dat06$`CAL-06`[1],
  vol = 0.2,
  r = 0,
  tdays = 250,
  daysleft = 500,
  tcost = 0,
  int = TRUE)

# the generic plot() method
plot(cal06_obpi_b, title = "", legend = "right", ylab.1 = "EUR/MWh")
```

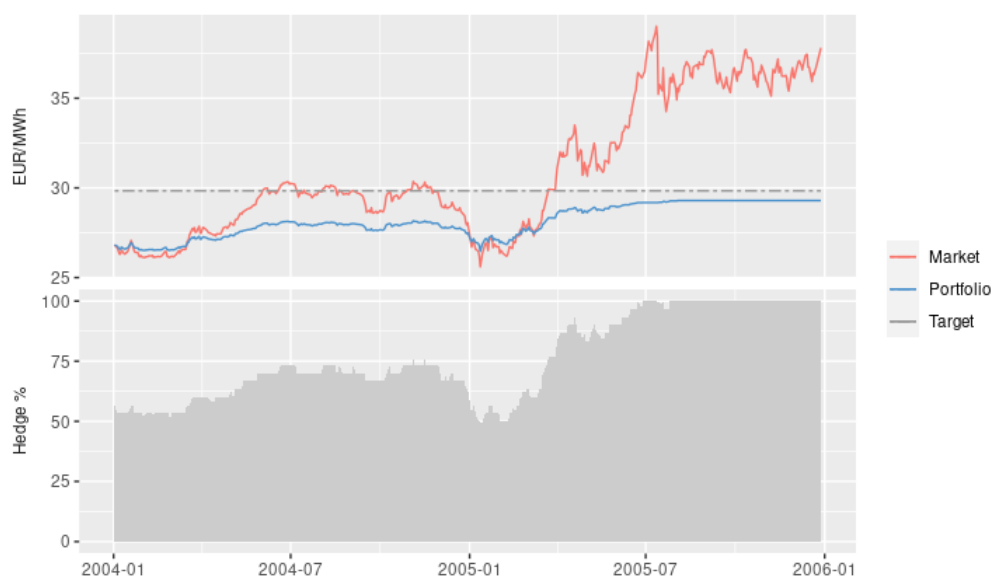


Figure 6: Option based portfolio insurance (OBPI) strategy for buyer CAL-06. Daily observations for prices (top panel) and hedge rate (bottom panel). As the market price continue to rise, the hedge rate is increased and the portfolio price is locked below the target price level.

An aggregated view of the trading activity over the 2 year period and final results can be retrieved by running the `summary()` method on the object created above:

```
> summary(cal06_obpi_b)
$Description
[1] "Hedging strategy of type OBPI and length 500"

$Volume
[1] 30

$Target
[1] 29.83626

$ChurnRate
[1] 4.333333
```

\$Stats

	Market	Trade	Exposed	Position	Hedge	Target	Portfolio
First	26.82	17	13	17	0.5666667	29.83626	26.82000
Max	39.01	17	17	30	1.0000000	29.83626	29.29433
Min	25.60	-3	0	13	0.4333333	29.83626	26.46833
Last	37.81	0	0	30	1.0000000	29.83626	29.29433

We note from the "ChurnRate" that the underlying 30 MW volume had to be traded 4.33 times in order to synthesize the call option and achieve the results summarised in "Stats". By also considering the trading costs in the calculations, the user can get valuable inputs when considering alternatives, such as simply buying the option in the market. However, such contract may not always be available.

Finally, the show() method provide details regarding daily values for market price, transactions, exposed volume, futures contract position, hedge rate, the target price and the calculated portfolio price:

```
> head(show(cal06_obpi_b))
```

	Date	Market	Trade	Exposed	Position	Hedge	Target	Portfolio
1	2004-01-02	26.82	17	13	17	0.5666667	29.83626	26.82000
2	2004-01-05	26.63	-1	14	16	0.5333333	29.83626	26.73767
3	2004-01-07	26.31	0	14	16	0.5333333	29.83626	26.58833
4	2004-01-08	26.31	0	14	16	0.5333333	29.83626	26.58833
5	2004-01-09	26.54	0	14	16	0.5333333	29.83626	26.69567
6	2004-01-12	26.32	0	14	16	0.5333333	29.83626	26.59300

For the sake of comparison, the OBPI strategy for CAL-06 from a sellers point of view can be implemented with similar assumptions by setting the volume to $q = -30$. Using the default at-the-money strike price, the seller calculates an expected target floor to protect at 23.80 EUR/MWh and an initial hedge rate of 43 percent. As the market starts to rise, the hedge is reduced. The seller increases the hedge in late 2004 to dampen the effect from the market drop, and finally exits the forward market positions as the price increases during 2005. The portfolio price follows the market upwards with a premium for the put option replication, as expected for an insurance scheme. The CAL-06 contract closes at 37.81 EUR/MWh, and the seller has a portfolio price of 35.34 EUR/MWh, which may be locked in on the final trading day.

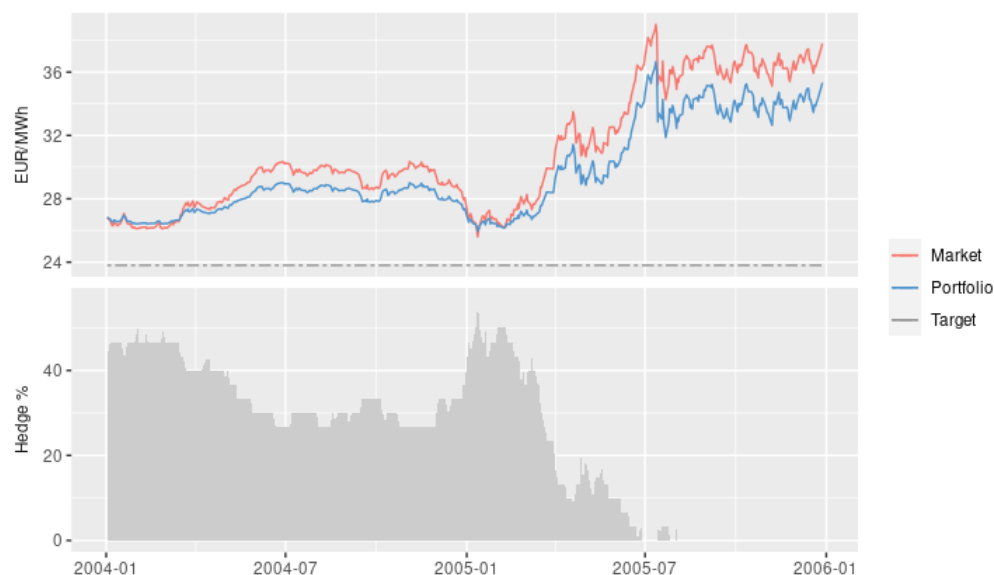


Figure 7: Option based portfolio insurance (OBPI) strategy for seller CAL-06. Daily observations for prices (top panel) and hedge rate (bottom panel). The hedge rate is lowered in the upwards trending market, and the portfolio price continue to increase.


```

# instance of the OBPI class
cal06_obpi_s <- obpi(q = - 30,
                    tdate = dat06$Date,
                    f = dat06$`CAL-06`,
                    k = dat06$`CAL-06`[1],
                    vol = 0.2,
                    r = 0,
                    tdays = 250,
                    daysleft = 500,
                    tcost = 0,
                    int = TRUE)

# the generic plot() method
plot(cal06_obpi_s, title = "", legend = "right", ylab.1 = "EUR/MWh")

> summary(cal06_obpi_s)
$Description
[1] "Hedging strategy of type OBPI and length 500"

$Volume
[1] -30

$Target
[1] 23.80374

$ChurnRate
[1] 4.2

$Stats
      Market Trade Exposed Position    Hedge    Target Portfolio
First  26.82   -13    -17     -13 0.4333333 23.80374  26.82000
Max    39.01     2    -13      0 0.5666667 23.80374  36.64867
Min    25.60   -13   -30     -17 0.0000000 23.80374  25.95167
Last   37.81     0   -30      0 0.0000000 23.80374  35.33567

```

In the examples above, we have implicitly assumed that both the consumer and the seller have a flat volume corresponding to 30 MW over the entire year which can be covered by a base load contract such as the CAL-06. In practice, this is typically not the case. Industrial energy consumers will have consumption profiles determined by the activity level in their production facilities, and often face seasonal shifts due to variation in demand, or holidays. Weather also play a large role, both for consumers and producers such as hydroelectric plants. In order to hedge the predicted volume more precisely, some of the other contract types presented in Table 2 will need to be included in the portfolio. Market players will "roll forward" and start trading contracts covering shorter periods such as quarters, months and weeks, as they become available. The mandate for the energy portfolio will typically be broken down into smaller time intervals with expected volume and required hedge levels. All strategies presented here may be used to make decisions for several years and their sub periods, and the market value of a specific volume prognosis and corresponding futures positions can be evaluated using the forward curve discussed in previous sections.

In order to maintain focus on the strategies themselves, we continue with the baseload example with 30 MW. In Figure 8 we plot results for the remaining four strategies for the consumer hedging with CAL-06. The benchmark strategies "SHPI", and "SLPI" follow simple, mechanical patterns. The "SHPI" builds a full hedge gradually over the trading period, ending at either the average forward market price for the period, or the target price. This approach will always ensure a full hedge at expiry, without intervention. The "SLPI" does not take any positions unless the target is reached, ending either at the target level, or leaving an option to close at the contract expiry price. As the CAL-06 increase significantly during 2005, both end up at the target level.

The "CPPI", and "DPPI" strategies are more dynamic and adjust hedge rate according to market developments and the model parameters. As the "DPPI" implements a dynamic risk factor, μ_t , the strategies are geared differently. In this example, the "DPPI" successfully adjusts the target price downward on one of the first trading days, and achieves a lower portfolio price on last trading day.

A similar overview from a seller's perspective is provided in Figure 9. As the market trends upwards during the hedging period, none of the strategies end up at the initial target price. The "SHPI" builds the hedge positions in a step-wise manner, ending up with a portfolio price equal to

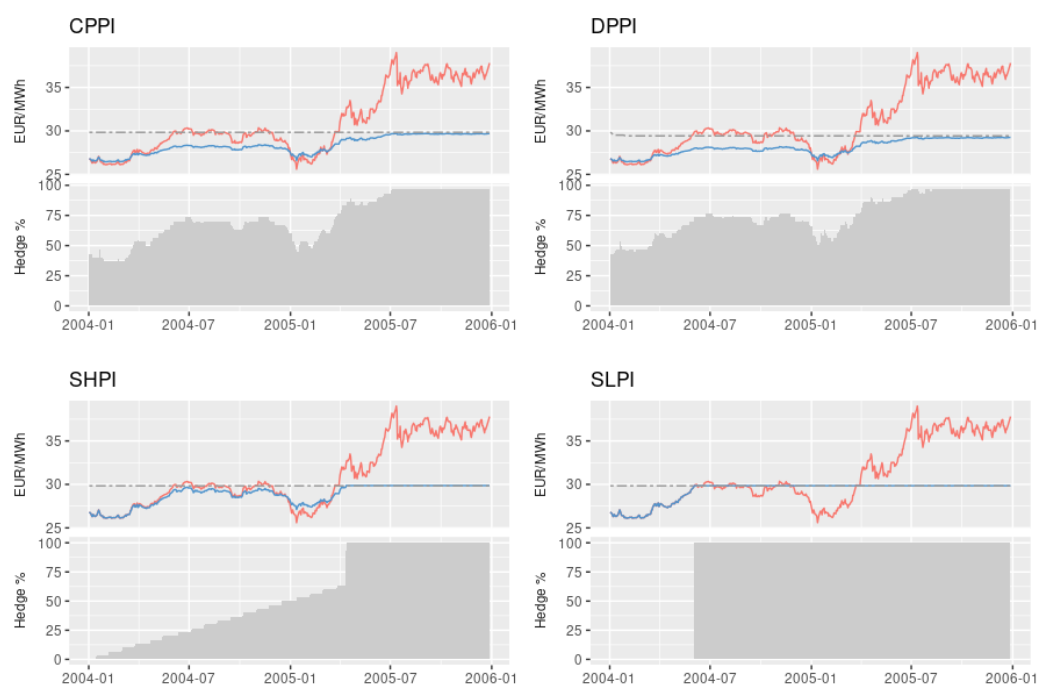


Figure 8: Achieved results for the strategies CPPI, DPPI, SHPI and SLPI for buyer CAL-06. Daily observations for prices (top panels) and hedge rate (bottom panels).

the average futures market price for the period. The "SLPI" does not enter any positions, leaving an option to lock in market price at expiry. Finally, we can also here see some differences between "CPPI", and "DPPI". This is due to the dissimilar gearing of the portfolios, but also because of the rather frequent adjustments of the target price by "DPPI". In order to protect the higher targets, hedge rate must be increased and "DPPI" falls behind "CPPI" and ends up at a lower portfolio price for the seller.

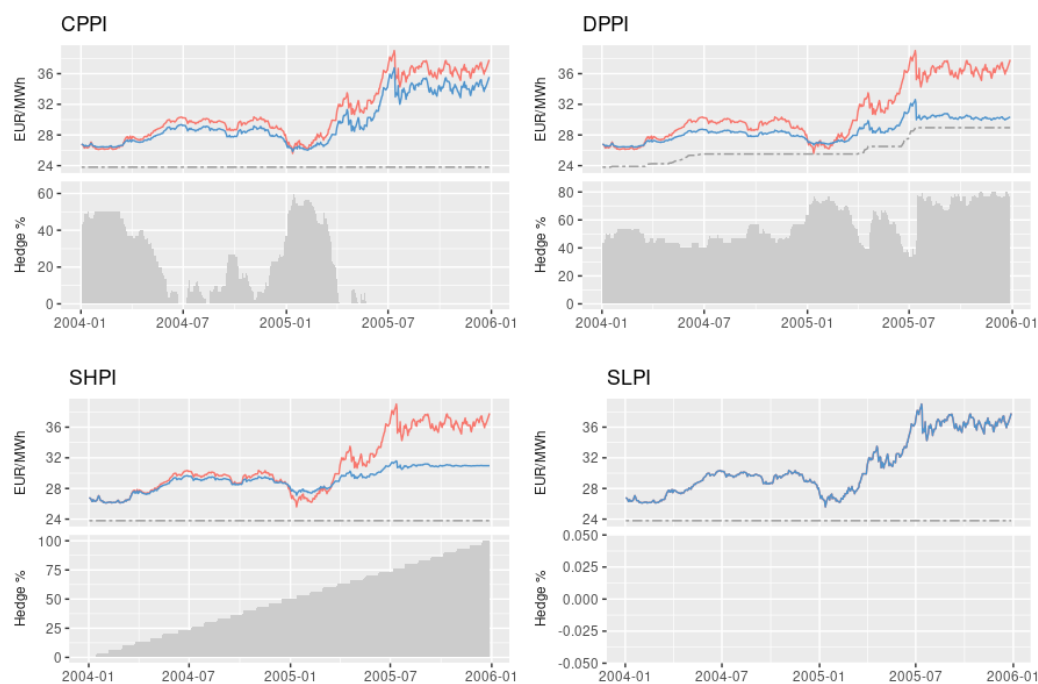


Figure 9: Achieved results for the strategies CPPI, DPPI, SHPI and SLPI for seller CAL-06. Daily observations for prices (top panels) and hedge rate (bottom panels).

etrm can also be used in conjunction with other R packages to evaluate risks related to energy procurement. Metrics such as *Value-at-Risk* and *Expected Shortfall* can for example be calculated using the **PerformanceAnalytics** package. We will proceed with a simple, illustrative example. Consider the OBPI portfolios "cal06_obpi_b" and "cal06_obpi_s" in the code example above. If we need to calculate risk measures at a specific point in time, say at day 350 in the trading period, we can execute the following code:

```
library(PerformanceAnalytics)

# CAL-06 returns prior to t=350
ret_06 <- head(diff(log(show(cal06_obpi_b)$Market)), 349)

# portfolio status at t=350
pdat <- rbind(
  Buyer = show(cal06_obpi_b)[350,],
  Seller = show(cal06_obpi_s)[350,]
)

# add risk measures to pdat
pdat <- cbind(pdat,
  VaR = abs(rep(VaR(ret_06, p=.95, method="historical"), 2)*pdat$Market*pdat$Exposed*8760),
  ES = abs(rep(ES(ret_06, p=.95, method="historical"), 2)*pdat$Market*pdat$Exposed*8760))
```

The calculation above evaluate market risk related to the unhedged volume (exposed MW \times 8760 hours in the year 2006) at current market prices under the (simplistic) assumption of symmetry in the returns distribution. The portfolio status, including risk metrics is

```
> pdat[c(-1, -3)]
```

	Market	Exposed	Position	Hedge	Target	Portfolio	VaR	ES
Buyer	32.54	3	27	0.9	29.84	28.98	10671.55	18237.06
Seller	32.54	-27	-3	0.1	23.81	30.38	96043.94	164133.52

Overview of the etrm package

Package **etrm** offers an open source implementation of core functionalities of an ETRM system:

- Construction of forward curves
- Strategies for price risk management

Functions included in the package are listed in Table 5.

Function	Description
msfc()	Maximum Smoothness Forward Curve
cppi()	Constant Proportion Portfolio Insurance
dppi()	Dynamic Proportion Portfolio Insurance
obpi()	Option Based Portfolio Insurance
shpi()	Step Hedge Portfolio Insurance
slpi()	Stop Loss Portfolio Insurance

Table 5: Overview of **etrm** package functions

All functions act as constructors for their corresponding S4 classes, as described in further detail in previous sections. The classes all have generic methods `plot()`, `summary()` and `show()`. Unit tests covering all functions in **etrm** have been implemented using the **testthat** framework introduced in Wickham (2011b).

Three synthetic data sets are included in the package, see Table 6. The "powfutures130513" and "powpriors130513" data may be used to create forward curves with the `msfc()` function for the trading date 2013-05-13. The portfolio insurance strategies may be tested on the "powcal" data set, which contains historical prices for 11 base load power futures.

Data set	Description
powfutures130513	Synthetic data for a set of electricity base load futures quoted at 2013-05-13. Closing prices for contracts with weekly, monthly, quarterly and yearly settlement periods
powpriors130513	Two simple priors for forward market price curve Daily values for calculation to be used with powfutures130513
powcal	Synthetic data set with daily closing prices for 11 electricity base load futures with yearly settlement periods for 2006-2016

Table 6: Overview of `etrm` package data sets

Summary and suggestions for future work

This paper introduces `etrm`, an R package for energy market risk management. The package contains tools previously not available in the R ecosystem, such as the `msfc()` function for building a forward curve for energy commodities with flow delivery contracts and strong seasonality. The forward curve is a key decision making tool with many uses, such as pricing non-standard supply agreements, investment decisions and risk management. `etrm` also provides implementations of portfolio insurance strategies for handling price risk, suitable for both long and short hedgers. The functions can be used for back testing strategies on historical futures price data, risk and strategy evaluations, and as decision support tools for trade execution.

The `etrm` package may be developed further by incorporating new elements. First, the forward curve calculation may be done on an hourly level. The bid-ask spread can be used as price constraint for the optimization, as an extension of the current solution based on closing prices. Competing forward curve calculation methods can also be added to the package, and new asset allocation strategies for price risk management could be included.

A further extension of `etrm` functionality can be to implement a "PORTFOLIO" class, consisting of a daily volume prognosis covering the full management horizon, supplemented with authorized volumes per (sub)period and hedging strategy objects implemented in accordance with these authorizations. The portfolio object could contain multiple strategy objects for contracts such as "year", "quarter", "month" and "week", depending on the shape of the volume prognosis. This construction can be priced using the forward curve, and portfolio wide risk measures could be calculated via Monte Carlo simulations on the curve.

Acknowledgments

We thank the editor and two anonymous reviewers for their constructive comments, which helped us to improve the manuscript. This work has been supported by The Norwegian Research Council via the industrial Ph.D. scheme. The author also wish to thank Montel AS and former colleagues in Kinect Energy Markets AS for interesting discussions and helpful comments.

Bibliography

K. J. Adams and D. R. Van Deventer. Fitting yield curves and forward rate curves with maximum smoothness. *Journal of Fixed Income*, pages 52–62, 1994. URL <https://doi.org/10.3905/jfi.1994.408102>. [p325]

N. Anderson, F. Breedon, M. Deacon, A. Derry, and G. Murphy. *Estimating and Interpreting the Yield Curve*, volume 6. John Wiley & Sons Incorporated, 1996. [p325]

M. Benini, M. Marracci, P. Pelacchi, and A. Venturini. Day-ahead market price volatility analysis in deregulated electricity markets. In *IEEE Power Engineering Society Summer Meeting*, volume 3, pages 1354–1359 vol.3, 2002. URL <https://doi.org/10.1109/PESS.2002.1043596>. [p324]

F. E. Benth and M. Schmeck. Pricing and hedging options in energy markets using black-76. *Journal of Energy Markets*, 7(2), 2014. URL <https://doi.org/10.21314/JEM.2014.114>. [p324]

- F. E. Benth, J. Kallsen, and T. Meyer-Brandis. A non-gaussian ornstein–uhlenbeck process for electricity spot price modeling and derivatives pricing. *Applied Mathematical Finance*, 14(2):153–169, 2007a. URL <https://doi.org/10.1080/13504860600725031>. [p324]
- F. E. Benth, S. Koekkebakker, and F. Ollmar. Extracting and applying smooth forward curves from average-based commodity contracts with seasonal variation. *The Journal of Derivatives*, 15(1):52–66, 2007b. URL <https://doi.org/10.3905/jod.2007.694791>. [p324, 325]
- F. E. Benth, J. S. Benth, and S. Koekkebakker. *Stochastic Modelling of Electricity and Related Markets*, volume 11. World Scientific, 2008. URL <https://doi.org/10.1142/6811>. [p324, 327]
- H. Bessembinder and M. L. Lemmon. Equilibrium pricing and optimal hedging in electricity forward markets. *The Journal of Finance*, 57(3):1347–1382, 2002. URL <https://doi.org/10.1111/1540-6261.00463>. [p324]
- T. Bjork. *Arbitrage Theory in Continuous Time*. Oxford University Press, 3 edition, 2009. URL <https://EconPapers.repec.org/RePEc:oxp:obooks:9780199574742>. [p335]
- F. Black. The pricing of commodity contracts. *Journal of Financial Economics*, 3(1):167–179, 1976. URL [https://doi.org/10.1016/0304-405X\(76\)90024-6](https://doi.org/10.1016/0304-405X(76)90024-6). [p334]
- F. Black and R. W. Jones. Simplifying portfolio insurance. *The Journal of Portfolio Management*, 14(1):48–51, 1987. URL <https://doi.org/10.3905/jpm.1987.409131>. [p334]
- S. Borak and R. Weron. *A Semiparametric Factor Model for Electricity Forward Curve Dynamics*. Humboldt-Universität zu Berlin, Wirtschaftswissenschaftliche Fakultät, 2008. URL <http://dx.doi.org/10.18452/4142>. [p325]
- M. Burger, B. Klar, A. Müller, and G. Schindlmayr. A spot market model for pricing derivatives in electricity markets. *Quantitative Finance*, 4:109–122, 2004. URL <https://doi.org/10.1088/1469-7688/4/1/010>. [p324]
- J.-S. Chen, C.-L. Chang, J.-L. Hou, and Y.-T. Lin. Dynamic proportion portfolio insurance using genetic programming with principal component analysis. *Expert Systems with Applications*, 35(1):273–278, 2008. URL <https://doi.org/10.1016/j.eswa.2007.06.030>. [p334]
- A. Deoras. Electricity load and price forecasting webinar case study, 2021. <https://www.mathworks.com/matlabcentral/fileexchange/28684-electricity-load-and-price-forecasting-webinar-case-study> [Accessed: 2021-04-06]. [p324]
- A. Eydeland and K. Wolyniec. *Energy and Power Risk Management: New Developments in Modeling, Pricing, and Hedging*, volume 97. John Wiley & Sons, 2002. [p324, 325, 333]
- S. Farrington. Energy risk software rankings: A different world, 2020. <https://www.risk.net/commodities/energy/7511636/energy-risk-software-rankings-a-different-world> [Accessed: 2021-04-05]. [p324]
- S.-E. Fleten and J. Lemming. Constructing forward price curves in electricity markets. *Energy Economics*, 25(5):409–424, 2003. URL [https://doi.org/10.1016/S0140-9883\(03\)00039-2](https://doi.org/10.1016/S0140-9883(03)00039-2). [p325]
- M. Hildmann, E. Kaffe, Y. He, and G. Andersson. Combined estimation and prediction of the hourly price forward curve. In *2012 IEEE Power and Energy Society General Meeting*, pages 1–8, 2012. URL <https://doi.org/10.1109/PESGM.2012.6345333>. [p325]
- J. Janczura, S. Trück, R. Weron, and R. C. Wolff. Identifying spikes and seasonal components in electricity spot price data: A guide to robust modeling. *Energy Economics*, 38:96–110, 2013. [p324]
- D. S. Kirschen and G. Strbac. *Fundamentals of Power System Economics*. John Wiley & Sons, 2018. [p324, 325, 333]
- H.-I. Lee, M.-H. Chiang, and H. Hsu. A new choice of dynamic asset management: the variable proportion portfolio insurance. *Applied Economics*, 40(16):2135–2146, 2008. URL <https://doi.org/10.1080/00036840600949280>. [p334]
- H. Leland and M. Rubinstein. The evolution of portfolio insurance. *Dynamic Hedging*, 01 1976. URL https://www.researchgate.net/publication/265430746_The_Evolution_of_Portfolio_Insurance. [p324, 333, 334]
- H. E. Leland. Who should buy portfolio insurance? *The Journal of Finance*, 35(2):581–594, 1980. URL <http://www.jstor.org/stable/2327419>. [p324, 333]

- K. G. Lim and Q. Xiao. Computing maximum smoothness forward rate curves. *Statistics and Computing*, 12(3):275–279, 2002. URL <https://doi.org/10.1023/A:1020707028156>. [p326]
- J. H. McCulloch. Measuring the term structure of interest rates. *The Journal of Business*, 44(1):19–31, 1971. URL <http://www.jstor.org/stable/2351832>. [p325]
- M. Nicolosi. Wind power integration and power system flexibility—an empirical analysis of extreme events in germany under the new negative price regime. *Energy Policy*, 38(11):7257–7268, 2010. ISSN 0301-4215. URL <https://www.sciencedirect.com/science/article/pii/S0301421510005860>. Energy Efficiency Policies and Strategies with regular papers. [p324]
- F. Ollmar. An analysis of derivative prices in the Nordic power market. 2003. URL <http://hdl.handle.net/11250/164248>. [p324, 325, 326, 327]
- A. Perold. Constant proportion portfolio insurance. Harvard business school. *Unpublished manuscript*, 1986. [p334]
- A. F. Perold and W. F. Sharpe. Dynamic strategies for asset allocation. *Financial Analysts Journal*, 44(1): 16–27, 1988. URL <http://www.jstor.org/stable/4479087>. [p324, 333]
- S. Sundar. Energy trading & risk management with Matlab webinar case study, 2021. <https://www.mathworks.com/matlabcentral/fileexchange/28056-energy-trading-risk-management-with-matlab-webinar-case-study> [Accessed: 2021-04-06]. [p324]
- H. Wickham. ggplot2. *WIREs Computational Statistics*, 3(2):180–185, 2011a. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/wics.147>. [p328]
- H. Wickham. testthat: Get Started with Testing. *The R Journal*, 3(1):5–10, 2011b. URL <https://doi.org/10.32614/RJ-2011-002>. [p341]

Anders D. Sleire
Department of Mathematics
University of Bergen
P.O.Box 7803
N-5020 Bergen
Norway
Anders.Sleire@uib.no