Reservoir Simulator extensibility as the enabler for solving field management challenges

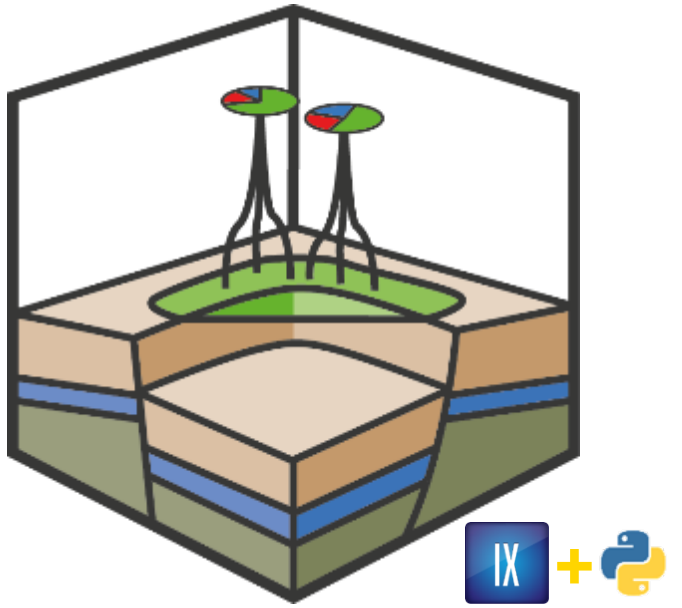
Federica Caresani (speaker), Laura Dovera, Alberto Cominelli – Eni

Daniel Dias, Paolo Delbosco– Schlumberger

SIS Forum 2019, September 17th 2019

Outline

- Introduction
- INTERSECT Field Management
- Field Management extensibility points
- Eni applications:
  - Field A – dual production target
  - Reservoir X – CO2 management in ultra-sour reservoir



Introduction

- Numerical simulation of the reservoir fluid dynamics allows evaluating the risks and optimizing the reservoirs development and the management.

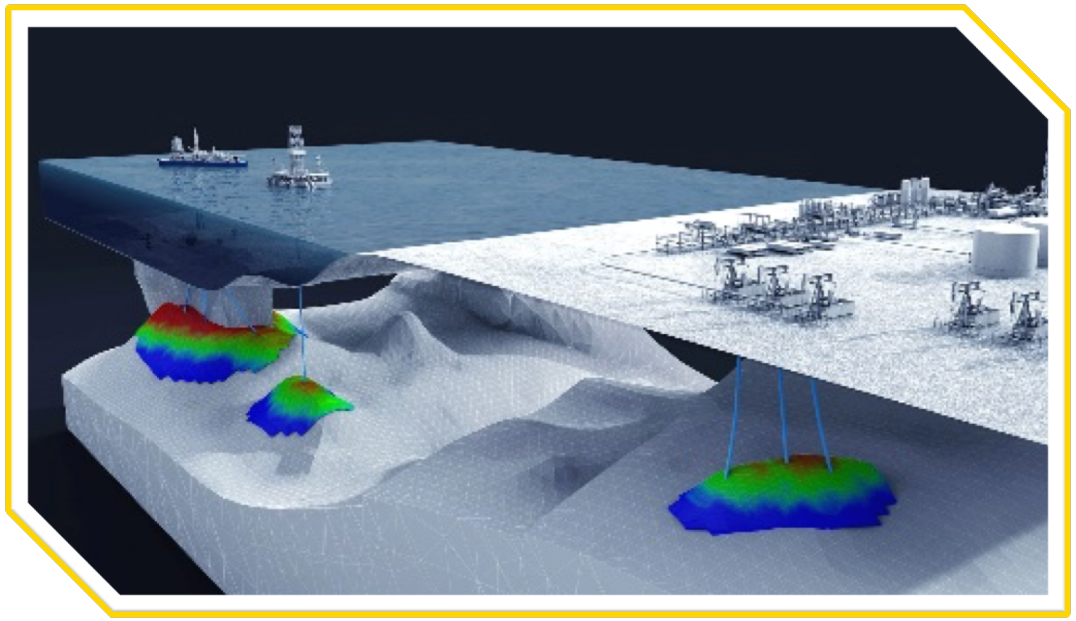
- Commercial simulators, like INTERSECT, are suitable for evaluating the reservoirs performance in many operational scenarios. However, reservoir engineers may face particularly challenging tasks that are not directly available in the simulation package.

- It is sometimes necessary to implement original and specific solutions in the simulator, leveraging on the flexibility and the openness of the modelling tool.

INTERSECT Field Management (1/2)

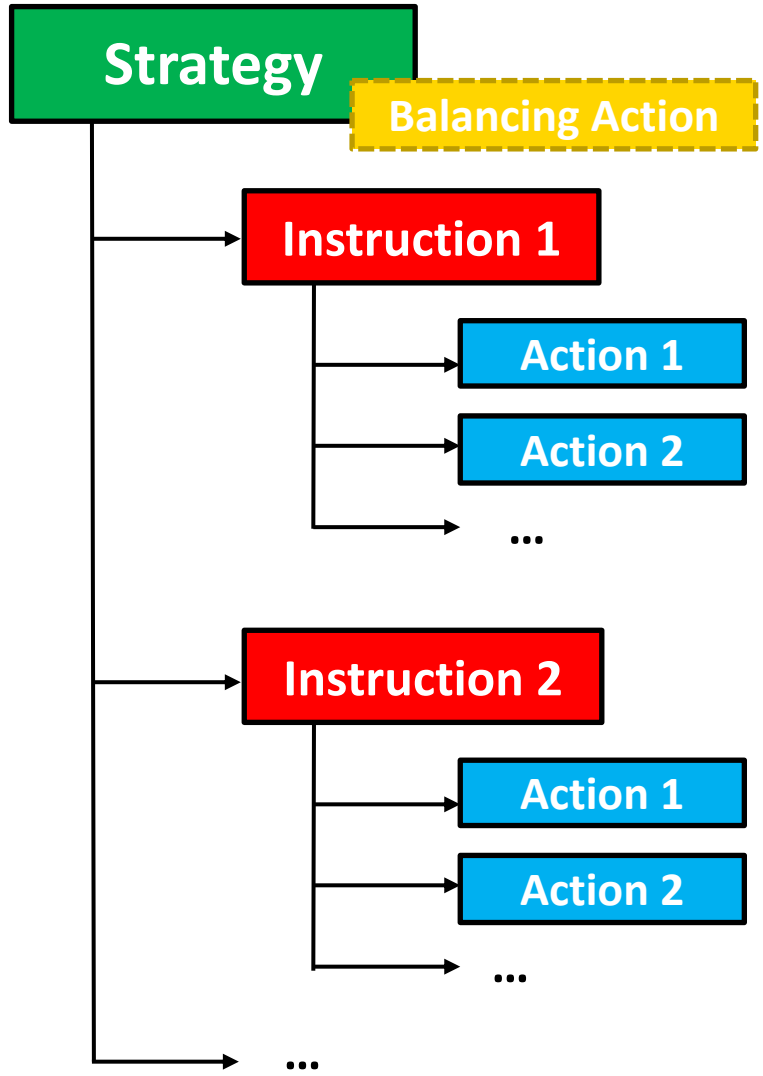
INTERSECT Field Management

- Focuses the scheduling and control of reservoir production operations to evaluate different development strategies and to maximize the production.
- Provides a framework to model all the operational constraints and complex operating logic required to manage the asset.
- Achieves the production targets while honoring all the physical flow and pressure constraints imposed throughout the production system.



INTERSECT Field Management (2/2)

- **Strategy**
  - Set of conditional control logic instructions used to express the field development plan
- **Instructions**
  - Directive to perform a series of actions, either unconditionally or when a criterion is met
- **Expressions**
  - Predefined standard quantities used in instructions
- **Actions**
  - Basic set of operations from which a Field Management strategy is built
- **Balancing Actions**
  - Allocation of individual well rates in a hierarchy of wells and groups such that all user-supplied group constraints are honored



Comparing ECLIPSE and INTERSECT

ECLIPSE

- ACTIONX

Set of SCHEDULE section keywords stored for later processing when a set of conditions are met

- UDQs

User-defined summary quantities

- UDAs

User-defined arguments for keywords of the SCHEDULE section

INTERSECT

- Field Management
  - Entail hierarchy of wells, groups and platforms
  - Formulate field development rules (Strategy)
  - Flexible and extensible
- Flexible and powerful language
  - Simulation data are defined using Nodes, Fields and Commands.
  - Python can be encapsulated to manipulate objects and trigger actions/strategies

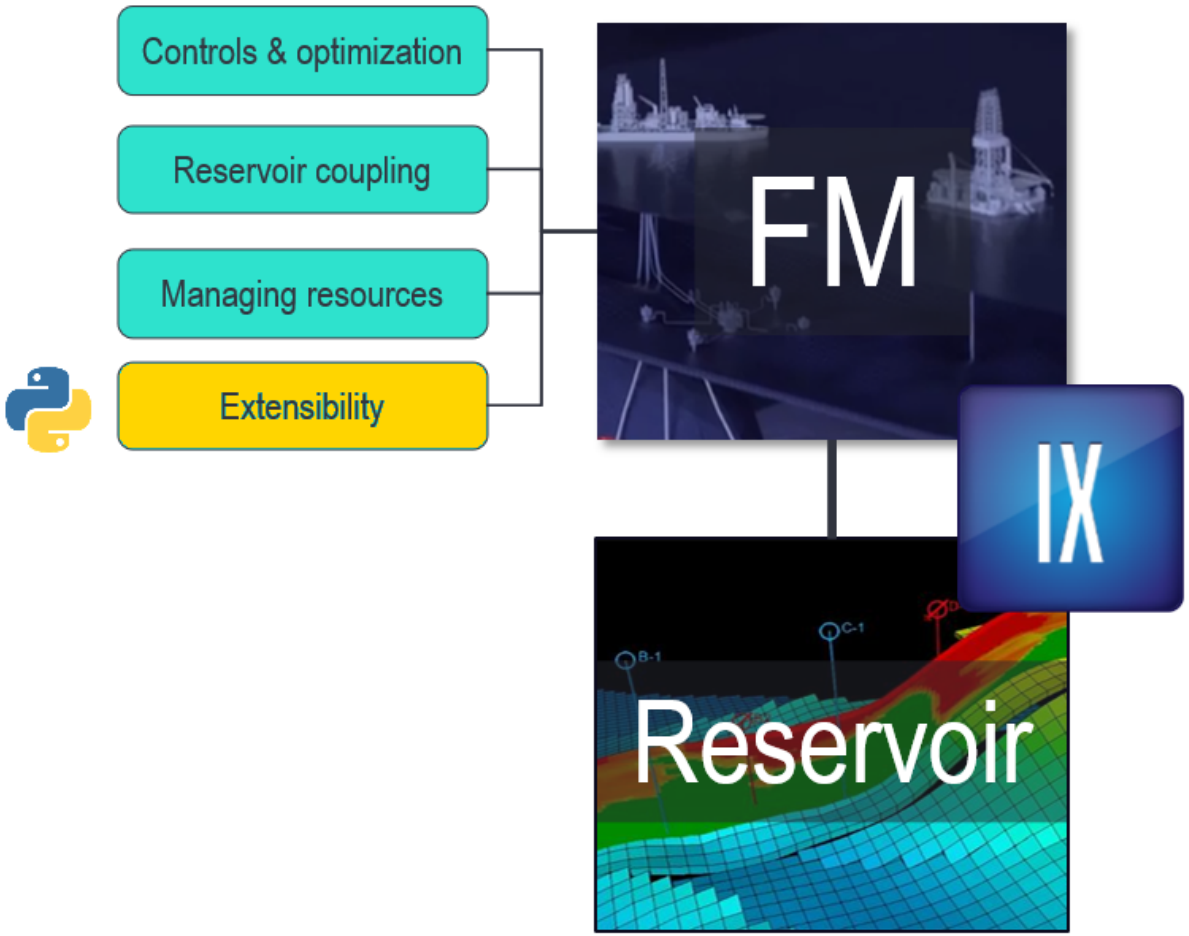
Extensibility of INTERSECT Field Management

Leverage on the INTERSECT simulator flexibility:

- Production and injection constraints can be added to the development strategy
- Simulator Python scripting capabilities can be used to integrate constraint definition

INTERSECT simulator has the capability to incorporate Python scripting, a component of the modelling workflow which enables:

- Expanding the field management capabilities
- Integrating custom solutions to specific engineering tasks not directly available in the simulator



INTERSECT Python scripting

Field management extensibility points:

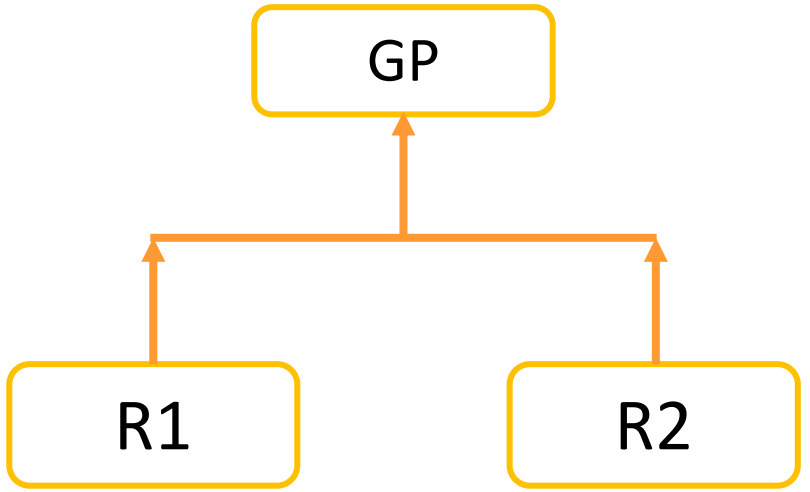
- **CustomScript**
  - Scripts that **may be executed** by the user at prescribed points of the simulation
- **CustomControl**
  - Scripts that are **automatically executed** at particular algorithmic positions throughout the simulation
- **CustomAction**
  - Scripts that are **dynamically executed** by Field Management Instructions along with the other list of actions
- **CustomVariable**
  - Scripts that **may be embedded** in Field Management expressions

INTERSECT Python scripting – an example

```
CustomAction "Cutback" {
ScriptRequiresEntity = TRUE
OneTime=TRUE
ArgNames = ["Multiplier"]
ArgTypes = ["double"]
Script = @{
New_OPR = $arg1*entity.get_property(OIL_PRODUCTION_RATE).value
entity.set_constraint((New_OPR, OIL_PRODUCTION_RATE))
print('cut')
}@
}

CustomControl "test"{
ExecutionPosition=ALL_POSITIONS
Script = @{
print('PROD1',Well('PROD1').get_constraint(OIL_PRODUCTION_RATE))
}@
}

DATE  "22-Oct-1982"
DynamicList "High_GOR" {
   InitialList= Group('FIELD')
   EntityLevel = WELL
   SelectionCriteria = ["GAS_OIL_RATIO > 1.2"]
}
CustomAction "Cutback" {
   EntityList = DynamicList('High_GOR')
   Args = ["0.5"]
}
Instruction "Cutback_high_GOR" {
   Actions = [ CustomAction('Cutback') ]
}
Strategy "Strategy" {
   Instructions = [ Instruction("Cutback_high_GOR") ]
}
```

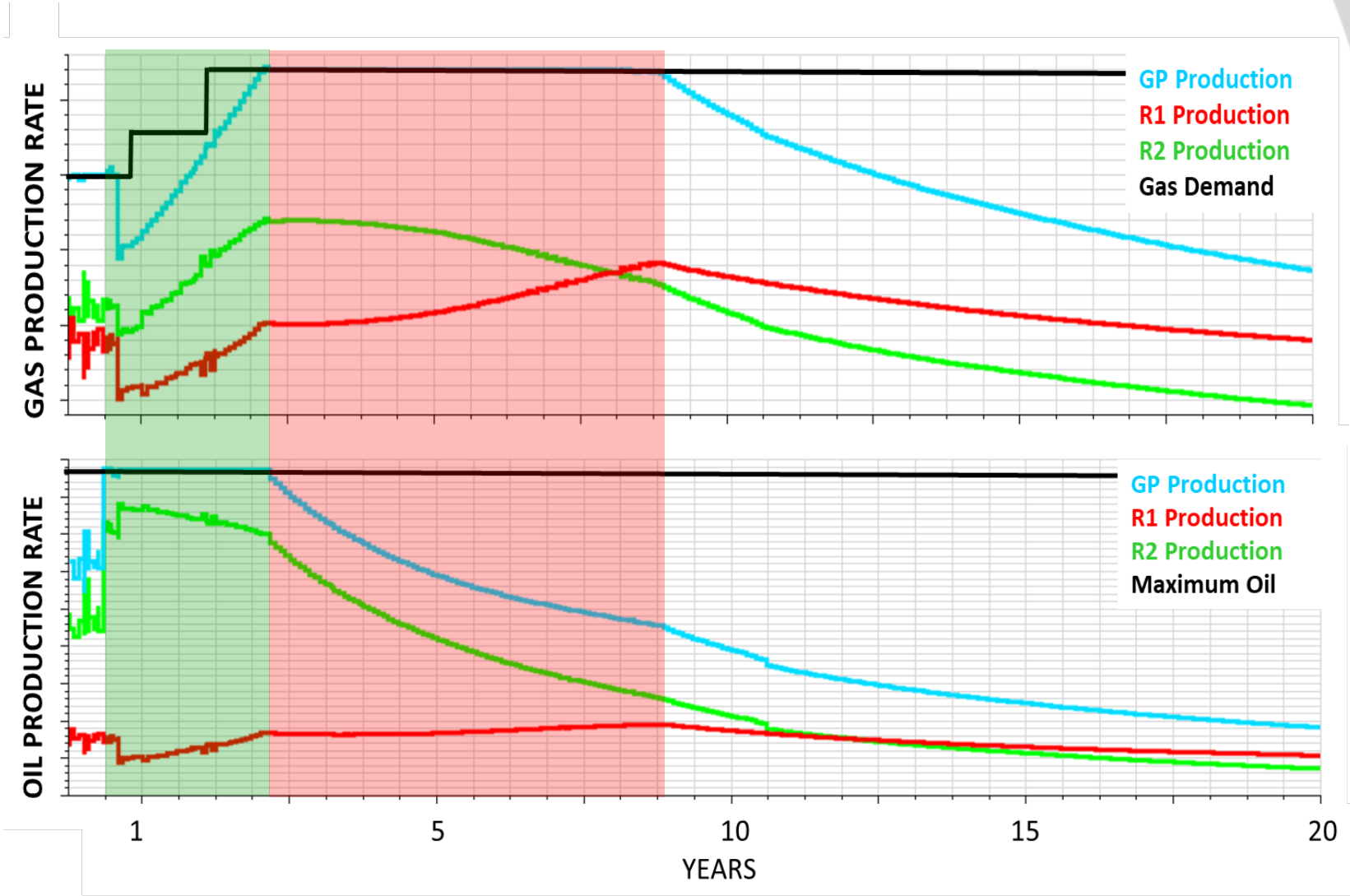
Field A case study: introduction

- Reservoir1 (R1) and Reservoir2 (R2) are a gas condensate and an oil field located off-shore.
- R1 and R2 are linked to a gas plant (GP) which collects production from all gas wells of R1 and from oil wells of R2.
- Gas delivered to GP plant is used to satisfy Take or Pay demand but oil treatment is constrained to a maximum liquid capacity.



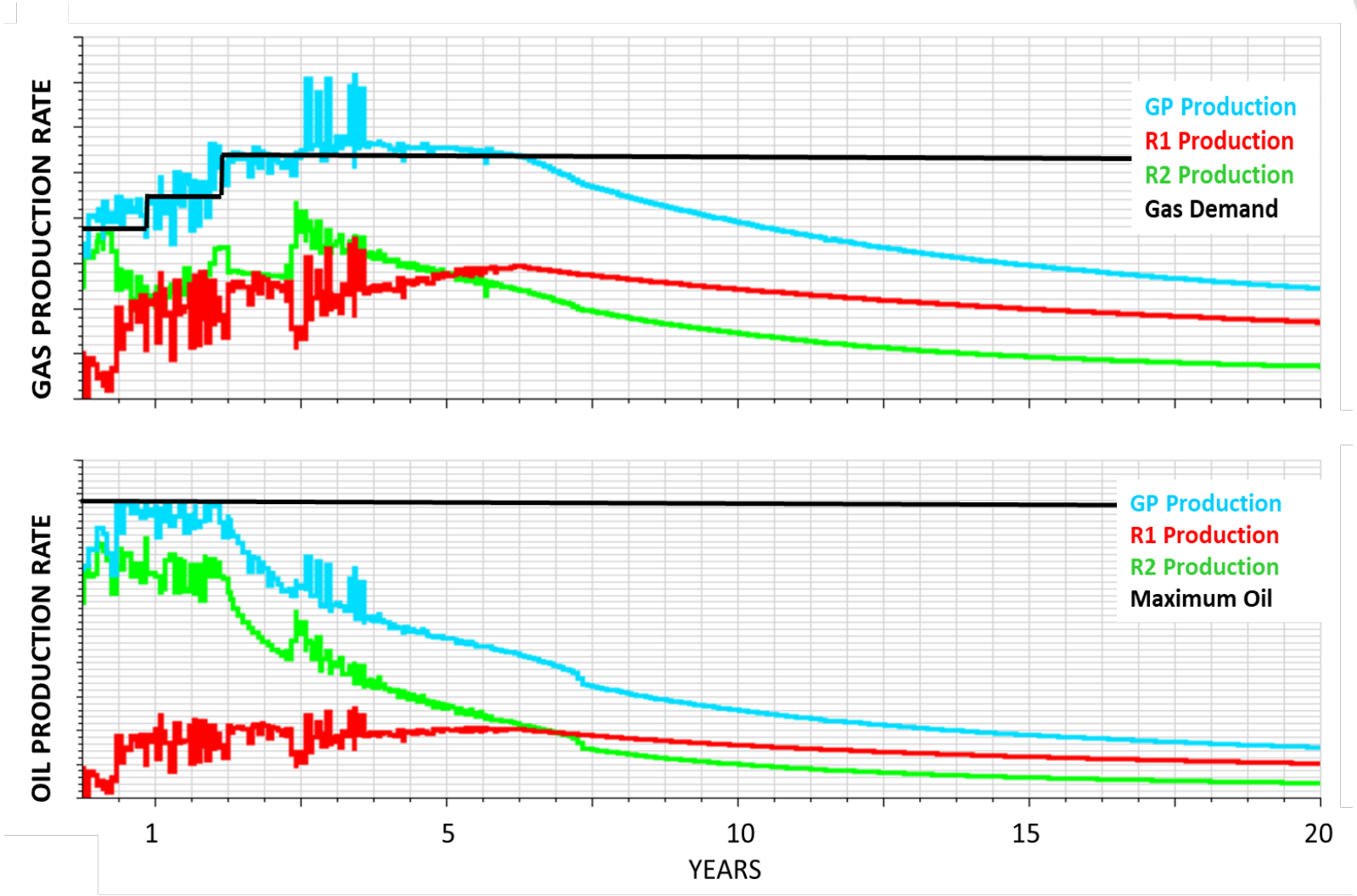
Field A case study: base case

- To model the network interaction, an INTERSECT reservoir coupling model is used with two group targets assigned to GP plant:
  - Oil production under maximum capacity constraint
  - Gas demand requirement.
- The standard INTERSECT allocation algorithm based on potential can impose only one target.



Field A case study: tentative solution with linear optimizer

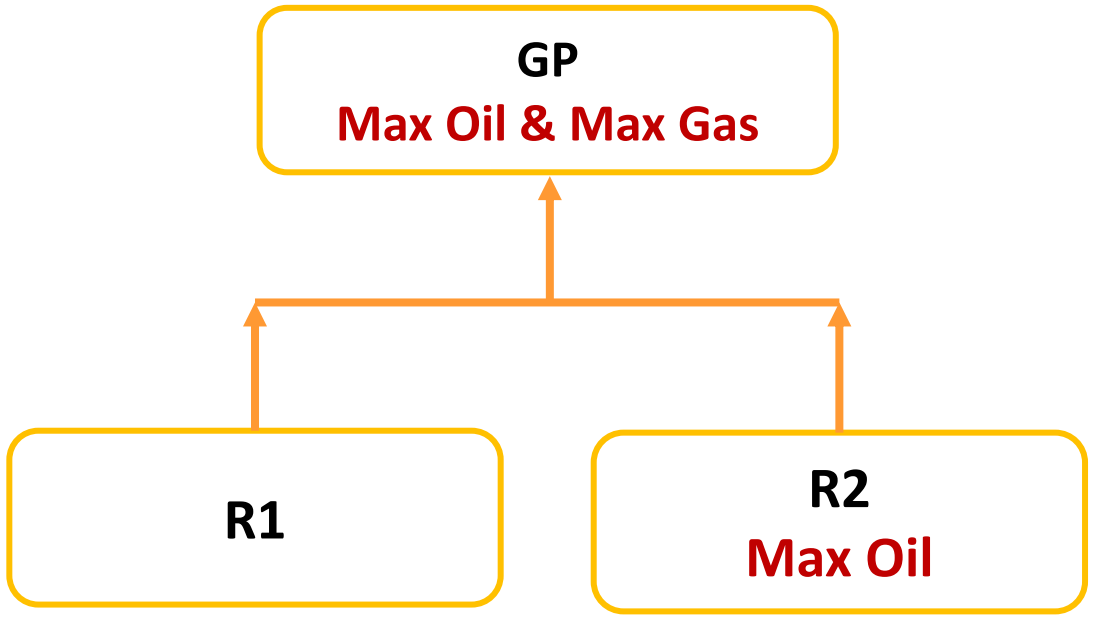
- Formally, the issue of a dual production constraint is a double objective optimization problem.
- INTERSECT FM implements a local linear optimizer method but, since it is an optimizer, the solution is unstable
- There is an effective potentiality to achieve both targets.



Field A case study: *ad-hoc* allocation rule

- In order to impose an *ad-hoc* allocation rule capable of maintaining the dual production target, a Python script has been implemented.

- The main idea is to define the child group R2 as independent entity of the father group GP with its own oil target.



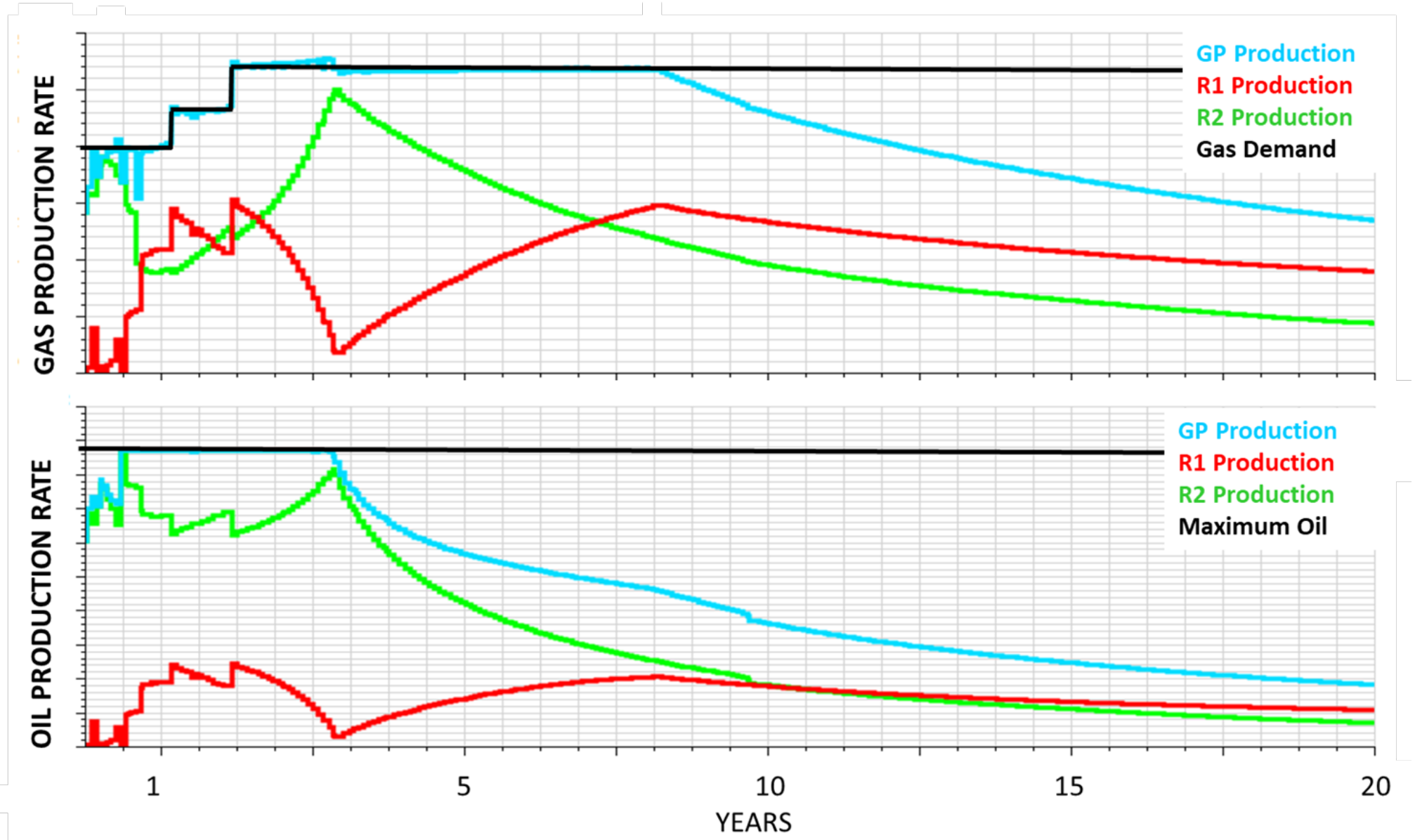
Field A case study: custom implementation

- The oil target of R2 is set by the Python script based on a specific allocation rule: reduce/increase R2 oil target to compensate for both oil and gas mismatch at gas plant GP level

```
# Expression CONDITIONAL_OIL_TARGET defines the maximum oil for R2

Expression "CONDITIONAL OIL TARGET" {
    Definition =  ZZZ
}

Expression "MAX_R2_OPR_HGC" {
    Definition="OIL_PRODUCTION_RATE(Group('R2')) <= Expression('CONDITIONAL OIL TARGET')"
}

# CustomAction SET_CONDITIONAL_OIL_TARGET_ACTION modifies the CONDITIONAL_OIL_TARGET value accoridng to specific allocation rule

CustomAction "SET_CONDITIONAL_OIL_TARGET_ACTION" {

ScriptIsTopologyModifying="True"
Script=@{

GP_GAS_TARGET = XXX
GP_OIL_TARGET = YYY

GP_GAS_PRODUCTION = Group('GP').get_property(GAS_PRODUCTION_RATE).value
GP_OIL_PRODUCTION = Group('GP').get_property(OIL_PRODUCTION_RATE).value

if ( GP_GAS_PRODUCTION < GP_GAS_TARGET ):
    if ( GP_OIL_PRODUCTION >= GP_OIL_TARGET ):
        R2_OIL_TARGET = GP_OIL_TARGET - (GP_GAS_TARGET - GP_GAS_PRODUCTION) * GP_CGR
    else:
        R2_OIL_TARGET = GP_OIL_TARGET + ( GP_OIL_TARGET - GP_OIL_PRODUCTION )
else:
     if ( GP_OIL_PRODUCTION >= GP_OIL_TARGET ):
        R2_OIL_TARGET = GP_OIL_TARGET
    else:
        R2_OIL_TARGET = GP_OIL_TARGET + ( GP_OIL_TARGET - GP_OIL_PRODUCTION )

Expression('CONDITIONAL_OIL_TARGET').Definition.set('R2_OIL_TARGET')

}@
}
```

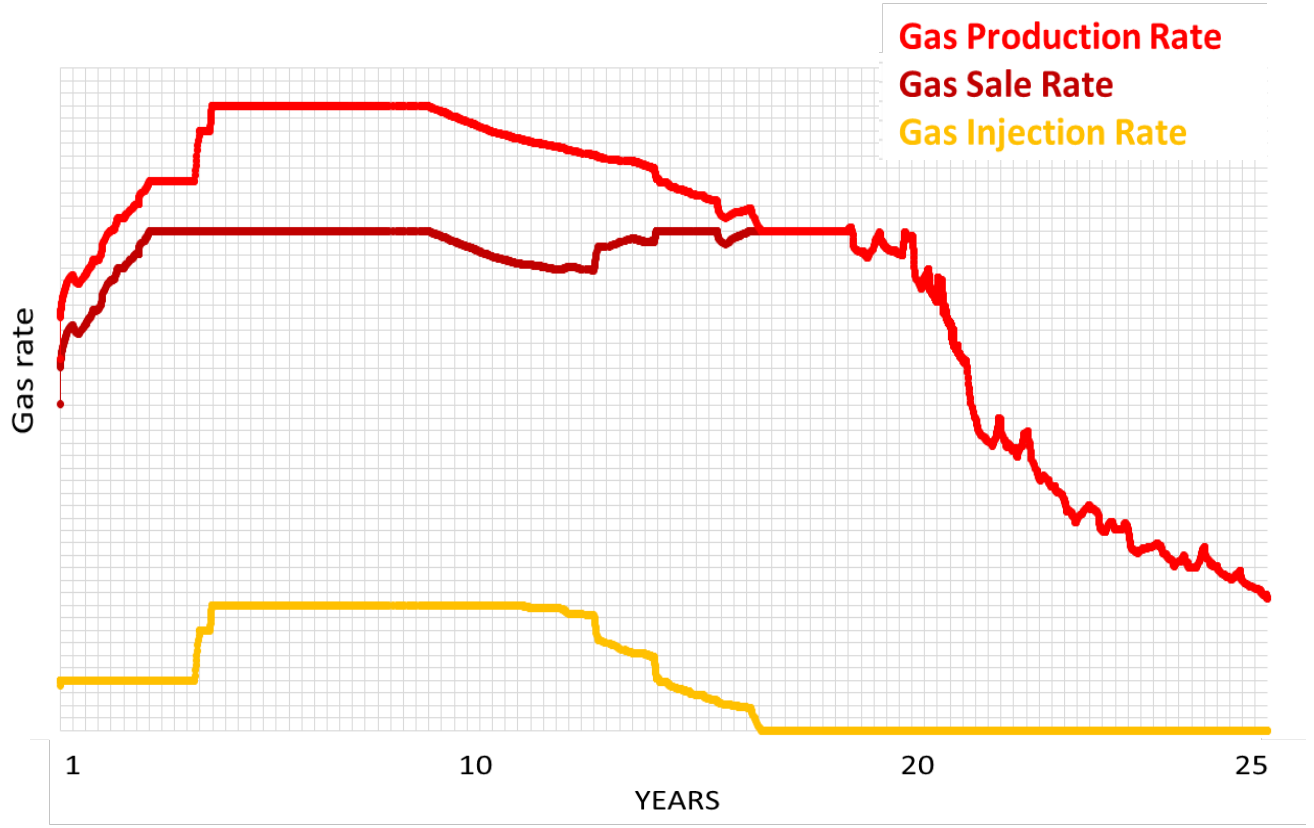
R2 Oil target		GP gas target: Met	GP gas target: Not met
GP oil target	Met	Maintain	Choke
GP oil target	Not met	Ramp up	Ramp up

Field A case study: results



Reservoir X case study: introduction

- Reservoir X is an oil and ultra-sour gas field off-shore
- Compositional model
- Current development strategy:
  - Oil production target
  - Maximum gas production rate
  - Gas sale target
  - Water injection and gas injection for pressure maintenance
    - Maximum water injection
    - Maximum gas injection



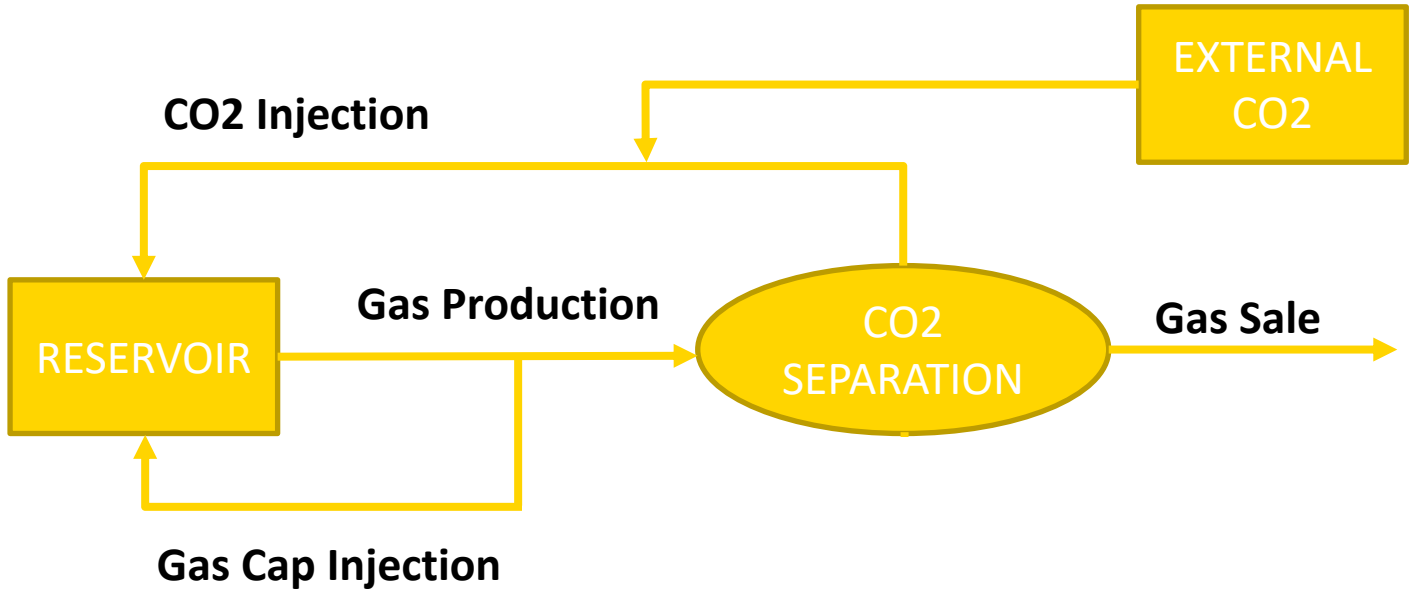
Reservoir X case study: CO2 injection scenario

- The gas in Reservoir X is rich of non hydrocarbon components: N2, CO2, H2S.

- To sustain reservoir pressure, a CO2 injection scenario was proposed
  - A stream of CO2 coming from the surrounding fields of the area to be directly injected into the oil rim.
  - CO2 injection system capacity is a new constraint.

- The injection of CO2 in Reservoir X will determine an increase in CO2 fraction in the produced gas
  - The excess CO2 must be separated from the stream to respect sale gas specifics
  - CO2 injection rate will increase in time
  - We assume we can increase plant gas production capacity to handle CO2 volumes

Reservoir X case study: CO2 management



- We want to understand how much CO2 can be injected (how long) before CO2 cannot be handle any longer.
  - A custom script is necessary to setup a production and CO2 injection strategy which allows to maximize gas sale plateau

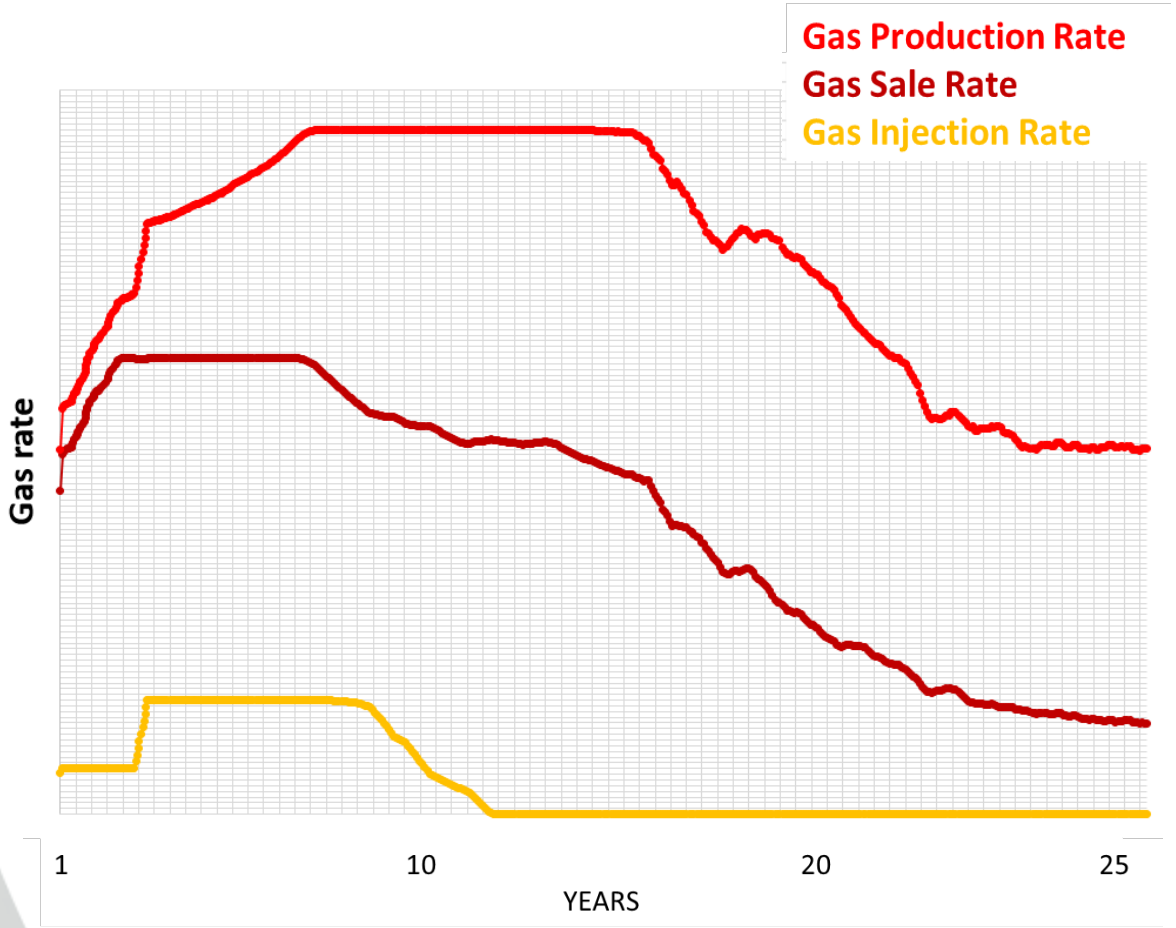
Reservoir X case study: custom implementation

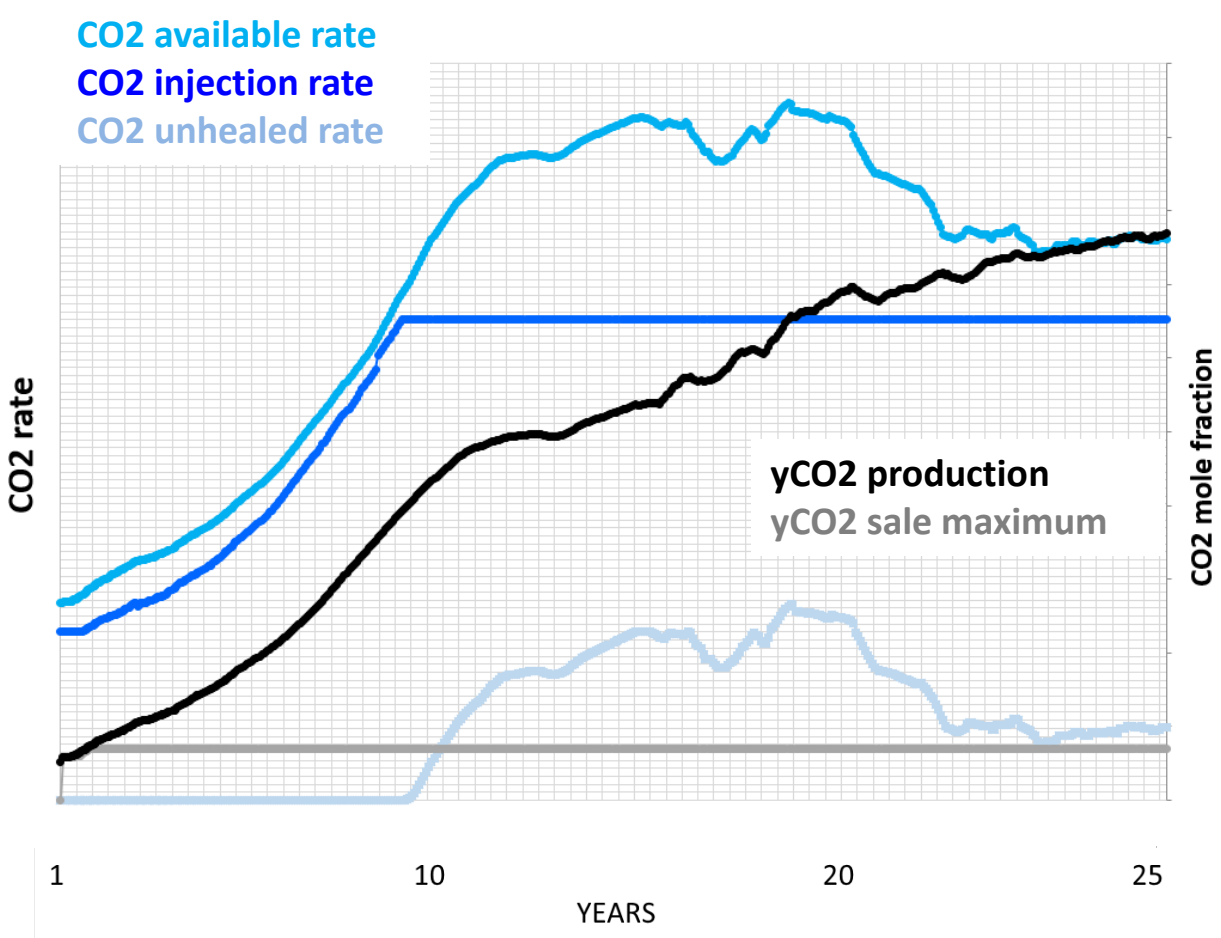
- Custom solution:
  1. Get gas production rate
  2. Get moles and composition of gas production stream
  3. Get moles available for sale: remove from produced moles the amount of gas cap for reinjection
  4. Evaluate excess CO2 moles

$$N_{CO2,EXCESS} = \frac{N_{CO2} - N_{tot} * y_{CO2,LIMIT}}{1 - y_{CO2,LIMIT}}$$

  5. Evaluate moles and rates available for sale: remove excess CO2
  6. Compute moles and rate for CO2 injection
  7. Set new gas production target and CO2 injection target

Reservoir X case study: results





Conclusions

- INTERSECT capability to incorporate Python scripting is a powerful modelling component
- It allows leveraging on INTERSECT Field Management flexibility and extensibility to develop solutions to non-standard reservoir engineering challenges
- Two Eni real-field applications were presented, where the use of custom scripts allowed to disclose the value of complex development scenarios
- Other applications were implemented in Eni with successful and valuable results
  - Gas balance
  - Flexible Integrated Asset Model for deep water development (Selvaggio P. *et al*., SPE 192603-MS)
- Way forward: develop an internal library of solutions to a large spectrum of reservoir engineering challenges to be included in Eni's reservoir studies