Production optimization

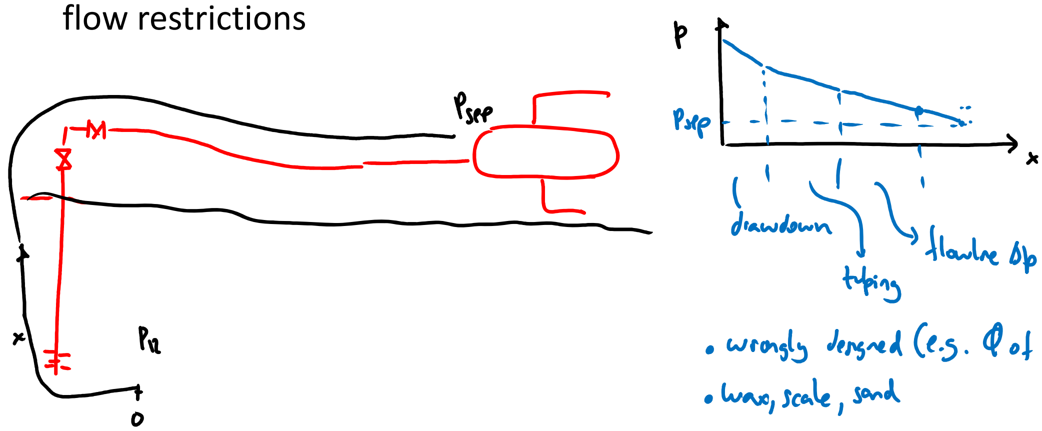
Prof. Milan Stanko (NTNU)

*Chapter 3 of compendium

1

Production optimization – what is it?

- Detect locations in the system with abnormally high-pressure loss and flow restrictions



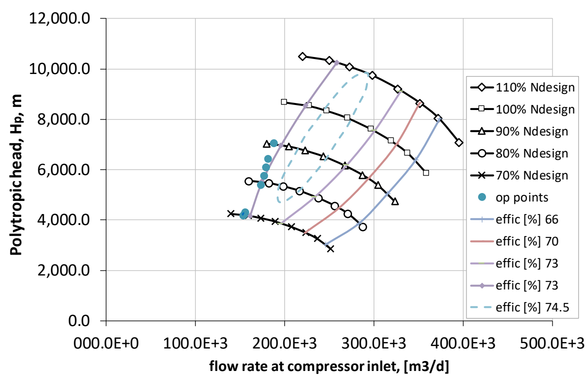
- wrongly designed (e.g. Φ of tubing, pipe)
- wax, scale, sand

Result: increase in production

2

Production optimization – what is it?

- Verification of equipment design conditions vs actual operating conditions



subsea compressor

3

Production optimization – what is it?

- Identification and addressing fluid sources that have "disadvantageous" characteristics (e.g. high water cut, high H_2S content)
- Identify and correct system malfunctions and unintended behavior
- Analyze and improve the logistics and planning of maintenance, replacement and installation of equipment or in the execution of field activities.
- Review the occurrence of failures and recognize patterns (data analytics?)

4

Production optimization – what is it?

- Calibration of instrumentation
- Identification of operational constraints (e.g. water handling capacity, power capacity)
- Observe and analyze the response of the system when changes are introduced
- Find control settings of equipment (or system characteristics) that give a production higher than current (or, preferably, that give maximum production possible)
- Find control settings of equipment (or system characteristics) that maximize an objective KPI
- Identify bottlenecks
- Identifying and monitoring Key Performance Indicators (KPIs)

5

Production optimization – what is it?

- Calibration of instrumentation
- Identification of operational constraints (e.g. water handling capacity, power capacity)
- Observe and analyze the response of the system when changes are introduced
- **Find control settings of equipment (or system characteristics) that give a production higher than current (or, preferably, that give maximum production possible)**
- **Find control settings of equipment (or system characteristics) that maximize an objective KPI**
- Identify bottlenecks
- Identifying and monitoring Key Performance Indicators (KPIs)

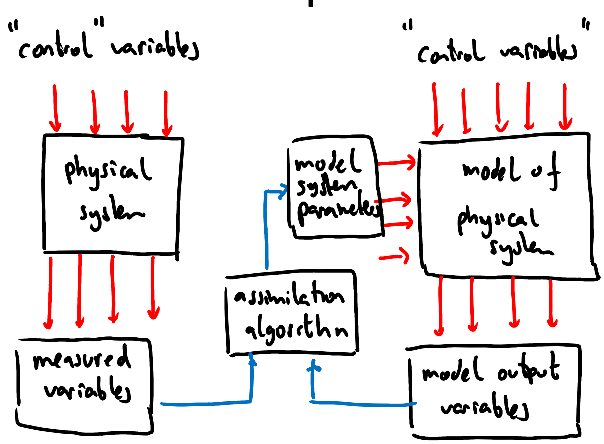
6

Time scales of production optimization

Long term	Short term	Shorter term
• Years, months	• Daily, weekly	• Seconds, minutes, hours

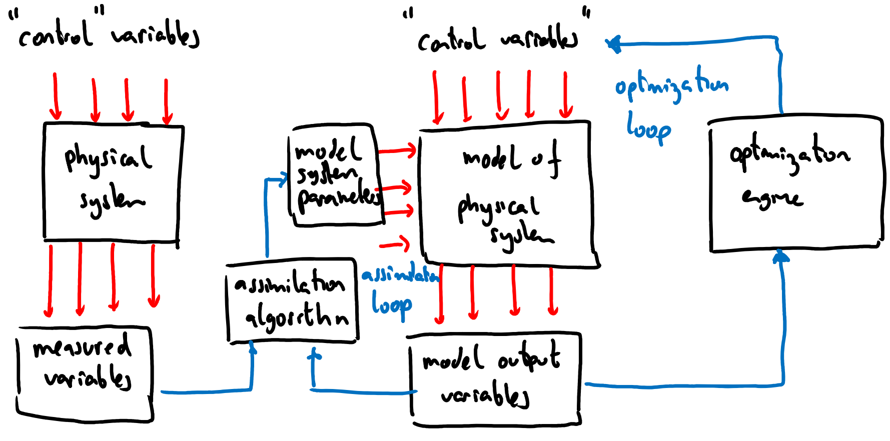
7

Model-based production optimization: fidelity



8

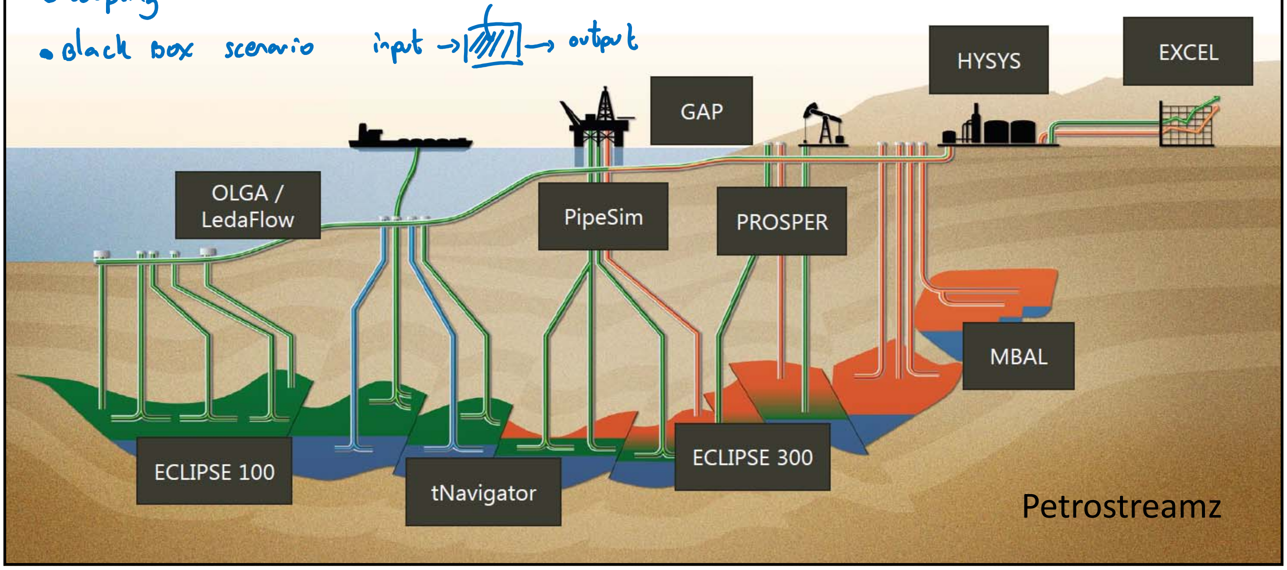
Model-based production optimization: optimization



9

Integrated asset modeling (IAM)

- consistency !!
- coupling
- black box scenario input → output



10

Time scales of production optimization and models

Long term	Short term	Shorter term
• Years, months Models are highly uncertain (limited data) Models are typically transient (reservoir model) but in IAM also steady-state models are included	• Daily, weekly There is data to tune models Models are typically steady state (network, well, processing plant)	• Seconds, minutes, hours • Can we use steady state models? Or do we need transient models? • Why to use models? We can develop optimization strategies on the actual system

11

Time scales of production optimization and examples

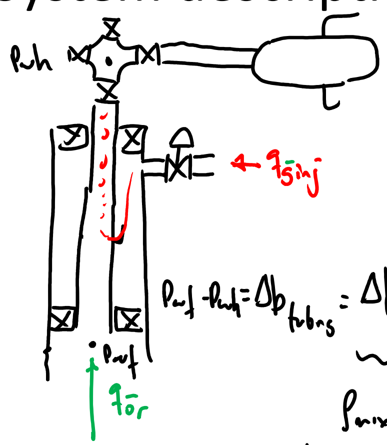
Long term	Short term	Shorter term
• Maximize recovery factor and NPV, reduce water cut and GOR • Control variables: well placement, well rates, well "status", well routing, "presence" of equipment (processing, ICD)	• Maximize oil production, condensate production, gas production, revenue • How to allocate a scarce resource (gas injection, power) • Variables: choke opening, gas lift rates, pump frequency, well routing	• Maximize production, revenue • Reduce and mitigate fluctuations • Variables: ~~choke~~ control valve opening, gas lift rates, pump frequency,

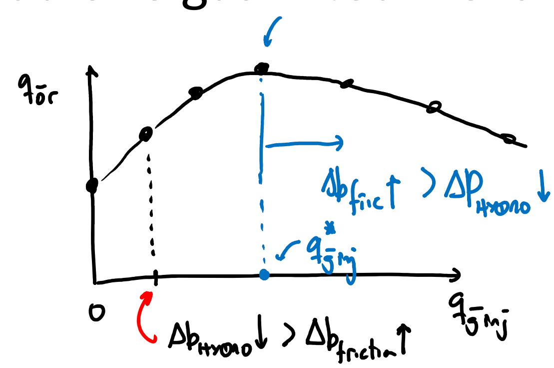
12

Examples

1. (Short term) Two standalone gas-lifted wells

System description



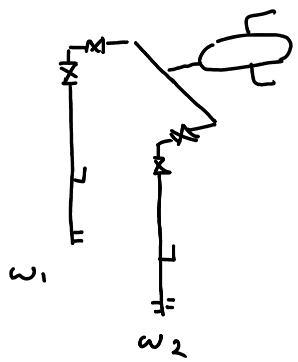


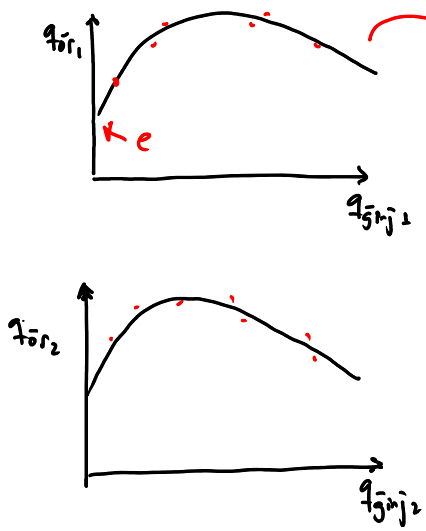
$$P_{wf} - P_{wh} = \Delta p_{tubing} = \underbrace{\Delta p_{hydro}}_{\rho_{mix} \cdot g} + \Delta p_{fric}$$

$$v^2 \quad \left(\frac{q_o + q_g}{A_p}\right)^2 \rightsquigarrow \Delta p_{fric} \uparrow$$

when q_{ginj} increases $\rightsquigarrow \rho_{mix} \rightarrow \rho_{gas} \rightsquigarrow \Delta p_{hydro} \downarrow$

1. Two standalone gas-lifted wells: modeling strategy





$q_{or} = a\, q_{g\,inj}^4 + b\, q_{g\,inj}^3 + c\, q_{g\,inj}^2 + d\, q_{g\,inj} + e$

when is $q_{o\,total} = q_{or_1} + q_{or_2}$ maximum?

at which $q_{g\,inj_1}$? $q_{g\,inj_2}$?

what if there is a constraint on gas available? $q_{g\,inj\,tot} \geq q_{g\,inj_1} + q_{g\,inj_2}$

15

1. Two standalone gas-lifted wells: objective function behavior – brute force color map

$q_{o\,TOTAL} = q_{or_1}(q_{g\,inj_1}) + q_{or_2}(q_{g\,inj_2})$

$q_{o\,TOTAL} = f(q_{g\,inj_1}, q_{g\,inj_2})$

$z = f(x, y)$

max $\quad x^*, y^*$

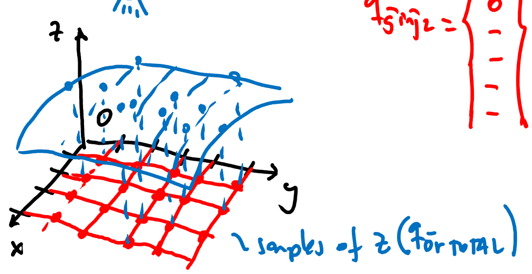
$q_{g\,inj_1} = \{ - \}$

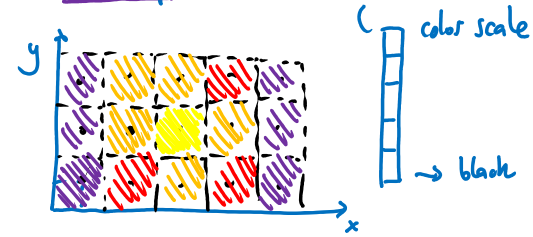
$q_{g\,inj_2} = \{ 0 \}$

1) element-wise combination of all

2) evaluate z ($q_{o\,TOTAL}$) for all

3) make a color map





16

1. Two standalone gas-lifted wells: objective function behavior – brute force color map

```
#CASE: SYSTEM OF TWO STANDALONE GAS-LIFTED WELLS
#AUTHOR: MILAN STANKO, NTNU, COURSE: TPG4230
#IMPORTING NEEDED LIBRARIES
import numpy as np
from scipy.optimize import fsolve
import matplotlib.pyplot as plt
```

```
def GLPerf_qo(a, b, c, d,e, qgi):
    res =a*np.power(qgi,4)+ b*np.power(qgi,3)+c*np.power(qgi,2)+d*qgi+e  #performance curve fitted to 4th degree polynomial
    return res
```

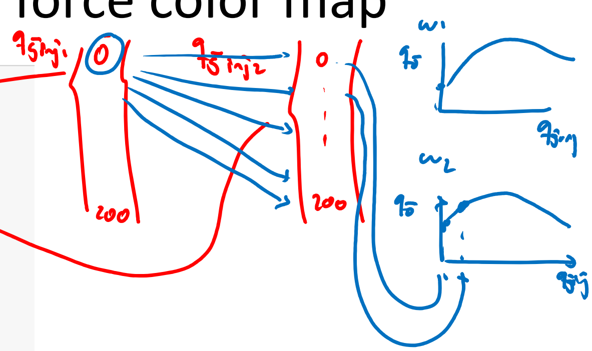
```
#gas lift performance curve data for two wells
a1=-3.9e-7  #(1/1e03 Sm3/d)^3
b1=2.1e-4   #(1/1e03 Sm3/d)^2
c1=-0.043   #(1/1e03 Sm3/d)
d1=3.7      #
e1=12       #(1e03 Sm3/d)
a2=-1.3e-7  #(1/1e03 Sm3/d)^3
b2=1e-4     #(1/1e03 Sm3/d)^2
c2=-0.028   #(1/1e03 Sm3/d)
d2=3.1
e2=-17       #(1e03 Sm3/d)
Po=6.29*80   #oil price USD/Sm3
Pg=2/8     #gas price, USD/Sm3
```

17

1. Two standalone gas-lifted wells: objective function behavior – brute force color map

```
#BRUTE-FORCE COMPUTING ALL COMBINATIONS
npoints=100
qgi_max=200  #[1e03 Sm3/d]
qgi_w1=np.linspace(0,qgi_max,npoints)
qgi_w2=np.linspace(0,qgi_max,npoints)
qotot=[]
qgitot=[]
revenue=[]
#computing objective (total oil production or revenue) and
#constraint, total gas injected
for qgi1 in qgi_w1:
    for qgi2 in qgi_w2:
        qo1=GLPerf_qo(a1,b1,c1,d1,e1,qgi1)
        qo2=GLPerf_qo(a2,b2,c2,d2,e2,qgi2)
        qotot=np.append(qotot,qo1+qo2)
        qgitot=np.append(qgitot,qgi1+qgi2)
        revenue=np.append(revenue,(qo1+qo2)*Po-(qgi1+qgi2)*Pg)
revenue=revenue/1e3
```

[1E03 USD]



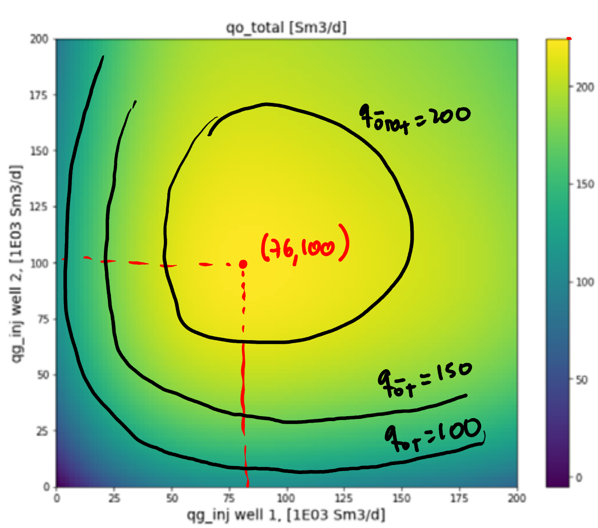
18

1. Two standalone gas-lifted wells: objective function behavior – brute force color map

```
#CREATING COLORMAPS AND CONTOUR PLOTS OF OBJECTIVE VARIABLE (total oil production)
#AND CONSTRAINED VARIABLE (total gas injection rate)
obj_opt=2  #1 if oil rate, 2 if revenue
if obj_opt==2:
    obj=revenue
    tag='revenue [1e3 USD]'
    levels_obj=np.linspace(50,110,5)
elif obj_opt==1:
    obj=qotot
    tag='qo_total [Sm3/d]'
    levels_obj=np.linspace(50,210,5)
constr=qgitot
#specifying desired number and range of contour lines
levels_qgi=np.linspace(100,200,4)
plt.figure(figsize=(10,8))
#creating mesh of qgin1,qgin2 to plot
xi,yi=np.mgrid[qgi_w1.min():qgi_w1.max():npoints*1j,qgi_w2.min():qgi_w2.max():npoints*1j]#ar
#Contour plot of objective function, total oil production
contour_obj=plt.contour(xi,yi,obj.reshape(xi.shape),levels=levels_obj,colors='black')
plt.clabel(contour_obj, inline=True, fmt='%1.0f',fontsize=12)
#Plot contour of constraint variable, total gas injection
contour_qgi=plt.contour(xi,yi,constr.reshape(xi.shape),levels=levels_qgi,colors='maroon')
plt.clabel(contour_qgi, inline=True,fmt='%1.0f', fontsize=12)
#plot color map of objective function, total oil production
plt.pcolormesh(xi,yi,obj.reshape(xi.shape))
#axis labels and plot title
plt.xlabel('qg_inj well 1, [1E03 Sm3/d]',fontsize=14)
plt.ylabel('qg_inj well 2, [1E03 Sm3/d]',fontsize=14)
plt.title(tag,fontsize=14)
plt.colorbar()
plt.show()
```

19

1. Two standalone gas-lifted wells: objective function behavior – brute force color map



where is q_{total} maximum? i believe:

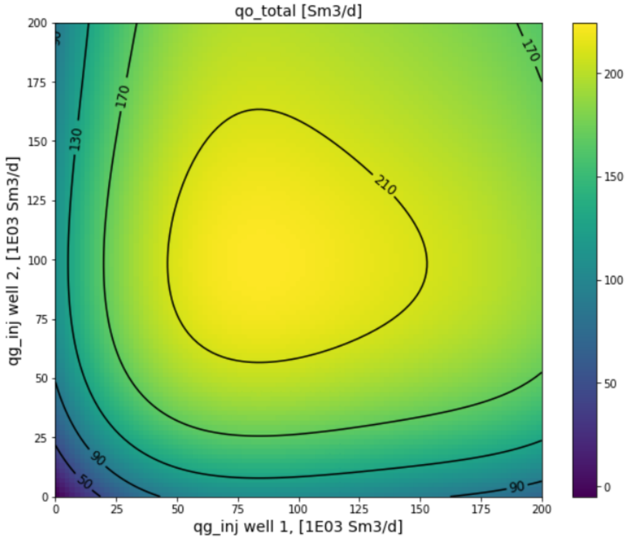
$q_{TOTAL} = 225$ Sm³/d $q_{inj1} \simeq 76$E3 Sm³/d

$q_{inj2} \simeq 100$E3 Sm³/d

contour plot ⤳ join all points with a line that have the same q_{TOTAL}

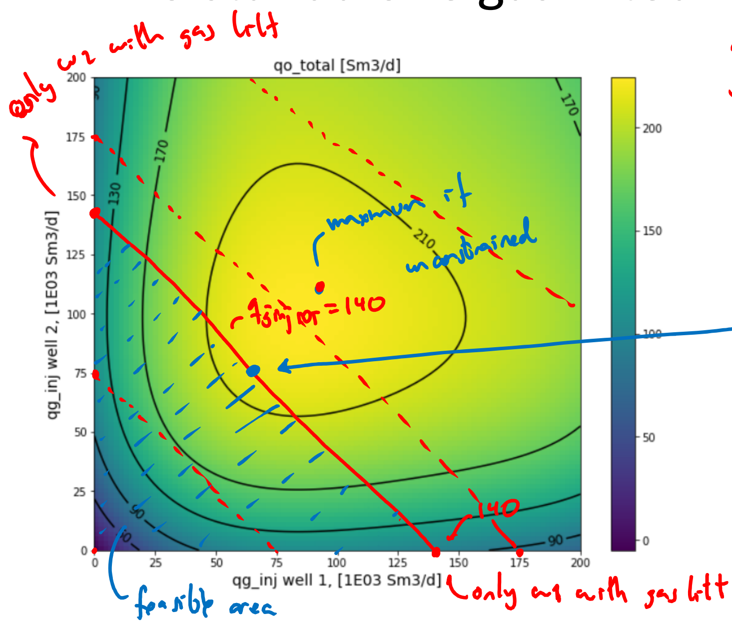
20

1. Two standalone gas-lifted wells: objective function behavior – contour lines



21

1. Two standalone gas-lifted wells: constraint



$$\underbrace{q_{g\,inj\,TOTAL}}_{140\ E03\ Sm^3/d} \geq \underbrace{q_{g\,inj\,1} + q_{g\,inj\,2}}$$

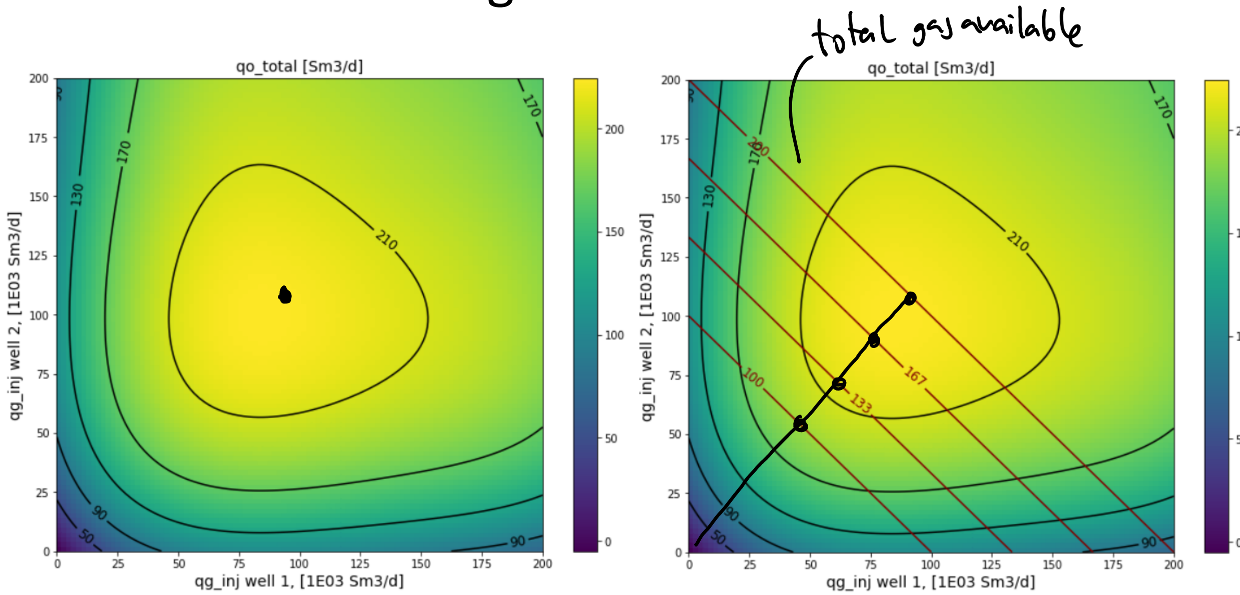
$x + y = \text{constant}$

$y = \text{const} - x$

maximum if constrained

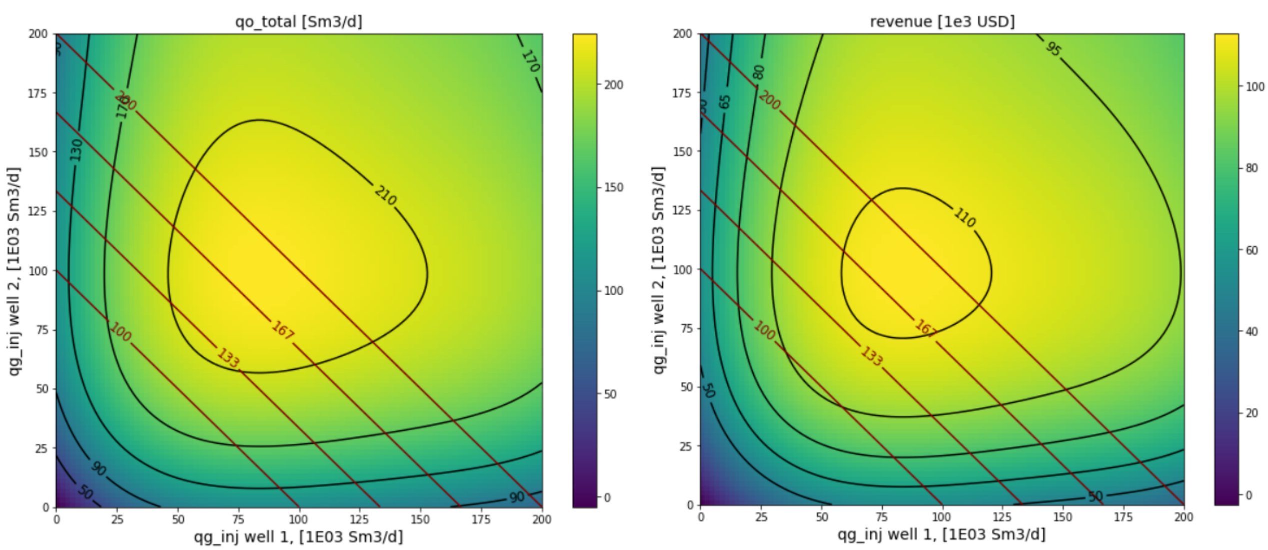
22

1. Two standalone gas-lifted wells: constraint



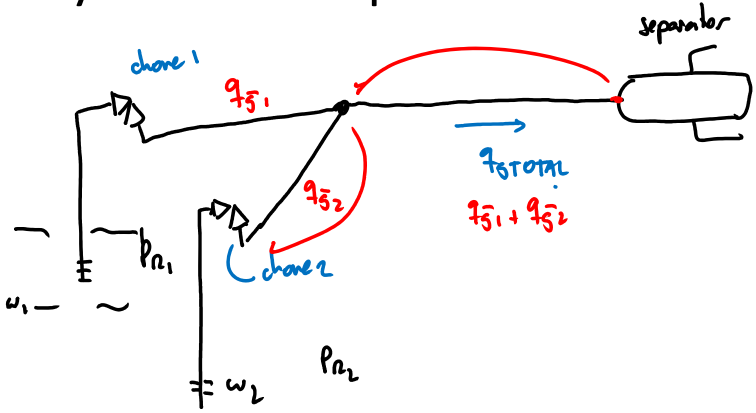
23

1. Two standalone gas-lifted wells: change of objective function



24

2. (Short term)Two gas wells in a network System description



each well produces some condensate

$CGR_1 = 0$

$CGR_2 =$ non zero

maximize $q_{g\,TOTAL}$, revenue, condensate

constraints in $q_{g1} \leq q_{g\,max}$

↳ to avoid sand production

25

2. Two gas wells in a network – modeling approach

Dry gas equations :

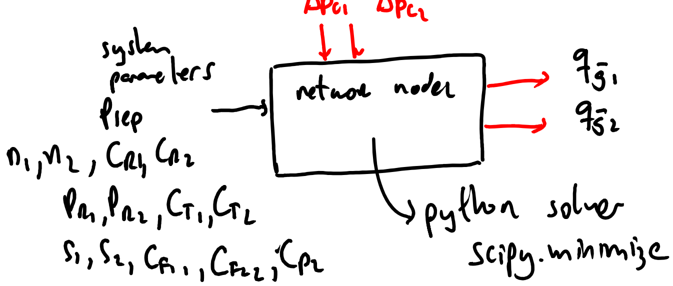
$$q_g = C_R \left(p_R^2 - p_{wf}^2\right)^n$$

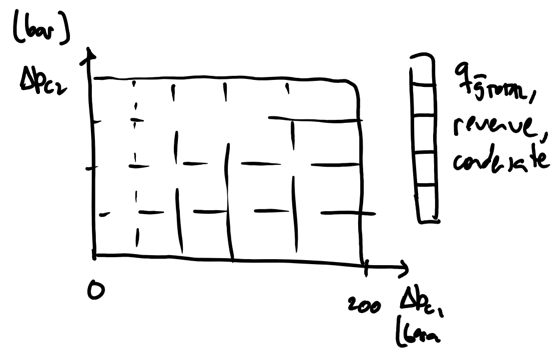
$$q_g = \left(\frac{p_{wf}^2}{e^s} - p_{wh}^2\right)^{0.5} \cdot C_T$$

$$q_g = \left(p_{in}^2 - p_{out}^2\right)^{0.5} C_{FL}$$

$$q_g = f\left(\Delta p_{choke1}, \Delta p_{choke2}\right)$$

Δp_{c1} Δp_{c2}



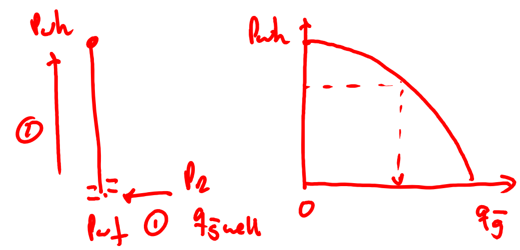


26

2. Two gas wells in a network – modeling approach

```
#AUTHOR: MILAN STANKO, NTNU, COURSE: TPG4230
#IMPORTING NEEDED LIBRARIES
import numpy as np
from scipy.optimize import minimize
import matplotlib.pyplot as plt
```

```
#FUNCTIONS
def IPRqg(CR, n, pR, pwf):
    a = CR * np.power((np.power(pR, 2) - np.power(pwf,2)),n)
    return a

def IPRpwf(CR, n, pR, qg):
    a = np.power(((np.power(pR, 2) - np.power((qg / CR), (1 / n)))) , 0.5)
    return a

def Tubingqg(CT, s, p1, p2):
    a = CT * np.power((np.power(p1, 2) / np.exp(s) - np.power(p2, 2)), 0.5)
    return a

def Tubingp1(CT, s, p2, qg):
    a = math.exp(s / 2) * np.power((np.power(p2, 2) + np.power((qg / CT),2)), 0.5)
    return a

def Tubingp2(CT, s, p1, qg):
    a = np.power((np.power(p1, 2) / np.exp(s) - np.power((qg / CT), 2)),0.5)
    return a
#WELLHEAD PERFORMANCE RELATIONSHIP
def WPR_pwh(pR,CR,n,CT,S,qg):
    pwf=IPRpwf(CR, n, pR, qg)    ← 1
    pwh=Tubingp2(CT,S,pwf,qg)    ← 2
    return pwh

v_WPR_pwh=np.vectorize(WPR_pwh)
```

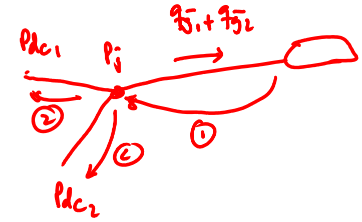


27

2. Two gas wells in a network – modeling approach

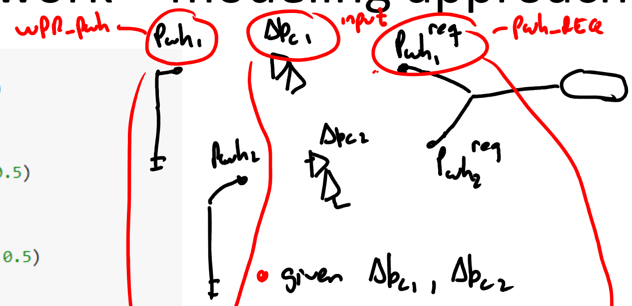
$q_g = C_{PL}(p_1^2 - p_2^2)^{0.5}$

```
def Lineqg(Cfl, p1, p2):
    a = Cfl * np.power(np.power(p1, 2) - np.power(p2, 2), 0.5)
    return a

def Linep1(Cfl, p2, qg):
    a = np.power((np.power(p2, 2) + np.power((qg / Cfl), 2)),0.5)
    return a

def Linep2(Cfl, p1, qg):
    a = np.power((np.power(p1, 2) - np.power((qg / Cfl), 2)), 0.5)
    return a

v_Linep1=np.vectorize(Linep1)
#REQUIRED PRESSURE AT CHOKE DISCHARGE CALCULATED FROM SEPARATOR FOR ALL WELLS
def pwh_REQ(Cpl,Cfl,psep,qg):
    pj=Linep1(Cpl,psep,np.sum(qg))  ← 1
    pwh=v_Linep1(Cfl,pj,qg)  ← 2
    return pwh

def error(qg,pR,CR,n,CT,S,Cpl,Cfl,psep,DP):
    pavail=v_WPR_pwh(pR,CR,n,CT,S,qg)
    preq=pwh_REQ(Cpl,Cfl,psep,qg)
    a=pavail-DP-preq
    a=np.power(a,2)
    return np.sum(a)
```



28

2. Two gas wells in a network – modeling approach




```
def Lineqg(Cfl, p1, p2):
    a = Cfl * np.power(np.power(p1, 2) - np.power(p2, 2), 0.5)
    return a

def Linep1(Cfl, p2, qg):
    a = np.power((np.power(p2, 2) + np.power((qg / Cfl), 2)),0.5)
    return a

def Linep2(Cfl, p1, qg):
    a = np.power((np.power(p1, 2) - np.power((qg / Cfl), 2)), 0.5)
    return a

v_Linep1=np.vectorize(Linep1)
#REQUIRED PRESSURE AT CHOKE DISCHARGE CALCULATED FROM SEPARATOR FOR ALL WELLS
def pwh_REQ(Cpl,Cfl,psep,qg):
    pj=Linep1(Cpl,psep,np.sum(qg))
    pwh=v_Linep1(Cfl,pj,qg)
    return pwh

def error(qg,pR,CR,n,CT,S,Cpl,Cfl,psep,DP):
    pavail=v_WPR_pwh(pR,CR,n,CT,S,qg)
    preq=pwh_REQ(Cpl,Cfl,psep,qg)
    a=pavail-DP-preq
    a=np.power(a,2)
    return np.sum(a)
```

- given Δp_{c_1}, Δp_{c_2}
- find q_{g_1}, q_{g_2} such that

$$P_{wh_1} - \Delta p_{c_1} = P_{wh_1}^{req}$$

$$P_{wh_2} - \Delta p_{c_2} = P_{wh_2}^{req}$$

calculate P_{wh_1}, P_{wh_2}

calculate $P_{wh_1}^{req}$, $P_{wh_2}^{req}$

$$error = \left(P_{wh_1} - \Delta p_{c_1} - P_{wh_1}^{req}\right)^2 + \left(P_{wh_2} - \Delta p_{c_2} - P_{wh_1}^{req}\right)^2$$

29

2. Two gas wells in a network – modeling approach

solver changing q_{g_1}, q_{g_2}, such that

$$\min \quad \left(P_{wh_1} - \Delta p_{c_1} - P_{wh_1}^{req}\right)^2 + \left(P_{wh_2} - \Delta p_{c_2} - P_{wh_2}^{req}\right)^2$$

30

2. Two gas wells in a network – modeling approach

```
#INPUT DATA
pR1=240  #bara
pR2=210  #bara
pR=[pR1,pR2]
CR1=1000 #Sm3/bar2n
CR2=700 #Sm3/bar2n
CR=[CR1,CR2]
n1=0.8
n2=0.75
n=[n1,n2]
S1=0.43
S2=0.34
S=[S1,S2]
CT1=38152 #Sm3/bar2
CT2=41163 #Sm3/bar2
CT=[CT1,CT2]
qg=[10,10]  #initial seed for well rate Sm3/d
Cpl=49406 #Sm3/bar2
Cfl1=70152.7 #Sm3/bar2
Cfl2=69883.2 #Sm3/bar2
Cfl=[Cfl1,Cfl2]
psep=60  #bara
CGR1=0  #Sm3/Sm3
CGR2=1/3000  #Sm3/Sm3
CGR=[CGR1,CGR2]
Po=6.29*80   #oil price USD/Sm3
Pg=1.5/8     #gas price, USD/Sm3
```

31

2. Two gas wells in a network – modeling approach

```
#BRUTE FORCE SOLVING ALL COMBINATIONS
npoints=20
DP1max=150  #bara
DP1=np.linspace(0,DP1max,npoints)
DP2max=150  #bara
DP2=np.linspace(0,DP2max,npoints)
qgtotal=[]
qctotal=[]
qg1=[]
qg2=[]
for dp1 in DP1:
    for dp2 in DP2:
        x=minimize(error,qg,args=(pR,CR,n,CT,S,Cpl,Cfl,psep,[dp1,dp2]),method='Nelder-Mead')
        qg1=np.append(qg1,x.x[0])
        qg2=np.append(qg2,x.x[1])
        qctotal=np.append(qctotal,np.dot(CGR,x.x))
        qgtotal=np.append(qgtotal,np.sum(x.x))
revenue=qctotal*Po+qgtotal*Pg
#converting output to millions
revenue=revenue/1e06
qg1=qg1/1e06
qg2=qg2/1e06
qgtotal=qgtotal/1e06
```

Δp_{c1} $\begin{bmatrix} 0 \\ - \\ - \\ - \\ - \\ 150 \end{bmatrix}$ Δp_{c2} $\begin{bmatrix} 0 \\ - \\ - \\ - \\ - \\ 150 \end{bmatrix}$

~ line excel solver

by changing $\begin{bmatrix} q_{g1} \\ q_{g2} \end{bmatrix}$

collection of results $\begin{bmatrix} q_{g1} \\ q_{g2} \end{bmatrix} \cdot \begin{bmatrix} CGR_1 \\ CGR_2 \end{bmatrix}$ +

32

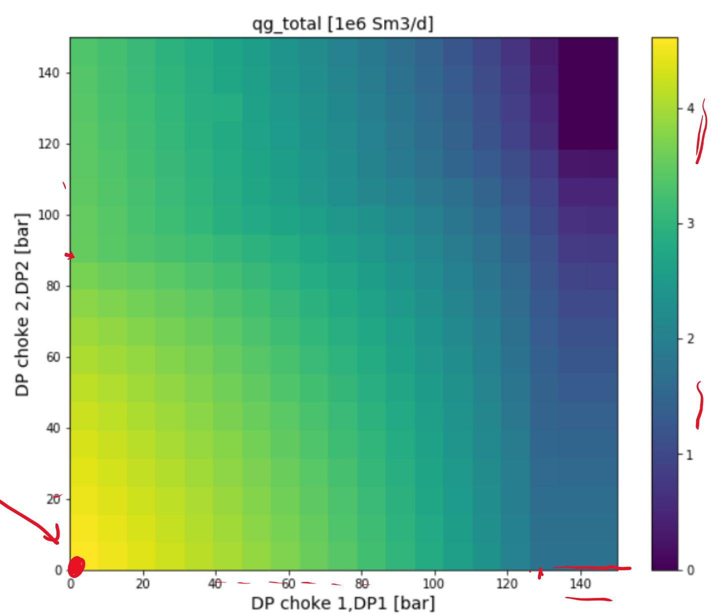
2. Two gas wells in a network – plotting

```python
#CREATING COLORMAPS AND CONTOUR PLOTS OF OBJECTIVE VARIABLE
#AND CONSTRAINED VARIABLE
obj_opt=3  #1 if revenue, 2 if gas, 3 if condensate
if obj_opt==1:
    obj=revenue
    tag='revenue [1e6 USD]'
elif obj_opt==2:
    obj=qgtotal
    tag='qg_total [1e6 Sm3/d]'
elif obj_opt==3:
    obj=qctotal
    tag='qc_total[Sm3/d]'
const_opt=1 #1 if well 1 gas rate, 2 if gas rate of well 2, 3 if total gas rate
if const_opt==1:
    constr=qg1
elif const_opt==2:
    constr=qg2
elif const_opt==3:
    constr=qgtotal
plt.figure(figsize=(10,8))
#creating mesh of DP1,DP2 to plot
xi,yi=np.mgrid[DP1.min():DP1.max():npoints*1j,DP2.min():DP2.max():npoints*1j]#another option to this is to use X,Y=np.meshgr
#Contour plot of objective function
contour_obj=plt.contour(xi,yi,obj.reshape(xi.shape),4,colors='black')
plt.clabel(contour_obj, inline=True, fmt='%1.1f',fontsize=12)
#Contour plot of constraint
contour_constr=plt.contour(xi,yi,constr.reshape(xi.shape),4,colors='maroon')
plt.clabel(contour_constr, inline=True, fmt='%1.1f',fontsize=12)
#plot color map of objective function,
plt.pcolormesh(xi,yi,obj.reshape(xi.shape))
#axis labels and plot title
plt.xlabel('DP choke 1,DP1 [bar]',fontsize=14)
plt.ylabel('DP choke 2,DP2 [bar]',fontsize=14)
plt.title(tag,fontsize=14)
plt.colorbar()
plt.show()
```

33

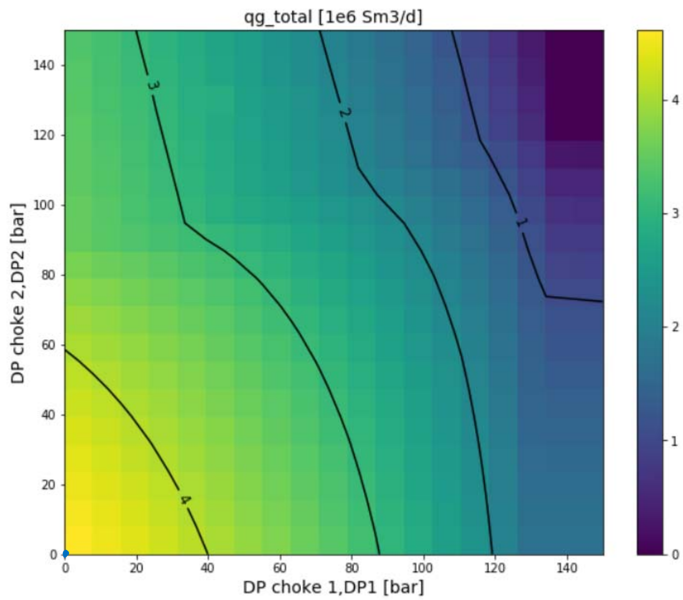
2. Two gas wells in a network – objective function: gas rate

maximum total gas production



34

2. Two gas wells in a network – objective function: gas rate



35

2. Two gas wells in a network – effect of constraint: gas rate or well 1

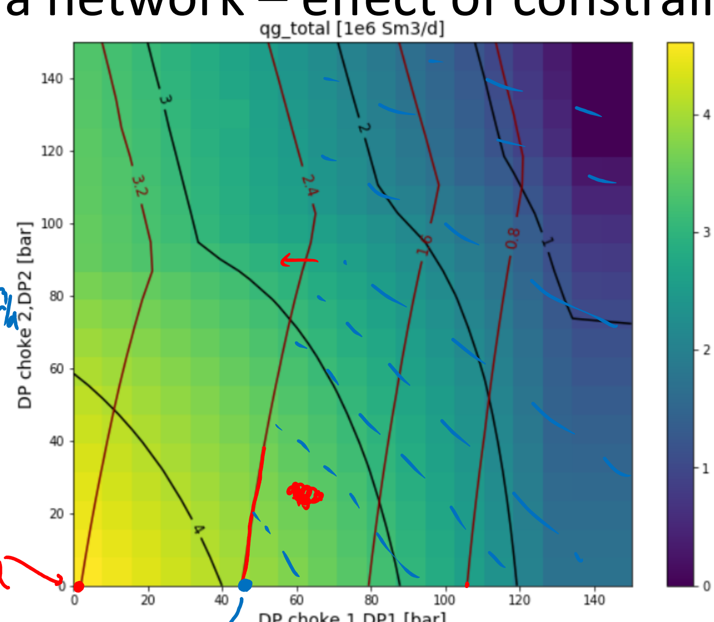
2.4E06

$q_{g_1} \leq$ [illegible] Sm^3/d

due to sand production

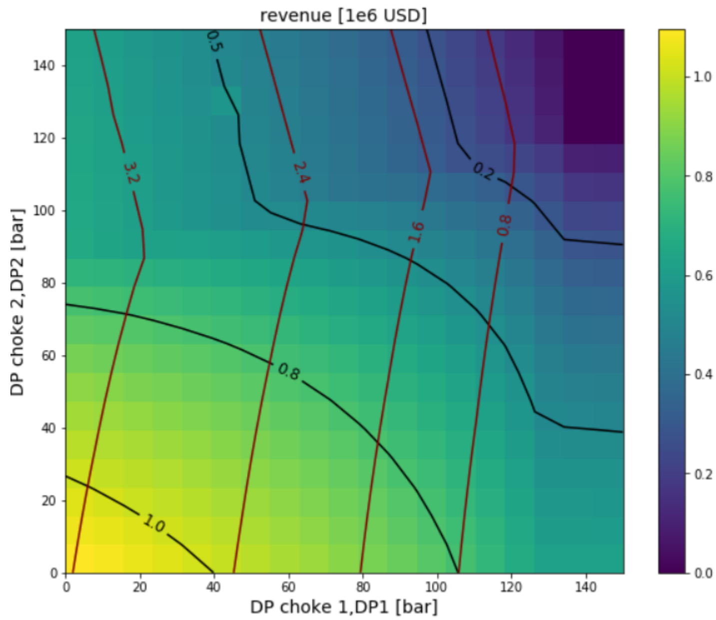
max q_g if constrained by $q_{g_1} \leq 2.4E6\ Sm^3/d$

max q_g if unconstrained



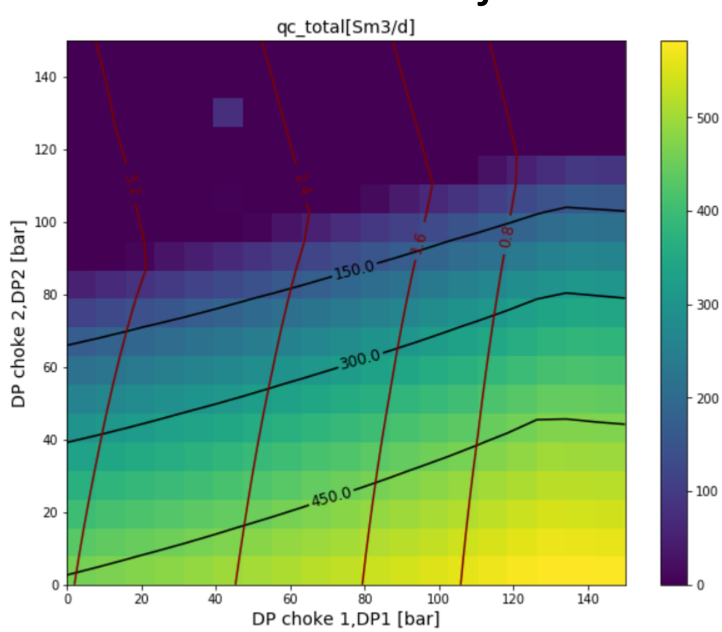
36

2. Two gas wells in a network – objective function: revenue



37

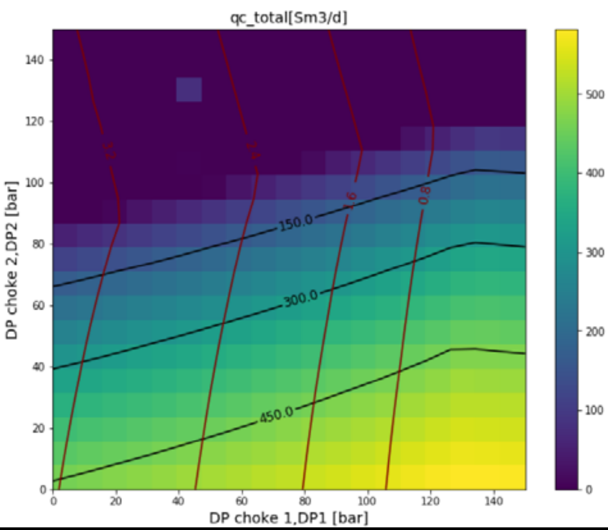
2. Two gas wells in a network – objective function: condensate



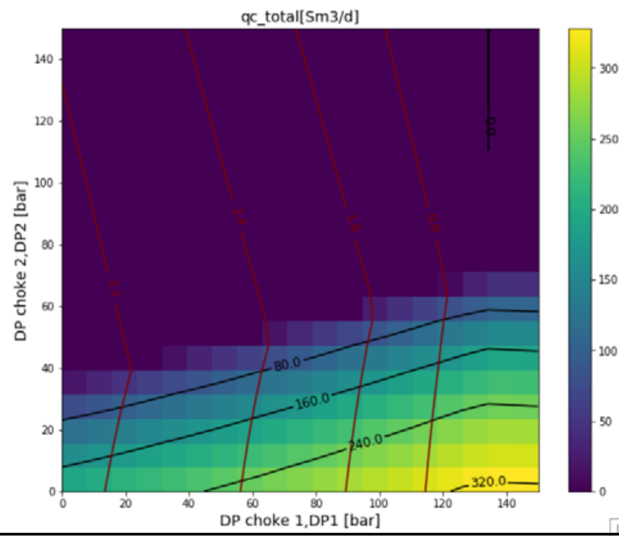
38

2. Two gas wells in a network – effect of depletion

p_{R_2} = 210 bara



p_{R_2} = 150 bara



39