Steam Assisted Gravity Drainage

May 18, 2016

Lewis Li, Daniel Brito

1 Reservoir Parameters

In this notebook, we will analyze the recovery and production rates in a steam assisted gravity drainage (SAGD) project. The analysis uses the data from Butler, 1991. The reservoir parameters are specified below:

Symbol	Units	Value	Description
T_r	C	15	Reservoir temperature
T_s	C	188	Steam temperature
m_{exp}	m	3.4	Kinematic viscosity exponent
$\mu_{Bitumen,188}$	cs	7.8	Bitumen kinematic viscosity @ 188C
$r_{thickness}$	m	20	Reservoir thickness
α	$\frac{m^2}{D}$	0.07	Thermal diffusivity
ϕ	%	0.33	Porosity
S_o	%	0.75	Initial oil saturation
S_{or}	%	0.13	Residual oil saturation
K_{eff}	D	0.4	Effective permeability for oil flow
D_{well}	m	2.5	Distance between reservoir's base and producers
w	m	75	Spacing between wells
T_{total}	years	7	Evaluation period
g	$\frac{m}{s^2}$	9.81	Accelerationb due to gravity

```
In [1]: % Reservoir temperature (C)
        Tr = 15;
        % Steam temperature (C)
        Ts = 188;
        % Kinematic viscosity exponent (m)
        m_exp = 3.4;
        % Bitumen kinematic  viscosity @ 188C (cs)
        mu_Bitumen188 = 7.8;
        % Reservoir thickness (m)
        resThickness = 20;
        % Thermal diffusivity (m^2/D)
```

```
alpha = 0.07;
% Porosity
phi = 0.33;
% Initial oil saturation
S_o = 0.75;
% Residual oil saturation
S_or = 0.13;
% Effective permeability for oil flow (Darcy)
K_eff = 0.4;
% Distance between reservoir's base and producers (m)
baseWellDistance = 2.5;
% Spacing between wells (m)
w = 75;
% Evaluation period (years)
TotalTime = 7;
% Gravity (m/s^2)
g = 9.81;

%% Conversion Factors
DarcyToM2 = 9.869233e-13;
SecondsToDays = 24*60*60;
DaysToSeconds = SecondsToDays;
YearsToDays = 365;
centistokesToM2S= 1e-6;
```

1.1 Computing Depletion Rates

We will first compute the depletion rates, which depends on the position of the interface. We will express time as a dimensionless unit T^*:

$$T^* = \frac{t}{w}\sqrt{\frac{kg\alpha}{2\phi\Delta S_o m\nu_o(T_s)h}}$$

```
In [2]: h = resThickness - baseWellDistance;     % Height

        tStarConstant = (2.0/w) * sqrt (((K_eff * DarcyToM2)*g*...
            alpha/(SecondsToDays) ) / ( phi * (S_o - S_or) * ...
            m_exp * mu_Bitumen188*centistokesToM2S * h));
```

To simplify the problem of solving for the interface position numerically, we approximate the dimensionless production rate using a polynomial:

$$q* = \sqrt{3/2} - T^{*2}\sqrt{\frac{2}{3}}$$

We will compute this over the entire time period of the project (7 years) discretized into 100 time steps.

```
In [3]: n = 100;
```

```
%Time discretization (s)
t = linspace(0,TotalTime*YearsToDays*SecondsToDays,n);

%Time discretization (years)
tYears = t./(YearsToDays*SecondsToDays);

tStar = tStarConstant*t;
qStar = sqrt(3/2) - sqrt(2/3)*tStar.^2;
```

Likewise, the production is given by:

$$q = \frac{2 \times q^*}{F}$$

where F is a dimensionless factor given by:

$$F = \sqrt{\frac{m\nu_s}{kg\alpha\phi h\Delta S_o}}$$

```
In [4]: FFactor = sqrt( (m_exp*mu_Bitumen188*centistokesToM2S) /...
            ( (K_eff * DarcyToM2) * g *(alpha/(SecondsToDays)) * ...
            h * phi * (S_o - S_or) ) );
        q = 2*qStar*SecondsToDays./FFactor;
```

The recovery is then estimated as:

$$Rev = \int q^* dt^*$$

In this case, we can analytically integrate our expression for q* to obtain:

$$Rev = \sqrt{\frac{3}{2}}T^* - \frac{1}{3}T^{*3}\sqrt{\frac{2}{3}}$$

```
In [5]: Recovery = sqrt(3/2)*tStar - tStar.^3*sqrt(2/3)/3;

        DepletionResults = [tYears' tStar' qStar' Recovery' q'];
        ColumnNames = {'Time, years', 't*', 'q*','Recovery','q,m^3/(m day)'};

        DepletionResults = [ColumnNames; num2cell(DepletionResults)];
        display(DepletionResults)

DepletionResults =

    'Time, years'    't*'        'q*'        'Recovery'    'q,m^3/(m day)'
    [          0]    [     0]    [1.2247]    [       0]    [       0.1377]
    [     0.0707]    [0.0108]    [1.2246]    [  0.0132]    [       0.1377]
    [     0.1414]    [0.0216]    [1.2244]    [  0.0265]    [       0.1377]
    [     0.2121]    [0.0324]    [1.2239]    [  0.0397]    [       0.1376]
    [     0.2828]    [0.0432]    [1.2232]    [  0.0529]    [       0.1376]
    [     0.3535]    [0.0540]    [1.2224]    [  0.0661]    [       0.1375]
```

[0.4242]	[0.0649]	[1.2213]	[0.0794]	[0.1374]
[0.4949]	[0.0757]	[1.2201]	[0.0925]	[0.1372]
[0.5657]	[0.0865]	[1.2186]	[0.1057]	[0.1371]
[0.6364]	[0.0973]	[1.2170]	[0.1189]	[0.1369]
[0.7071]	[0.1081]	[1.2152]	[0.1320]	[0.1367]
[0.7778]	[0.1189]	[1.2132]	[0.1452]	[0.1364]
[0.8485]	[0.1297]	[1.2110]	[0.1583]	[0.1362]
[0.9192]	[0.1405]	[1.2086]	[0.1713]	[0.1359]
[0.9899]	[0.1513]	[1.2060]	[0.1844]	[0.1356]
[1.0606]	[0.1621]	[1.2033]	[0.1974]	[0.1353]
[1.1313]	[0.1729]	[1.2003]	[0.2104]	[0.1350]
[1.2020]	[0.1837]	[1.1972]	[0.2234]	[0.1346]
[1.2727]	[0.1946]	[1.1938]	[0.2363]	[0.1343]
[1.3434]	[0.2054]	[1.1903]	[0.2492]	[0.1339]
[1.4141]	[0.2162]	[1.1866]	[0.2620]	[0.1335]
[1.4848]	[0.2270]	[1.1827]	[0.2748]	[0.1330]
[1.5556]	[0.2378]	[1.1786]	[0.2876]	[0.1326]
[1.6263]	[0.2486]	[1.1743]	[0.3003]	[0.1321]
[1.6970]	[0.2594]	[1.1698]	[0.3130]	[0.1316]
[1.7677]	[0.2702]	[1.1651]	[0.3256]	[0.1310]
[1.8384]	[0.2810]	[1.1603]	[0.3381]	[0.1305]
[1.9091]	[0.2918]	[1.1552]	[0.3507]	[0.1299]
[1.9798]	[0.3026]	[1.1500]	[0.3631]	[0.1293]
[2.0505]	[0.3135]	[1.1445]	[0.3755]	[0.1287]
[2.1212]	[0.3243]	[1.1389]	[0.3879]	[0.1281]
[2.1919]	[0.3351]	[1.1331]	[0.4001]	[0.1274]
[2.2626]	[0.3459]	[1.1271]	[0.4124]	[0.1268]
[2.3333]	[0.3567]	[1.1209]	[0.4245]	[0.1261]
[2.4040]	[0.3675]	[1.1145]	[0.4366]	[0.1253]
[2.4747]	[0.3783]	[1.1079]	[0.4486]	[0.1246]
[2.5455]	[0.3891]	[1.1011]	[0.4605]	[0.1238]
[2.6162]	[0.3999]	[1.0942]	[0.4724]	[0.1231]
[2.6869]	[0.4107]	[1.0870]	[0.4842]	[0.1223]
[2.7576]	[0.4215]	[1.0797]	[0.4959]	[0.1214]
[2.8283]	[0.4324]	[1.0721]	[0.5075]	[0.1206]
[2.8990]	[0.4432]	[1.0644]	[0.5191]	[0.1197]
[2.9697]	[0.4540]	[1.0565]	[0.5305]	[0.1188]
[3.0404]	[0.4648]	[1.0484]	[0.5419]	[0.1179]
[3.1111]	[0.4756]	[1.0401]	[0.5532]	[0.1170]
[3.1818]	[0.4864]	[1.0316]	[0.5644]	[0.1160]
[3.2525]	[0.4972]	[1.0229]	[0.5755]	[0.1150]
[3.3232]	[0.5080]	[1.0140]	[0.5865]	[0.1140]
[3.3939]	[0.5188]	[1.0050]	[0.5974]	[0.1130]
[3.4646]	[0.5296]	[0.9957]	[0.6082]	[0.1120]
[3.5354]	[0.5404]	[0.9863]	[0.6189]	[0.1109]
[3.6061]	[0.5512]	[0.9766]	[0.6295]	[0.1098]
[3.6768]	[0.5621]	[0.9668]	[0.6401]	[0.1087]
[3.7475]	[0.5729]	[0.9568]	[0.6504]	[0.1076]

[3.8182]	[0.5837]	[0.9466]	[0.6607]	[0.1065]
[3.8889]	[0.5945]	[0.9362]	[0.6709]	[0.1053]
[3.9596]	[0.6053]	[0.9256]	[0.6810]	[0.1041]
[4.0303]	[0.6161]	[0.9148]	[0.6909]	[0.1029]
[4.1010]	[0.6269]	[0.9038]	[0.7007]	[0.1017]
[4.1717]	[0.6377]	[0.8927]	[0.7105]	[0.1004]
[4.2424]	[0.6485]	[0.8813]	[0.7200]	[0.0991]
[4.3131]	[0.6593]	[0.8698]	[0.7295]	[0.0978]
[4.3838]	[0.6701]	[0.8581]	[0.7388]	[0.0965]
[4.4545]	[0.6810]	[0.8461]	[0.7481]	[0.0952]
[4.5253]	[0.6918]	[0.8340]	[0.7571]	[0.0938]
[4.5960]	[0.7026]	[0.8217]	[0.7661]	[0.0924]
[4.6667]	[0.7134]	[0.8092]	[0.7749]	[0.0910]
[4.7374]	[0.7242]	[0.7965]	[0.7836]	[0.0896]
[4.8081]	[0.7350]	[0.7837]	[0.7921]	[0.0881]
[4.8788]	[0.7458]	[0.7706]	[0.8005]	[0.0867]
[4.9495]	[0.7566]	[0.7573]	[0.8088]	[0.0852]
[5.0202]	[0.7674]	[0.7439]	[0.8169]	[0.0837]
[5.0909]	[0.7782]	[0.7302]	[0.8249]	[0.0821]
[5.1616]	[0.7890]	[0.7164]	[0.8327]	[0.0806]
[5.2323]	[0.7999]	[0.7024]	[0.8403]	[0.0790]
[5.3030]	[0.8107]	[0.6882]	[0.8479]	[0.0774]
[5.3737]	[0.8215]	[0.6738]	[0.8552]	[0.0758]
[5.4444]	[0.8323]	[0.6592]	[0.8624]	[0.0741]
[5.5152]	[0.8431]	[0.6444]	[0.8695]	[0.0725]
[5.5859]	[0.8539]	[0.6294]	[0.8764]	[0.0708]
[5.6566]	[0.8647]	[0.6142]	[0.8831]	[0.0691]
[5.7273]	[0.8755]	[0.5989]	[0.8896]	[0.0674]
[5.7980]	[0.8863]	[0.5833]	[0.8960]	[0.0656]
[5.8687]	[0.8971]	[0.5676]	[0.9022]	[0.0638]
[5.9394]	[0.9079]	[0.5517]	[0.9083]	[0.0620]
[6.0101]	[0.9187]	[0.5355]	[0.9142]	[0.0602]
[6.0808]	[0.9296]	[0.5192]	[0.9199]	[0.0584]
[6.1515]	[0.9404]	[0.5027]	[0.9254]	[0.0565]
[6.2222]	[0.9512]	[0.4860]	[0.9307]	[0.0547]
[6.2929]	[0.9620]	[0.4691]	[0.9359]	[0.0528]
[6.3636]	[0.9728]	[0.4521]	[0.9409]	[0.0508]
[6.4343]	[0.9836]	[0.4348]	[0.9457]	[0.0489]
[6.5051]	[0.9944]	[0.4174]	[0.9503]	[0.0469]
[6.5758]	[1.0052]	[0.3997]	[0.9547]	[0.0450]
[6.6465]	[1.0160]	[0.3819]	[0.9589]	[0.0429]
[6.7172]	[1.0268]	[0.3638]	[0.9629]	[0.0409]
[6.7879]	[1.0376]	[0.3456]	[0.9668]	[0.0389]
[6.8586]	[1.0485]	[0.3272]	[0.9704]	[0.0368]
[6.9293]	[1.0593]	[0.3086]	[0.9738]	[0.0347]
[7]	[1.0701]	[0.2898]	[0.9771]	[0.0326]

1.2 Rates From Rising Steam

We next compute the rates for the rising steam chamber using:

$$q_{cum} = 2.25\left(\frac{kg\alpha}{m\nu_s}\right)^{\frac{2}{3}}(\phi\Delta S_o)^{\frac{1}{3}}t^{\frac{4}{3}}$$

$$q = 3\left(\frac{kg\alpha}{m\nu_s}\right)^{\frac{2}{3}}(\phi\Delta S_o)^{\frac{1}{3}}t^{\frac{1}{3}}$$

and recovery is given by:

$$Rev = \frac{q_{cum}}{h\phi\Delta S_o(spacing)}$$

```
In [6]: Coef1 = (((K_eff * DarcyToM2)*g*alpha/(SecondsToDays))/...
            ( m_exp * mu_Bitumen188*centistokesToM2S ) )^(2/3);
        Coef2 = (phi * (S_o - S_or) )^(1/3);

        qCumRise = 2.25*Coef1*Coef2*t.^(4/3)*SecondsToDays;
        qRise = 3*Coef1*Coef2*t.^(1/3)*SecondsToDays;
        RecoveryRise = qCumRise./(h*phi*(S_o - S_or)*w*SecondsToDays);

        SteamResults = [tYears' qRise' RecoveryRise'];
        ColumnNames = {'Time, years', 'q,m^3/(m day)','Recovery',};

        EndIndex = sum(RecoveryRise<=1)+1;
        Results = [ColumnNames; num2cell(SteamResults)];
        display(Results(1:EndIndex,:))
```

```
'Time, years'    'q,m^3/(m day)'    'Recovery'
    [         0]    [            0]    [      0]
    [    0.0707]    [       0.0481]    [ 0.0035]
    [    0.1414]    [       0.0606]    [ 0.0087]
    [    0.2121]    [       0.0694]    [ 0.0150]
    [    0.2828]    [       0.0763]    [ 0.0220]
    [    0.3535]    [       0.0822]    [ 0.0296]
    [    0.4242]    [       0.0874]    [ 0.0378]
    [    0.4949]    [       0.0920]    [ 0.0464]
    [    0.5657]    [       0.0962]    [ 0.0555]
    [    0.6364]    [       0.1000]    [ 0.0649]
    [    0.7071]    [       0.1036]    [ 0.0747]
    [    0.7778]    [       0.1069]    [ 0.0848]
    [    0.8485]    [       0.1101]    [ 0.0952]
    [    0.9192]    [       0.1131]    [ 0.1060]
    [    0.9899]    [       0.1159]    [ 0.1170]
    [    1.0606]    [       0.1186]    [ 0.1282]
    [    1.1313]    [       0.1212]    [ 0.1397]
    [    1.2020]    [       0.1236]    [ 0.1515]
    [    1.2727]    [       0.1260]    [ 0.1635]
```

[1.3434]	[0.1283]	[0.1757]
[1.4141]	[0.1305]	[0.1882]
[1.4848]	[0.1327]	[0.2008]
[1.5556]	[0.1347]	[0.2137]
[1.6263]	[0.1368]	[0.2267]
[1.6970]	[0.1387]	[0.2400]
[1.7677]	[0.1406]	[0.2534]
[1.8384]	[0.1425]	[0.2670]
[1.9091]	[0.1443]	[0.2808]
[1.9798]	[0.1460]	[0.2947]
[2.0505]	[0.1477]	[0.3088]
[2.1212]	[0.1494]	[0.3231]
[2.1919]	[0.1511]	[0.3375]
[2.2626]	[0.1527]	[0.3521]
[2.3333]	[0.1542]	[0.3669]
[2.4040]	[0.1558]	[0.3818]
[2.4747]	[0.1573]	[0.3968]
[2.5455]	[0.1588]	[0.4120]
[2.6162]	[0.1602]	[0.4274]
[2.6869]	[0.1617]	[0.4428]
[2.7576]	[0.1631]	[0.4584]
[2.8283]	[0.1645]	[0.4742]
[2.8990]	[0.1658]	[0.4900]
[2.9697]	[0.1672]	[0.5060]
[3.0404]	[0.1685]	[0.5222]
[3.1111]	[0.1698]	[0.5384]
[3.1818]	[0.1710]	[0.5548]
[3.2525]	[0.1723]	[0.5713]
[3.3232]	[0.1735]	[0.5879]
[3.3939]	[0.1748]	[0.6047]
[3.4646]	[0.1760]	[0.6215]
[3.5354]	[0.1772]	[0.6385]
[3.6061]	[0.1783]	[0.6556]
[3.6768]	[0.1795]	[0.6728]
[3.7475]	[0.1806]	[0.6901]
[3.8182]	[0.1818]	[0.7075]
[3.8889]	[0.1829]	[0.7250]
[3.9596]	[0.1840]	[0.7426]
[4.0303]	[0.1851]	[0.7604]
[4.1010]	[0.1861]	[0.7782]
[4.1717]	[0.1872]	[0.7961]
[4.2424]	[0.1883]	[0.8142]
[4.3131]	[0.1893]	[0.8323]
[4.3838]	[0.1903]	[0.8506]
[4.4545]	[0.1913]	[0.8689]
[4.5253]	[0.1924]	[0.8873]
[4.5960]	[0.1934]	[0.9059]
[4.6667]	[0.1943]	[0.9245]

```
[      4.7374]    [        0.1953]    [   0.9432]
[      4.8081]    [        0.1963]    [   0.9621]
[      4.8788]    [        0.1972]    [   0.9810]
[      4.9495]    [        0.1982]    [   1.0000]
```

1.3 Computing the time changeover point

The time changeover is the intersection point of both curves.

```
In [7]: warning('off','all')
        addpath('../matlab')
        [RecoveryIntersection,qIntersection] = ...
            intersections(Recovery',q,RecoveryRise',qRise,1);

        % Determine time step where intersect occurs for depletion
        RecoveryRow = sum(Recovery<=RecoveryIntersection);

        % Determine time step where intersect occurs for depletion
        RecoveryRiseRow = sum(RecoveryRise<=RecoveryIntersection);

        timeCalcDepletion = YearsToDays * tYears(RecoveryRow-1);
        timeCalcSteam = YearsToDays * tYears(RecoveryRiseRow-1);
        deltaTime = timeCalcSteam - timeCalcDepletion;

        display(['The changeover point occurs at ' num2str(qIntersection) ...
            ' and recovery of ' num2str(RecoveryIntersection)]);
        display(['Corresponds to ' num2str(timeCalcSteam) ...
            ' days production from steam']);
        display(['Corresponds to ' num2str(timeCalcDepletion) ...
            ' days production from depletion']);
        display(['Yields a difference of ' num2str(deltaTime) ' days']);
```

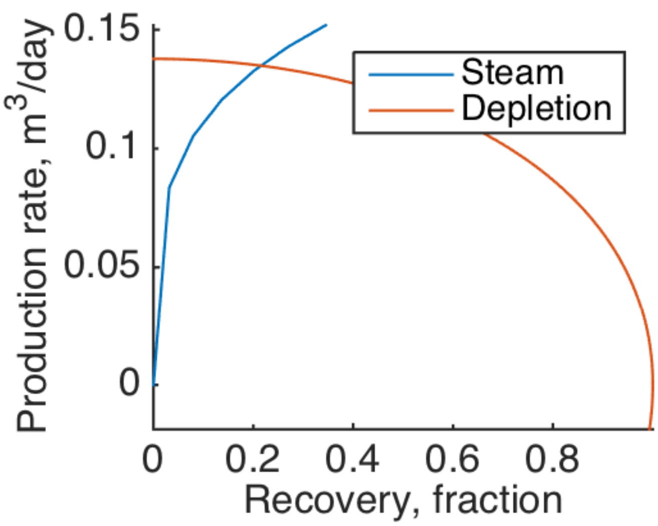
```
The changeover point occurs at 0.13488 and recovery of 0.21457
Corresponds to 541.9697 days production from steam
Corresponds to 387.1212 days production from depletion
Yields a difference of 154.8485 days
```

1.4 Computing the Response Rates

We can now plot production rate versus recovery and visualize where the crossover point occurs.

```
In [12]: % The rates for production are from steam until the changeover point, and
         % then the rates are from depeletion.
         ChangeOverTime = timeCalcSteam/YearsToDays;
         StepSize = 0.25;

         % Steam-based production (computed as function of time in seconds)
```

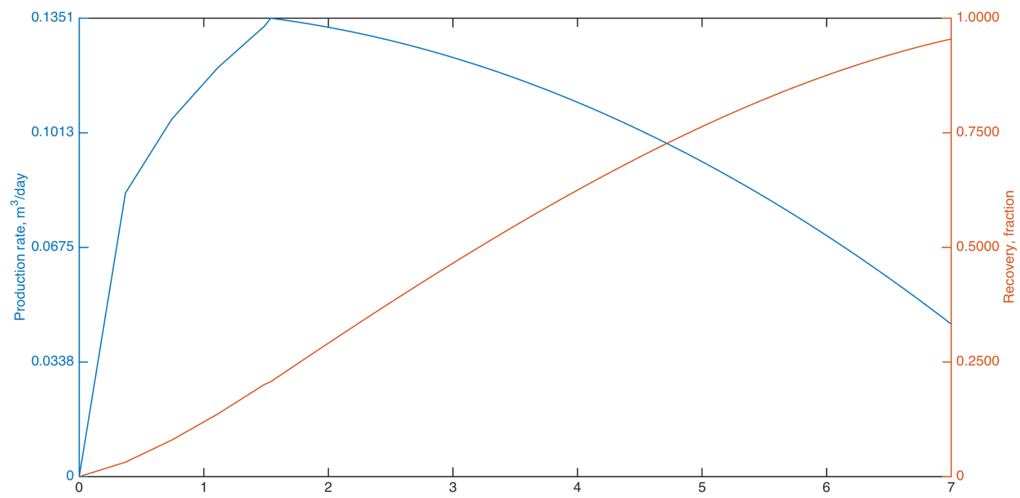
```
timeSteam = (0:StepSize:1.5)*ChangeOverTime*YearsToDays*DaysToSeconds;
qCumSteam = (2.25 * Coef1 * Coef2 * (timeSteam).^(4/3)) * DaysToSeconds;
qSteam = (3 * Coef1 * Coef2 * (timeSteam).^(1/3)) * DaysToSeconds;
RecoverySteam = qCumSteam ./ ( h*phi*(S_o - S_or)*w*DaysToSeconds );

% Set up time for depeletion (including time correction)
StepSize = 0.05;
timeDepletion = (0:(ChangeOverTime*StepSize):(TotalTime+2) - ...
    deltaTime/YearsToDays) * YearsToDays*DaysToSeconds;
tStarDepletion = tStarConstant * timeDepletion;
qStarDepletion = sqrt(1.5) - (tStarDepletion.^2)*sqrt(2/3);
q = 2 * qStarDepletion * (DaysToSeconds)/( FFactor );
recoveryFactor = sqrt(3/2)*tStarDepletion - ...
    (1/3)*(tStarDepletion.^3)*sqrt(2/3);

figure;
FontSize = 15;
hold on;
plot(RecoverySteam,qSteam);
plot(recoveryFactor,q);
axis tight;
ylabel('Production rate, m^3/day','Fontsize', FontSize) % label left y-axi
xlabel('Recovery, fraction','Fontsize', FontSize) % label right y-axis
legend('Steam','Depletion');
set(gca,'FontSize',FontSize);
```



We will next plot the production from steam up to the changeover point, then the remaining rates are calculated from depletion after adjusting the time.

```
In [14]: %plot inline -s 2000,1000
```

```
% Convert time from seconds to years and remove time correction
timeSteamYears = (1/(YearsToDays*DaysToSeconds)) * timeSteam;
timeDepletionYears = (1/(YearsToDays*DaysToSeconds)) * ...
    timeDepletion + deltaTime/YearsToDays;

% Find when steam time ends and depeletion starts
SteamEndTime = sum(timeSteamYears<=ChangeOverTime);
DepletionStartTime = find(timeDepletionYears>=ChangeOverTime,1);

qTotal = [(qSteam(:,1:SteamEndTime))';(q(:,DepletionStartTime:end))'];
timeTotal = [(timeSteamYears(:,1:SteamEndTime))' ; ...
     (timeDepletionYears(:,DepletionStartTime:end))'];

qStarTotal = [ zeros(SteamEndTime,1); ...
     (qStarDepletion(:,DepletionStartTime:end))'];
tStarTotal = [ zeros(SteamEndTime,1); ...
     (tStarDepletion(:,DepletionStartTime:end))'];
RecoveryTotal = [ (RecoverySteam(:,1:SteamEndTime))'; ...
     (recoveryFactor(:,DepletionStartTime:end))'];

figure
[haxes,hline1,hline2] = plotyy(timeTotal,qTotal,timeTotal, ...
    RecoveryTotal,'plot','plot');
ylabel(haxes(1),'Production rate, m^3/day','Fontsize', ...
    FontSize) % label left y-axis
ylabel(haxes(2),'Recovery, fraction','Fontsize', ...
    FontSize) % label right y-axis
xlabel(haxes(2),'Time (years)', 'Fontsize', ...
    FontSize) % label x-axis

Nticks = 5;
FontSize = 12;
set(haxes,'XLim',[0 TotalTime])
set(haxes(1),'YLim',[0 max(qTotal)]);
set(haxes(1),'ytick',linspace(0, max(qTotal), Nticks));
set(haxes(2),'YLim',[0 max(RecoveryTotal)]);
set(haxes(2),'ytick',linspace(0, max(RecoveryTotal), Nticks));
set(gcf,'color','w');
set(haxes(1),'FontSize',FontSize);
set(haxes(2),'FontSize',FontSize);
```



Finally, we disply the full result.

```
In [15]: TotalResults = [timeTotal qTotal tStarTotal qStarTotal RecoveryTotal];
         ColumnNames = {'Time, years', 'q,m^3/(m day)','t*','q*','Recovery'};
         TotalResults = [ColumnNames; num2cell(TotalResults)];

         EndIndex = sum(timeTotal<=TotalTime)+1;
         display(TotalResults(1:EndIndex,:))
```

```
'Time, years'    'q,m^3/(m day)'    't*'         'q*'         'Recovery'
    [         0]    [             0]    [      0]    [      0]    [        0]
    [    0.3712]    [        0.0836]    [      0]    [      0]    [   0.0316]
    [    0.7424]    [        0.1053]    [      0]    [      0]    [   0.0797]
    [    1.1136]    [        0.1205]    [      0]    [      0]    [   0.1368]
    [    1.4848]    [        0.1327]    [      0]    [      0]    [   0.2008]
    [    1.5379]    [        0.1351]    [0.1702]    [1.2011]    [   0.2072]
    [    1.6121]    [        0.1347]    [0.1816]    [1.1978]    [   0.2208]
    [    1.6864]    [        0.1343]    [0.1929]    [1.1944]    [   0.2343]
    [    1.7606]    [        0.1339]    [0.2043]    [1.1907]    [   0.2479]
    [    1.8348]    [        0.1335]    [0.2156]    [1.1868]    [   0.2614]
    [    1.9091]    [        0.1330]    [0.2270]    [1.1827]    [   0.2748]
    [    1.9833]    [        0.1325]    [0.2383]    [1.1784]    [   0.2882]
    [    2.0576]    [        0.1320]    [0.2497]    [1.1738]    [   0.3016]
    [    2.1318]    [        0.1315]    [0.2610]    [1.1691]    [   0.3149]
    [    2.2061]    [        0.1309]    [0.2724]    [1.1642]    [   0.3281]
    [    2.2803]    [        0.1304]    [0.2837]    [1.1590]    [   0.3413]
    [    2.3545]    [        0.1297]    [0.2951]    [1.1537]    [   0.3544]
    [    2.4288]    [        0.1291]    [0.3064]    [1.1481]    [   0.3675]
    [    2.5030]    [        0.1285]    [0.3178]    [1.1423]    [   0.3805]
    [    2.5773]    [        0.1278]    [0.3291]    [1.1363]    [   0.3934]
```

[2.6515]	[0.1271]	[0.3405]	[1.1301]	[0.4063]
[2.7258]	[0.1264]	[0.3518]	[1.1237]	[0.4190]
[2.8000]	[0.1256]	[0.3632]	[1.1171]	[0.4318]
[2.8742]	[0.1249]	[0.3745]	[1.1102]	[0.4444]
[2.9485]	[0.1241]	[0.3859]	[1.1032]	[0.4570]
[3.0227]	[0.1233]	[0.3972]	[1.0959]	[0.4694]
[3.0970]	[0.1224]	[0.4086]	[1.0884]	[0.4818]
[3.1712]	[0.1216]	[0.4199]	[1.0808]	[0.4941]
[3.2455]	[0.1207]	[0.4313]	[1.0729]	[0.5064]
[3.3197]	[0.1198]	[0.4426]	[1.0648]	[0.5185]
[3.3939]	[0.1188]	[0.4540]	[1.0565]	[0.5305]
[3.4682]	[0.1179]	[0.4653]	[1.0480]	[0.5425]
[3.5424]	[0.1169]	[0.4767]	[1.0392]	[0.5543]
[3.6167]	[0.1159]	[0.4880]	[1.0303]	[0.5661]
[3.6909]	[0.1148]	[0.4994]	[1.0211]	[0.5777]
[3.7652]	[0.1138]	[0.5107]	[1.0118]	[0.5892]
[3.8394]	[0.1127]	[0.5221]	[1.0022]	[0.6007]
[3.9136]	[0.1116]	[0.5334]	[0.9924]	[0.6120]
[3.9879]	[0.1105]	[0.5448]	[0.9824]	[0.6232]
[4.0621]	[0.1093]	[0.5561]	[0.9722]	[0.6343]
[4.1364]	[0.1082]	[0.5675]	[0.9618]	[0.6453]
[4.2106]	[0.1070]	[0.5788]	[0.9512]	[0.6561]
[4.2848]	[0.1058]	[0.5902]	[0.9404]	[0.6669]
[4.3591]	[0.1045]	[0.6015]	[0.9293]	[0.6775]
[4.4333]	[0.1033]	[0.6129]	[0.9181]	[0.6879]
[4.5076]	[0.1020]	[0.6242]	[0.9066]	[0.6983]
[4.5818]	[0.1007]	[0.6356]	[0.8949]	[0.7085]
[4.6561]	[0.0993]	[0.6469]	[0.8830]	[0.7186]
[4.7303]	[0.0980]	[0.6583]	[0.8710]	[0.7286]
[4.8045]	[0.0966]	[0.6696]	[0.8587]	[0.7384]
[4.8788]	[0.0952]	[0.6810]	[0.8461]	[0.7481]
[4.9530]	[0.0937]	[0.6923]	[0.8334]	[0.7576]
[5.0273]	[0.0923]	[0.7037]	[0.8205]	[0.7670]
[5.1015]	[0.0908]	[0.7150]	[0.8073]	[0.7762]
[5.1758]	[0.0893]	[0.7264]	[0.7940]	[0.7853]
[5.2500]	[0.0878]	[0.7377]	[0.7804]	[0.7942]
[5.3242]	[0.0862]	[0.7491]	[0.7666]	[0.8030]
[5.3985]	[0.0846]	[0.7604]	[0.7526]	[0.8116]
[5.4727]	[0.0831]	[0.7717]	[0.7384]	[0.8201]
[5.5470]	[0.0814]	[0.7831]	[0.7240]	[0.8284]
[5.6212]	[0.0798]	[0.7944]	[0.7094]	[0.8365]
[5.6955]	[0.0781]	[0.8058]	[0.6946]	[0.8445]
[5.7697]	[0.0764]	[0.8171]	[0.6795]	[0.8523]
[5.8439]	[0.0747]	[0.8285]	[0.6643]	[0.8599]
[5.9182]	[0.0730]	[0.8398]	[0.6488]	[0.8674]
[5.9924]	[0.0712]	[0.8512]	[0.6332]	[0.8746]
[6.0667]	[0.0694]	[0.8625]	[0.6173]	[0.8817]
[6.1409]	[0.0676]	[0.8739]	[0.6012]	[0.8887]

```
 [      6.2152]    [         0.0658]   [0.8852]   [0.5849]   [  0.8954]
 [      6.2894]    [         0.0639]   [0.8966]   [0.5684]   [  0.9019]
 [      6.3636]    [         0.0620]   [0.9079]   [0.5517]   [  0.9083]
 [      6.4379]    [         0.0601]   [0.9193]   [0.5347]   [  0.9145]
 [      6.5121]    [         0.0582]   [0.9306]   [0.5176]   [  0.9204]
 [      6.5864]    [         0.0563]   [0.9420]   [0.5002]   [  0.9262]
 [      6.6606]    [         0.0543]   [0.9533]   [0.4827]   [  0.9318]
 [      6.7348]    [         0.0523]   [0.9647]   [0.4649]   [  0.9372]
 [      6.8091]    [         0.0503]   [0.9760]   [0.4469]   [  0.9423]
 [      6.8833]    [         0.0482]   [0.9874]   [0.4287]   [  0.9473]
 [      6.9576]    [         0.0461]   [0.9987]   [0.4103]   [  0.9521]
```

```
In [ ]:
```