Comprehensive Analysis of a Slanted Gate

Consider the diagrams sketched in Figs. 1 and 2. The right wall is a moveable metal gate that is hinged at the bottom. The gate has weight W, length L, and depth B into the page. Initially, the gate is vertical and the water level reaches the top of the gate. Thus, the volume of water contained behind the gate is $V_w = aLB$. The rod that holds the gate up has length $L\sqrt{2}$ and is initially hinged at an angle of 45 degrees as shown in Fig. 1. The force of the rod on the gate is directed along the rod and is denoted as F_{rod}. A stop holds the opposite end of the rod in place. However, this stop can be moved to the right, allowing the gate to be rotated clockwise to some angle θ about its hinge as shown in Fig. 2.

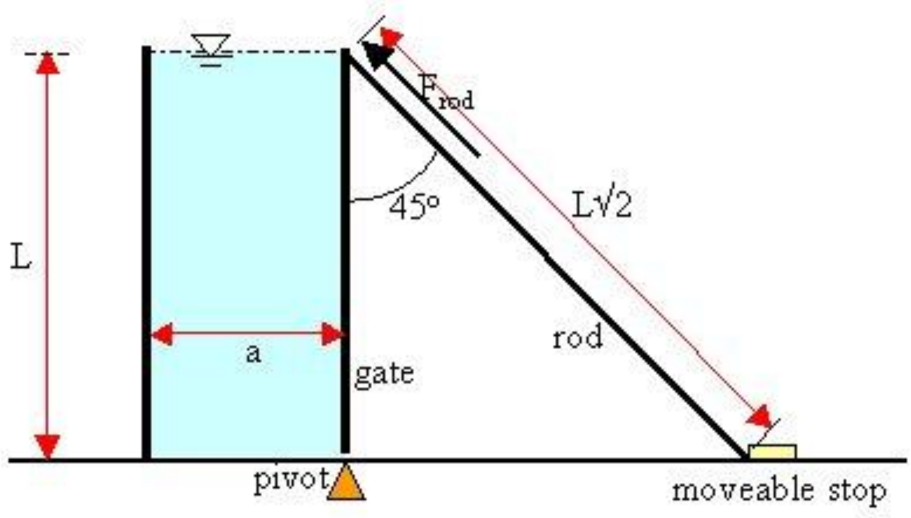


Fig. 1 Initial configuration of gate.

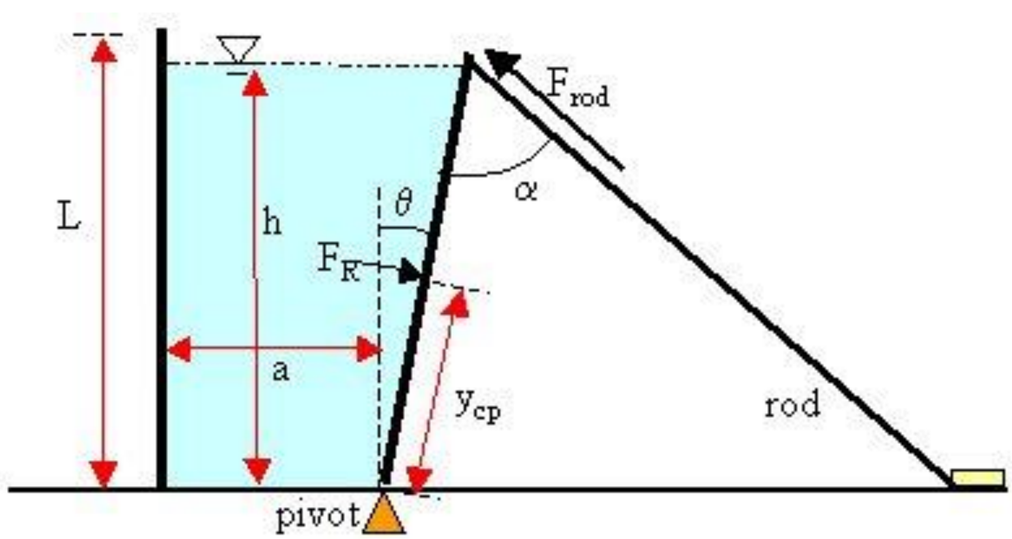
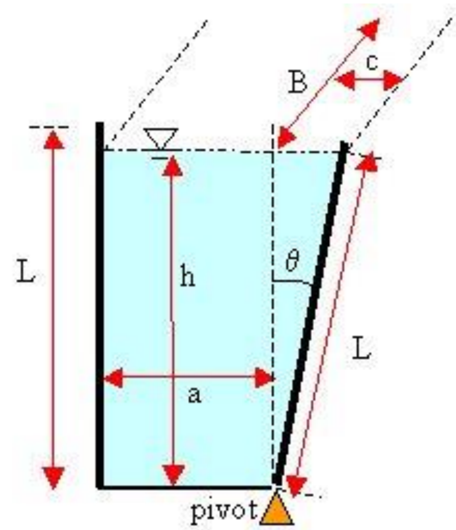


Fig. 2 Gate at some angle θ.

The goal of this particular example is to analyze this system in detail and to answer several questions concerning the system when the gate is at various angles, θ, as follows:

I. Determine the volume of the reservoir vs. θ. What is the maximum volume that can be maintained? What angle, θ_{max}, gives the maximum volume, V_{max}?

To answer the above questions, we need to determine the maximum volume bounded by the reservoir walls. This is a geometry problem that can be easily visualized in the diagram to the right. Here we see that the total volume is comprised of two volumes that can be easily determined:



$$V = (ah)B + (\frac{1}{2}hc)B$$

where $c = L\sin\theta$ and $h = L\cos\theta$. Inserting the expressions for c and h and recalling that $V_w = aLB$, we have

$$V = aLB\cos\theta + \frac{1}{2}BL^2\cos\theta\sin\theta$$

or

$$V = V_w \cos\theta\left(1 + \frac{L\sin\theta}{2a}\right) \tag{1}$$

To determine the maximum volume allowed, we can simply plot V(θ) and determine the values of V_{max} and θ_{max} visually. An alternative is to use a formal 1-D optimization routine to find the numerical maximum of V(θ) subject to the bounds $0 \le \theta \le \pi/2$. And, of course, we could use both the graphical and numerical approach to get the best of both worlds. In fact, this was done in the first part of **slanted_gate_1.m** (see listing at the end of this section of notes).

Note: Although we will not formally discuss optimization methods as part of this course (this is a very interesting and important subject, but we simply do not have sufficient time), you should at least be aware that these methods exist, and that, in general, they are only a little more difficult to use than the ***fzero*** root finding routine that we have discussed as part of Lesson 5. In particular, the routine used here is ***fminbnd***, which finds the local minimum of some function, f(x), in the interval $a \le x \le b$. The call to ***fminbnd*** is of the form

[xmin,fmin] = fminbnd(@function_name,a,b,options,p1,p2,…)

where the function file describing f(x) has the form

function F = function_name(x,p1,p2,…).

with p1, p2, …, being additional parameters that can be passed to the function (see ***help fminbnd*** for more details). Note that we can also use an ***anonymous function*** if f(x) is relatively simple!

In our case, we want to find the maximum of eqn. (1), so we simply search for the minimum of $-V(\theta)$. Also, since the equation for $V(\theta)$ is quite straightforward, the ***anonymous function*** option was chosen for illustration purposes. As seen in the first part of the listing for **slanted_gate_1.m** (see Table 1 at the end of this section), the actual syntax used was

```
[thmax,Vmax] = fminbnd(VofTh,0,pi/2);
```

where the function, ***VofTh***, is an ***anonymous function*** of the form

```
VofTh = @(th) -Vw*cos(th).*(1 + L*sin(th)/(2*a));
```

where ***th*** represents the independent variable θ.

Figure 3 summarizes nicely the results to our Part I questions. Here we see that the ratio V/V_w rises to a maximum of about 1.3 at θ_{max} = 30 degrees and then continually decreases as the clockwise angle of rotation approaches 90 degrees. Note that the actual numerical values of V_{max} and θ_{max} labeled on the plot come directly from the ***fminbnd*** optimization routine. These results indicate that at an angle of 30 degrees we could get 30% more water into the reservoir relative to the original value of V_w when the gate is upright (θ = 0 degrees).

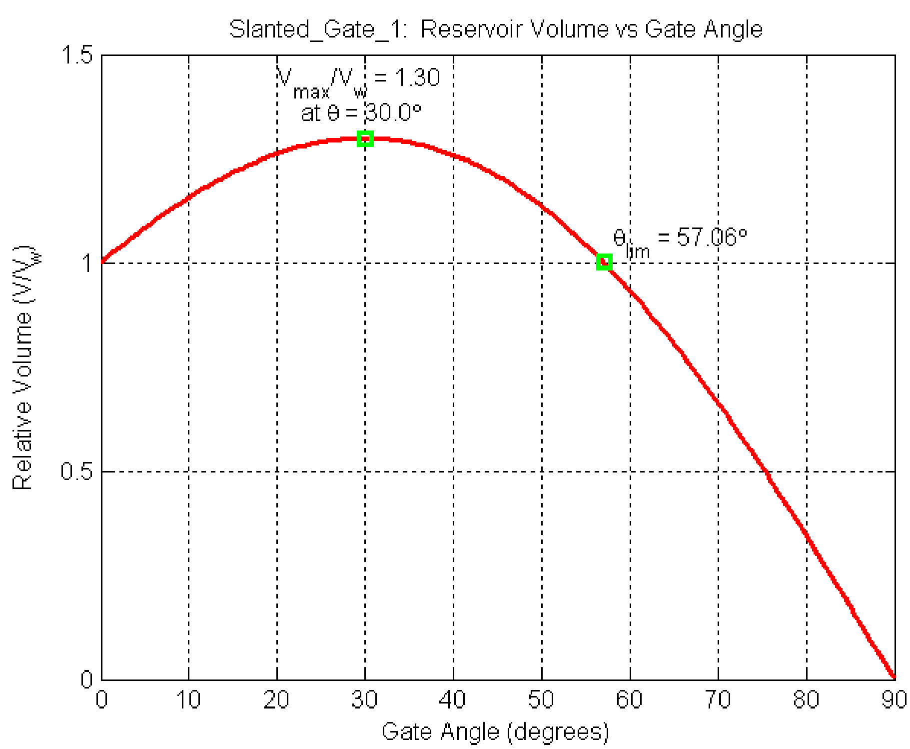


Fig. 3 Visualization of V(θ) from Part I of the slanted_gate_1.m program.

II. Find the limiting angle so that the original volume of water, V_w, does not spill over the gate.

Here, we note that the water spills over the gate when $V < V_w$ or when $V/V_w < 1$. To answer the above question, we could simply pull this value off the plot of $V(\theta)$ via visual observation -- i.e. what is θ when $V = V_w$?. We can also formulate this question in the form of a root finding problem (i.e. what is the value of θ such that $f(\theta) = V(\theta) - V_w = 0$?) and use Matlab's ***fzero*** routine to find a precise numerical value for θ_{lim}. Part II of **slanted_gate_1.m** , along with an anonymous function for $f(\theta)$, takes this approach, and the result, $\theta_{lim} = 57.06$ degrees, is noted directly on the plot of $V(\theta)$ in Fig. 3. It is pretty easy to see that for any angle greater than θ_{lim}, the dam will surely overflow, since the geometry simply cannot hold all the original volume of fluid, V_w.

III. Determine the water height versus angle for $V = V_w$.

Here we are interested in finding h vs. θ when the actual volume of water within the reservoir is V_w (in fact, we use the assumption that the volume of water is constant at V_w for the rest of the analyses in this Case Study). Except at the endpoints, $\theta = 0$ and $\theta = \theta_{lim}$, the reservoir capacity is greater than V_w over the range $0 \le \theta \le \theta_{lim}$. Thus the expressions, $c = L\sin\theta$ and $h = L\cos\theta$, are no longer valid, since the water volume does not fill the reservoir volume (i.e. the slanted gate is not fully wetted along its length L). The water height is still denoted as h, but now we want to write the length c in the geometry sketch on page 2 as $c = h\tan\theta$. With this expression, the actual water volume versus angle can be written as

$$V = (ah)B + (\frac{1}{2}h^2 \tan\theta)B = V_w$$

where the last equality states that the actual volume is that associated with the original water volume, V_w. The quantity of interest in this expression is the water height, h, and we could solve this problem with ***fzero*** by asking the question, "What is the value of h such that $f(h) = V(h) - V_w = 0$ for each value of θ?" (this is the preferred approach here). However, just to illustrate the use of Matlab's ***roots*** command, we will take an alternative path. In particular, we can also write the above expression as a quadratic polynomial in h,

$$\frac{B}{2}\tan\theta\, h^2 + aBh - V_w = 0 \tag{2}$$

Now, for each angle in the range $0 \le \theta \le \theta_{lim}$, we can determine the roots of eqn. (2) via Matlab's ***roots*** command, where the polynomial is simply represented by the coefficients, with the highest order term first. The algorithm actually implemented in **slanted_gate_1.m** uses the following structure:

```
loop over all θ
    p = [(B/2)*tanθ   a*B   -Vw];   r = roots(p),   h = max(r);
end loop
```

where the command ***max(r)*** extracts the positive root of the two roots generated with the ***roots*** command (in this case, one root is positive and the other is negative). The value of h for each angle was saved in a vector and plotted to give the desired relationship, $h(\theta)$ vs. θ, where the resultant plot is shown in Fig. 4. Although there is nothing surprising here -- that is, we expected the water height to decrease monotonically with increasing angle -- actually generating this result is nontrivial. We see that over the range of valid angles for no spillage of the original water volume, the actual water height decreases by nearly a factor of two from its original maximum value.

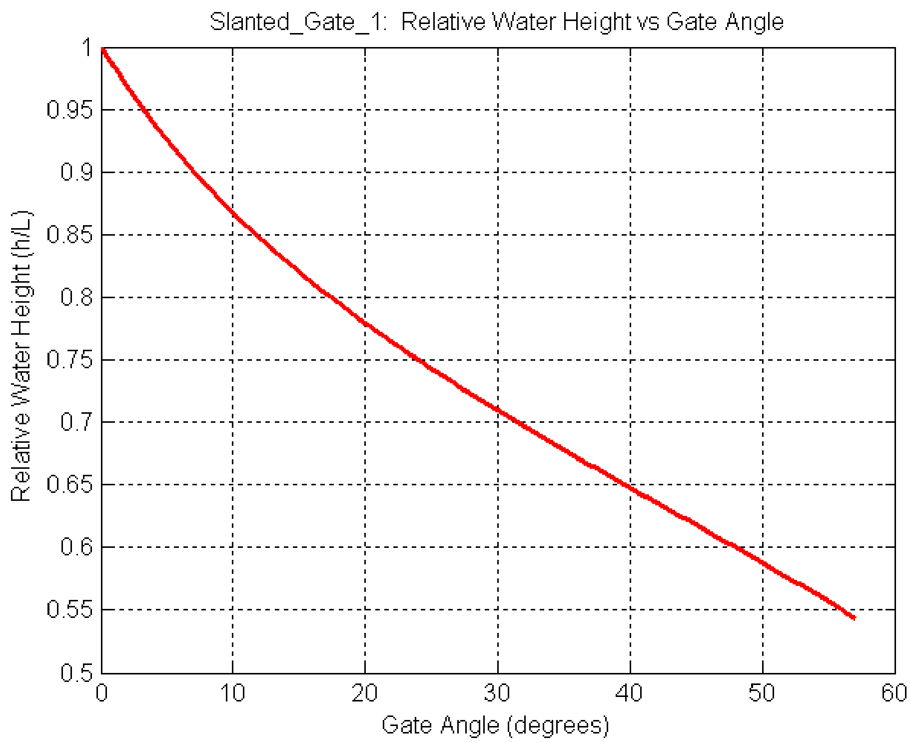


Fig. 4 Relative water height vs. θ from Part III of the slanted_gate_1.m program.

IV. Determine the resultant force, F_R, vs. gate angle and the location of the center of pressure, y_{cp}, relative to the hinge (for fixed $V = V_w$).

Now that we know the water height versus angle, from a study of Fluid Statics (see Chapter 4 of the 6th Ed. of Mott's "Applied Fluid Mechanics", for example), we can determine the force exerted on the gate and the location where the resultant force, F_R, produces the same moment as the actual distributed force -- where this location is referred to as the center of pressure, y_{cp}.

Recalling that the gate has depth B into the page, we can use a side view of the system as sketched in Fig. 5 to help define the notation used to determine F_R and y_{cp}. In particular, it is important to note that the area, A, where the force is applied only includes the submerged portion of the gate. Using the notation in Fig. 5, this can be written as

$$A = \xi B = \frac{hB}{\cos\theta}$$

Now, the resultant force can be written as

$$F_R = \gamma h_c A = \gamma \frac{h}{2} A = \gamma \frac{h}{2} \frac{hB}{\cos\theta}$$

or

$$F_R = \frac{\gamma B h^2}{2\cos\theta} \tag{3}$$

where γ is the specific weight of the fluid and h_c is the fluid depth at the centroid of the submerged area.

Also, from Fig. 5, we see that y_c is the distance along the plane from the surface to the centroid and that y_R is the distance from the surface to the center of pressure. From eqn. 4-5 in the 6th Ed. of Mott's text (noting the slightly different notation used here), we can define y_R as

$$y_R = y_c + \frac{I_{xc}}{y_c A}$$

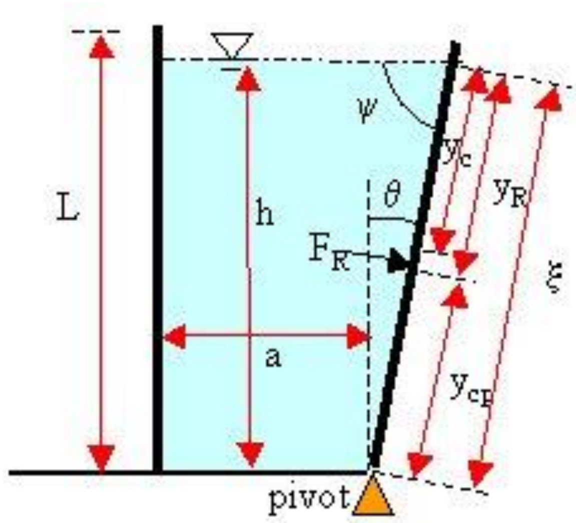


Fig. 5 Notation for determining F_R and y_{cp}.

where I_{xc} is the moment of inertia about the centroid of the area of interest. For the rectangular area in this problem, we have (see Appendix L in Mott's text, for example)

$$I_{xc} = \frac{1}{12} B\xi^3$$

Now, with $y_c = \xi/2 = h/2\cos\theta$, we can write y_R as

$$y_R = \frac{h}{2\cos\theta} + \frac{1}{12}\frac{Bh^3/\cos^3\theta}{(h/2\cos\theta)(hB/\cos\theta)} = \frac{1}{2}\frac{h}{\cos\theta} + \frac{1}{6}\frac{h}{\cos\theta}$$

or

$$y_R = \frac{2}{3}\frac{h}{\cos\theta} \tag{4}$$

and, finally, from Fig. 5, the center of pressure relative to the hinge is given by

$$y_{cp} = \xi - y_R = \frac{h}{\cos\theta} - \frac{2}{3}\frac{h}{\cos\theta} = \frac{1}{3}\frac{h}{\cos\theta} \tag{5}$$

Equations (3) and (5) were evaluated and plotted in Part IV of **slanted_gate_1.m** and the results are presented in Fig. 6. As expected, the resultant force on the gate tends to decrease with angle since the depth of water also decreases (see Fig. 4). However, F_R is related to the product of h and the area, A, of the submerged portion of the gate. Although h is monotonically decreasing, A peaks at the two limits of the rotation angle (0 degrees and 57.06 degrees), with a minimum in the center of the θ range. Thus, we see that F_R tends to flatten out in the 40 –50 degree range, and even increase slightly near θ_{lim}. The behavior of the center of pressure is also a little complicated. However, in this case, y_{cp} is simply proportional to the submerged planar area (i.e. $y_{cp} = h/3\cos\theta = A/3B$), so we do indeed expect the peaks at the endpoints and a dip in the middle of the θ range as seen in the lower part of Fig. 6 (and as discussed above for A vs. θ).

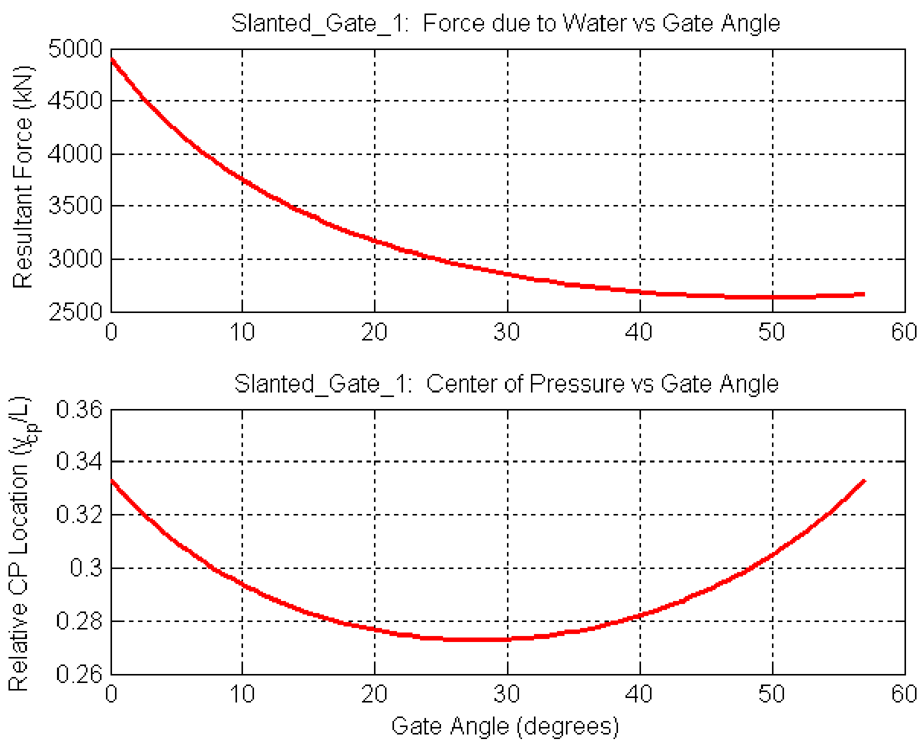


Fig. 6 F_R and y_{cp} vs. θ from Part IV of the slanted_gate_1.m program.

V. Determine the rod force, F_{rod}, vs. gate angle and find an optimum angle that minimizes the rod force.

The rod force, F_{rod}, can be determined by taking moments about the hinge at the bottom of the gate. The three forces of interest are F_R, W, and F_{rod}, and their location of application and direction are indicated in the sketch in Fig. 7. Since a moment involves the product of a perpendicular force and the distance to the pivot point, we can write the three moments of interest here as follows:

moment due to water pressure: $F_R\, y_{cp}$

moment due to weight of gate: $W\sin\theta\ (L/2)$

moment due to force exerted by rod: $-F_{rod}\cos\beta\ (L) = -F_{rod}\sin\alpha\ (L)$

where we have noted that $\beta + (180 - \alpha) = 90$, which gives $\beta = \alpha - 90$, and from a basic trigonometric identity, we have that

$$\cos(\alpha - 90) = \cos\alpha\cos 90 + \sin\alpha\sin 90 = \sin\alpha$$

Since, for static equilibrium, the sum of the moments must vanish, we have

$$F_R y_{cp} + W\sin\theta\frac{L}{2} - F_{rod}\sin\alpha\, L = 0$$

or
$$F_{rod} = \frac{F_R y_{cp} + W\sin\theta\frac{L}{2}}{L\sin\alpha} \tag{6}$$

where F_R is given by eqn. (3) and y_{cp} by eqn. (5). Thus, the only remaining task is to determine the relationship between angles α and θ.

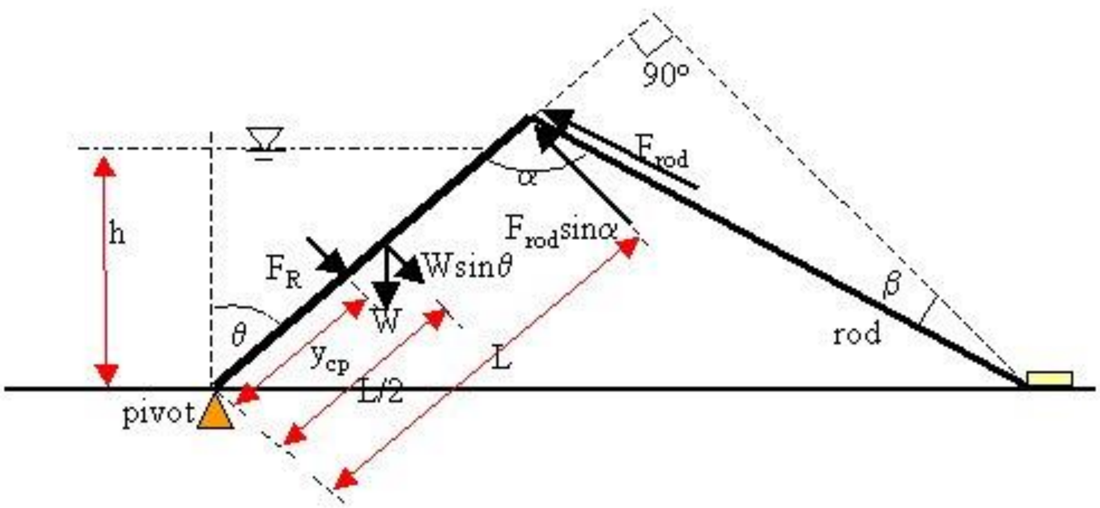


Fig. 7 Notation for writing a moment balance for the slanted gate.

To develop the desired relationship, we can use the law of sines, which states that, in a simple triangle with side lengths a, b, and c opposite angles A, B, and C, respectively, the following relationship is valid

$$\frac{a}{\sin A} = \frac{b}{\sin B} = \frac{c}{\sin C}$$

Now, applying the law of sines to our problem, we have that

$$\frac{L\sqrt{2}}{\sin(90-\theta)} = \frac{L}{\sin(180-(90-\theta)-\alpha)} = \frac{L}{\sin(90-(\alpha-\theta))} \tag{7}$$

which gives

$$\sqrt{2}\sin(90-(\alpha-\theta)) = \sin(90-\theta)$$

or, with some additional trigonometric identities, we have

$$\sqrt{2}\left[\sin 90\cos(\alpha-\theta) - \cos 90\sin(\alpha-\theta)\right] = \sin 90\cos\theta - \cos 90\sin\theta$$

which reduces to

$$\sqrt{2}\cos(\alpha-\theta)=\cos\theta$$

and, finally

$$\alpha=\theta+\cos^{-1}\left[\frac{\cos\theta}{\sqrt{2}}\right] \tag{8}$$

Thus, knowing θ, we can easily compute angle α via eqn. (8) -- and F_{rod}, which is the real goal here, via eqn. (6).

As before, we can find the angle that gives the minimum rod force from a plot of F_{rod} vs. θ or by a formal optimization routine that finds the minimum of eqn. (6). In fact, both procedures were used in the last part of **slanted_gate_1.m**. We again used Matlab's ***fminbnd*** routine to do the formal optimization, where the rod force, F_{rod}, for any rotation angle, θ, is evaluated in function routine **slanted_gate_1a.m**. The result of this exercise is plotted in Fig. 8, which displays F_{rod} vs. θ and identifies the desired minimum rod force and corresponding rotation angle. We see that the rod force needed to hold up the gate changes considerably with angle (by nearly a factor of three), and that a minimum occurs at θ = 36.2 degrees -- where the minimum value of F_{rod} is about 790 kN.

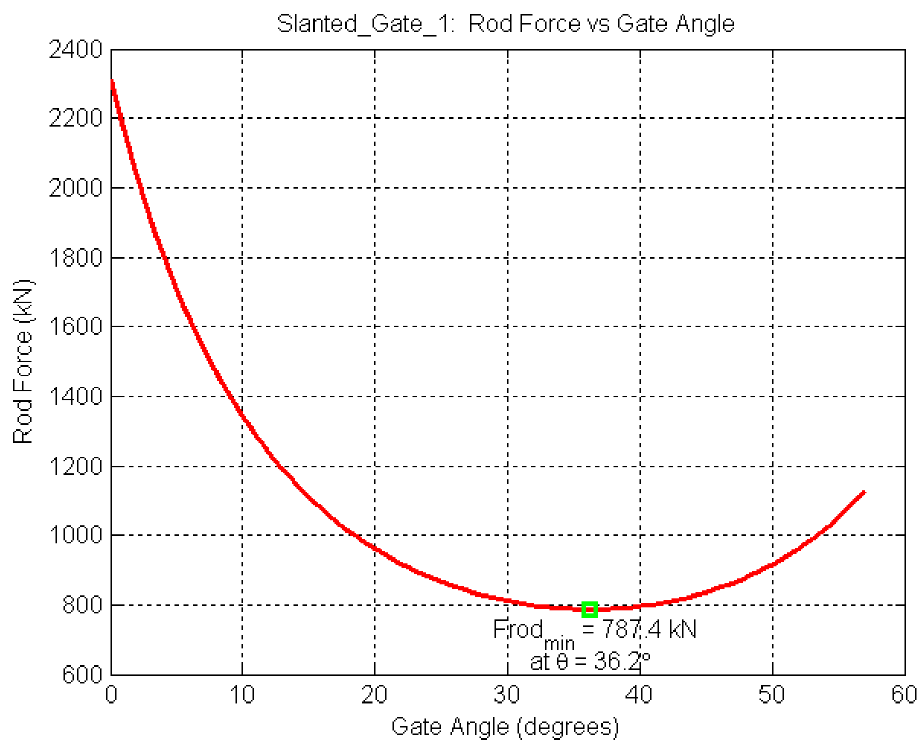


Fig. 8 Rod force versus rotation angle for the slanted gate problem.

Well, we have finally completed our analysis of the slanted gate problem. There was a fair amount of development and analysis required to fully understand this particular system. The equations and inter-relationships among the variables were somewhat complicated and not easy to visualize without the help of Matlab's evaluation and plotting capabilities. Thus, we have put Matlab's power to good use in this problem, and this example application should serve as a good illustration of how to use Matlab to advantage in a variety of similar circumstances.

Finally, in closing our treatment of this subject, we provide a listing of the Matlab programs written as part of this application (note also that anonymous functions, **VofTh** and **fth**, were defined and used within the main program **slanted_gate_1.m**):

slanted_gate_1.m -- main program that organizes all the computational and visualization tasks

slanted_gate_1a.m -- function for use with the ***fminbnd*** routine in Part V of the main program

A full listing of these m-files is contained in Table 1.

Reference: The basic idea for this problem was obtained from Example 11.1.2 on pages 481-485 in "An Engineer's Guide to Matlab" by Magrab, et. al. (2000, Prentice Hall).

Table 1 Listing of the Matlab programs written as part of the slanted gate application.

```
%
%   SLANTED_GATE_1.M       Comprehensive Application
%              Analysis of a Slanted Gate on a Water Reservoir
%
%   This file does some computational analysis for the Slanted Gate Problem
%   discussed in the notes.  It illustrates several of Matlab's capabilities
%   including the use of anonymous functions, function subprograms, the fzero and
%   fminbnd optimization routines, and finding roots of polynomials with the
%   roots command.  It also uses several simple plotting features including the
%   hold on/off command, subplots, and the text command to place text at a specific
%   location on the plot.  Finally, it also illustrates proper file documentation
%   using internal comments and the fprintf command and it provides a number of
%   fairly complicated arithmetic expressions where "dot" arithmetic is needed.
%   Overall, this program should give a good view of some of the analysis
%   capabilities that are easily implemented in Matlab.  A good understanding of
%   this example will go a long way in helping you apply Matlab in a variety of
%   different situations.
%
%   The basic idea for this problem was obtained from Example 11.1.2 on
%   pages 481-485 in "An Engineer's Guide to Matlab" by Magrab, et. al.
%   (2000, Prentice Hall).
%
%   File prepared by J. R. White, UMass-Lowell  (last update:  Nov. 2017)
%

      clear all,  close all,  nfig = 0;
%
%   identify basic problem data
      a = 5;                        % initial width of reservoir (m)
      B = 10;                       % depth of reservoir into page (m)
      L = 10;                       % vertical length of gate (m)
      Vw = a*L*B;                   % initial water volume
      W = 100;                      % weight of gate (kN)
      spwt = 9.81;                  % specific weight of water (kN/m^3)
%
%   Part I:  Determine total volume vs gate angle.  Also find max volume
%            and associated angle.
%
%   graphical analysis
      VofTh = @(th) -Vw*cos(th).*(1 + L*sin(th)/(2*a));  % anonymous function
      th = linspace(0,pi/2,181);
```

```
      Vol = -VofTh(th);
      nfig = nfig+1;   figure(nfig)
      plot(th*180/pi,Vol/Vw,'r-','LineWidth',2),grid
      title('Slanted\_Gate\_1:  Reservoir Volume vs Gate Angle')
      xlabel('Gate Angle (degrees)'),ylabel('Relative Volume (V/V_w)')
%
%   formal optimization routine for finding Vmax
      [thmax,Vmax] = fminbnd(VofTh,0,pi/2);
      Vmax = -Vmax/Vw;   thmax = thmax*180/pi;
      fprintf(1,'\n       Summary Results from the Slanted_Gate_1.m Program \n\n');
      fprintf(1,'  Max possible value of V/Vw is %6.3f \n',Vmax);
      fprintf(1,'  Max V/Vw occurs at a gate angle of %6.3f degrees \n',thmax);
      r = axis;   r(4) = 1.5;   axis(r);
      hold on
      plot(thmax,Vmax,'gs','LineWidth',2)
      text(20,Vmax+.13,['V_{max}/V_{w} = ',num2str(Vmax,'%5.2f')])
      text(20,Vmax+.06,['    at \theta = ',num2str(thmax,'%5.1f'),'\circ'])
%
%   Part II:  Find the limiting angle so the water does not spill (put on above plot)
%
      fth = @(th) Vw*cos(th)*(1 + L*sin(th)/(2*a)) - Vw;
      [thlim_r] = fzero(fth,[pi/6 pi/2]);
      Vlim = -VofTh(thlim_r)/Vw;  thlim_d = thlim_r*180/pi;
      fprintf(1,'\n');
      fprintf(1,'  Limiting angle for water not to spill is %6.3f degrees\n',thlim_d);
      fprintf(1,'  The value of V/Vw at this point is %6.3f \n',Vlim);
      plot(thlim_d,Vlim,'gs','LineWidth',2)
      text(thlim_d+1,Vlim+.05,['\theta_{lim} = ',num2str(thlim_d,'%5.2f'),'\circ'])
      hold off
%
%   Part III:  Find water height vs gate angle for no spillage (via roots command)
%
      Nth = 100;  th = linspace(0,thlim_r,Nth);  h = zeros(1,Nth);
      for i = 1:Nth
        p = [B*tan(th(i))/2 a*B -Vw];  rr = roots(p);  h(i) = max(rr);
      end
      nfig = nfig+1;   figure(nfig)
      plot(th*180/pi,h/L,'r-','LineWidth',2),grid
      title('Slanted\_Gate\_1:  Relative Water Height vs Gate Angle')
      xlabel('Gate Angle (degrees)'),ylabel('Relative Water Height (h/L)')
%
%   Part IV:  Determine resultant force, FR, vs gate angle and location of
%             the center of pressure relative to the hinge (for fixed V = Vw).
%
      FR = spwt*B*h.*h./(2*cos(th));  ycp = h./(3*cos(th));
      nfig = nfig+1;   figure(nfig)
      subplot(2,1,1),plot(th*180/pi,FR,'r-','LineWidth',2),grid
      title('Slanted\_Gate\_1:  Force due to Water vs Gate Angle')
      ylabel('Resultant Force (kN)')
      subplot(2,1,2),plot(th*180/pi,ycp/L,'r-','LineWidth',2),grid
      title('Slanted\_Gate\_1:  Center of Pressure vs Gate Angle')
      xlabel('Gate Angle (degrees)'),ylabel('Relative CP Location (y_{cp}/L)')
%
%   Part V:  Determine rod force, Frod, vs gate angle and find an optimum
%               angle which minimizes the rod force.
%
%   Note:  Since I plan to use fminbnd to find the formal minimum of Frod vs theta,
%          let's put all the needed computations in a function file.  This is a
%          little redundant since most of the information is already available.
%          However, since efficiency is not the real goal here, let's just focus
%          on the current analysis (and forget that we already computed most of the
%          needed parameters).  Here we will simply implement eqn. (6) from the notes
%          -- and any intermediate results that may be needed...
%
      Nth = 100;  th = linspace(0,thlim_r,Nth);
      Frod = slanted_gate_1a(th,a,B,L,Vw,W,spwt);
      nfig = nfig+1;   figure(nfig)
      plot(th*180/pi,Frod,'r-','LineWidth',2),grid
      title('Slanted\_Gate\_1:  Rod Force vs Gate Angle')
      xlabel('Gate Angle (degrees)'),ylabel('Rod Force (kN)')
      hold on
```

```
%
      [thmin,Frodmin] = fminbnd(@slanted_gate_1a,0,thlim_r,[],a,B,L,Vw,W,spwt);
      fprintf(1,'\n');
      fprintf(1,'  Minimum value of Frod is %6.3f kN \n',Frodmin);
      fprintf(1,'  Minimum Frod occurs at a gate angle of %6.3f degrees \n',thmin*180/pi);
      plot(thmin*180/pi,Frodmin,'gs','LineWidth',2)
      text(29,Frodmin-75,['Frod_{min} = ',num2str(Frodmin,'%5.1f'),' kN'])
      text(29,Frodmin-150,['      at \theta = ',num2str(thmin*180/pi,'%5.1f'),'\circ'])
%
%  end of problem
```

```
%
%   SLANTED_GATE_1A.M   Function file for use in FMINBND
%
%   This file evaluates the rod force, Frod, for any gate angle, th.
%   This function is used in the last part of the slanted_gate_1.m file.
%   It contains much of the calculations done in the earlier parts of
%   the program.  If our real focus had only been the rod force, this
%   would have been the only major component needed.
%
%   File prepared by J. R. White, UMass-Lowell (last update:  Nov. 2017)
%
      function Frod = slanted_gate_1a(th,a,B,L,Vw,W,spwt)
%
%   compute h  (see eqn. 2 in notes)
      Nth = length(th);  h = zeros(1,Nth);
      for i = 1:Nth
        p = [B*tan(th(i))/2 a*B -Vw];  rr = roots(p);  h(i) = max(rr);
      end
%
%   compute FR and ycp  (see eqns. 3 and 5 in notes)
      FR = spwt*B*h.*h./(2*cos(th));  ycp = h./(3*cos(th));
%
%   compute angle alpha  (see eqn. 8 in notes)
      alpha = th + acos(cos(th)/sqrt(2));
%
%   now compute Frod  (see eqn. 6 in notes)
      Frod = (FR.*ycp + W*sin(th)*L/2)./(L*sin(alpha));
%
%   end of function
```