The Dead Oil Viscosity Correlations
A C-Sharp Simulation Approach

Hassan S. Naji

Petroleum Geology and Sedimentology Dept, Faculty of Earth Sciences, King Abdulaziz University, Jeddah, Saudi Arabia

email: hnaji@kau.edu.sa

Abstract. Over the years, different authors have proposed many dead oil viscosity correlations for various crude oil mixtures from all over the world. The majority of correlations fall into a category that uses system temperature and oil API gravity as the only required correlation parameters. This category may be designated as the least accurate and it might not successfully correlate viscosity since it ignores the *paraffinicity* or *character of the crude oil*. Another set of correlations offers increased accuracy in calculating dead oil viscosity. It requires, however, the Watson characterization factor as an extra parameter. This factor provides a means of identifying the *paraffinicity* or *character of the crude oil*. It stays reasonably constant for the chemically similar hydrocarbons.

The Watson factor and the boiling point are related together through the API gravity. In this paper, two regression equations, for evaluating the boiling point of hydrocarbons (pure components and crude oils) as a function of API gravity, are proposed. Therefore, the Watson factor will be evaluated as a function of boiling point; which, in turn will be evaluated as a function of API gravity. It has also been concluded from this study that the boiling point equation for pure components may be used for crude oils having API gravity greater than 30. Another aspect of the paper is that all correlations were coded in an object-oriented manner, which offers flexibility in programming and allows the different parts of the code to be described easily and in a natural manner as if they were real world objects. This allows for their simple inclusion in commercial black oil or compositional reservoir simulators. Eight correlations of the first category and two of the second category were considered for a sharp review from a simulation perspective. Eight experimentally-generated data sets were used to test all correlations and to quality check the validity of the proposed equations.

1. Introduction

Crude oil viscosity is an important physical property that controls and influences the flow of oil through porous media and pipes. It is defined as the internal resistance of the fluid to flow. During the course of black-oil or compositional reservoir simulation, the task of calculating crude oil viscosity is required at each pressure step. This task is accomplished by a three-stage process, which starts with the estimation of dead oil (gas-free) viscosity at atmospheric pressure and reservoir temperature. The dead oil viscosity is taken as input for the estimation of the gas-saturated crude oil viscosity; which, in turn, is taken as input for the estimation of the (undersaturated) crude oil viscosity at reservoir pressure. Figure 1 depicts the three viscosity areas. Viscosity starts at the dead viscosity condition; where it is evaluated at atmospheric pressure and system temperature. As the pressure increases, the free gas in contact with oil dissolves into the oil until all gas is dissolved. The pressure increases until it gets to the bubble point pressure, where no more gas can dissolve into the oil. This region represents the saturated region. If pressure increases above the bubblepoint pressure, no more gas can dissolve, and therefore the viscosity starts to increase due to the increase in pressure. This region represents the undersaturated region.
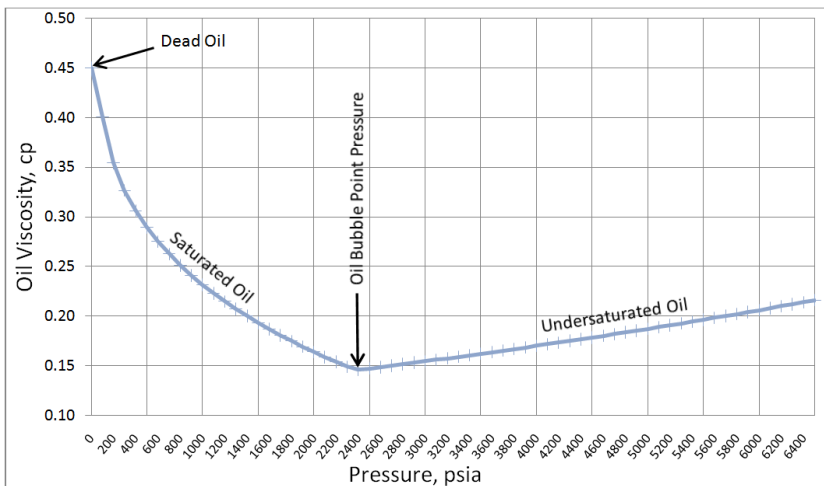

Fig. 1. Oil viscosity versus pressure showing the dead, saturated, and undersaturated regions.

Viscosity correlations that are applicable to a wide range of pressure and temperature conditions are needed. Unfortunately, such methods require composition of the hydrocarbon mixture. Correlations that do not require the compositions of hydrocarbon mixtures are applicable only to narrow ranges of operating conditions. Crude oil viscosity is affected by oil composition (specific gravity and characterization factor), solution gas/oil ratio, system temperature and pressure.

2. Dead Oil Viscosity Correlations

Many correlations were proposed to estimate viscosity of dead oils at atmospheric pressures and reservoir temperatures. Most common correlations require the dead oil API gravity and reservoir temperature as the only correlating parameters. Such correlations are least accurate and might not successfully correlate viscosity since they ignore the *paraffinicity* or *character of the crude oil*. Correlations, however, that utilize the Watson characterization factor for defining or characterizing the crude oil have been proposed. Such correlations have been mistakenly ignored due to their lengthy procedure. However, such correlations show stable viscosity predictions over a wide range of prevailing temperature.

In this paper, the following correlations that belong to the first category were considered: Beal, Beggs and Robinson, Dindoruk and Christman, Egbogah and Ng, Glaso, Kartoatmodjo and Schmidt, Al-Khafaji, and Petrosky [1-8]. In addition, the following two correlations, which belong to the second category, were considered: Twu and Bergman & Sutton [9-10]. Table 1 summarizes those correlations along with a C# snippet of their codes. Twu[9] has proposed a correlation that requires the dead oil API gravity, reservoir temperature, and Watson characterization factor. Bergman and Sutton[10] revised Twu[9] coefficients and proposed regression equations, to the kinematic viscosity at the reference temperatures of 100 °F and 210 °F, that enhance predicted dead oil viscosity values of the correlation. In this paper, two expressions are proposed for the estimation of the boiling point T_b in terms of API gravity for both pure components and crude oils. Since the Watson factor is related to the boiling point through API gravity, this will render the second category to only require API gravity and system temperature as the only correlation parameters.

Hassan S. Naji

Table 1. Dead oil viscosity correlations (μ_{od})

Author(s)	Correlation	C# Code Snippet	#

Beal[1]	$$\mu_{od} = \left(0.32 + \frac{1.8 \times 10^7}{API^{4.53}}\right)\left(\frac{360}{T+200}\right)^X$$ $$X = 10^{\left(0.43 + \frac{8.33}{API}\right)}$$	`internal class Beal : Model`
		`{`
		`    public Beal(double T, double API)`
		`    {`
		`        double X = System.Math.Pow(10, 0.43 + 8.33 / API);`
		`        this.Uod = (0.32 + 1.8e7 * System.Math.Pow(API, -4.53))`
		`* System.Math.Pow(360.0 / (T + 200.0), X);`
		`    }`
		`}`
			1

Beggs & Robinson[2]	$$\mu_{od} = 10^X - 1$$ $$X = 10^{(3.0324 - 0.02023\,API)}\,T^{-1.163}$$	`internal class Beggs : Model`
		`{`
		`    public Beggs(double T, double API)`
		`    {`
		`        double X = System.Math.Pow(10.0, 3.0324 - 0.02023 * API)`
		`* System.Math.Pow(T, -1.163);`
		`        this.Uod = System.Math.Pow(10.0, X) - 1.0;`
		`    }`
		`}`
			2

Bergman & Sutton[10]	$$\mu_{od} = \gamma_{oT}\,\nu_T$$ $$\gamma_{oT} = 0.999012\gamma_{o\,60}VCF_T$$ $$\gamma_{o\,60} = \gamma_o \text{ @ } 60°F$$ $$VCF_T = e^{[-\alpha_{60}\Delta T(1+0.8\alpha_{60}\Delta T)]}$$ $$\alpha_{60} = \frac{K_0 + K_1\gamma_{o\,60}}{\gamma_{o\,60}^2}$$ For crude oils:
	$K_0 = 2.5042 \times 10^{-4}$
	$K_1 = 8.3020 \times 10^{-5}$
	For pure components:
	$K_0 = 3.4175 \times 10^{-4}$
	$K_1 = -4.542 \times 10^{-5}$ $$\Delta T = T - 60$$ $$\nu_T = \psi - e^X$$
		`internal class BergmanCrude : Model`
		`{`
		`    public BergmanCrude(double T, double API)`
		`    {`
		`        double SGo = 141.5 / (131.5 + API);`
		`        double SGo60 = 0.999012 * SGo;`
		`        double[] K = { 2.5042e-4, 8.302e-5 };`
		`        double Alpha60 = (K[0] + K[1] * SGo60) / (SGo60 *`
		`SGo60);`
		`        double dT = T - 60.0;`
		`        double VCFT = System.Math.Exp(-Alpha60 * dT * (1.0 + 0.8`
		`* Alpha60 * dT));`
		`        double SGoT = SGo60 * VCFT;`
		`        // The Tb equation was obtained by fitting the Bergman-`
		`Sutton plot of the`
		`        // Watson factor versus API gravity.`
		`        //`
		`        // Tb = 540.39 - 1776.8 * SGo + 2744.7 * SGo * SGo`
		`        double Tb = 540.39 - 1776.8 * SGo + 2744.7 * SGo * SGo;`
			3

```csharp
double Taw = 0.533272 + 1.91017e-4 * Tb + 7.79681e-8 *
Tb * Tb
          - 2.84376e-11 * Tb * Tb * Tb + 9.59468e27 *
System.Math.Pow(Tb, -13);
double Alpha = 1.0 - Taw;
double SGo0 = 0.843593 - 0.128624 * Alpha - 3.36159 *
System.Math.Pow(Alpha, 3) - 13749.5 * System.Math.Pow(Alpha,
12);

double dSGo = SGo - SGo0;
double TbSq = System.Math.Sqrt(Tb);
double x = System.Math.Abs(2.68316 - 62.0863 / TbSq);
double f2 = x * dSGo - 47.6033 * dSGo * dSGo / TbSq;
double f1 = 0.980633 * x * dSGo - 47.6033 * dSGo * dSGo
/ TbSq;

double v2 = System.Math.Exp(2.40219 - 9.59688 * Alpha +
3.45656 * Alpha * Alpha -
143.632 * Alpha * Alpha * Alpha) + 0.152995;
double logv2 = System.Math.Log(v2);
double v1 = System.Math.Exp(0.701254 + 1.38359 * logv2 +
0.103604 * logv2 * logv2);

double TbT = 232.442 / Tb;
double v210 = System.Math.Exp(System.Math.Log(v2 + TbT)
          * System.Math.Pow((1.0 + 2.0 * f2) / (1.0 - 2.0 *
f2), 2)) - TbT;
double v100 = System.Math.Exp(System.Math.Log(v1 + TbT)
          * System.Math.Pow((1.0 + 2.0 * f1) / (1.0 - 2.0 *
f1), 2)) - TbT;

double Z210 = (v210 + 0.7) + System.Math.Exp(-1.47 +
1.84 * v210 - 0.51 * v210 * v210);
double Z100 = (v100 + 0.7) + System.Math.Exp(-1.47 +
1.84 * v100 - 0.51 * v100 * v100);

double Tabs = T + 459.67;
double B = (System.Math.Log(System.Math.Log(Z210)) -
System.Math.Log(System.Math.Log(Z100))) /
          (System.Math.Log(669.67) -
System.Math.Log(559.67));
double H = System.Math.Log(System.Math.Log(Z100)) + B *
(System.Math.Log(Tabs) - System.Math.Log(559.67));
```

$$\chi = -0.7487 - 3.295\psi + 0.6119\psi^2 - 0.3193\psi^3$$

$$\psi = (Z_T - 0.7)$$

$$Z_T = e^{\Gamma}$$

$$\Gamma = e^{H}$$

$$H = ln[ln(Z_{100})] + B[ln(T_{Abs}) - ln(559.67)]$$

$$B = \frac{ln[ln(Z_{210})] - ln[ln(Z_{100})]}{ln(669.67) - ln(559.67)}$$

$$T_{Abs} = T + 459.67$$

$$Z_{100} = (v_{100} + 0.7) + e^{[-1.47 - 1.84(v_{100}) - 0.51(v_{100})^2]}$$

$$Z_{210} = (v_{210} + 0.7) + e^{[-1.47 - 1.84(v_{210}) - 0.51(v_{210})^2]}$$

$$v_{100} = e^{ln\left(v_1 + \frac{232.442}{T_b}\right)\left(\frac{1+2f_1}{1-2f_1}\right)^2} - \frac{232.442}{T_b}$$

$$v_{210} = e^{ln\left(v_2 + \frac{232.442}{T_b}\right)\left(\frac{1+2f_2}{1-2f_2}\right)^2} - \frac{232.442}{T_b}$$

$$v_1 = e^{0.701254 + 1.38359\,ln(v_2) + 0.103604\,[ln(v_2)]^2}$$

$$v_2 = e^{2.40219 - 9.59688\,\alpha + 3.45656\alpha^2 - 143.632\alpha^4} + 0.152995$$

$$f_1 = 0.980633\,\xi\,\Delta\gamma_o - 47.6033\frac{(\Delta\gamma_o)^2}{\sqrt{T_b}}$$

$$f_2 = \xi\,\Delta\gamma_o - 47.6033\frac{(\Delta\gamma_o)^2}{\sqrt{T_b}}$$

```
double G = System.Math.Exp(H);
double ZT = System.Math.Exp(G);

double Psi = ZT - 0.7;
double Chi = -0.7487 - 3.295 * Psi + 0.6119 * Psi * Psi
- 0.3193 * Psi * Psi * Psi;
double vT = Psi - System.Math.Exp(Chi);

this.Uod = SGoT * vT;
}
}
```

$$\xi = \left| 2.68316 - \frac{62.0863}{\sqrt{T_b}} \right|$$

$$\Delta \gamma_o = \gamma_o - \gamma_o^0$$

$$\gamma_o^0 = 0.843593 - 0.128624\alpha - 3.36159\alpha^3 - 13749.5\alpha^{12}$$

$$\alpha = 1 - \tau$$

$$\tau = 0.533272 + 1.91017 \times 10^{-4} T_b + 7.79681 \times 10^{-8} T_b^2 - 2.84376 \times 10^{-11} T_b^3 + 9.59468 \times 10^{27} T_b^{-13}$$

$$T_b = 540.39 - 1776.8 \gamma_o + 2744.7 \gamma_o^2$$

Dindoruk & Christman[3]

$$\mu_{od} = \frac{a_3 T^{a_4} (\log(API))^A}{a_5 P_b^{a_6} + a_7 R_{sb}^{a_8}}$$

$$A = a_1 \log(T) + a_2$$

a_1	14.505357625	a_5	-3.1461171e-9
a_2	-44.868655416	a_6	1.517652716
a_3	9.36579e9	a_7	0.010433654
a_4	-4.194017808	a_8	-0.00077688

```
internal class DindorukChristman : Model
{
    private double[] a = { 14.505357625, -44.868655416,
9.36579e9, -4.194017808,        -3.1461171e-9, 1.517652716,
0.010433654, -0.000776880 };

    public DindorukChristman(double T, double SGg, double API,
double Pb)
    {
        GasSolubility.Model hRso = new
GasSolubility.DindorukChristman(T, SGg, API);

        double Rsb = hRso.GetRso(Pb);
        double A = this.a[0] * System.Math.Log10(T) + this.a[1];
        double N = this.a[2] * System.Math.Pow(T, this.a[3]) *
System.Math.Pow(System.Math.Log10(API), A);
        double D = this.a[4] * System.Math.Pow(Pb, this.a[5]) +
this.a[6] * System.Math.Pow(Rsb, this.a[7]);
        this.Uod = N / D;
    }
}
```

4

Egbogah & Ng[4]

$$\mu_{od} = 10^X - 1$$

$$X = 10^{1.8653 - 0.025086 API - 0.56441 \log(T)}$$

```
internal class Egbogah : Model
{
    public Egbogah(double T, double API)
    {
        double X = System.Math.Pow(10.0, 1.8653 - 0.025086 * API
```

5

	Code	Equation	
| Glaso[5] | ```cs
- 0.56441 * System.Math.Log10(T));
    this.Uod = System.Math.Pow(10.0, X) - 1.0;
  }
}

internal class Glaso : Model
{
  public Glaso(double T, double API)
  {
    double X = 10.313 * System.Math.Log10(T) - 36.447;
    this.Uod = 3.141e10 * System.Math.Pow(T, -3.444) *
System.Math.Pow(System.Math.Log10(API), X);
  }
}
``` | $\mu_{od} = 3.141 \times 10^{10} T^{-3.444} \left(log\left(API\right)\right)^{X}$ <br><br> $X = 10.313\, log(T) - 36.447$ | 6 |
| Kartoatmodjo & Schmidt[6] | ```cs
internal class KartoatmodjoSchmidt : Model
{
  public KartoatmodjoSchmidt(double T, double API)
  {
    double A = 1.6e9 * System.Math.Pow(T, -2.8177);
    double B = System.Math.Pow(System.Math.Log10(API),
5.7526 * System.Math.Log10(T) - 26.9718);
    this.Uod = A * B;
  }
}
``` | $\mu_{od} = A \times B$ <br><br> $A = 1.6 \times 10^{9}\, T^{-2.8177}$ <br><br> $B = [log(API)]^{5.7526\, log(T) - 26.9718}$ | 7 |
| Al-Khafaji[7] | ```cs
internal class Khafaji : Model
{
  public Khafaji(double T, double API)
  {
    double N = System.Math.Pow(10, 4.9563 - 0.00488 * T);
    double D = System.Math.Pow(API + T / 30.0 - 14.29,
2.709);
    this.Uod = N / D;
  }
}
``` | $\mu_{od} = \dfrac{N}{D}$ <br><br> $N = 10^{(4.9563 - 0.00488\, T)}$ <br><br> $D = \left(API + \dfrac{T}{30} - 14.29\right)^{2.709}$ | 8 |
| Petrosky[8] | ```cs
internal class Petrosky : Model
{
  public Petrosky(double T, double API)
  {
    double A = 2.3511e7 * System.Math.Pow(T, -2.10255);
    double B = System.Math.Pow(System.Math.Log10(API),
4.59388 * System.Math.Log10(T) - 22.82792);
    this.Uod = A * B;
  }
}
``` | $\mu_{od} = A \times B$ <br><br> $A = 2.3511 \times 10^{7} T^{-2.10255}$ <br><br> $B = [log(API)]^{(4.59388\, log(T) - 22.82792)}$ | 9 |

Twu[9]

$$\mu_{od} = \gamma_{oT}\, \nu_T$$

$$\gamma_{oT} = 0.999012\,\gamma_{o\,60}\,VCF_T$$

$$\gamma_{o\,60} = \gamma_o\ @\ 60\,°F$$

$$VCF_T = e^{[-\alpha_{60}\Delta T(1+0.8\alpha_{60}\Delta T)]}$$

$$\alpha_{60} = \frac{K_0 + K_1\,\gamma_{o\,60}}{\gamma_{o\,60}^2}$$

For crude oils:
$K_0 = 2.5042 \times 10^{-4}$
$K_1 = 8.3020 \times 10^{-5}$
For pure components:
$K_0 = 3.4175 \times 10^{-4}$
$K_1 = -4.542 \times 10^{-5}$

$$\Delta T = T - 60$$

$$\nu_T = \psi - e^\chi$$

$$\chi = -0.7487 - 3.295\psi + 0.6119\psi^2 - 0.3193\psi^3$$

$$\psi = (Z_T - 0.7)$$

$$Z_T = e^\Gamma$$

$$\Gamma = e^H$$

$$H = ln[ln(Z_{100})] + B\,[ln(T_{Abs}) - ln(559.67)]$$

$$B = \frac{ln[ln(Z_{210})] - ln[ln(Z_{100})]}{ln(669.67) - ln(559.67)}$$

$$T_{Abs} = T + 459.67$$

$$Z_{100} = (\nu_{100} + 0.7) + e^{[-1.47 - 1.84(\nu_{100}) - 0.51(\nu_{100})^2]}$$

10

```csharp
internal class TwuCrude : Model
{
    public TwuCrude(double T, double API)
    {
        double SGo = 141.5 / (131.5 + API);
        double SGo60 = 0.999012 * SGo;
        double[] K = { 2.5042e-4, 8.302e-5 };
        double Alpha60 = (K[0] + K[1] * SGo60) / (SGo60 *
SGo60);

        double dT = T - 60.0;
        double VCFT = System.Math.Exp(-Alpha60 * dT * (1.0 + 0.8
* Alpha60 * dT));
        double SGoT = SGo60 * VCFT;

        // The Tb equation was obtained by fitting the Bergman-
Sutton plot of the
        // Watson factor versus API gravity.
        //
        // Tb = 540.39 - 1776.8 * SGo + 2744.7 * SGo * SGo
        //
        double Tb = 540.39 - 1776.8 * SGo + 2744.7 * SGo * SGo;
        double Taw = 0.533272 + 1.91017e-4 * Tb + 7.79681e-8 *
Tb * Tb
            - 2.84376e-11 * Tb * Tb * Tb + 9.59468e27 *
System.Math.Pow(Tb, -13);
        double Alpha = 1.0 - Taw;
        double SGo0 = 0.843593 - 0.128624 * Alpha - 3.36159 *
System.Math.Pow(Alpha, 3) - 13749.5 * System.Math.Pow(Alpha,
12);
        double dSGo = SGo - SGo0;
        double TbSq = System.Math.Sqrt(Tb);
        double x = System.Math.Abs(1.99873 - 56.7394 / TbSq);
        double f2 = x * dSGo - 21.1141 * dSGo * dSGo / TbSq;
        double f1 = 1.33932 * x * dSGo - 21.1141 * dSGo * dSGo /
TbSq;
        double v2 = System.Math.Exp(4.73227 - 27.0975 * Alpha +
49.4491 * Alpha * Alpha -
            50.4706 * Alpha * Alpha * Alpha * Alpha) - 1.5;
        double logv2 = System.Math.Log(v2);
```

```
            double v1 = System.Math.Exp(0.801621 + 1.37179 * logv2);

            double TbT = 450.0 / Tb;
            double v210 = System.Math.Exp(System.Math.Log(v2 + TbT)
                * System.Math.Pow((1.0 + 2.0 * f2) / (1.0 - 2.0 *
f2), 2)) - TbT;
            double v100 = System.Math.Exp(System.Math.Log(v1 + TbT)
                * System.Math.Pow((1.0 + 2.0 * f1) / (1.0 - 2.0 *
f1), 2)) - TbT;

            double Z210 = (v210 + 0.7) + System.Math.Exp(-1.47 -
1.84 * v210 - 0.51 * v210);
            double Z100 = (v100 + 0.7) + System.Math.Exp(-1.47 -
1.84 * v100 - 0.51 * v100);

            double Tabs = T + 459.67;
            double B = (System.Math.Log(System.Math.Log(Z210))) -
System.Math.Log(System.Math.Log(Z100))) /
(System.Math.Log(669.67) -
System.Math.Log(559.67));
            double H = System.Math.Log(System.Math.Log(Z100)) + B *
(System.Math.Log(Tabs) - System.Math.Log(559.67));
            double G = System.Math.Exp(H);
            double ZT = System.Math.Exp(G);

            double Psi = ZT - 0.7;
            double Chi = -0.7487 - 3.295 * Psi + 0.6119 * Psi * Psi
- 0.3193 * Psi * Psi * Psi;
            double vT = Psi - System.Math.Exp(Chi);

            this.Uod = SGoT * vT;

        }
    }
```

$$Z_{210} = (v_{210} + 0.7) + e^{\left[-1.47 - 1.84(v_{210}) - 0.51(v_{210})^2\right]}$$

$$v_{100} = e^{\ln\left(v_1 + \frac{450}{T_b}\right)\left(\frac{1+2f_1}{1-2f_1}\right)^2} - \frac{450}{T_b}$$

$$v_{210} = e^{\ln\left(v_2 + \frac{450}{T_b}\right)\left(\frac{1+2f_2}{1-2f_2}\right)^2} - \frac{450}{T_b}$$

$$v_1 = e^{0.801621 + 1.37179\,\ln(v_2)}$$

$$v_2 = e^{4.73227 - 27.0975\,\alpha + 49.4491\,\alpha^2 - 50.4706\,\alpha^4} + 1.5$$

$$f_1 = 1.39932\,\xi\,\Delta\gamma_o - 21.1141\,\frac{(\Delta\gamma_o)^2}{\sqrt{T_b}}$$

$$f_2 = \xi\,\Delta\gamma_o - 21.1141\,\frac{(\Delta\gamma_o)^2}{\sqrt{T_b}}$$

$$\xi = \left| 1.99873 - \frac{56.7394}{\sqrt{T_b}} \right|$$

$$\Delta\gamma_o = \gamma_o - \gamma_o^0$$

$$\gamma_o^0 = 0.843593 - 0.128624\,\alpha - 3.36159\,\alpha^3 - 13749.5\,\alpha^{12}$$

$$\alpha = 1 - \tau$$

$$\tau = 0.533272 + 1.91017 \times 10^{-4}\,T_b + 7.79681 \times 10^{-9}\,T_b^2 - 2.84376 \times 10^{-11}\,T_b^3 + 9.59468 \times 10^{27}\,T_b^{-13}$$

$$T_b = 540.39 - 1776.8\,\gamma_o + 2744.7\,\gamma_o^2$$

3. Characterizing Petroleum Fractions

The physical properties of pure components were measured and compiled over the years. Properties include specific gravity, normal boiling point, molecular weight, critical properties and acentric factor. Katz and Firoozabadi[11] presented a generalized set of properties for pure components with carbon number in the range 6-45. Whitson[12] modified this set to make its use more consistent. Riazi and Daubert[13] developed a set of equations to evaluate properties of pure and undefined petroleum fractions. They expressed the molecular weight of a pure or undefined petroleum fraction (MW) in terms of specific gravity (γ) and boiling point (T_b) as follows:

$$MW = 42.965\kappa^{1.26007}\gamma^{4.98308}$$
$$e^{[2.097x10^{-4}\kappa - 7.78712\gamma + 2.08476x10^{-3}\gamma\kappa]}$$

(1)

Where $\kappa = \frac{T_b}{1.8}$. Kesler and Lee[14] developed a set of equations to evaluate properties of pure and undefined petroleum fractions. They expressed the molecular weight of a pure or undefined petroleum fraction in terms of specific gravity (γ) and boiling point (T_b) as follows:

$$MW = -12,272.6 + 9,486.4\gamma + (8.3741 - 5.9917\gamma)\kappa$$
$$+(1 - 0.77084\gamma - 0.02058\gamma^2)\left[0.7465 - \frac{222.466}{\kappa}\right]\frac{10^7}{\kappa}$$
$$+(1 - 0.80882\gamma + 0.02226\gamma^2)\left[0.3228 - \frac{17.335}{\kappa}\right]\frac{10^{12}}{\kappa^3}$$

(2)

Søreide[15] expressed the boiling point (T_b) of a pure or undefined petroleum fraction in terms of specific gravity (γ) and molecular weight (MW) as follows:

$$\kappa = 1071.28 - 9.417x10^4 MW^{-0.03522}\gamma^{3.266}$$
$$e^{(-4.922x10^{-3}MW - 4.7685\gamma + 3.462x10^{-3}MW\gamma)}$$

(3)

Twu[16] used the critical properties back-calculated from vapor pressure data to get correlations for the undefined petroleum fractions. The molecular weight is estimated as follows:

$$MW = e^{\beta\left[\frac{(1+2f_M)}{(1-2f_M)}\right]^2}$$

(4)

Where β is obtained by solving the following objective function:

$$f(\beta) = e^{5.12640+2.71579\beta-0.286590\beta^2-\frac{39.8544}{\beta}+\frac{0.122488}{\beta^2}} \tag{5}$$

$$-13.7512\beta + 19.6197\beta^2 - \kappa = 0$$

Other parameters of the equation are given by:

$$f_M = \Delta\gamma_M \left[\chi + \Delta\gamma_M \left(\frac{0.143979}{\kappa^{0.5}} - 0.0175691\right)\right] \tag{6}$$

$$\chi = \left|0.012342 - \frac{0.244541}{\kappa^{0.5}}\right| \tag{7}$$

$$\Delta\gamma_M = e^{5(\gamma^0-\gamma)} - 1 \tag{8}$$

$$\gamma^0 = 0.843593 - 0.128624\alpha - 3.36159\alpha^3 - 13749.5\alpha^{12} \tag{9}$$

$$\alpha = 1 - \left(\begin{array}{c}0.533272 + 0.34383 \times 10^{-3}\kappa + 2.52617 \times 10^{-7}\kappa^2 \\ -1.658481 \times 10^{-10}\kappa^3 + 4.60773 \times 10^{24}\kappa^{-13}\end{array}\right) \tag{10}$$

Naji[17] has presented a set of regression equations for the estimation of physical properties of pure components as a function of carbon number. Those equations were built upon Katz and Firoozabadi[11] dataset. From a simulation perspective, the fit data is more consistent than the original data. The regression equations, for the specific gravity, boiling point, and molecular weight, are given by:

$$\gamma(n) = 0.6839638(n-5)^{0.08661026} \tag{11}$$

$$T_b(n) = -2.238720 \times 10^{-4}(n-5)^4 + 2.916260 \times 10^{-2}(n-5)^3 \\ -1.593227(n-5)^2 + 54.72655(n-5) + 553.347 \tag{12}$$

$$MW(n) = -5.763156 \times 10^{-6}(n-5)^5 + 7.293105 \times 10^{-4}(n-5)^4 \\ -0.03341596(n-5)^3 + 0.5740517(n-5)^2 \\ +10.24725(n-5) + 72.53757 \tag{13}$$

4. Characterizing Crude Oils

As mentioned before, the dead oil viscosity correlations that offer increased accuracy in calculating viscosity require the Watson characterization factor as an extra parameter. This factor provides a means of identifying the *paraffinicity* or *character of the crude oil*. It stays reasonably constant for the chemically similar hydrocarbons; *i.e.*

higher values (greater than 11.9) indicate paraffinic hydrocarbons, whereas lower values indicate naphthenic or aromatic components. Values less than 10 indicate highly aromatic hydrocarbons. Bergman & Sutton[10] used nonlinear regression techniques to update the correlation for the Watson characterization factor proposed by Riazi-Daubert[13] and later updated by Riazi[18]. They suggested the use of the following updated equation to calculate the Watson characterization factor, K_W:

$$K_W = \left[b_1 e^{(b_2 M_o + b_3 \gamma_o + b_4 M_o \gamma_o)} M_o^{b_5} \gamma_o^{b_6} \right]^{b_7} \gamma_o^{b_8} \tag{14}$$

Where:

$b_1 = 2012.84$	$b_2 = -1.8519 \times 10^{-3}$	$b_3 = -3.70833$	$b_4 = 1.31441 \times 10^{-3}$
$b_5 = 0.589485$	$b_6 = 3.36211$	$b_7 = 0.3333$	$b_8 = -1.0$

The molecular weight of the crude oil (M_o) is often measured for the oil itself or derived from the plus fraction values reported in the PVT report of the crude oil. If the chemical nature of the crude oil is known, it is defined as a function of API gravity. If the chemical nature of the crude oil is unknown, however, Bergman & Sutton[10] presented a plot of Watson factor versus crude oil API gravity (Fig. 2). This plot has originated from a database compiled from all over the world for over 3000 crude oil assays, conventional PVT reports, and literature data. The plot should provide guidance for typical values of Watson factors as a function of API gravity. For simulation purposes, the linear fitting equation of this plot suffices; which is given by:

$$K_W = 11.2 + 0.0192164(\text{API}) \tag{15}$$

The Watson factor is related to the average boiling point and API gravity of the crude oil through the following relation:

$$K_W = \left(\frac{131.5 + \text{API}}{141.5} \right) (T_b)^{\frac{1}{3}} \tag{16}$$

Plugging (16) into (15) and rearranging, yields the average boiling point expressed in terms of the API gravity of the crude oil:

$$T_b = \left[\frac{1584.8 + 2.7191206 \text{API}}{131.5 + \text{API}} \right]^3 \tag{17}$$

Alternatively, equation (17) may be expressed as a forth-order polynomial as follows:

$$T_b = 1748 - 30.05\text{API} + 0.3451\text{API}^2 - 0.002416\text{API}^3 + 7.397 \times 10^{-6}\text{API}^4 \tag{18}$$
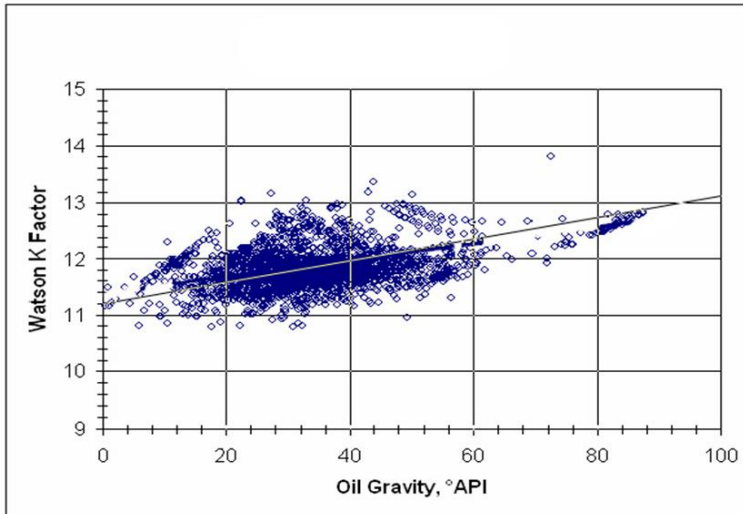
Fig. 2. Plot of Watson characterization factor versus crude oil API gravity. (Bergman & Sutton[10]).

Figure 3 is a plot of average boiling point (T_b) versus (API) for both pure components and crude oils. It is clear from the plot, that the T_b curves, for pure components and crude oils, tend to converge for API gravities higher than 30. On the other side, for API gravities lower than 30; *i.e.* for heavy fractions, the T_b curve for pure components gets higher than the T_b curve for crude oils as API gravities decrease. This suggests that T_b equation for pure components may be suitable for crude oils having API gravities higher than 30.
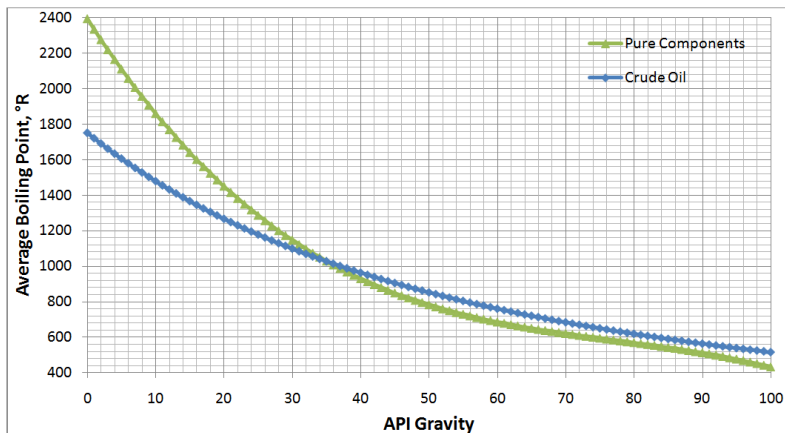
Fig. 3. Plot of average boiling point versus API gravity for pure components and crude oils.

5. Methodology

In the category of correlations utilizing the Watson characterization factor, we follow the procedure of Twu[9] for the estimation of dead oil viscosity; which can be summarized as follows:

$$\mu_{od} = \gamma_{oT}\ \nu_T \tag{19}$$

γ_{oT} is the crude oil specific gravity as affected by the temperature T and is given by:

$$\gamma_{oT} = 0.999012\gamma_{o\ 60}VCF_T \tag{20}$$

$\gamma_{o\ 60}$ is the crude oil specific gravity (γ_o) at 60 °F.

VCF_T is the crude oil *volume correction factor* with a base temperature of 60 °F:

$$VCF_T = e^{[-\alpha_{60}\Delta T(1+0.8\alpha_{60}\Delta T)]} \tag{21}$$

α_{60} is the thermal expansion coefficient with a base temperature of 60 °F:

$$\alpha_{60} = \frac{K_0+K_1\ \gamma_{o\ 60}}{\gamma_{o\ 60}^2} \tag{22}$$

The coefficients K_0 and K_1 are given for each liquid of interest. ASTM[20] provides values for both pure components and generalized crude oils; which were updated by Bergman & Sutton[10] as:

For pure components:

$$K_0 = 3.4175 \times 10^{-4} \tag{23}$$

$$K_1 = -4.542 \times 10^{-5} \tag{24}$$

For generalized crude oils:

$$K_0 = 2.5042 \times 10^{-4} \tag{25}$$

$$K_1 = 8.3020 \times 10^{-5} \tag{26}$$

$\Delta T = T - 60$ is the temperature difference between base and current temperatures.

v_T is the crude oil kinematic viscosity as affected by the temperature T. It is evaluated as follows:

$$v_T = \psi - e^{\chi} \tag{27}$$

Other parameters are given by:

$$\chi = -0.7487 - 3.295\psi + 0.6119\psi^2 - 0.3193\psi^3 \tag{28}$$

$$\psi = (Z_T - 0.7) \tag{29}$$

$$Z_T = e^{\Gamma} \tag{30}$$

$$\Gamma = e^{H} \tag{31}$$

$$H = ln[ln(Z_{100})] + B\,[ln(T_{Abs}) - ln(559.67)] \tag{32}$$

$$B = \frac{ln[ln(Z_{210})] - ln[ln(Z_{100})]}{ln(669.67) - ln(559.67)} \tag{33}$$

$$T_{Abs} = T + 459.67 \tag{34}$$

$$Z_{100} = (v_{100} + 0.7) + e^{\left[-1.47 - 1.84(v_{100}) - 0.51(v_{100})^2\right]} \tag{35}$$

$$Z_{210} = (v_{210} + 0.7) + e^{\left[-1.47 - 1.84(v_{210}) - 0.51(v_{210})^2\right]} \tag{36}$$

Twu[9] has adopted the following expressions for calculating kinematic viscosities of the crude oil at two reference temperatures of 100 °F and 210 °F:

$$v_{100} = e^{ln\left(v_1 + \frac{450}{T_b}\right)\left(\frac{1+2f_1}{1-2f_1}\right)^2} - \frac{450}{T_b} \tag{37}$$

$$v_{210} = e^{ln\left(v_2 + \frac{450}{T_b}\right)\left(\frac{1+2f_2}{1-2f_2}\right)^2} - \frac{450}{T_b} \tag{38}$$

Where:

$$v_1 = e^{0.801621 + 1.37179\,ln(v_2)} \tag{39}$$

$$v_2 = e^{4.73227 - 27.0975\,\alpha + 49.4491\alpha^2 - 50.4706\alpha^4} + 1.5 \tag{40}$$

$$f_1 = 1.33932\,\xi\,\Delta\gamma_o - 21.1141\frac{(\Delta\gamma_o)^2}{\sqrt{T_b}} \tag{41}$$

$$f_2 = \xi \, \Delta\gamma_o - 21.1141 \frac{(\Delta\gamma_o)^2}{\sqrt{T_b}} \tag{42}$$

$$\xi = \left| 1.99873 - \frac{56.7394}{\sqrt{T_b}} \right| \tag{43}$$

Using nonlinear regression techniques on various forms from Twu[9], Bergman & Sutton[10] have updated those expressions to minimize the error in calculated dead oil viscosity. The final resulting equations are given by:

$$v_{100} = e^{ln\left(v_1 + \frac{232.442}{T_b}\right)\left(\frac{1+2f_1}{1-2f_1}\right)^2} - \frac{232.442}{T_b} \tag{44}$$

$$v_{210} = e^{ln\left(v_2 + \frac{232.442}{T_b}\right)\left(\frac{1+2f_2}{1-2f_2}\right)^2} - \frac{232.442}{T_b} \tag{45}$$

Where:

$$v_1 = e^{0.701254 + 1.38359 \, ln(v_2) + 0.103604[ln(v_2)]^2} \tag{46}$$

$$v_2 = e^{2.40219 - 9.59688 \, \alpha + 3.45656\alpha^2 - 143.632\alpha^4} + 0.152995 \tag{47}$$

$$f_1 = 0.980633 \, \xi \, \Delta\gamma_o - 47.6033 \frac{(\Delta\gamma_o)^2}{\sqrt{T_b}} \tag{48}$$

$$f_2 = \xi \, \Delta\gamma_o - 47.6033 \frac{(\Delta\gamma_o)^2}{\sqrt{T_b}} \tag{49}$$

$$\xi = \left| 2.68316 - \frac{62.0863}{\sqrt{T_b}} \right| \tag{50}$$

The rest of parameters are given by:

$$\Delta\gamma_o = \gamma_o - \gamma_o^0 \tag{51}$$

$$\gamma_o^0 = 0.843593 - 0.128624\alpha - 3.36159\alpha^3 - 13749.5\alpha^{12} \tag{52}$$

$$\alpha = 1 - \tau \tag{53}$$

$$\tau = 0.533272 + 1.91017 \times 10^{-4} T_b + 7.79681 \times 10^{-8} T_b^2$$
$$-2.84376 \times 10^{-11} T_b^3 + 9.59468 \times 10^{27} T_b^{-13} \tag{54}$$

In this paper, the following expressions are used to calculate the boiling point (T_b) in terms of API gravity:

For pure components (and hence for crude oils having API gravities > 30):

$$T_b = 2393.806 - 60.19172API + 0.7137592API^2 - 3.078513 \times 10^{-3}API^3 \quad (55)$$

For crude oils, we use equation 17; which is rewritten for convenience:

$$T_b = \left[\frac{1584.8 + 2.7191206API}{131.5 + API}\right]^3 \quad (56)$$

Or as a forth-order polynomial as expressed by equation 18 as follows:

$$T_b = 1748 - 30.05API + 0.3451API^2 - 0.002416API^3 + 7.397 \times 10^{-6}API^4 \quad (57)$$

6. Input Data

Table 2 presents the experimental input data of two oil samples that were presented by Dindoruk & Christman[3]. Table 3 presents the experimental input data of six oil samples that were presented by Ahmed[19]. The eight samples were used to generate the dead oil viscosities for all correlations considered in this paper.

Table 2: Experimental input data used for generating the dead oil viscosity for the various correlations .

Parameter	Oil A	Oil B
$T, °F$	160	230
$\gamma_g, (air = 1)$	0.7310	0.7072
$°API$	27.4	33.3
$P_b, psia$	5000	5000
$R_{sb}, scf/STB$	813	1049
$T_{sep}, °F$	68	60
$p_{sep}, psia$	114.7	114.7

Source: Dindoruk and Christman[3] .

Table 3: Experimental input data used for generating the dead oil viscosity for the various correlation .

Parameter	Oil #1	Oil #2	Oil #3	Oil #4	Oil #5	Oil #6
$T, °F$	250	220	260	237	218	180
$\gamma_g, (air = 1)$	0.851	0.855	0.911	0.898	0.781	0.848
$°API$	47.1	40.7	48.6	40.5	44.2	27.3
$P_b, psia$	2377	2620	2051	2884	3065	4239
$R_{sb}, scf/STB$	751	768	693	968	943	807
$T_{sep}, °F$	60	75	72	120	60	173
$p_{sep}, psia$	150	100	100	60	200	85

Source: Ahmed[19], chapter 2, pp. 79-80.

7. Results and Discussions

Table 1 summarizes the describing equations and C# codes of all dead oil viscosity correlations presented in this study. As it can be seen from the C# code snippets that all classes are derived from a single parent class called *"Model"*. This behavior guarantees that all correlations refer to and rely on the same procedure and data set. Inside each class, there exists a class constructor, having the same name as the class. The constructor performs all calculations at the construction stage of the class. This process has an excellent impact on saving calculation timing. In addition, this allows for simple and straight forward implementation of such correlations in predefined reservoir simulators.

Figure 1 shows the three oil viscosity regions as a function of pressure. The dead oil viscosity calculations are performed at surface pressures. Comes next is the saturated oil viscosity region, which extends to the bubblepoint pressure of the crude oil. It requires the dead oil viscosity as input. Above bubblepoint pressure, comes the undersaturated oil viscosity region; which depends on the saturated values as input. Figure 2, which is taken from Bergman & Sutton[10], has originated from a database compiled from all over the world for over 3000 crude oil assays, conventional PVT reports, and literature data. The plot provides guidance for typical values of Watson factors as a function of API gravity. For simulation purposes, the linear fitting equation of this plot suffices, which is given by equation 15. Figure 3 is a plot of average boiling point (T_b) versus API for both pure components and crude oils. It is clear from the plot, that the T_b curves, for pure components and crude oils, tend to converge for API gravities higher than 30. On the other hand, for API gravities lower than 30; *i.e.* for heavy fractions, the T_b curve for pure components gets higher than the T_b curve for crude oils as API gravities decrease. This may suggest that T_b equation for pure components is suitable for crude oils having API gravities higher than 30.

Figures 4-11 are plots of the dead oil viscosity versus temperature for all correlations considered in this study. From the first glance of all figures, it is apparent that as the temperature increases, the dead oil viscosity decreases and tends to converge for all correlations. As the temperature decreases, however, the dead viscosity values tend to diverge for the different correlations. Note, however, how the second category of correlations, namely Twu[9] and Bergman & Sutton[10], are calculated.

They are calculated in exactly the same way as the first category; *i.e.* as a function of temperature and oil API gravity. The Watson characterization factor is implicitly calculated as a function of oil API gravity. Also note that these correlations tend to predict values within the expected range of dead viscosities. Beggs[2] correlation tends to predict high dead oil viscosity values at lower temperatures for all studied samples. Bergman & Sutton[10] and Twu[9] correlations for crude oils tend to predict good estimates for all samples considered in this study. For pure components, however, they predict high values for heavy crude oils (samples A and 6: their API < 30). Whereas they tend to predict good estimates for other samples (samples B & 1 to 5: their API > 30).

8. Conclusions

1.   Two categories of the dead oil viscosity correlations were revised and programmed in an object-oriented fashion.

2.   The first category, which requires only oil API gravity and system temperature, neglects the oil character (paraffinic, naphthenic, or aromatic). This negligence may render the category as least accurate.

3.   The second category accounts for crude oil character and tends to give more accurate viscosity predictions. It requires the Watson characterization factor as an extra parameter. This factor is related to the oil API gravity and boiling point.

4.   Two regression equations were used to express the oil boiling point, for both pure components and crude oils, as a function of API gravity. The crude oil regression equation has stemmed from a database compiled from all over the world for over 3000 crude oil assays, conventional PVT reports, and literature data.

5.   The pure components regression equation may be used for crude oils having API gravities higher than 30. Eight experimentally-generated data sets were used to test all viscosity correlations.
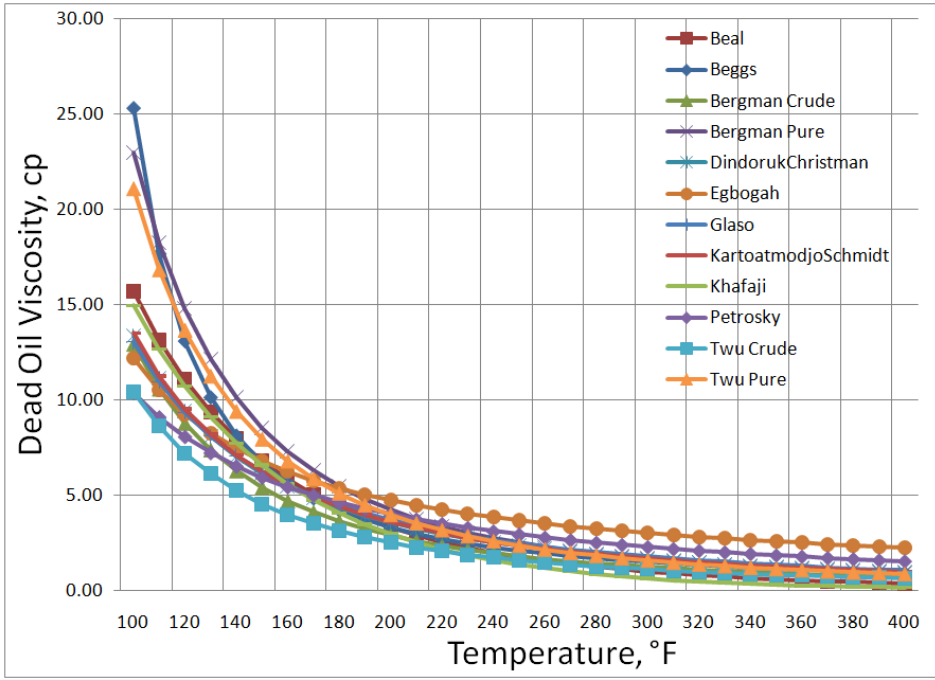
Fig. 4. Plot of dead oil viscosity versus temperature for the various correlations (Dindoruk and Christman[3], Oil A).
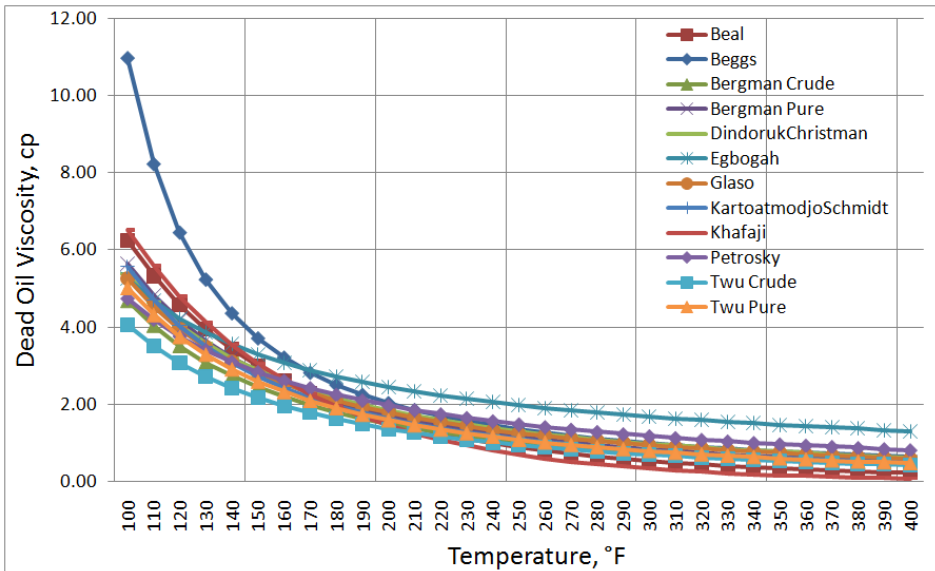

Fig. 5. Plot of dead oil viscosity versus temperature for the various correlations (Dindoruk and Christman[3], Oil B).
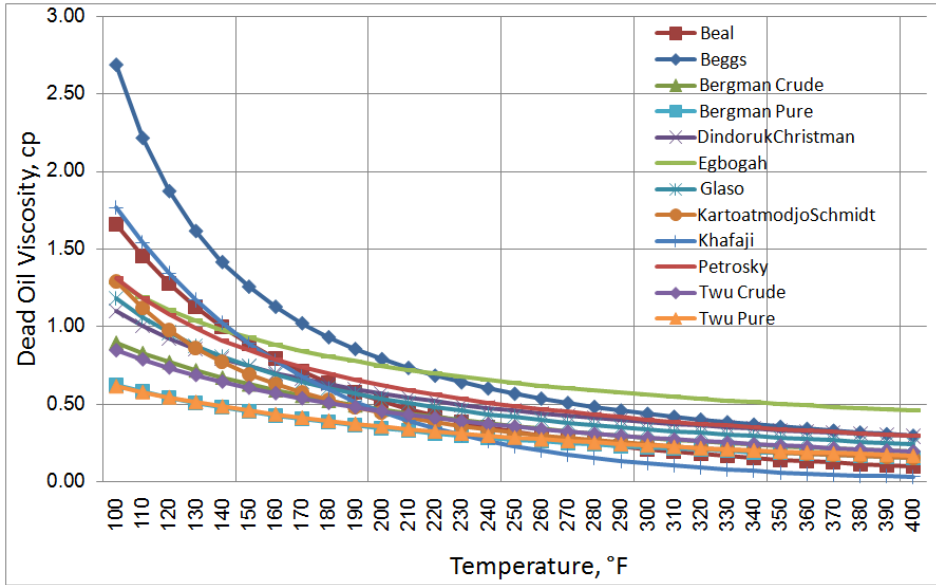
Fig. 6. Plot of dead oil viscosity versus temperature for the various correlations (Ahmed[19], Oil #1).
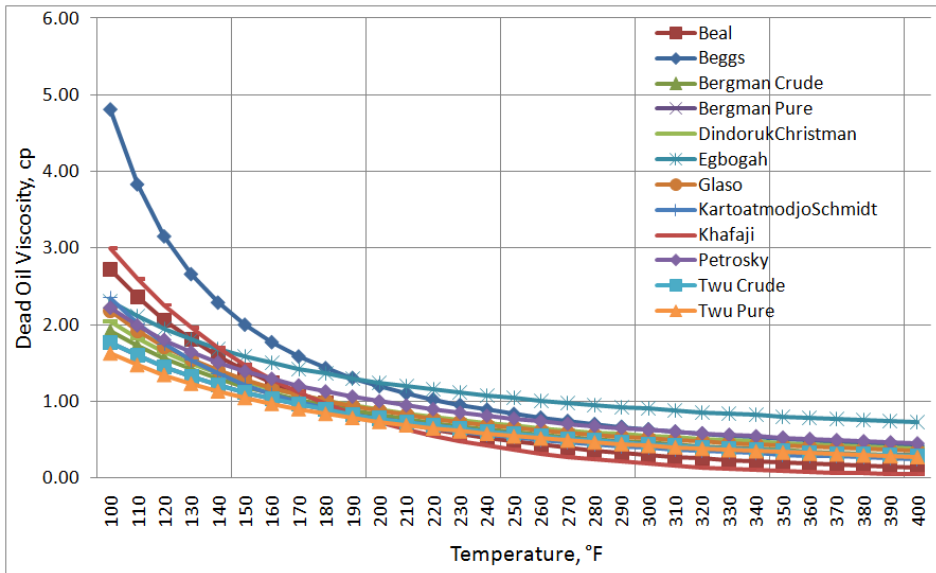

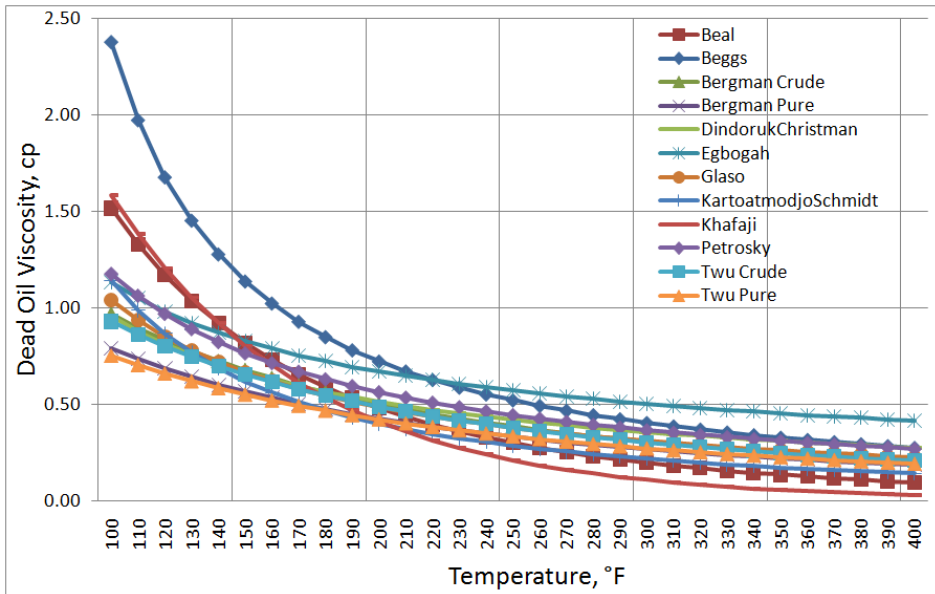**Fig. 7. Plot of dead oil viscosity versus temperature for the various correlations (Ahmed[19], Oil #2).**

Fig. 8. Plot of dead oil viscosity versus temperature for the various correlations (Ahmed[19], Oil #3).
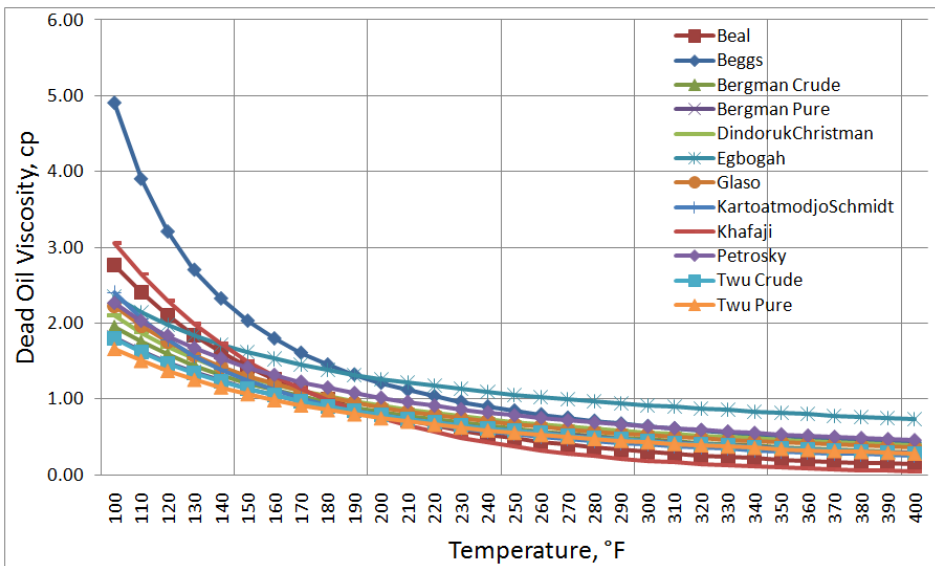

Fig. 9. Plot of dead oil viscosity versus temperature for the various correlations (Ahmed[19], Oil #4).
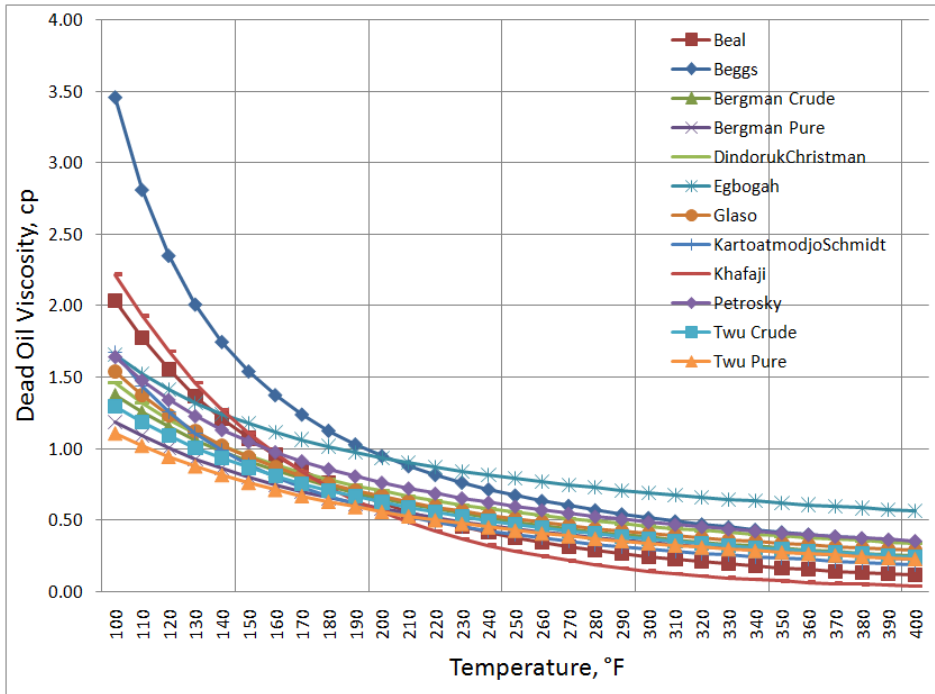
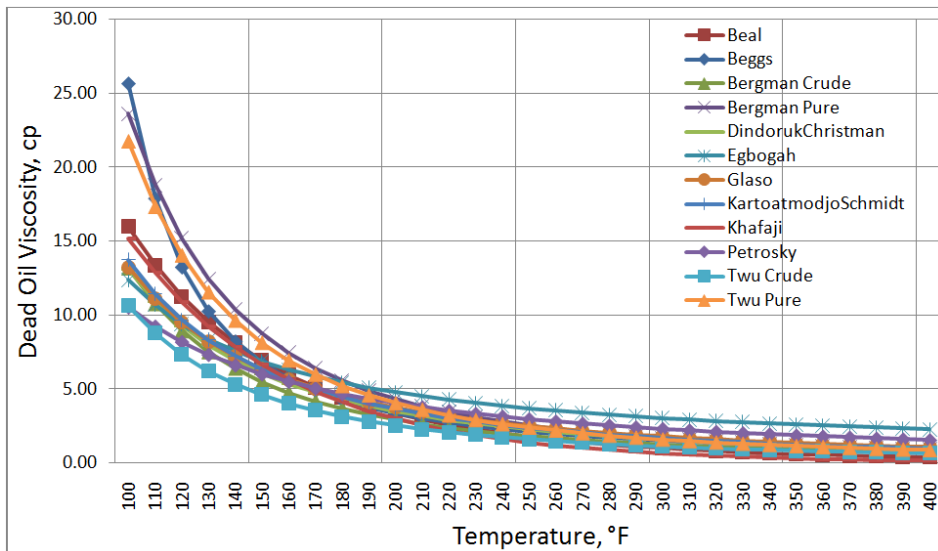**Fig. 10. Plot of dead oil viscosity versus temperature for the various correlations (Ahmed[19], Oil #5).**

Fig. 11. Plot of dead oil viscosity versus temperature for the various correlations (Ahmed[19], Oil #6).

Nomenclature

API	= oil specific gravity in API units, °API
B_o	= oil formation volume factor, bbl/STB
P	= current system pressure, psia
P_b	= oil bubble point pressure, psia
P_{sep}	= separator pressure, psia
R_{sb}	= solution gas/oil ratio at the bubble-point pressure, SCF/STB
R_{so}	= solution gas/oil ratio at current pressure P, SCF/STB
T	= system temperature, °F
T_{sep}	= separator temperature, °F
γ_o	= oil specific gravity
γ_g	= gas specific gravity
ρ_o	= oil density, lbs
ρ_g	= gas density, lbs
μ_o	= undersaturated oil viscosity, cp
μ_{ob}	= saturated oil viscosity, cp
μ_{od}	= dead oil viscosity, cp

References

[1] **Beal, C.,** The Viscosity of Air, Water, Natural Gas, Crude Oils and its Associated Gases at Oil Field Temperatures and Pressures, *Trans. AIME*, **165**: 94-112 (1946).

[2] **Beggs, H.D.** and **Robinson, J.R.,** Estimating the Viscosity of Crude Oil Systems, *JPT*, September: 1140-1141(1975).

[3] **Dindoruk, B.** and **Christman, P.G.,** PVT Properties and Viscosity Correlations for Gulf of Mexico Oils, *December 2004 SPE Reservoir Evaluation & Engineering*, paper SPE 89030.

[4] **Egbogah, E.O.** and **Ng, J.T.,** An Improved Temperature-Viscosity Correlation for Crude Oil Systems, *J. Pet. Sci. Eng.:* 197-200, July (1990).

[5] **Glasø, O.,** Generalized Pressure-Volume-Temperature Correlations, *JPT:* 785-795, May (1980).

[6] **Kartoatmodjo, T.** and **Schmidt, Z.,** Large Data Bank Improves Crude Physical Property Correlations, *Oil and Gas J.,*: 51-55, July 4 (1994).

[7] **Al-Khafaji, A.H., Abdul-Majeed, G.H.** and **Hassoon, S. F.,** Viscosity Correlation For Dead, Live, and Undersaturated Crude Oils, *J. Pet. Res.*:1-16, Dec. (1987).

[8] **Petrosky, G.E. Jr.,** PVT Correlations for Gulf of Mexico Crude Oils, *M.S. Thesis*, University of Southwestern Louisiana (1990).

[9] **Twu, C.H.,** Internally Consistent Correlation for Predicting Liquid Viscosities of Petroleum Fractions, *Ind. Eng. Chem. Process Des. Dev.* **34** (4)**:** 1287-1293(1985).

[10] **Bergman, D.F.** and **Sutton, R.P.,** A Consistent and Accurate Dead-Oil-Viscosity Method, paper SPE-110194 , *The 2007 SPE Annual Technical Conference and Exhibition held in Anaheim, California, USA, 11-14 November 2007*.

[11] **Katz, D.L.** and **Firoozabadi, A.,** Predicting phase behavior of condensate/crude oil systems using methane interaction coefficients, *JPT*:1649-1655(1978).

[12] **Whitson, C.H.,** Characterizing hydrocarbon plus fractions, *SPEJ* , **23**: 683-694(1983).

[13] **Riazi, M. R.** and **Daubert, T. E.,** Characterizing Parameters for Petroleum Fractions, Ind. *Eng. Chem. Res.,* **26**(24): 755-759(1987).

[14] **Kesler, M. G.** and **Lee. B. I.,** Improved Prediction of Enthalpy of Fractions, *Hydrocarbon Processing*, March: 153-158(1976).

[15] **Søreide, I.,** Improved Phase Behavior Predictions of Petroleum Reservoir Fluids from a Cubic Equation of State, *Dr. Ing. Dissertation*, Norwegian Institute of Technology (NTH), Trondheim, Norway, (1989).

[16] **Twu, C.H.,** An Internally Consistent Correlation for Predicting the Critical Properties and Molecular Weights of Petroleum and Coal-Tar Liquids, *Fluid Phase Equilibria,* **16**: 137 (1984).

[17] **Naji, H.S.,** Characterizing Pure and Undefined Petroleum Components, *International Journal of Engineering & Technology IJET - IJENS.* **10**(2): 39-68, April 10 (2010).

[18] **Riazi, M.R.,** *Characterization and Properties of Petroleum Fractions*, First Edition, ASTM, West Conshohocken, PA Chapter 2 (2005).

[19] **Ahmed, T.,** *Reservoir Engineering Handbook*, Third Edition, Gulf Professional Publishing GPP, Elsevier Inc. (2006).

[20] Petroleum Measurement Tables – Volume Correction Factors, Volume X-Background, Development and Program Documentation, ASTM D 1250-80, **ASTM**, Philadelphia, PA. 73-80.

معادلات رياضية لحساب لزوجة الزيت الخام عند الظروف السطحية – نظرة من جانب النمذجة الرياضية لمكامن البترول

حسن سليمان ناجي

قسم جيولوجيا البترول والترسبات، كلية علوم الأرض، جامعة الملك عبدالعزيز، جدة، المملكة العربية السعودية

email: hnaji@kau.edu.sa

المستخلص. من بادي الوقت، والباحثون في مجال هندسة البترول يعملون حثيثاً لإيجاد طرق مضاهاة لحساب لزوجة الزيت الخام في الظروف السطحية (Dead Oil) والذي يحتوي على نسبة قليلة من الغاز المذاب، وذلك اعتماداً على عينات من مواقع جغرافية مختلفة. معظم هذه الطرق تصنف في مجموعة تعتمد على درجة الحرارة، والثقل النوعي للزيت، كمتطلبين وحيدين للمضاهاة. وهناك مجموعة أخرى من الطرق، وهي أكثر دقة من سابقتها، تتطلب عامل واتسون لتصنيف الزيت الخام (Watson Characterization Factor) كمتطلب إضافي للمضاهاة. إن عامل واتسون لتصنيف الزيت الخام، يمكِن الباحث من التعرف على برافينية، أو صبغة الزيت الخام (Paraffinicity or Character of the Crude Oil)، حيث يبقى هذا العامل ثابتاً للمركبات الهيدروكربونية المتشابهة. ونظراً لتعقيد الحسابات في هذه المجموعة، يميل معظم الباحثين خطأً إلى استخدام طرق المجموعة الأولى الأقل دقة.

في هذه الورقة أتقدم بمقترح لمعادلتين رياضيتين لتقييم درجة غليان المركبات الهيدروكربونية النقية، والزيت الخام، من خلال

معرفة الثقل النوعي فقط. وحيث أن عامل واتسون ودرجة الغليان، يرتبطان بعلاقة مشتركة مع الثقل النوعي، عليه يمكن حساب قيمة عامل واتسون من معرفة قيمة الثقل النوعي فقط، والذي بدوره يسهل استخدام المجموعة الثانية، بنفس الطريقة المتبعة لعمل حسابات المجموعة الأولى. أيضاً أدت هذه الدراسة إلى استنتاج إمكانية استخدام معادلة درجة غليان المركبات الهيدروكربونية النقية، وإمكانية استخدامها للزيت الخام الذي يزيد ثقله النوعي عن 30 درجة بمقياس المعهد الأمريكي للدراسات البترولية API.

من الإسهامات العلمية الأخرى لهذه الورقة، أن جميع طرق المضاهاة التي تم عرضها، تم برمجتها باستخدام طريقة فصل البرنامج إلى كيانات مستقلة (Object-Oriented)، والذي بدوره يضفي مرونةَ كبيرةَ في عملية البرمجة، ويمكِن بسهولة من التغيير أو الإضافة للبرنامج بلا حدود، وينبغي التنويه هنا إلى أن هذه الطريقة في البرمجة هي الطريقة المتبعة حالياً في كتابة أعقد البرامج في جميع المجالات بيسرٍ وسهولةٍ. لقد تم عرض 8 طرق مضاهاة من المجموعة الأولى، وطريقتين من المجموعة الثانية، من وجهة نظر نمذجة ومحاكاة مكامن البترول. كما تم اختبار جميع الطرق، والتأكد من المعادلات المقترحة، باستخدام 8 أمثلة عملية.