

Project memo

Treating nonconvex PQ-curves in ProdRisk final simulation

Subtitle

VERSION
1.0

DATE
2017-11-20

AUTHOR(S)
Arild Helseth

CLIENT(S)
IBM project steering group

CLIENTS REF.

PROJECT NO.
502000395

NO. OF PAGES AND APPENDICES:
12 + appendices

ABSTRACT

The ProdRisk model requires a convex model formulation and is based on linear programming. This requirement is important to obey in the computation of the strategy (cuts). Once the cuts have been computed, nonconvex relationships can be treated in the final simulation to improve the accuracy of the simulation results. This memo documents the implementation details of the treatment of nonconvex PQ-curves in ProdRisk, and provides a basic user guide for the implemented functionality.


PREPARED BY
Arild Helseth

SIGNATURE



APPROVED BY
Birger Mo

SIGNATURE



PROJECT MEMO NO.
AN.17.12.60

CLASSIFICATION
Unrestricted

Document history

VERSION	DATE	VERSION DESCRIPTION
1.0	2017-11-21	First version

Table of contents

1	Introduction	4
2	Treatment of PQ-curves in ProdRisk.....	4
3	Mathematical modeling.....	7
4	On the choice of MIP-solver	8
5	Testing	8
6	Using the functionality.....	12

1 Introduction

In the LTM/ProdRisk context, a PQ-curve describes the functional relationship between the power output (MW) from a power station and the corresponding water discharge (m³/s) through the station. A power station can comprise multiple generators, and the individual PQ-curves for each generator should be joined into one PQ-curve representing the station.

ProdRisk is based on the SDDP method. Convexity is a basic requirement for the SDDP method; using a nonconvex¹ PQ-curve would violate this requirement.

In this document, we describe a ProdRisk extension allowing treatment of nonconvex PQ-curves in the final simulation. Simulation with a nonconvex PQ-curve may improve accuracy and realism of the simulation results, but will not impact the strategy (cuts) computed in ProdRisk.

This work was carried out as a part of the research project "Integrating Balancing Markets in Hydropower Scheduling Methods". In that context, the treatment of nonconvex PQ-curves is of particular importance when studying a system's capability to operate below the best efficiency points, e.g. to deliver down-regulation capability from rotating machinery.

2 Treatment of PQ-curves in ProdRisk

For the further discussion, we divide ProdRisk's tasks in two; an *optimization* part and a *final simulation* part. The optimization part comprises both forward and backward iterations. The PQ-curves that are provided by the model user are "convexified" internally in ProdRisk before used in the optimization part of the program. That is, the first PQ-segment has the highest efficiency, and the subsequent segments have lower efficiencies than the previous ones. If a nonconvex PQ-curve is defined by the user, ProdRisk establishes a convex approximation for this purpose by omitting nonconvex points on the curve.

Consider a PQ-curve, as defined by the user, in Figure 1. The curve comprises 3 segments, each described by a discharge variable q and a corresponding efficiency (MW/m³/s) x . This curve is not convex, since $\eta_2 > \eta_1$. If this curve is used directly in an LP problem, segment 2 (with η_2 and q_2) will be loaded before segment 1 (x_1 and q_1). There are no exact ways of telling the LP solver to always use segment 1 before segment 2.

¹ In reality the PQ curve should be *concave*, i.e. an decreasing efficiency with increasing discharge. In this document we use the term *convex* instead to indicate if the curve supports a convex formulation of the optimization problem.

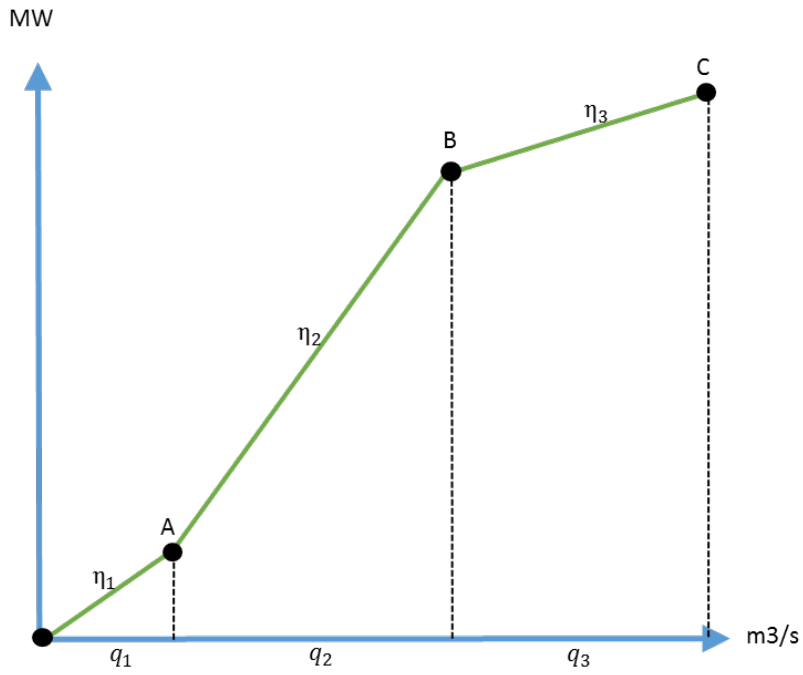


Figure 1 PQ-curve example.

The curve in Figure 1 is convexified in the optimization part. This is done as illustrated in Figure 2, by omitting point A from Figure 1. Now $\eta_1 > \eta_2$.

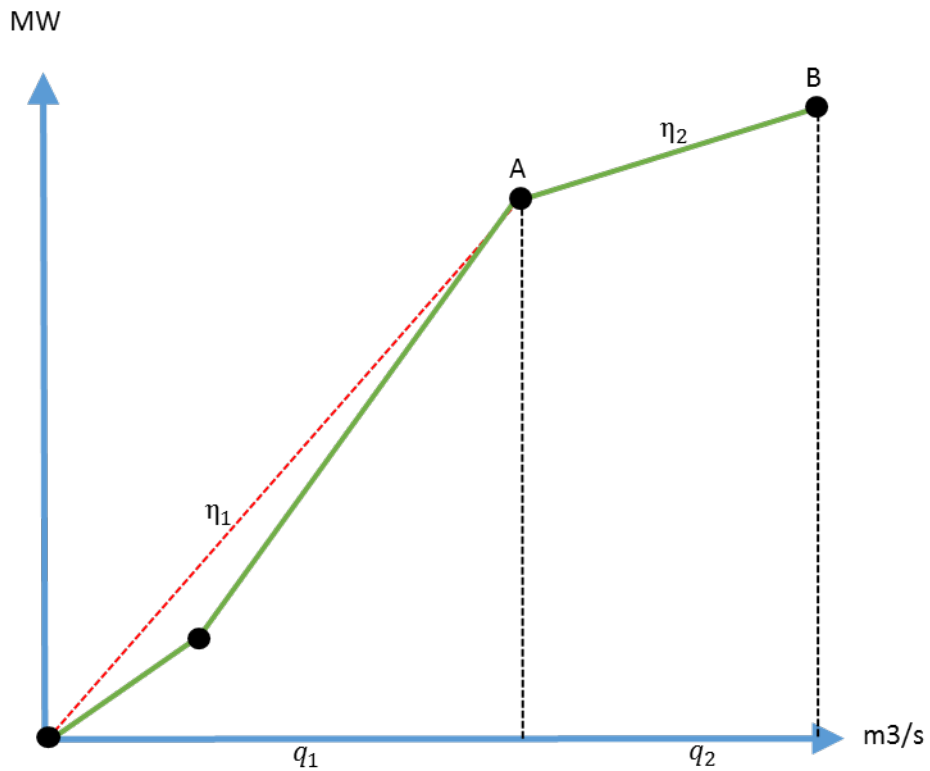


Figure 2 Convexified PQ-curve.

Once the cuts have been computed in the optimization part, a final simulation is carried out. The results (generation, water discharges etc.) from this simulation are written to result files and are those seen by the user. Since the simulation part is decoupled from the cut creation in the optimization part, there are no longer a strict requirement on convexity. Currently, the final simulation in ProdRisk is also based on LP and uses the convexified PQ-curve. ProdRisk allows the user to settle ("avregne") the production for each station according to the original PQ-curve. This functionality is governed by the parameter `PQValg` in `ProdRisk.CPAR`. In case the original PQ-curve is nonconvex, the total settled production will be lower than what is actually found when solving the LP problem in the final simulation. If `PQValg` is set to 0, ProdRisk will use the convex approximation in the final simulation.

The settlement process is illustrated in Figure 3. Consider a minimum discharge requirement Q_{min} . From the simulation ProdRisk finds that the station generates p_1 when running at Q_{min} , based on the convexified PQ-curve. In the settlement, the real curve is used, yielding p_2 . From this illustration it is clear that the use of convexified PQ-curves overestimates the generation capability of the station.

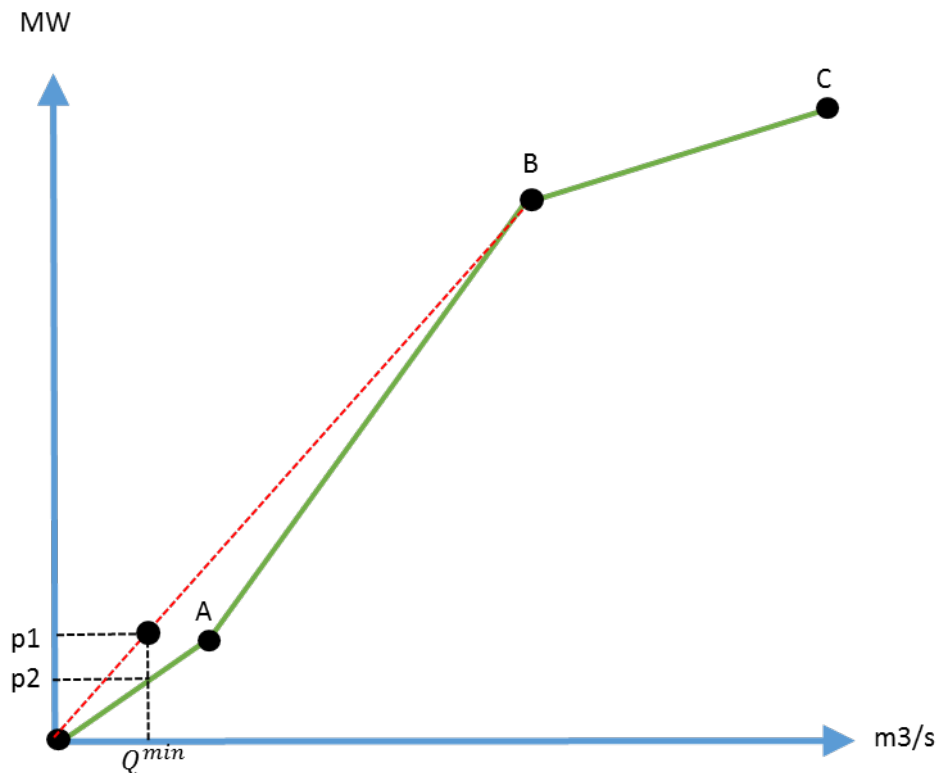


Figure 3 Final simulation settlement.

In this document we describe how the "true" shape of the PQ-curves can be represented by use of MIP formulation in the final simulation in ProdRisk. The impact of overestimated production capability will persist in the optimization part of ProdRisk, .i.e. the cuts will be not be affected when using the new functionality.

3 Mathematical modelling

In order to force ProdRisk to discharge according to the piecewise linear and nonconvex curve in Figure 1, one needs to reformulate the LP problem to an MIP problem. There are several known techniques for this type of transformation. We will not go into details about the strengths and weaknesses of the different techniques here, but point the interested reader to the literature, e.g. the following reference:

Padberg, M.: ‘*Approximating separable nonlinear functions via mixed zero-one programs*’, Oper. Res. Lett., 2000, 27, (1), pp. 1–5.

We chose a formulation presented as "Model 1" in the reference above. A short description of the resulting mathematical formulation in ProdRisk is presented below.

In an LP model the power generation is found by summing the product of efficiency (η) and discharge (q) for each PQ-curve segment $s=1..NS$, as shown in Equation (1).

$$P = \sum_{s=1}^{NS} \eta_s q_s \quad (1)$$

As discussed previously, this formulation fails to load the discharge segments in the right order if the PQ curve is nonconvex. Thus, we need a set of auxiliary variables enforcing the segments to be loaded in the correct sequence.

We introduce one binary variable x_s per segment $s=1..NS$ and add the following equations to constrain the discharge variables q_s :

$$\begin{aligned} x_1 \bar{Q}_1 &\leq q_1 \leq \bar{Q}_1 \\ x_2 \bar{Q}_2 &\leq q_2 \leq \bar{Q}_2 x_1 \\ &\dots \\ x_s \bar{Q}_s &\leq q_s \leq \bar{Q}_s x_{s-1} \\ &\dots \\ 0 &\leq q_{NS} \leq \bar{Q}_{NS} x_{NS-1} \end{aligned} \quad (2)$$

In order to use q_2 , x_1 should be 1, forcing q_1 to be at its upper boundary \bar{Q}_1 . In order to use q_3 , x_2 should be 1, forcing q_2 to be at its upper boundary \bar{Q}_2 , and so on. The set of equations in (2) forces

$x_1 \geq x_2 \geq \dots \geq x_s \geq \dots \geq x_{NS}$. Note that variables q_1 and q_{NS} do not have binary variables associated with their upper and lower boundary, respectively. In practice we split the range constraints in (2) in one lower- and one upper-boundary constraint.

The auxiliary variables x and constraints in (2) are added per module with nonconvex PQ-curve and per load period ("prisavsnitt"). For each module with nonconvex PQ-curve NS binary variables x_s and $2*(NS-1)$ equations of type (2) are added per load period.

4 On the choice of MIP-solver

The resulting MIP problem to be solved in the final simulation in ProdRisk can be solved by a variety of optimization solvers. We considered CPLEX (commercial) and Coin Cbc (open source, see <https://projects.coin-or.org/Cbc>).

Generally speaking, the difference in performance between commercial and open source solvers seems to be larger for MIP problems than for LP problems. Solution algorithms and heuristics for solving MIP problems have seen significant improvements the last decades, unlike algorithms for solving small- and medium-scale LP problems. Thus, our initial recommendation is to let commercial solvers take care of MIP problems to ensure computational speed and robustness.

On the other hand, by choosing an open source solver, all ProdRisk users will have the opportunity to apply the new functionality to their scheduling problems. Although many users already have a CPLEX version of ProdRisk, most of these have a limited license that does not allow the use of MIP functionality.

To ensure that the functionality can be taken directly in use, we chose to add Coin Cbc solver to ProdRisk to solve the MIP problems in case of nonconvex PQ curves in the final simulation. The Coin Cbc libraries are statically linked with the executable program. We have limited experience with use of the Coin Cbc library, and therefore cannot guarantee for the robustness and computational performance of the Coin Cbc library.

5 Testing

In the following we report test results illustrating the impact of the new functionality on a fictitious test system. The test system comprises two hydropower modules in a cascade, as illustrated in Figure 4. Both PQ-curves are nonconvex, and there is a minimum discharge constraint for the lower module.

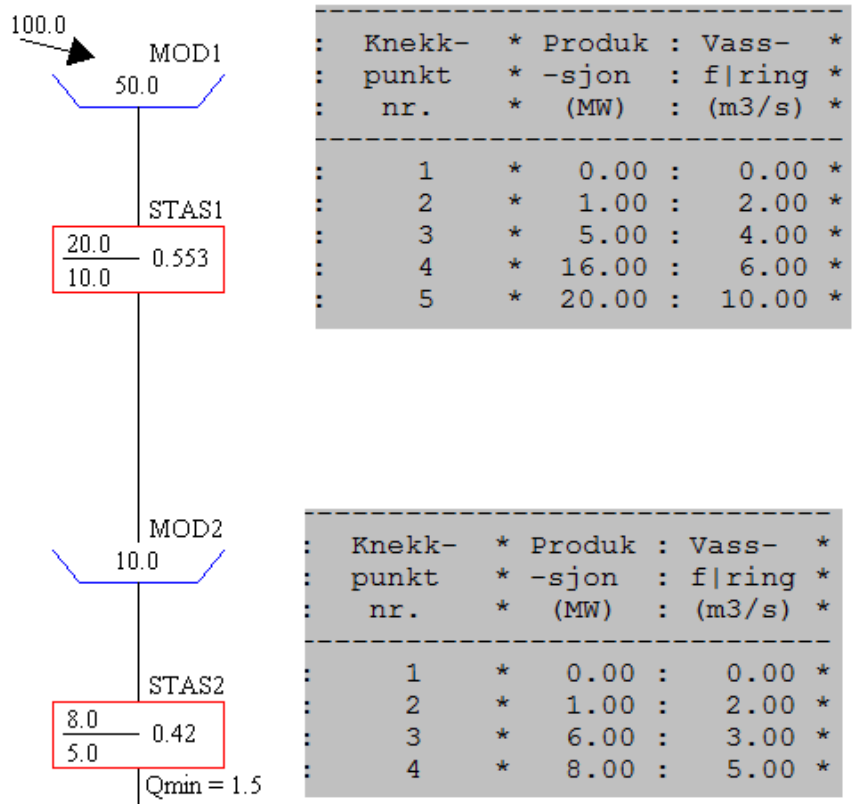


Figure 4 Test system details.

ProdRisk is run with 50 inflow records for a period of 104 weeks considering 5 load periods and reservoir balances within the week. The reservoir trajectories (min, mean, max) for the lower reservoir is shown in Figure 5, comparing a final simulation with LP and MIP using the same set of cuts. It seems like the water values (represented by cuts) are too low for the MIP-model, since the reservoir level on average decreases with time. This type of behaviour may be seen when simulating with finer details than what is allowed in the backward pass in ProdRisk.

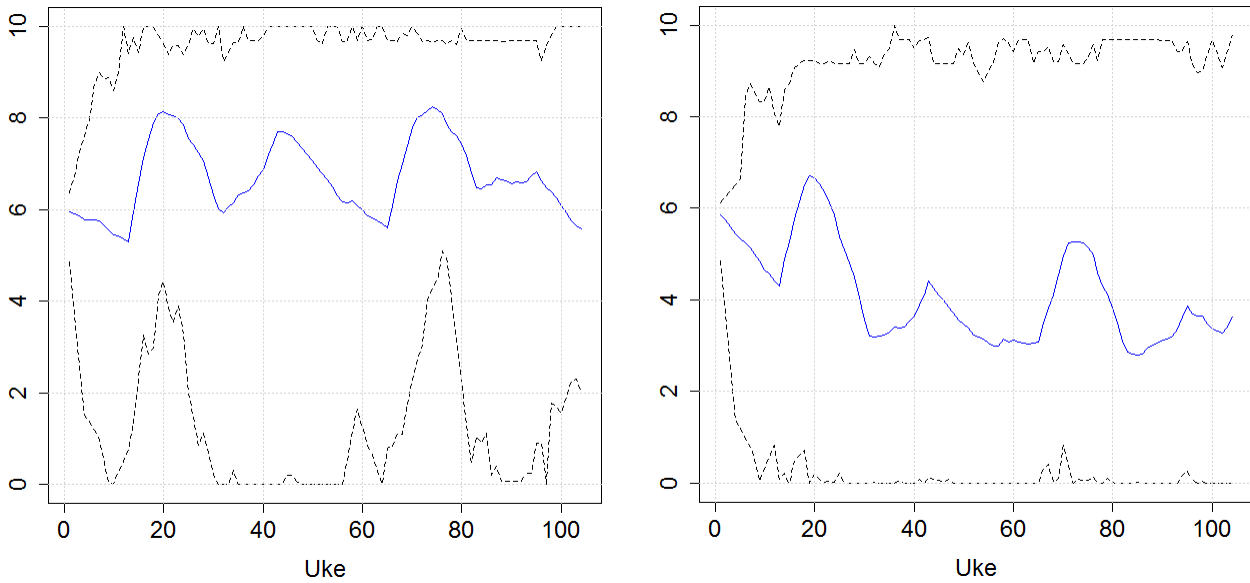


Figure 5 Reservoir trajectories (min, mean, max) for the upper reservoir. Simulation with linear model to the left, and with MIP model to the right.

Duration curves for generation at the lower module are presented in Figure 6. The best efficiency point at 6 MW is more frequently used when simulating with the MIP model. The MIP model discovers that running 1.5 m³/s to meet the minimum discharge requirement rarely pays off, and prefers using more water to operate at a higher efficiency.

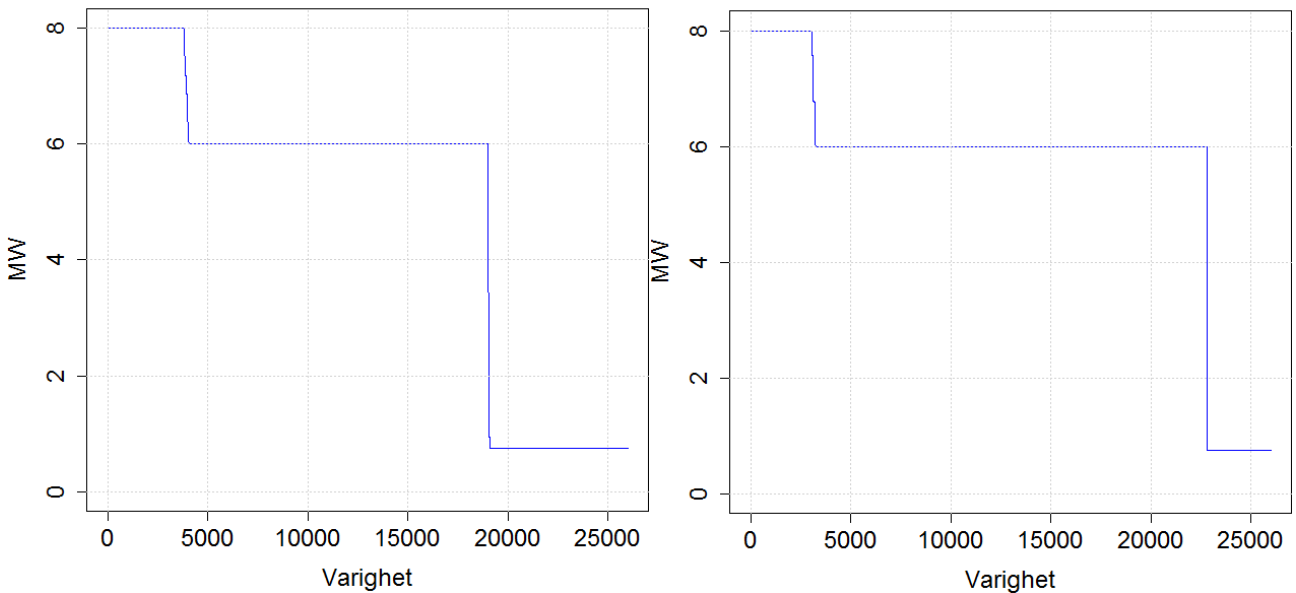


Figure 6 Duration curve for generation for the lower module. Simulation with linear model to the left, and with MIP model to the right.

The final simulation can be run in parallel. We ran the tests on a PC with an Intel Core i7-4940MX processor with 3.30 GHz and 32 GB RAM using 8 cores (1 master). The final simulation times were:

Case	Aggregated (5 load periods)	Sequential (19 load periods)
LP	0.9 sec	1.7 sec
MIP	6.8 sec	573 sec

Note that the MIP solution times tend to grow rapidly with the number of modules with nonconvex PQ-curves and with the number of load periods. This is obvious in the case study results, where an increase of binary variables by a factor of approximately 4 (when going from 5 to 19 load periods) leads to an extreme increase in computation time. To overcome the high computation times, a future solution may be to allow the user to specify which PQ-curves that should be exactly represented in the simulation. We also believe that the use of commercial solvers will lead to much lower computation times for large problems.

In addition to the testing reported here, the new functionality has been tested on a set of representative cases provided by the users.

6 Using the functionality

The functionality is protected by an environmental variable `LTM_PRODRISK_NCPQ` and a corresponding password obtained from SINTEF.

Use the key `PQValg` in ProdBRisk to indicate that a final simulation with nonconvex PQ-curves is desired. Traditionally this key has allowed the values 0 (remove non-convex PQ points in the final simulation) and 1 (use original PQ points in the settlement after each optimization problem has been solved in the final simulation). Key 1 is the default choice, please see the user documentation for more details. We extended the range for this key, so by entering the number 2, one will simulate with explicit representation of the nonconvex PQ points.

As an indication that the functionality has been successfully initiated, the following message will be printed to screen:

```
Avsluttende simulering
MIP solver Cbc+Clp
```

The computation time will normally be significantly higher than the default (LP) final simulation. No extra results are printed, and the result programs can be used as before.

Although the final simulation uses an open source solver, one can still run the new functionality and use CPLEX for the optimization part.



Technology for a better society

www.sintef.no