Modelling European Power Supply and demand with MATLAB and Parallel Computing

Lea Bloechlinger

Bank of America
Merrill Lynch

Power Trading and Analysis

Trading

- Commodities
  - Power (DE, FR, UK, BE, NL,..) Base, Peak, Offpeak
  - Coal, Gas, Emissions
- Futures and Forwards (Financial and Physical), Options
- Intraday, Day-Ahead, Weeks, Months, Quarter, Year

Analysis

- Fundamental approach
- Model the whole power system, estimate the marginal costs, compare with market prices, derive trade ideas

Fundamental Analysis

- What are the costs of meeting all power demand in CWE
  - using the available generation capacity
  - taking into account the different characteristics of units (efficiency, ramping flexibility, etc.)
  - taking into account the expected renewable production
  - assuming optimal border flows within CWE
  - assuming border flows from external countries according to price differentials in the forward curves

→ Stack model which provides an hourly cost curve for the BoY and YA

Compare with market prices and follow day on day changes

Stack Model

1. **Input Data**
2. Optimization Model
3. Results

Generation Units

1700 Units wo wind/solar/hydro

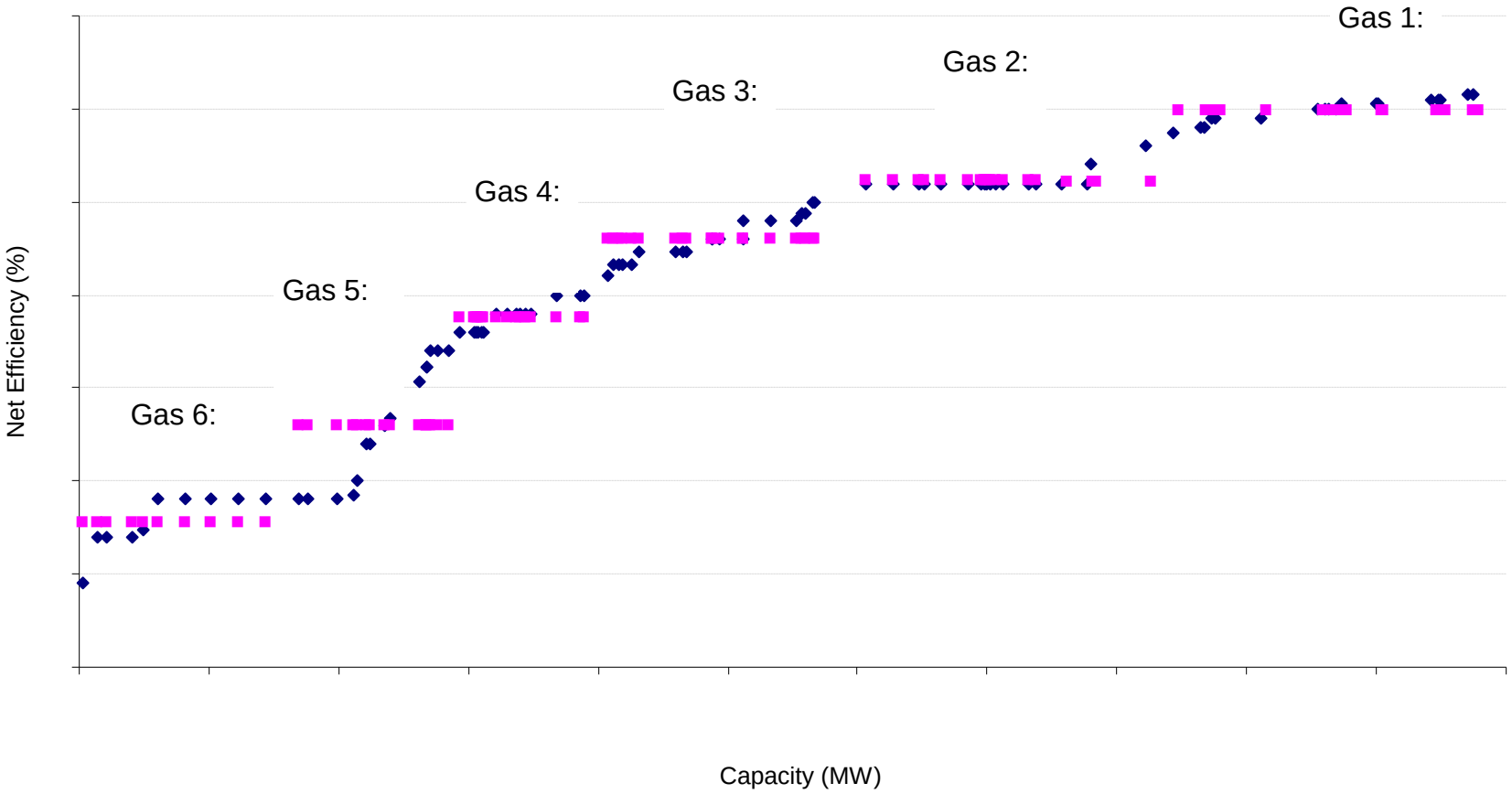
- Characterised by
  - Fuel Type: Nuke, Lignite, Coal, Gas, Fuel Oil, Diesel Oil, Biomass
  - Technology: ST, GT, CCGT, CHP
  - Capacity
  - Efficiency / Age
  - Operating Modus
- Which implies different
  - Variable Costs
  - Startup Costs
  - Ramping Flexibility

Generation Units – Daily Changes/Uncertainty

- Variable costs depend on fuel and emission price which permanently changes

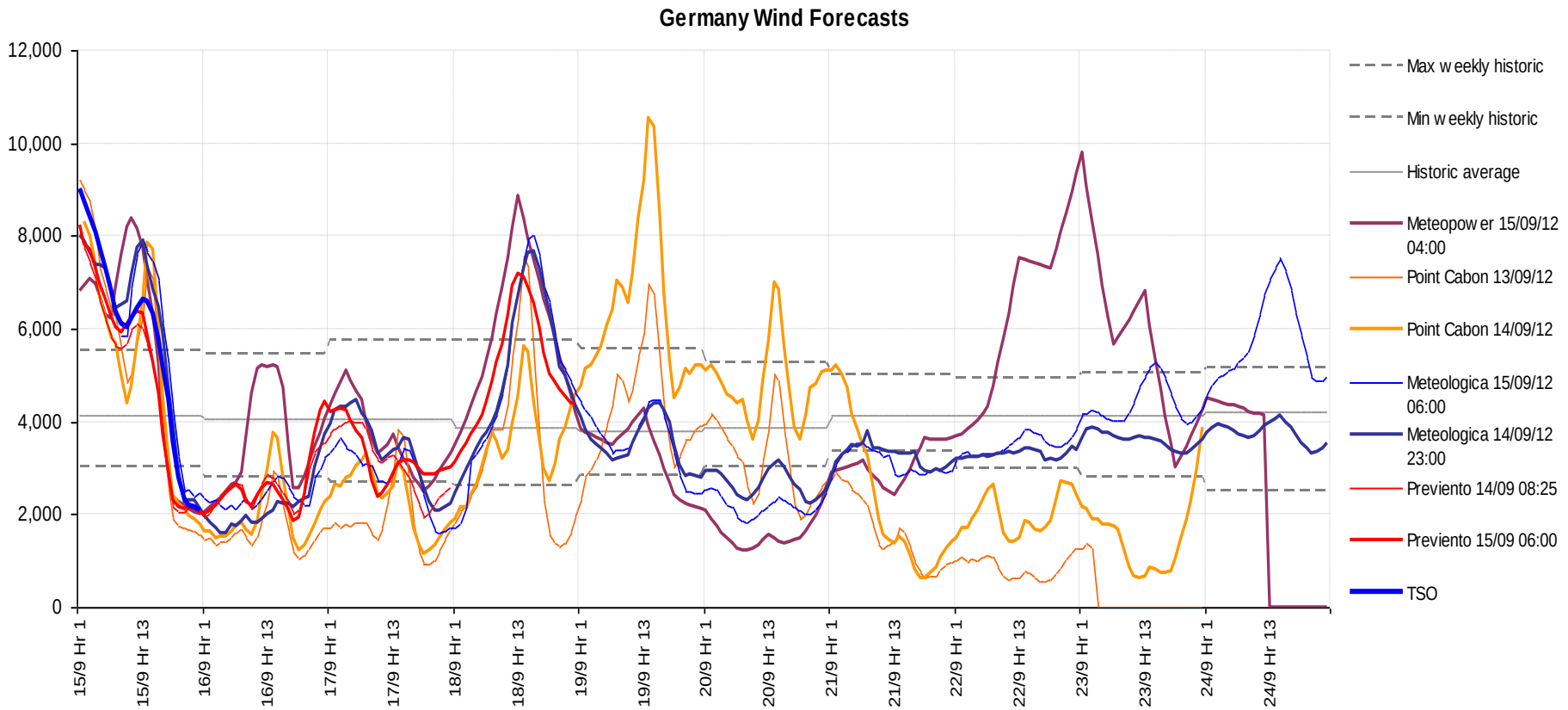
- Available capacity
  - Planned Maintenance
    - Daily unit-specific information provided by some utilities
    - Daily fuel-specific information provided by TSO covering majority of plants
    - Own assumptions, seasonal shapes for not-covered unit and the longer end of the curve
  - Unplanned Outages
    - Realtime UMM (urgent market messages)

Abstraction in Model

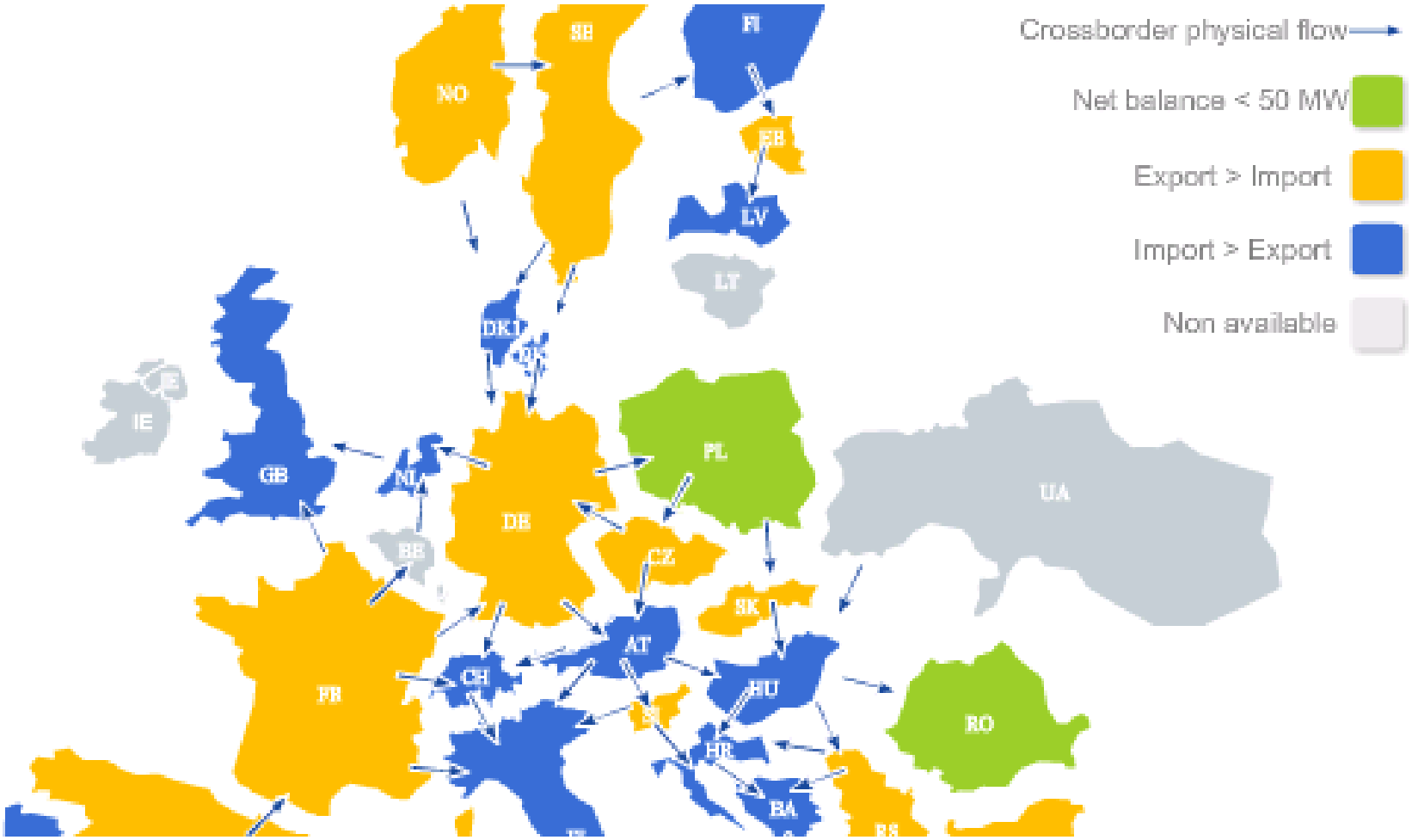


Renewable Generation

- Wind, Solar, Hydro
- Zero Variable Costs



European Power Grid



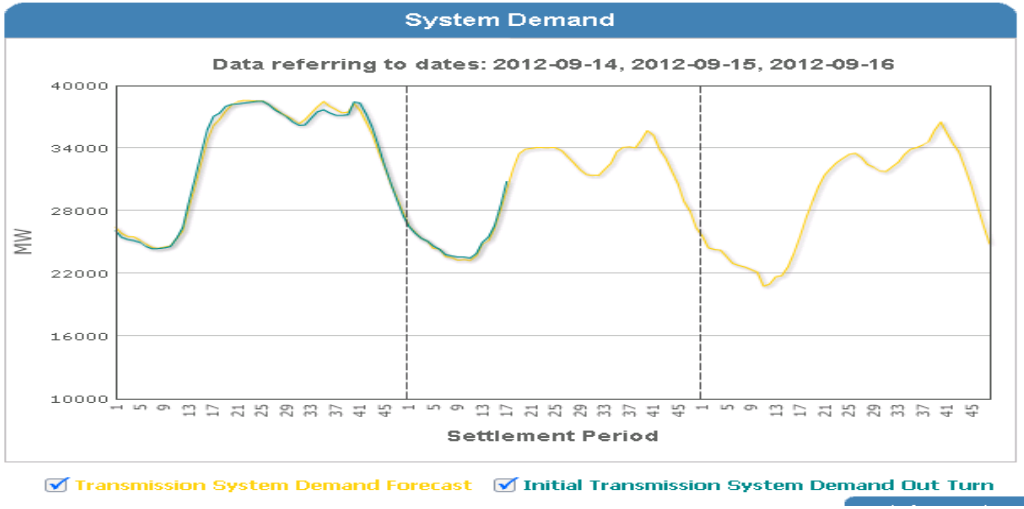
Crossborder Transmission Lines

- Crossborder Transmission Lines
  - 11 external borders

    NO, SE, DK1, DK2, PL, CZ, HU, SI, IT, ES, IE
  - 9 internal borders
- Characterised by
  - Installed Capacity
  - Available Capacity - Daily updates published by TSO
  - Price Differential - Permanently changing

Demand

- Short-/Mid-term Demand
  - Weather data (temperature, cloud cover,..)
  - Calendar (hour, weekday, month)
- Long-term Demand
  - Economy, IP Numbers
  - Efficiency measures



Stack Model

1. Input Data
2. **Optimization Model**
3. Results

Optimization Problem - Linear Program

min f‘x			minimize costs
s.t.	A x	< = b	e.g. Production < Available Capacity
	A_eq x	= b_eq	e.g. Demand = Supply
	x	< = ub	e.g. Available Cap < Installed Cap
	-x	< = lb	e.g. Production > 0

Some numbers for an optimization period of 1 month (DE, FR, NL, BE, UK, CH)

- No decision variables x: 262’080
- No of <= restrictions: 253’440
- No of = restrictions: 48’960
  - A: [253’440x262’080 double]
  - b: [253’440x1 double]
  - f: [262’080x1 double]

A_eq: [48’960x262’080 double]

b_eq: [48’960x1 double]

lb, ub, x: [262’080x1 double]

Implementation Details

- Object-Oriented Approach
  - OptProg
    - Contains all matrices going into the Linear Program
    - Use of dependent properties (easily link parts of matrices)
  - Stack
    - Contains all input and output
    - Intuitively organized

- Advantages of Object-Orient Approach
  - Maintainability
  - Easily extentable

Use of LinProg

- options =

  optimset('LargeScale', 'on','Algorithm', 'interior-point', 'TolFun', 0.00001);
- [x,fval,exitflag,output,lambda] =

```
linprog(        OptProg.d_f.Value,          f
                OptProg.A,                  A
                OptProg.d_b.Value,          b
                OptProg.A_eq                Aeq
                OptProg.d_b_eq.Value,       beq
                OptProg.d_lb.lb,            LB
                OptProg.d_ub.ub,            UB
               [],
               options);
```

Implementation Details

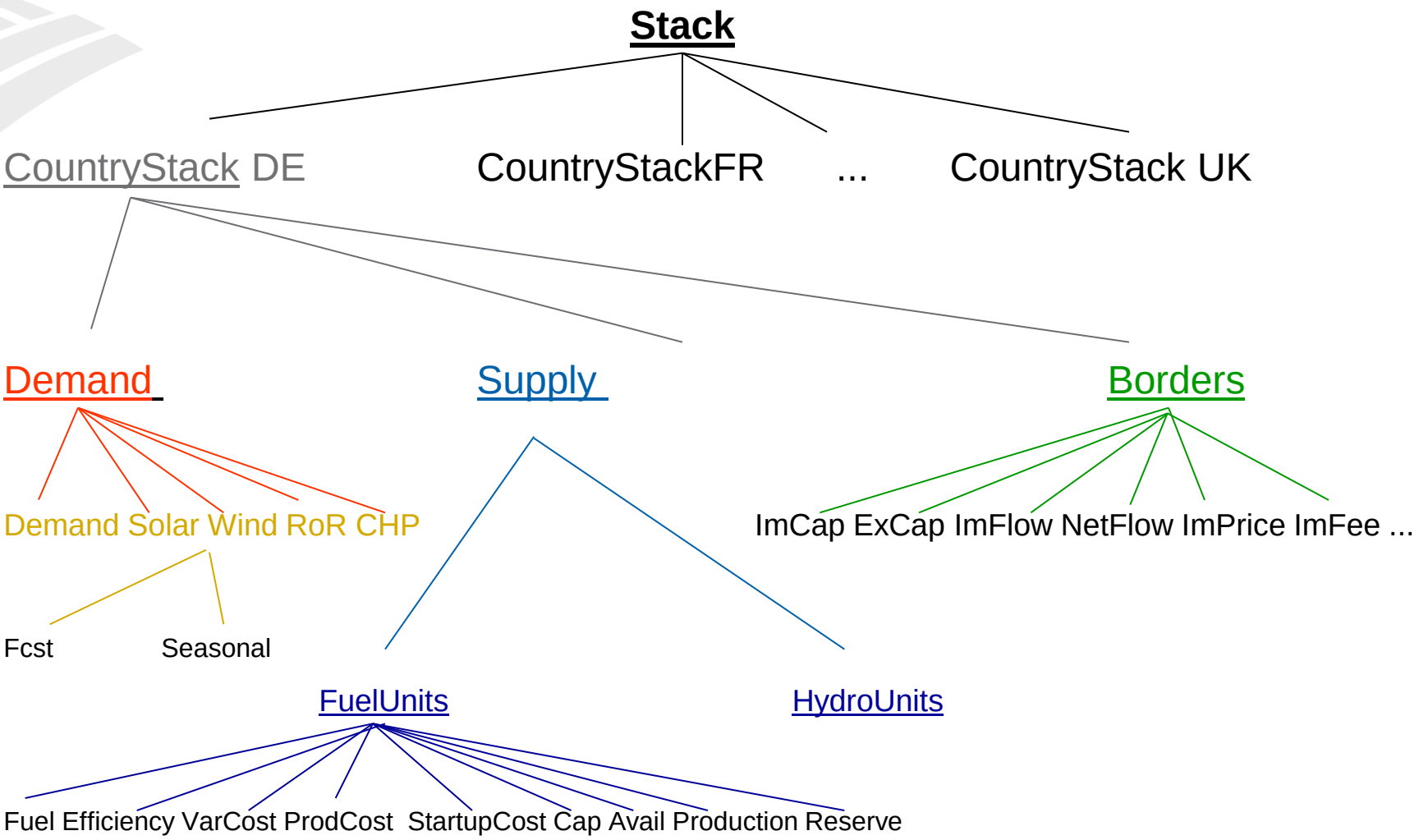
- Parallel Computing
  - Linprog not possible to run in parallel but model can be cut along time-dimension
  - Depending on ratio of time window and number of scenarios parfor loop either over time or scenarios
  - Easy implementation, see next slide

Easy Implementation of Parallel Computing

```
if  nR  < 3                 % if less then three time periods
   for i = 1:nR             % normal loop over time but
        [obj, optp] = runBase(countries, datestr(sdts(i)), datestr(edts(i)),
                              publishDate, scenID, ExtractID);
        obj         = addDummies(obj);
        parfor j = 1:length(scenIds) % but parfor-loop over scenarios
               runDem(obj, optp, scenIds(j), fullDataSet, modelIds(j), series)
        end
    end
 else                       % if more then three time periods
    parfor i=1:nR           % parfor-loop over time periods
        [obj, optp] = runBase(countries, datestr(sdts(i)), datestr(edts(i)),
                              publishDate, scenID, ExtractID);
        obj         = addDummies(obj);
        for j = 1:length(scenIds) % and normal loop over dem scenarios
               runDem(obj, optp, scenIds(j), fullDataSet, modelIds(j), series)
        end
    end
 end
```

Agenda

1. Input Data
2. Optimization Model
3. **Results**


Stack
CountryStack DE
CountryStackFR
...
CountryStack UK
Demand
Supply
Borders
Demand Solar Wind RoR CHP
ImCap ExCap ImFlow NetFlow ImPrice ImFee ...
Fcst
Seasonal
FuelUnits
HydroUnits
Fuel Efficiency VarCost ProdCost StartupCost Cap Avail Production Reserve

Results

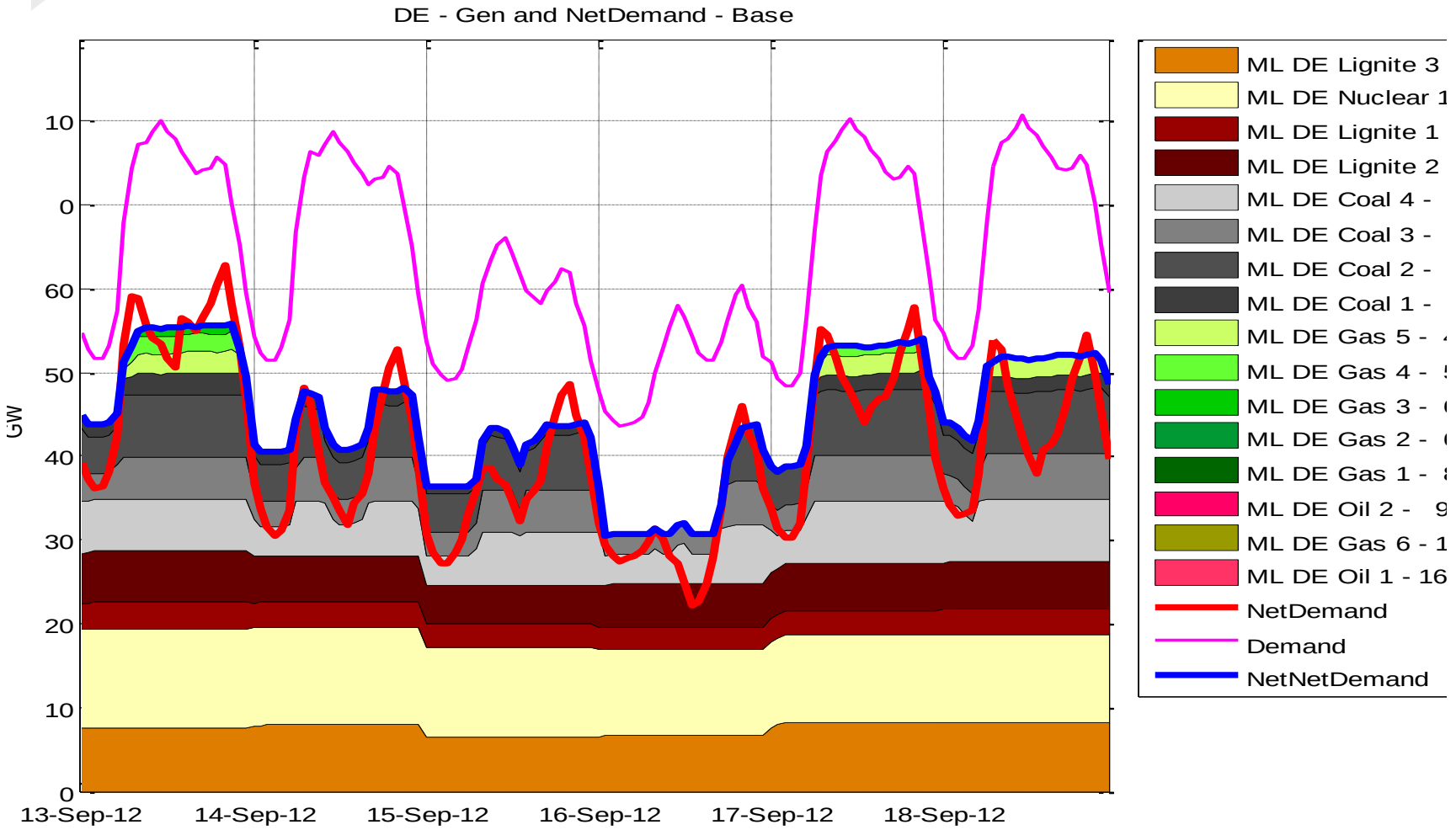
- Due to the object-oriented approach easy but still flexible way to illustrate results also for unexperienced Matlab users

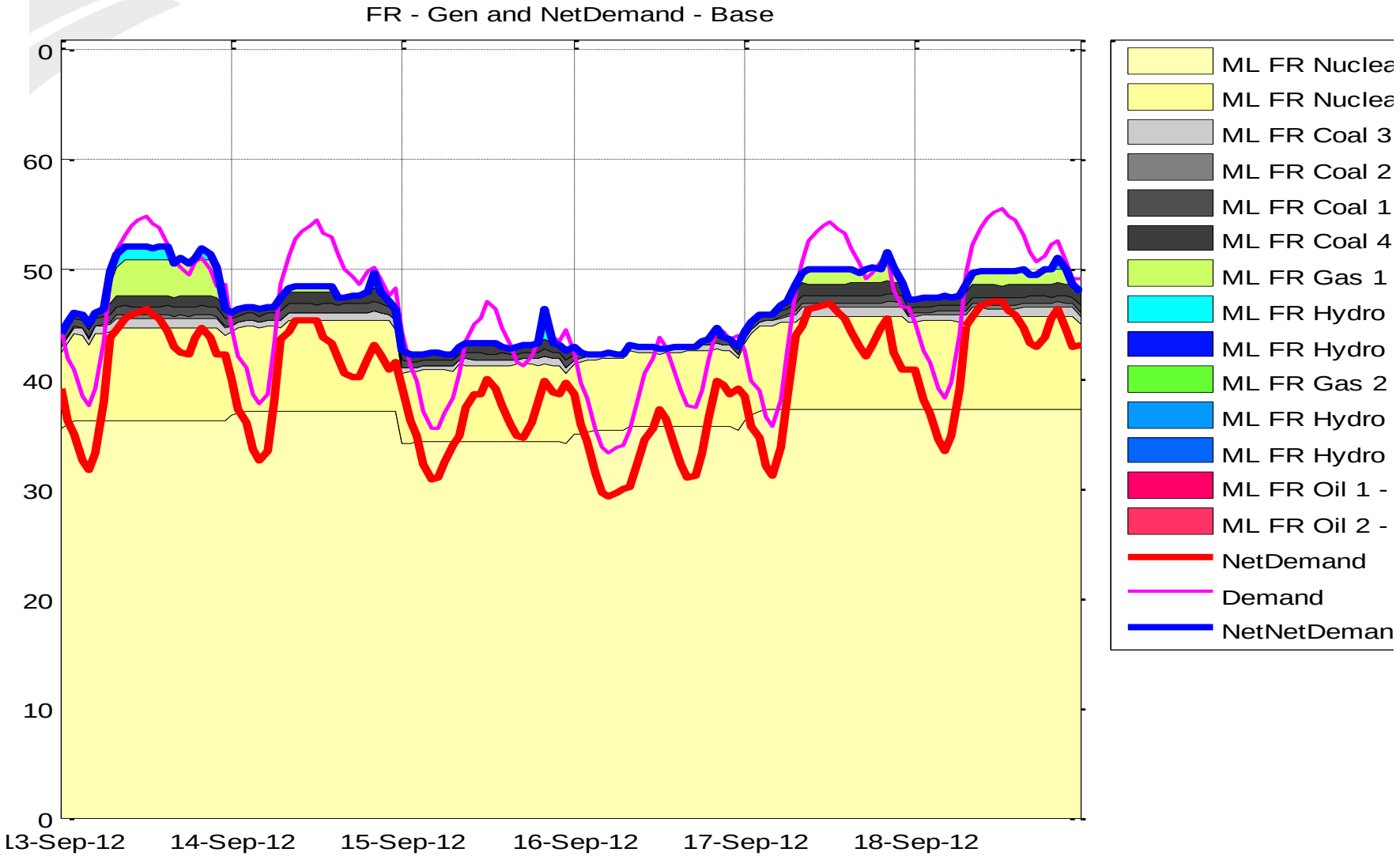
```
>> obj = stacks(countries, startDate, endDate, publishDate, scenarioID)

>> obj = create_opt_stack(obj)

>> help stack/plot_generation

    out = plot_generation(obj, Country, varargin)

    Plots the generation by unit together with the netdemand for the chosen country.
    Returns the plotted data as a dataset.
```

Variable	Description	Default/List of Choice
obj	stack	mandatory
Country	Country	mandatory
frequency	Time Frequency	*Hourly, Daily, Weekly, Monthly
quality	Quality	*Base, Peak, OffPeak, OffPeak5D, WE
gen	Switch Gen/Avail/Cap	*Gen/Avail/Cap
startDate	Start of plot	*obj.StartDate
endDate	End of plot	*obj.EndDate
fuellevel	fuel or unit level	*false/true
addprice	add prices	*false/true

```
    See also plot_demand, plot_stack, plot_border, plot_allborders, plot_price
```

Results


DE - Gen and NetDemand - Base
GW
10
0
60
50
40
30
20
10
0
13-Sep-12
14-Sep-12
15-Sep-12
16-Sep-12
17-Sep-12
18-Sep-12
ML DE Lignite 3
ML DE Nuclear 1
ML DE Lignite 1
ML DE Lignite 2
ML DE Coal 4 -
ML DE Coal 3 -
ML DE Coal 2 -
ML DE Coal 1 -
ML DE Gas 5 -
ML DE Gas 4 -
ML DE Gas 3 -
ML DE Gas 2 -
ML DE Gas 1 -
ML DE Oil 2 -
ML DE Gas 6 - 1
ML DE Oil 1 - 16
NetDemand
Demand
NetNetDemand


FR - Gen and NetDemand - Base
0
60
50
40
30
20
10
0
13-Sep-12
14-Sep-12
15-Sep-12
16-Sep-12
17-Sep-12
18-Sep-12
ML FR Nuclea
ML FR Nuclea
ML FR Coal 3
ML FR Coal 2
ML FR Coal 1
ML FR Coal 4
ML FR Gas 1
ML FR Hydro
ML FR Hydro
ML FR Gas 2
ML FR Hydro
ML FR Hydro
ML FR Oil 1 -
ML FR Oil 2 -
NetDemand
Demand
NetNetDeman

Benefits of Matlab

- Model moved from xls/vba to Matlab/Database
  - Much more flexibility regarding
    - time aggregation
    - geographic coverage
    - scenario analysis
  - Increase in performance

- Much easier handling of large dataset