Case Study – Rotate and mirror image charts

How does one calculate and plot a graph after rotating it through some angle or after creating a mirror image of it? See Figure 1 for examples of both. The normal curve in red above the horizontal axis is the original curve. The curve in green, which is a normal curve shown sideways with its base along the vertical axis, is the original curve rotated 90° counter-clockwise. The curve in blue, which looks like an upside down normal curve, is the mirror image of the original curve with the mirror placed aligned with the x-axis.

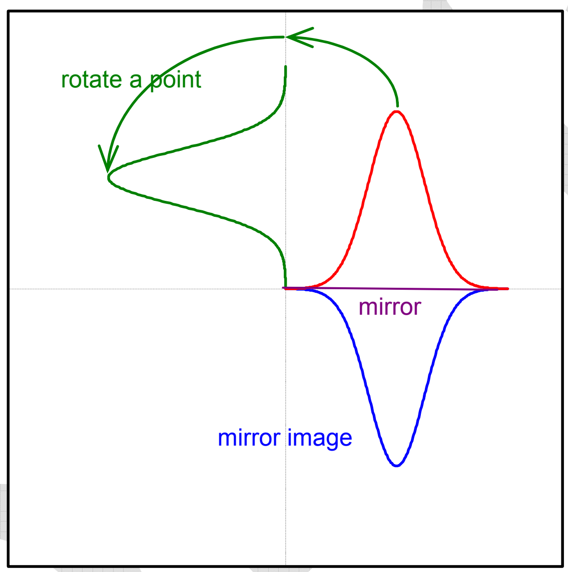


Figure 1

This is easy to do once one knows the associated mathematical mappings.

Theory

As with other cases, the reader doesn't have to go through the theory section to implement the process in Excel. This section is for those interested in the underlying analytics.

Consider any point in Cartesian coordinate (x_0, y_0). In Polar terms that corresponds to $r = \sqrt{x_0^2 + y_0^2}$ and $\theta = \tan^{-1}\left(\frac{y_0}{x_0}\right)$.

Rotation: After rotation by θ', the polar coordinate of the new point is $(r, \theta + \theta')$, and the Cartesian coordinate of the new point is $x_1 = r\cos(\theta + \theta')$ and $y_1 = r\cos(\theta + \theta')$.

Mirror image: See Figure 2. Suppose the mirror goes through the origin at an angle of $\theta^{'}$. Then, the original point, the mirror image point, and the origin form an isosceles triangle. From geometry, the polar coordinate of the new point is $\left(r, 2\theta^{'} - \theta\right)$ and the Cartesian coordinate can be computed as before resulting in $x_1 = r\cos\left(2\theta^{'} - \theta\right)$and $y_1 = r\cos\left(2\theta^{'} - \theta\right)$

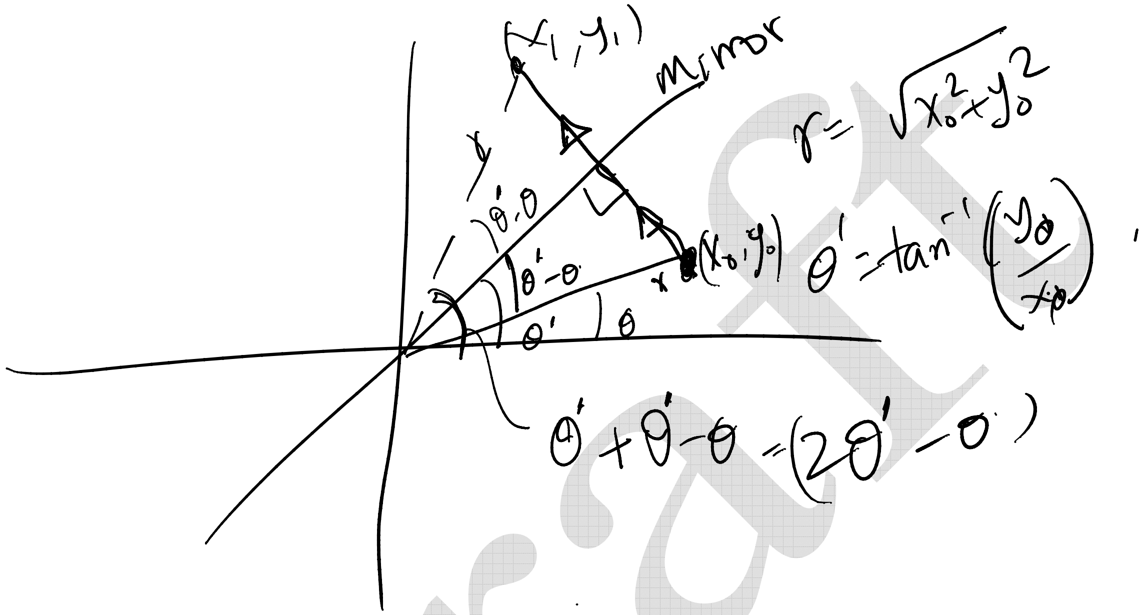


Figure 2

Now, suppose the mirror doesn't go through the origin but is given by the equation $y = mx + c$. The easiest way to handle this is to shift the measurement system so that the origin coincides with the y intercept of the mirror as in Figure 3.

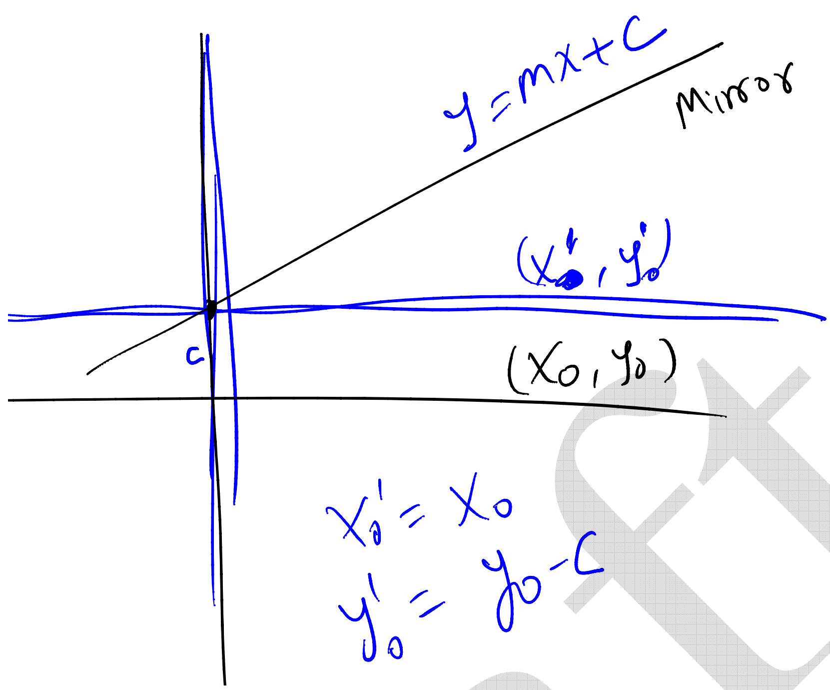


Figure 3

Any point in the original system (x_0, y_0) becomes the point $(x_0^{'}, y_0^{'})$ in the new system given by the transformation $x_0^{'} = x_0$ and $y_0^{'} = y_0 - c$. Now that the mirror passes through the origin, the angle $\theta^{'} = \tan^{-1}(m)$. With this information, the mirror coordinate are calculated as above. This coordinate is transformed back into the original system with the transformation

$x_1^{'} = x_1$ and $y_1^{'} = y_1 + c$.

The calculations in Excel

While the above calculations can be done in Excel, it is probably easier to understand, audit, implement and maintain a system based on User Defined Functions.

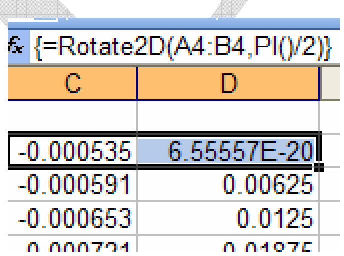
```
Option Explicit

Function Rotate2D(X0Y0 As Range, RotateRadians As Double)
    'X0Y0 is expected to contain 2 cells with the original _
     x and y coordinates _
     RotateRadians is the angular *counter-clockwise* rotation specified _
     in radians _
     Returns X1,Y1 as a two element array
    Dim Theta As Double, Rslt(1 To 2) As Double, X0 As Double, Y0 As Double
    X0 = X0Y0.Cells(1).Value
    Y0 = X0Y0.Cells(2).Value
    With Application.WorksheetFunction
    Theta = .Atan2(X0, Y0) + RotateRadians
        End With
    Rslt(1) = Sqr(X0 ^ 2 + Y0 ^ 2) * Cos(Theta)
```

```
    Rslt(2) = Sqr(X0 ^ 2 + Y0 ^ 2) * Sin(Theta)
    If Application.Caller.Columns.Count > 1 Then
        Rotate2D = Rslt
    Else
        Rotate2D = Application.WorksheetFunction.Transpose(Rslt)
        End If
    End Function
Function MirrorImage2D(X0Y0 As Range, _
        MirrorSlope As Double, MirrorIntercept As Double)
    'X0Y0 is expected to contain 2 cells with the original _
     x and y coordinates _
     The mirror is given by the equation _
    y=MirrorSlope * x + MirrorIntercept _
     Returns X1,Y1 as a two element array
    Dim Theta As Double, Rslt(1 To 2) As Double, X0 As Double, Y0 As Double
    X0 = X0Y0.Cells(1).Value
    Y0 = X0Y0.Cells(2).Value - MirrorIntercept
    With Application.WorksheetFunction
    Theta = 2 * Atn(MirrorSlope) - .Atan2(X0, Y0)
        End With
    Rslt(1) = Sqr(X0 ^ 2 + Y0 ^ 2) * Cos(Theta)
    Rslt(2) = Sqr(X0 ^ 2 + Y0 ^ 2) * Sin(Theta) + MirrorIntercept
    If Application.Caller.Columns.Count > 1 Then
        MirrorImage2D = Rslt
    Else
        MirrorImage2D = Application.WorksheetFunction.Transpose(Rslt)
        End If
    End Function
```

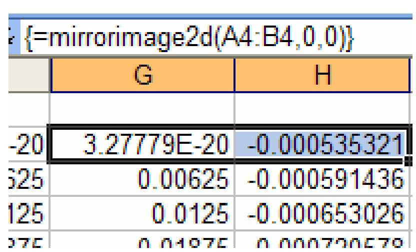
How the functions are used

Suppose the data for the original chart are in columns A and B starting with row 4. Then, select C4:D4 and array-enter `=Rotate2D(A4:B4,PI()/2)`. Copy the formula as far down columns C:D as needed. Similarly, select G4:H4 and array enter `=mirrorimage2d(A4:B4,0,0)`. Copy the formula as far down columns G:H as needed.



{=Rotate2D(A4:B4,PI()/2)}

C	D
-0.000535	6.55557E-20
-0.000591	0.00625
-0.000653	0.0125



{=mirrorimage2d(A4:B4,0,0)}

G	H
3.27779E-20	-0.000535321
0.00625	-0.000591436
0.0125	-0.000653026

Plot the data in columns A and B for the original curve, C and D for the rotated curve, and G and H for the mirror-image curve. Experiment with different values for *RotateRadians* and *MirrorSlope* and *MirrorIntercept* to see the effect of different rotations or of a mirror placed at different positions.