SUMMARY

This analysis calculated the mass moment of inertia of a non-viscous fluid in a slowly rotating rectangular tank. Given the dimensions of the tank in the x, y, and z coordinates, the axis of rotation, the percentage of the tank occupied by the fluid, and angle of rotation, an algorithm was written that could calculate the mass moment of inertia of the fluid. While not included in this paper, the change in the mass moment of inertia of the fluid could then be used to calculate the force exerted by the fluid on the container wall.

ANALYSIS

Depending on the dimensions of the tank, the amount of fluid in the tank, and the angle of rotation, the resulting shape of the fluid can be broken down into simple geometries. Since the fluid was assumed to be incompressible, non-viscous, and the rotational velocity small, the shape of the fluid was represented by rectangular and triangular prisms. The most simple fluid shapes were represented by a single rectangular or a single triangular prism, while the most complex geometry was represented by two rectangular and one triangular prism (See Fig. 1 and 2). The mass moment of inertia ("Iz" in this coordinate system) of each prism was first calculated relative to its centroid. Then the parallel axis theorem was used to calculate the Iz of the entire volume at either the fluid’s center of mass or the container’s center of rotation.

The mass moment of inertia of a rectangular prism with respect to its centroid and the axis shown was calculated by the following equation:

\[ I_z = \frac{1}{12} m (y^2 + z^2) \]

(NASA-CR-197777) THE CALCULATION OF THE MASS MOMENT OF INERTIA OF A FLUID IN A ROTATING RECTANGULAR TANK Final Report (California State Univ.) 46 p
\[ I_z = \frac{m(a^2 + b^2)}{12} \]

where:
- \( I_z \) = Moment of inertia with respect to the z-axis.
- \( a \) = Length of rectangular prism in x-axis.
- \( b \) = Length of rectangular prism in y-axis.
- \( m \) = Mass of prism.

The mass moment of inertia of a triangular prism with respect to its centroid and the axis shown was calculated by the following equations:

\[ I_z = \frac{m(a^2 + b^2)}{18} \]

The parallel axis theorem:

\[ I_z = I_z + m(x^2 + y^2) \]

where:
- \( x \) = Distance in x-direction from centroidal to arbitrary axis.
- \( y \) = Distance in y-direction from centroidal to arbitrary axis.

For comparison, the effective moment of inertia of fluid was calculated by the following formula:

\[ \frac{I_{FZ}}{I_{SZ}} = 1 - \frac{4I_z^3}{1 + I_z^3} + 2.510 \left( \frac{\tanh \frac{\pi}{2x_1} + 0.0045}{1 + I_z^3} \right) \]

where:
- \( I_{FZ} \) = Effective moment of inertia of fuel about z-axis.
- \( I_{SZ} \) = Moment of inertia of solidified fuel about z-axis.
- \( I_z \) \( \theta = 0 \).
- \( r_1 \) = b/a = tank aspect ratio in xy-plane.

RESULTS

As the partially filled rectangular tank shown in figures 1 or 2 rotates about its origin, the \( I_z \) of the fluid will change. Program MomentOfInertia (See Appendix B) calculated the \( I_z \) of the fluid relative to the centroid of the fluid. In tables one through
three, three different tank dimensions (all with unit depth) are shown: 1 x 2, 1 x 4, and 1 x 8, respectively. With each tank dimension, the mass moment of inertia for three different fluid volumes were tabulated. Only zero through 90 degrees were calculated since the mass moment of inertia for 90 through 180 degrees are mirror image of the shown data. The data then repeats every 180 degrees. This data is also shown graphically in figures 3 through 6.

As expected, the Iz of tanks with aspect ratio of 1/2 did not change significantly relative to tank rotation at any fluid level. This was due to the proximity of the aspect ratio to unity. With smaller (or greater) aspect ratios, the change in Iz increased significantly. For 50% volume, there was a 112% increase in Iz/rho for a/b = 1/2; while for a/b = 1/8, the change was 278%. Decreasing fluid volume also increased the change in Iz as the tank rotated. For a/b = 1/8, and the fluid volume was 80 percent, the change in Iz was 54%. For 20 percent fluid volume and the same aspect ratio, the change in Iz was 1,700%.

Program Moment1 (See Appendix B) calculated the Iz of the fluid relative to the center of rotation, which in this case was the origin. The calculations were very similar to the previous, except that the center of rotation was used as the axis instead of the C.G. of the fluid. Tables 3 through 6 shown the values calculated for 3 different aspect ratios and 3 different fluid volumes for each aspect ratio. These values are shown graphically in figures 7 thru 9.

As compared to the Iz relative to the C.G. of the fluid, the Iz relative to the center of rotation was greater, given the same aspect ratio and fluid volume. The greatest increase was the 20 percent fluid volume and aspect ratio = 1/2. The smaller percentage filled containers had greater increases than the higher percentage filled containers due to the greater change in the distance between the C.G. of the fluid and the center of rotation. For example, while the Iz of zero degrees, aspect ratio = 1/5, and 80 percent fluid volume changed 17% when the axis changed from C.G. of fluid to center of rotation, the Iz for the same angle and
aspect ratio, but only 20% fluid volume, changed 1.873%.

The change in the mass moment of inertia also increased with decreasing aspect ratio and decreasing fluid volume, however, the percent change was not as dramatic. The change in Iz for aspect ratio = 1/2 and 20% fluid volume relative to the C.G. was 248%, while the change in Iz relative to center of rotation was reduced to 48%.

The Iz of a tank with decreasing fluid level was also calculated by altering program MomentCG. The fluid level started at 100% at 0 degrees and decreased to 0% after 360 degrees. The results for three different aspect ratios are plotted in Figures 7 thru 9. This data demonstrated that even with an aspect ratio = 1/2, the change in Iz can be significant when the fluid level decreased. With even smaller aspect ratios, such as wing fuel tanks, the change in Iz was even greater. The "jumps" in the value of Iz corresponds to the a large change in the geometry of the fluid.

For comparison, the effective moment of inertia for various tank aspect ratios were also calculated. The effective moment of inertia is the moment of inertia of an equivalent mechanical system. However, the effective moment applies to small angular displacements only, which was quite different from the case that was analyzed here. These values are plotted in Figures 10 thru 12.

REFERENCES


FIGURE 1: TANK CONFIGURATION
FIGURE 2: MOST SIMPLE GEOMETRY.

FIGURE 3: MOST COMPLEX GEOMETRY.
### Table 1: Izz relative to fluid center of gravity for tank dimensions: $a = 1.00$, $b = 2.00$, and $c = 1.00$.  

<table>
<thead>
<tr>
<th>Theta</th>
<th>20% Full</th>
<th>50% Full</th>
<th>80% Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0387</td>
<td>0.1667</td>
<td>0.4747</td>
</tr>
<tr>
<td>10.0</td>
<td>0.0386</td>
<td>0.1677</td>
<td>0.4766</td>
</tr>
<tr>
<td>20.0</td>
<td>0.0385</td>
<td>0.1712</td>
<td>0.4829</td>
</tr>
<tr>
<td>30.0</td>
<td>0.0380</td>
<td>0.1780</td>
<td>0.4953</td>
</tr>
<tr>
<td>40.0</td>
<td>0.0361</td>
<td>0.1903</td>
<td>0.5178</td>
</tr>
<tr>
<td>50.0</td>
<td>0.0361</td>
<td>0.2125</td>
<td>0.5440</td>
</tr>
<tr>
<td>60.0</td>
<td>0.0411</td>
<td>0.2552</td>
<td>0.5703</td>
</tr>
<tr>
<td>70.0</td>
<td>0.0553</td>
<td>0.3154</td>
<td>0.5989</td>
</tr>
<tr>
<td>80.0</td>
<td>0.1019</td>
<td>0.3454</td>
<td>0.6183</td>
</tr>
<tr>
<td>90.0</td>
<td>0.1347</td>
<td>0.3542</td>
<td>0.6187</td>
</tr>
</tbody>
</table>

| % Change | 248 | 112 | 30 |

### Table 2: Izz relative to fluid center of gravity for tank dimensions: $a = 1.00$, $b = 4.00$, and $c = 1.00$.  

<table>
<thead>
<tr>
<th>Theta</th>
<th>20% Full</th>
<th>50% Full</th>
<th>80% Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.1093</td>
<td>0.8333</td>
<td>2.9973</td>
</tr>
<tr>
<td>10.0</td>
<td>0.1101</td>
<td>0.8358</td>
<td>3.0014</td>
</tr>
<tr>
<td>20.0</td>
<td>0.1126</td>
<td>0.8439</td>
<td>3.0147</td>
</tr>
<tr>
<td>30.0</td>
<td>0.1173</td>
<td>0.8599</td>
<td>3.0410</td>
</tr>
<tr>
<td>40.0</td>
<td>0.1256</td>
<td>0.8891</td>
<td>3.0894</td>
</tr>
<tr>
<td>50.0</td>
<td>0.1400</td>
<td>0.9450</td>
<td>3.1825</td>
</tr>
<tr>
<td>60.0</td>
<td>0.1642</td>
<td>1.0651</td>
<td>3.3819</td>
</tr>
<tr>
<td>70.0</td>
<td>0.2213</td>
<td>1.3867</td>
<td>3.6922</td>
</tr>
<tr>
<td>80.0</td>
<td>0.4158</td>
<td>2.3042</td>
<td>4.1547</td>
</tr>
<tr>
<td>90.0</td>
<td>1.0693</td>
<td>2.7083</td>
<td>4.4373</td>
</tr>
</tbody>
</table>

| % Change | 878 | 225 | 48 |
Table 3: Iz relative to fluid center of gravity for tank dimensions: \(a = 1.00, b = 8.00, \text{ and } c = 1.00\).

<table>
<thead>
<tr>
<th>Theta</th>
<th>20% Full</th>
<th>50% Full</th>
<th>80% Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.4747</td>
<td>5.6667</td>
<td>22.378</td>
</tr>
<tr>
<td>10.0</td>
<td>0.4766</td>
<td>5.6718</td>
<td>22.387</td>
</tr>
<tr>
<td>20.0</td>
<td>0.4829</td>
<td>5.6885</td>
<td>22.467</td>
</tr>
<tr>
<td>30.0</td>
<td>0.4953</td>
<td>5.7216</td>
<td>22.467</td>
</tr>
<tr>
<td>40.0</td>
<td>0.5180</td>
<td>5.7826</td>
<td>22.566</td>
</tr>
<tr>
<td>50.0</td>
<td>0.5610</td>
<td>5.9000</td>
<td>22.755</td>
</tr>
<tr>
<td>60.0</td>
<td>0.6519</td>
<td>6.1581</td>
<td>23.173</td>
</tr>
<tr>
<td>70.0</td>
<td>0.8833</td>
<td>6.8869</td>
<td>24.368</td>
</tr>
<tr>
<td>80.0</td>
<td>1.6633</td>
<td>10.522</td>
<td>28.232</td>
</tr>
<tr>
<td>90.0</td>
<td>8.5387</td>
<td>21.417</td>
<td>34.475</td>
</tr>
</tbody>
</table>

% Change 1699 278 54

Table 4: Iz relative to center of rotation for tank dimensions: \(a = 1.00, b = 2.00, \text{ and } c = 1.00\).

<table>
<thead>
<tr>
<th>Theta</th>
<th>20% Full</th>
<th>50% Full</th>
<th>80% Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.2947</td>
<td>0.4167</td>
<td>0.5387</td>
</tr>
<tr>
<td>10.0</td>
<td>0.2880</td>
<td>0.4167</td>
<td>0.5390</td>
</tr>
<tr>
<td>20.0</td>
<td>0.2780</td>
<td>0.4167</td>
<td>0.5453</td>
</tr>
<tr>
<td>30.0</td>
<td>0.2596</td>
<td>0.4167</td>
<td>0.5553</td>
</tr>
<tr>
<td>40.0</td>
<td>0.2387</td>
<td>0.4167</td>
<td>0.5737</td>
</tr>
<tr>
<td>50.0</td>
<td>0.2187</td>
<td>0.4167</td>
<td>0.5946</td>
</tr>
<tr>
<td>60.0</td>
<td>0.1986</td>
<td>0.4167</td>
<td>0.6147</td>
</tr>
<tr>
<td>70.0</td>
<td>0.1924</td>
<td>0.4167</td>
<td>0.6347</td>
</tr>
<tr>
<td>80.0</td>
<td>0.1987</td>
<td>0.4167</td>
<td>0.6409</td>
</tr>
<tr>
<td>90.0</td>
<td></td>
<td>0.4167</td>
<td>0.6347</td>
</tr>
</tbody>
</table>

% Change 34 0 18
Aspect Ratio = 1/5

<table>
<thead>
<tr>
<th>Theta</th>
<th>20% Full</th>
<th>50% Full</th>
<th>80% Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>2.1573</td>
<td>2.8333</td>
<td>3.5093</td>
</tr>
<tr>
<td>10.0</td>
<td>2.1542</td>
<td>2.8333</td>
<td>3.5124</td>
</tr>
<tr>
<td>20.0</td>
<td>2.1441</td>
<td>2.8333</td>
<td>3.5226</td>
</tr>
<tr>
<td>30.0</td>
<td>2.1240</td>
<td>2.8333</td>
<td>3.5427</td>
</tr>
<tr>
<td>40.0</td>
<td>2.0869</td>
<td>2.8333</td>
<td>3.5797</td>
</tr>
<tr>
<td>50.0</td>
<td>2.0153</td>
<td>2.8333</td>
<td>3.6514</td>
</tr>
<tr>
<td>60.0</td>
<td>1.8067</td>
<td>2.8333</td>
<td>3.8060</td>
</tr>
<tr>
<td>70.0</td>
<td>1.6238</td>
<td>2.8333</td>
<td>4.0428</td>
</tr>
<tr>
<td>80.0</td>
<td>1.2927</td>
<td>2.8333</td>
<td>4.3740</td>
</tr>
<tr>
<td>90.0</td>
<td>1.1973</td>
<td>2.8333</td>
<td>4.4693</td>
</tr>
</tbody>
</table>

% Change 44 0 27

Table 5: Iz relative to center of rotation for tank dimensions: a = 1.00, b = 4.00, and c = 1.00.

Aspect Ratio = 1/8

<table>
<thead>
<tr>
<th>Theta</th>
<th>20% Full</th>
<th>50% Full</th>
<th>80% Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>16.859</td>
<td>21.667</td>
<td>26.475</td>
</tr>
<tr>
<td>30.0</td>
<td>16.792</td>
<td>21.667</td>
<td>26.541</td>
</tr>
<tr>
<td>40.0</td>
<td>16.718</td>
<td>21.667</td>
<td>26.616</td>
</tr>
<tr>
<td>50.0</td>
<td>16.575</td>
<td>21.667</td>
<td>26.759</td>
</tr>
<tr>
<td>60.0</td>
<td>16.259</td>
<td>21.667</td>
<td>27.075</td>
</tr>
<tr>
<td>70.0</td>
<td>15.349</td>
<td>21.667</td>
<td>27.984</td>
</tr>
<tr>
<td>80.0</td>
<td>12.413</td>
<td>21.667</td>
<td>30.920</td>
</tr>
<tr>
<td>90.0</td>
<td>8.7947</td>
<td>21.667</td>
<td>34.539</td>
</tr>
</tbody>
</table>

% Change 48 0 31

Table 6: Iz relative to center of rotation for tank dimensions: a = 1.00, b = 8.00, and c = 1.00.
Figure 1: $I_z/I_h$ Relative to C.G. of Fluid for 1 x 2 Tank

- 20 percent
- 50 percent
- 80 percent
Figure 2: \( I_z/rho \) Relative to C.G. of Fluid for 1 x 4 Tank.
Figure 3: $l_z/\rho$ Relative to C.G. of Fluid for $1 \times 8$ Tank

- 20 percent
- 50 percent
- 80 percent
Figure 4: \( \frac{I_z}{I_{ho}} \) Relative to Center of Rotation for 1x2 Tank

- 20 percent
- 50 percent
- 80 percent

**Axes:**
- \( I_z/I_{ho} \) on the y-axis
- Angle Theta on the x-axis
Figure 5: \( \frac{I_z}{I_{tho}} \) Relative to Center of Rotation for 1 x 4 Tank

\[
\begin{array}{c|c|c|c|c}
\text{Angle } \theta & 0 & 5 & 10 & 15 \\
\hline
\frac{I_z}{I_{tho}} & 20 \text{ percent} & 50 \text{ percent} & 80 \text{ percent} \\
\end{array}
\]
Figure 6: tz/tho Relative to Center of Rotation for 1 x 8 Tank
Figure 7: tz/tho of 1 x 2 Tank with Decreasing Fluid
Figure 8: $l_z/rho$ of $1 \times 4$ Tank with Decreasing Fluid
Figure 9: Lz/tho of 1 x 8 Tank with Decreasing Fluid
Fig 10: \( \frac{f}{y} \) vs. Tank Aspect Ratio, \( r \)
Fig 11: Isy/ly vs. Tank Aspect Ratio, \( r \)
Fig. 13: Ily and Isy vs. Tank Aspect Ratio, r
This program calculates the mass moment of inertia of the fluid in a tank relative to the tv or the fluid. The tank is dimensioned as a in the x-dir, b in the y-dir, and c in the z-dir into the tank. The center of the tank is also the center of rotation and corresponds to the origin of the coordinates. The tank rotates counterclockwise and this angle is defined as theta. This version was last modified on 2/5/90.

```plaintext
const
pi := 3.141593;
rho = 1.0;

var

Number = array [1..90] of real;

ar inFile: text; // file pointer for output
outFile: text;
outFileName: string;
output: Number;
output2: Number;
printer: char;

a: (length of tank)
b: (height of tank)
c: (width of tank)
angle: (the angle of rotation in degrees)
area: (total area of fluid in xy plane)
increment: (increment by which angle increases)
areaTriangle: (Max area of triangular shape)
areaTriangle: (Max area of triangle and rectangle combined)
percent: (fluid level as a percent of height "h")
mass: (Mass of fluid)
moments: (Mass moment of inertia of entire shape)
theta: (angle)

*************

procedure CounterRotate(var ax, ay, ax1, ay1: real);
This procedure converts local Cord. to global Cord. given angle theta.:

var

ax := (ax1 * cos(theta)) - (ay1 * sin(theta));
ay := (ax1 * sin(theta)) + (ay1 * cos(theta));

end;

function IBarTri(var mTri, ta, tb: real): real;
This function will calculate the mass moment of inertia of a triangular prism given its mass, base, and height.:

gin
IBarTri := (mTri/18) * (sqr(ta) + sqr(ta));

end;

function IBarRect(var mRect, ra, rb: real): real;
This function will calculate the mass moment of inertia of a rectangular prism given its mass, length, and height.:

gin
IBarRect := (mRect/12) * (sqr(ra) + sqr(rb));

end;

procedure Centroid(var xBar, yBar,
a1.x1, y1,
a2.x2, y2,
a3.x3, y3: real);
```


```pascal
function WhatType:integer;
+This function determines what shape the fluid is in. Type 1 = triangle; *)
+type 2 = triangle and rectangles; and type 3 = one triangle and two *)
+rectangles. *)
var
area, alpha, bArea :real;
begin
    aTriangle := 0.5 * sqrt(a) * tangentTheta;
    aArea := aTriangle - (a * (t - (a * tangentTheta)));
    alpha := arctan(b/a);
    if tangentTheta <> 0.0 then
        begin
            bTriangle := abs((0.5 * sqrt(b))/tangentTheta);
            bArea := bTriangle + abs((b * (a - (b/tangentTheta))));
        end;
    if (angle = 0.0) then  (It's a rectangle.)
        WhatType := 4
    else if (angle = 90.0) then (It's still a rectangle.)
        WhatType := 5
    else if theta <= alpha then
        begin
            if (aTriangle > area) then
                WhatType := 1
            else if (aArea > area) and (bArea > area) then
                WhatType := 2
            else WhatType := 3;
        end
    else
        begin
            if (bTriangle > area) then
                WhatType := 1
            else if (bArea > area) then
                WhatType := 6
            else WhatType := 7;
        end;
end;

d procedure TypeOne;
+his procedure will calculate the mass moment of inertia of a triangular *)
+ism. *)
var
    a2, b2, h, j, cx, cy :real;
begin
    h := sqrt((2 * area)/tangentTheta) * tangentTheta;
    j := sqrt((2 * area)/tangentTheta);
    a2 := (j/3) - (a/2);
    b2 := (h/3) - (b/2);
    CounterRotate(cx, cy, a2, b2);
    moment := IBarTri(mass,h,j);
end;

procedure TypeTwo;
```
This Procedure will calculate the mass moment of inertia of a volume which can be broken down into one triangular and one rectangular prism.

```
function TypeThree:
    this Procedure will calculate the mass moment of inertia of a volume which can be broken down into one triangular and one rectangular prism.

    h, j, k, m,  
    areaTri,  
    areaRect1,  
    areaRect2,  
    massTri,  
    massRect1,  
    massRect2,  
    momentTri,  
    momentRect1,  
    momentRect2,  
    tri,  
    xCen, yCen,  
    x1, y1,  
    x2, y2,  
    x3, y3,  
    x4, y4,  
    x5, y5,  
    x6, y6,  
    x7, y7,  
    x8, y8,  
    x9, y9,  
    moment1, moment2:real;
```
\[
\begin{align*}
\text{a} &= (a + b) - \text{area}; \\
\text{i} &= \text{sqrt}(\text{i} \times \text{tangentTheta}); \\
h &= j \times \text{tangentTheta}; \\
k &= a - i; \\
m &= b - h; \\
\text{areaTri} &= 0.5 \times \text{r} \times j; \\
\text{areaRect1} &= a \times m; \\
\text{areaRect2} &= h \times k; \\
\text{massTri} &= \text{areaTri} \times c \times \rho; \\
\text{massRect1} &= \text{areaRect1} \times c \times \rho; \\
\text{massRect2} &= \text{areaRect2} \times c \times \rho; \\
x_2 &= k + j/3 - a/2; \\
y_2 &= b/2 - (2 \times h/3); \\
\text{CounterRotate}(x_1,y_1,x_2,y_2); \\
x_3 &= 0; \\
y_3 &= m/2 - b/2; \\
\text{CounterRotate}(x_5,y_5,x_3,y_3); \\
x_4 &= k/2 - a/2; \\
y_4 &= b/2 - h/2; \\
\text{CounterRotate}(x_4,y_4,x_3,y_3); \\
\text{Centroid}(\text{xCent}, \text{yCent}, \text{areaTri}, x_1,y_1, \text{areaRect1}, x_2,y_2, \text{areaRect2}, x_3,y_3); \\
x_7 &= \text{xCent} - x_1; \\
y_7 &= \text{yCent} - y_1; \\
x_8 &= \text{xCent} - x_2; \\
y_8 &= \text{yCent} - y_2; \\
x_9 &= \text{xCent} - x_3; \\
y_9 &= \text{yCent} - y_3; \\
\text{momentTri} &= \text{IBarTri}((\text{massTri},h,0)+(\text{massTri} \times (\text{sqr}(x_7) - \text{sqr}(y_7)))); \\
\text{momentRect1} &= \text{IBarRect}((\text{massRect1},a,m)+(\text{massRect1} \times (\text{sqr}(x_8) + \text{sqr}(y_8)))); \\
\text{momentRect2} &= \text{IBarRect}((\text{massRect2},n,k)+(\text{massRect2} \times (\text{sqr}(x_9) + \text{sqr}(y_9)))); \\
\text{moment} &= \text{momentTri} + \text{momentRect1} + \text{momentRect2};
\end{align*}
\]

\begin{verbatim}
procedure TypeFour;
arc
\text{c4: real;}
\begin{align*}
\text{c4} &= \text{percent} \times b; \\
\text{moment} &= \text{IBarRect}(\text{mass}, a, \text{c4});
\end{align*}
end;

procedure TypeFive;
arc
\text{c5: real;}
\begin{align*}
\text{c5} &= \text{percent} \times a; \\
\text{moment} &= \text{IBarRect}(\text{mass}, \text{c5}, b);
\end{align*}
end;

procedure TypeSix;
arc
\begin{align*}
\text{This Procedure will calculate the mass moment of inertia of a volume which}
\text{can be broken down into one triangular and one rectangular prism.}
\end{align*}
\begin{align*}
\text{area2,} &\quad (\text{Area of Rect}) \\
\text{areaTri,} &\quad (\text{Height of Rect}) \\
\text{areaRect1}, &\quad (\text{Mass of triangular prism}) \\
\text{massTri,} &\quad (\text{Mass of Rect. prism}) \\
\text{momentTri,} &\quad (\text{Moment of inertia of triangular prism rel. to origin}) \\
\text{momentRect}, &\quad (\text{Moment of inertia of Rect. prism rel. to origin}) \\
\text{J, gX, gY,} &\quad (\text{area of tank not filled with fluid})
\end{align*}
\end{verbatim}
begin
areaTri := 0.5 * sqrt(b) * tangentBeta;
areaRect1 := area - areaTri;
h := (area - areaTri)/b;
J := b * tangentBeta;
massTri := areaTri * c * rho;
massRect1 := areaRect1 * c * rho;
gx := h + j/3 - a/2;
gy := b/3 - b/2;
CounterRotate(x2,y2,gx,gy);
x := (h/2) - (a/2);
y := 0.0;
CounterRotate(x1,y1,x,y);
Centroid(xCen,yCen,AreaTri,x2,y2,AreaRect1,x1,y1,zero,zero,zero);
x3 := xCen - x2;
y3 := yCen - y2;
x4 := xCen - x1;
y4 := yCen - y1;
momentTri := iBarTri(massTri,b,j) + (massTri * (sqr(x3) + sqr(y3)));
momentRect := iBarRect(massRect1,b,h) + (massRect1 * (sqr(x4) + sqr(y4)));
moment := momentTri + momentRect;

procedure TypeSeven;
This Procedure will calculate the mass moment of inertia of a volume which
must be broken down into one triangular and two rectangular prisms.

ar
h, j, k, m,
areaTri,
areaRect1,
areaRect2,
massTri, (mass of triangle)
massRect1, (mass of rectangle below triangle)
massRect2, (mass of rectangle next to triangle)
momentTri,
momentRect1,
momentRect2,
tri, (the triangular area not filled with fluid)
xCen, yCen,
ax1, ay1,
ax2, ay2,
ax3, ay3,
x1, x2, x3, x4, x5, x6,
y1, y2, y3, y4, y5, y6,
moment1, moment2:real;

egin
tri := (a * b) - area;
j := sqrt((2 * tri)/tangentBeta);
h := j * tangentBeta;
k := b - j;
m := a - h;
areaTri := 0.5 * h * j;
areaRect1 := b * m;
areaRect2 := h * k;
massTri := areaTri * c * rho;
massRect1 := areaRect1 * c * rho;
massRect2 := areaRect2 * c * rho;
x2 := a/2 - (2 * h/3);
y2 := b/2 - (2 * j/3);
CounterRotate(x1,y1,x2,y2);
x5 := m/2 - a/2;
y5 := 0.0;
CounterRotate x, y, z, E, E;
\[ a := y_{-1} - y_{-2}; \]
\[ \theta := \frac{y_{-1}}{y_{-2}}; \]
CounterRotate x, y, z, \theta, \theta;

\text{Centroid}(x\text{Cen}, y\text{Cen}, a\text{Cen}, x_1, y_1, \text{areaTri}, n, x_2, y_2, \text{areaRect2}, x_3, y_3);
\[ a_{x1} := x\text{Cen} - x_1; \]
\[ a_{y1} := y\text{Cen} - y_1; \]
\[ a_{x2} := x\text{Cen} - x_2; \]
\[ a_{y2} := y\text{Cen} - y_2; \]
\[ a_{x3} := x\text{Cen} - x_3; \]
\[ a_{y3} := y\text{Cen} - y_3; \]
\[ \text{momentTri} := I_BaTri(massTri, n, j) \times (\text{massTri} \times (\text{areaTri} + \text{massTri} \times \text{areaRect1} + \text{massTri} \times \text{areaRect2} + \text{massTri} \times \text{areaRect3})); \]
\[ \text{momentRect1} := I_BaRect(massRect1, l, m) \times (\text{massRect1} \times (\text{momentRect1} + \text{massRect2} \times (\text{momentRect2} + \text{massRect2} \times (\text{momentRect2} + \text{massRect2} \times (\text{massRect2} \times \text{areaRect2})))))); \]
\[ \text{momentRect2} := I_BaRect(massRect2, h, k) \times (\text{massRect2} \times (\text{momentRect2} + \text{massRect2} \times (\text{momentRect2} + \text{massRect2} \times (\text{massRect2} \times \text{areaRect2})))))); \]
\[ \text{moment} := \text{momentTri} + \text{momentRect1} + \text{momentRect2}; \]

*****************************************************************************
*** (START MAIN PROGRAM) ***

ClearScreen;
zero := 0;
writeln('Enter dimensions for rectangular tank:');
readln(a, b, c);
writeln('Enter water level as a percentage of "b":');
readln(p_percent);
while (p_percent < 1) or (p_percent > 0) do
begin
  writeln('0.0 < Water Level < 1.0. Try again.');
  readln(p_percent);
end;
writeln('Enter angle theta increment.');
readln(increment);
writeln('Enter output file name.');
readln(outFile);
angle := 0.0;
area := a * b * percent;
mass := a * percent * b * c;
rewrote(outFile, outFile);
writeln(outFile, 'H = ', a * b * c);
writeln(outFile, 'B = ', b * c);
writeln(outFile, 'C = ', c);
writeln(outFile, 'Percent full = ', percent * b * c);
writeln(outFile, 'Type Theta Moment');
while (angle <= 90.0) do
begin
  moment := 0.0;
beta := (90 - angle) / 360 \times (2 \times pi);
theta := (angle / 360) \times (2 \times pi);
tangentBeta := sin(beta) / cos(beta);
tangentTheta := sin(theta) / cos(theta);
form := WhatType;
case (form) of
  1: TypeOne;
  2: TypeTwo;
  3: TypeThree;
  4: TypeFour;
  5: TypeFive;
  6: TypeSix;
  7: TypeSeven;
end;
writeln(outFile, form, ', ', angle:0.2, ', ', moment:8.4);
angle := angle + increment;
end;
close(outFile);
(*END MAIN PROGRAM*)
This program calculates the mass moment of inertia of the fluid in a tank relative to the center of the tank. The tank is dimensioned as \( a \) in the \( x \)-dir, \( b \) in the \( y \)-dir (up), and \( c \) in the \( z \)-dir (into the page). The center of the tank is also the center of rotation and corresponds to the origin of the coordinates. The tank rotates counterclockwise and this angle is defined as \( \theta \). This version was last modified on 2/4/90.

Uses

\text{PasPrinter;}
\text{Const}
\pi = 3.141593;
\rho = 1.0;
\text{type}
\textit{Number} = \text{array}[1..90]\text{ of real;}
\text{var}
\textit{outFile} : \text{text}; \text{(file pointer for output)}
\textit{outFileName} : \text{string}[15];
\textit{output1: Number;}
\textit{output2: Number;}
\textit{printer: char;}
\textit{a, b, c, area, angle, increment, aTriangle, bTriangle, percent, mass, moment, beta, tangentBeta, tangentTheta, theta: real;}
\textit{form: integer;}

\begin{itemize}
  \item \text{length of tank}
  \item \text{height of tank}
  \item \text{width of tank}
  \item \text{the angle of rotation in degrees}
  \item \text{increment by which angle increases}
  \item \text{Max area of triangular shape}
  \item \text{Max area of triangle and rectangle combined}
  \item \text{Fluid level as a percent of height \( b \)}
  \item \text{Mass of fluid}
  \item \text{Mass moment of inertia of entire shape}
  \item \text{\( 2 \theta \)}
  \item \text{\( \tan(\beta) \)}
  \item \text{\( \tan(\theta) \)}
  \item \text{Angle of rotation CCW in radians}
  \item \text{(defines the geometry)}
\end{itemize}

\text{procedure CounterRotate(var ax, ay, axl, ayl: real);}
\text{This procedure converts local Cord. to global Cord. given angle \( \theta \).}
\begin{align*}
\text{begin}
  ax &\leftarrow (axl * \cos(\theta)) \quad \text{(axl * sin(\theta))}; \\
  ay &\leftarrow (axl * \cos(\theta)) \quad \text{(ayl * sin(\theta))};
\end{align*}
\text{end;}

\text{function IBarTri(var mTri, ta, tb: real): real;}
\text{This function will calculate the mass moment of inertia of a triangular prism given its mass, base, and height.}
\begin{align*}
\text{begin}
  \text{IBarTri} &\leftarrow (mTri/18) \quad \text{*(sqr(ta) + sqr(tb))};
\end{align*}
\text{end;}

\text{function IBarRect(var mRect, ra, rb: real): real;}
\text{This function will calculate the mass moment of inertia of a rectangular prism given its mass, length, and height.}
\begin{align*}
\text{begin}
  \text{IBarRect} &\leftarrow (mRect/12) \quad \text{*(sqr(ra) + sqr(rb))};
\end{align*}
\text{end;}

\text{function WhatType: integer;}
\text{This function determines what shape the fluid is in. Type 1 = triangle; Type 2 = triangle and rectangle; and type 3 = one triangle and two}
\text{rectangular
area, alpha, bArea : real;

\begin{align*}
    aTriangle & := 0.5 \ast \text{sqr}(a) \ast \text{tangentTheta}; \\
    aArea & := aTriangle + (a \ast (b - (a \ast \text{tangentTheta}))) \\
    alpha & := \text{arctan}(b/a); \\
    \text{if tangentTheta} < 0.0 \text{ then} & \\
    \begin{align*}
        bTriangle & := \text{abs}(0.5 \ast \text{sqr}(b))/\text{tangentTheta}; \\
        bArea & := bTriangle + \text{abs}(b \ast (a - (b/\text{tangentTheta}))); \\
    \end{align*}
\end{align*}

\begin{align*}
    \text{if (angle} = 0.0 \text{) then} & \quad (\text{It's a rectangle.)} \\
    \text{WhatType} & := 4 \\
    \text{else if (angle} = 90.0 \text{) then} \quad (\text{It's still a rectangle.)} \\
    \text{WhatType} & := 5 \\
    \text{else if theta} \leq alpha \text{ then} & \\
    \begin{align*}
        \text{if (aTriangle} \ast \text{area) then} & \\
        \text{WhatType} & := 1 \\
        \text{else if (aArea} \ast \text{area) and (bArea} \ast \text{area) then} & \\
        \text{WhatType} & := 3 \\
        \text{else WhatType} & := 3; \\
    \end{align*}
\end{align*}

\begin{align*}
    \text{if (bTriangle} \ast \text{area) then} & \\
    \text{WhatType} & := 1 \\
    \text{else if (bArea} \ast \text{area) then} & \\
    \text{WhatType} & := 6 \\
    \text{else WhatType} & := 7; \\
\end{align*}

\text{nd;}

\text{procedure TypeOne;}
\text{This procedure will calculate the mass moment of inertia of a triangular prism.}
\text{ar a2, b2, h, j, cx, cy: real;}
\text{egiin h := sqrt((2 \ast \text{area})/\text{tangentTheta}) \ast \text{tangentTheta};}
\text{j := sqrt((2 \ast \text{area})/\text{tangentTheta});}
\text{a2 := (j/3) - (a/2);}
\text{b2 := (h/3) - (b/2);}
\text{CounterRotate(cx, cy, a2, b2);}
\text{moment := IBarTri(mass, h, j) + \text{(mass} \ast (\text{sqr}(\text{cx}) + \text{sqr}(\text{cy})))};
\text{nd;}

\text{procedure TypeTwo;}
\text{This Procedure will calculate the mass moment of inertia of a volume which can be broken down into one triangular and one rectangular prism.}
\text{ar area2, (Area of Rect) h, (Height of Rect) massTri, (Mass of triangular prism) massRect, (Mass of Rect. prism) momentTri, (Moment of inertia of triangular prism rel. to origin) momentRect, (Moment of inertia of Rect. prism rel. to origin) j, gx, gy, xbar, ybar, x, y, x3, y3\text{: real;}
\text{egiin h := (\text{area} - aTriangle)/a;}
\text{j := a \ast \text{tangentTheta};}
\text{massTri := aTriangle \ast c \ast \text{rho};}
\text{massRect := (area - aTriangle) \ast c \ast \text{rho};}
\text{gx := a/3 \ast a/2;}
\text{nx := h + i/3 \ast h/2;}
\text{nd;}

Original Page is of Poor Quality
CounterRotate(xbar,ybar,gx,gy);
momentTri := IBarTri(massTri,h,j) + (massTri * (sqr(xbar) + sqr(ybar)));
x := 0;
y := h/2 - b/2;
CounterRotate(x3,y3,x,y);
momentRect := IBarRect(massRect,a,h) + (massRect * (sqr(x3) + sqr(y3)));
moment := momentTri + momentRect;

procedure TypeThree;
{This Procedure will calculate the mass moment of inertia of a volume which
must be broken down into one triangular and two rectangular prisms.}
var
h, j, k, m,
massTri, {mass of triangle}
massRect1, {mass of rectangle below triangle}
massRect2, {mass of rectangle next to triangle}
momentTri,
momentRect1,
momentRect2,
tri, {the triangular area not filled with fluid}
x1, x2, x3, x4, x5, x6,
y1, y2, y3, y4, y5, y6,
moment1, moment2:real;
begin
tri := (a * b) - area;
j := sqrt((2 * tri)/tangentTheta);
h := j * tangentTheta;
k := a - j;
m := b - h;
massTri := 0.5 * h * j * c * rho;
massRect1 := a * m * c * rho;
massRect2 := h * k * c * rho;
x2 := k + j/3 - a/2;
y2 := b/2 - (2 * h/3);
CounterRotate(x1,y1,x2,y2);
momentTri := IBarTri(massTri,h,j) + (massTri * (sqr(x1) + sqr(y1)));
x5 := 0;
y5 := m/2 - b/2;
CounterRotate(x6,y6,x5,y5);
momentRect1 := IBarRect(massRect1,a,m)+(massRect1 * (sqr(x6) + sqr(y6)));
x4 := k/2 - a/2;
y4 := b/2 - h/2;
CounterRotate(x3,y3,x4,y4);
momentRect2 := IBarRect(massRect2,h,k)+(massRect2 * (sqr(x3) + sqr(y3)));
moment := momentTri + momentRect1 + momentRect2;
end;

procedure TypeFour;
var
c4:real;
begin
c4 := percent * b;
moment := IBarRect(mass,a,c4) + (mass * sqr((b/2) * (percent - 1)))
end;

procedure TypeFive;
var
c5:real;
begin
c5 := percent * a;
moment := IBarRect(mass,c5,b) + (mass * (sqr((a/2) * (percent - 1))))
end;
Procedure TypeSix;
(This Procedure will calculate the mass moment of inertia of a volume which)
(can be broken down into one triangular and one rectangular prism.)

var
  area2, {Area of Rect}
  areaTriangle, {Height of Rect}
  h, {Mass of triangular prism}
  massRect, {Mass of rectangular prism}
  momentTri, {Moment of inertia of triangular prism rel. to origin}
  momentRect, {Moment of inertia of Rect. prism rel. to origin}
  j, gx, gy, xbar, ybar,
  x, y, x3, y3:real;

begin
  areaTriangle := 0.5 * sqr(b) * tangentBeta;
  h := (area - areaTriangle)/b;
  j := b * tangentBeta;
  massTri := areaTriangle * c * rho;
  massRect := (area - areaTriangle) * c * rho;
  gx := h + j/3 - a/2;
  gy := b/3 - b/2;
  CounterRotate(xbar,ybar,gx,gy);
  momentTri := iBarTri(massTri,b,J);
  x := (h/2) - (a/2);
  y := 0.0;
  CounterRotate(x3,y3,x,y);
  momentRect := iBarRect(massRect,b,h);
  moment := momentTri + momentRect;
end;

Procedure TypeSeven;
(This Procedure will calculate the mass moment of inertia of a volume which)
(must be broken down into one triangular and two rectangular prisms.)

var
  h, j, k, m,
  massTri, {mass of triangle}
  massRect1, {mass of rectangle below triangle}
  massRect2, {mass of rectangle next to triangle}
  momentTri,
  momentRect1,
  momentRect2,
  tri, {the triangular area not filled with fluid}
  x1, x2, x3, x4, x5, x6,
  y1, y2, y3, y4, y5, y6,
  moment1, moment2:real;

begin
  tri := (a * b) - area;
  j := sqrt((2 * tri)/tangentBeta);
  h := j * tangentBeta;
  k := b - j;
  m := a - h;
  massTri := 0.5 * h * j * c * rho;
  massRect1 := b * m * c * rho;
  massRect2 := h * k * c * rho;
  x2 := a/2 - (2 * h/3);
  y2 := b/2 - (2 * j/3);
  CounterRotate(x1,y1,x2,y2);
  momentTri := iBarTri(massTri,h,j) + (massTri * (sqr(x1) + sqr(y1)));
  x5 := m/2 - a/2;
  y5 := 0.0;
  CounterRotate(x6,y6,x5,y5);
  momentRect1 := iBarRect(massRect1,b,m) + (massRect1 * (sqr(x6) + sqr(y6)));
  x4 := a/2 - h/2;
  y4 := k/2 - b/2;
  CounterRotate(x3,y3,x4,y4);
  momentRect2 := iBarRect(massRect2,b,h) + (massRect2 * (sqr(x3) + sqr(y3)));
  moment := momentTri + momentRect1 + momentRect2;
ClearScreen;
writeln ('Enter dimensions for rectangular tank:');
readln (a,b,c);
writeln ('Enter water level as a percentage of "b".');
readln (percent);
while ((percent) = 1) or (percent = 0) do
begin
  writeln ('0.0 < Water Level < 1.0. Try again.');
  readln (percent);
end;
writeln ('Enter angle theta increment.');
readln (increment);
writeln ('Enter output file name.');
readln (outFileName);
angle := 0.0;
area := a * b * percent;
mass := a * percent * b * c;
rewrite (outFile, outFileName);
writeln (outFile, 'A = ', a:8:4);
writeln (outFile, 'B = ', b:8:4);
writeln (outFile, 'C = ', c:8:4);
writeln (outFile, 'Percent full = ', percent:8:4);
writeln (outFile, 'Type Theta Moment');
while (angle <= 90.0) do
begin
  moment := 0.0;
  beta := ((90 - angle)/360)*(2 * pi);
  theta := (angle/360)*(2 * pi);
  tangentBeta := sin(beta)/cos(beta);
  tangentTheta := sin(theta)/cos(theta);
  form := WhatType;
  case (form) of
    1: TypeOne;
    2: TypeTwo;
    3: TypeThree;
    4: TypeFour;
    5: TypeFive;
    6: TypeSix;
    7: TypeSeven;
  end;
  writeln (outFile, form, ', ', angle:6:2, ', ', moment:8:4);
  angle := angle + increment;
end;
close (outFile);
end. (*END MAIN PROGRAM*)
APPENDIX B

PROGRAM LISTINGS
program momentOfInertia;

This program calculates the mass moment of inertia of the fluid in a tank.

var

ax1, ay1, ax2, ay2, ax3, ay3, a1, a2, a3, theta: real;

begin

ax := (ax1 * cos(theta)) - (ax2 * sin(theta));
ay := (ay1 * sin(theta)) + (ay2 * cos(theta));

function iBarTri(var mTri, ta, tb: real): real;

This function will calculate the mass moment of inertia of a triangular prism given its mass, base, and height.

begin

iBarTri := (mTri / 18) * (sqr(ta) + sqr(tb));

end;

function iBarRect(var mRect, ra, rb: real): real;

This function will calculate the mass moment of inertia of a rectangular prism given its mass, length, and height.

begin

iBarRect := (mRect / 12) * (sqr(ra) + sqr(rb));

end;

procedure Centroid(var xBar, yBar, a1, x1, y1,

a2, x2, y2,

a3, x3, y3: real);

end.
This procedure will calculate the x and y coord of the centroid of the fluid.

var
area, xBar, yBar:real;

begin
area := (a1 * v1) + (v2 * v2) + (v3 - v3);

xBar := xarea/area;
yBar := yarea/area;
end:

Function WhatType:integer;
This function determines what shape the fluid is in. Type 1 = triangle; *
Type 2 = triangle and rectangles; and type 3 = one triangle and two *
rectangles.
var
area, alpha, barea:real;

begin
triangle := 0.5 * sin(alpha) * tangentTheta;
area := triangle + a * (b - (a + tangentTheta));
alpha := atan(b/a);
if tangentTheta < 0.0 then
begin
triangle := abs(0.5 * sin(alpha) * tangentTheta);
area := triangle + abs(c * c - c * tangentTheta);
end;
if angle = 0.0 then (It's a rectangle.)
WhatType := 4
else if angle = 90.0 then (It's still a rectangle.)
WhatType := 5
else if theta = alpha then
begin
if (aTriangle > area) then
WhatType := 1
else if (bTriangle > area) then
WhatType := 2
else WhatType := 3;
end
else
begin
if (bTriangle > area) then
WhatType := 1
else if (barea > area) then
WhatType := 6
else WhatType := 7;
end;
end;

procedure TypeOne;
This procedure will calculate the mass moment of inertia of a triangular prism.
var
a2, b2, h, j, cx, cy:real;

begin
h := sqrt((2 * area)/tangentTheta) * tangentTheta;
j := sqrt((2 * area)/tangentTheta);
a2 := (j/3) - (a2/2);
b2 := (h/3) - (b2/2);
CounterRotate(cx, cy, a2, b2);
moment := 16aTriTricMass,h,j;
end;

procedure TypeTwo;

ORIGINAL PAGE IS OF POOR QUALITY
This procedure will calculate the mass moment of inertia of a volume which can be broken down into one triangular and one rectangular prism.

```plaintext
areaTri, areaRect1, h, (Height of Rect)
massTri, (mass of triangular prism)
massRect1, (mass of Rect. prism)
momentTri, (Moment of inertia of triangular prism rel. to origin)
momentRect1, (Moment of inertia of Rect. prism rel. to origin)
J, k1, k2, x1, y1,
x2, y2,
x3, y3,
x4, y4,
x5, y5,
x6, y6,
x7, y7,
x8, y8,
x9, y9,

;areaTri := area - areaTri
;areaRect1 := area - areaTri
;massTri := areaTri * (1 + rho);
massRect1 := areaRect1 * (1 + rho);
;px := a - b,.
;cy := h - (h - b)/2;
;counterRotate (E, px, cy);
;x1 := 0;
;x2 := h - h/2;
;counterRotate (E, x1, cy);

;Centroid (Cent, Cent, areaTri, x2, y2, areaRect1, x3, y3, zero, zero, zero);
;v := Cent - (x1);
;v4 := Cent - (x1);
;v5 := Cent - (x1);

;moment Tri := (massTri * (sin(v4) + sin(v5))/;
moment Rect1 := (massRect1 * (sin(v4) + sin(v5))/;
moment := momentTri + momentRect1 + momentRect2;

```

Procedure TypeThree:
This procedure will calculate the mass moment of inertia of a volume which must be broken down into one triangular and two rectangular prisms.

```plaintext
h, j, k, m,
areaTri, areaRect1, areaRect2,
massTri, (mass of triangle)
massRect1, (mass of rectangle below triangle)
massRect2, (mass of rectangle next to triangle)
momentTri,
momentRect1, momentRect2,
tri, (the triangular area not filled with fluid)
Cent, Cent, (centroids coord of the fluid)
x1, y1,
x2, y2,
x3, y3,
x4, y4,
x5, y5,
x6, y6,
x7, y7,
x8, y8,
x9, y9,
moment1, moment2, real;
```

ORIGINAL PAGE IS OF POOR QUALITY
\begin{verbatim}
BEGIN

n1 := (a * b) - area;

j := sqrt((b * tri)/tangentTheta);

t := j * tangentTheta;

k := a - t;

m := c - h;

areaTri := C.5 * h * j;

areaRect1 := a * m;

areaRect2 := h * k;

massTri := areaTri * c * rho;

massRect1 := areaRect1 * c * rho;

massRect2 := areaRect2 * c * rho;

x2 := k + j/3 - a/2;
y2 := b/2 - (2 * h/3);

CounterRotate(x1,x2,y2);

x3 := C;
y3 := m/2 - b/2;

CounterRotate(x2,y3,x5,y5);

x4 := a - a/2;
y4 := b/2 - h/2;

CounterRotate(x3,y3,x4,y4);

Centroid(x1, y1, x2, y2, x3, y3, x4, y4, x5, y5, x6, y6, areaRect1, x6, y6, areaRect2, x3, y3);

y7 := x6 - x1;
y8 := x6 - x1;
y9 := y6 - y1;
y10 := y6 - y1;

momentTri := (massTri - massTri) + (cos(y7) - cos(y8));

momentRect1 := (massRect1 - massRect1) * (cos(y1) + cos(y9));

momentRect2 := (massRect2 - massRect2) * (cos(y3) + cos(y5));

moment := momentTri + momentRect1 + momentRect2;

PROCEDURE TypeFour;

AR

c4: real;

BEGIN

c4 := percent + c;

moment := IBarRect(mass, c4);

END;

PROCEDURE TypeFive;

AR

c5: real;

BEGIN

c5 := percent + a;

moment := IBarRect(mass, c5, b);

END;

PROCEDURE TypeSix;

(* This Procedure will calculate the mass moment of inertia of a volume which can be broken down into one triangular and one rectangular prism. *)

AR

area2, (Area of Rect)

areaTri, (Area of Tri)

h, (Height of Rect)

massTri, (Mass of triangular prism)

massRect, (Mass of Rect. prism)

momentTri, (Moment of inertia of triangular prism rel. to origin)

momentRect, (Moment of inertia of Rect. prism rel. to origin)

J, g, v, x, y, z, tri, (area of tank not filled with fluid)

END.
\end{verbatim}
areaTr := 0.5 + a * b * tangentBeta;
areaRect1 := area - areaTri;
h := area = areaTri / b;
= c * tangentBeta;
massTri := areaTri * c * rho;
massRect1 := areaRect1 * c * rho;
gx := h - a / 3 - a / 2;
gz := a / 2 - b / 2;
CounterRotate(x2, z4, y3, y4);
:= x2 = a / 2;
y := (a / 2);

Center(x1, y1, z1, z2, z3, z4, z5, z6, z7, y5, y6, y7, y8);
in
:= areaTri := (massTri * h) + (massRect1 * (sin(x3) + sin(x4)));
moment := (momentRect1 + momentTri) - momentTri;

notate Thencever;
The Procedure will calculate the mass moment of inertia of a volume which
must be broken down into one triangular and two rectangular prisms.

eIn
tri := (a - b) * area;
j := sqrt((2 * tri) / tangentBeta);
h := j * tangentBeta;
= b = b - j;
m := a - h;
areaTri := 0.5 * h * j;
areaRect1 := b * m;
areaRect2 := h * j;
massTri := areaTri * c * rho;
massRect1 := areaRect1 * c * rho;
massRect2 := areaRect2 * c * rho;
:= b := (a / 2) - (c * h / 3);
y2 := b / 2 - (c / 2 * j / 3);
CounterRotate(x1, y1, z1, z2); x5 := m / 2 - a / 2;
v5 := 0;
**ORIGINAL PAGE IS OF POOR QUALITY**
Program Moment1:
(This program calculates the mass moment of inertia of the fluid in a tank )
relative the to the center of the tank. The tank is dimensioned as 'a' in the x-dir, 'b' in the y-dir(up), and 'c' in the z-dir(into the page). The center of the tank is also the center of rotation and corresponds to the origin of the coordinates. The tank rotates counterclockwise and this angle is defined as theta. This version was last modified on 2/4/90.
Uses PasPrinter;
Const
pi = 3.141593;  
rho = 1.0;
Type
Number = array [1..90] of real;
Var
outFile : text; (file pointer for output)
outFileName : string[15];
output1Number;
output2Number;
printer:char;
a,  (length of tank)
b,  (height of tank)
c,  (width of tank)
area,  (the angle of rotation in degrees)
angle,  (Increment by which angle increases)
ATriangle,  (Max area of triangular shape)
BTriangle,  (Max area of triangle and rectangle combined)
percent,  (Fluid level as a percent of height "b")
mass,  (Mass of fluid)
moment,  (Mass moment of inertia of entire shape)
beta,  (2 - theta)
tangentBeta,  (tan(beta))
tangentTheta,  (tan(theta))
theta:real;  (Angle of rotation CCW in radians)
form:integer;  (defines the geometry)
*******************************************************************************
Procedure CounterRotate(var ax, ay, axl, ayI:real);
(This procedure converts local Cord. to global Cord. given angle theta.)
Begin
(ax := (axl * cos(theta)) - (ayl * sin(theta));
(ay := (axl * sin(theta)) ÷ (ayl * cos(theta));
End;

Function IBarTri(var mTri,ta, tb:real):real;
(This function will calculate the mass moment of inertia of a triangular prism given its mass, base, and height.)
Begin
IBarTri := (mTri/18) * (sqr(ta) + sqr(tb));
End;

Function IBarRect(var mRect,ra,rb:real):real;
(This function will calculate the mass moment of inertia of a rectangular prism given its mass, length, and height.)
Begin
IBarRect := (mRect/12) * (sqr(ra) + sqr(rb));
End;

Function WhatType:integer;
(This function determines what shape the fluid is in. Type 1 = triangle; *)
*type 2 = triangle and rectangle; and type 3 = one triangle and two *)
  *insertions)
Begin
End;
ar

\begin{verbatim}

\def \aArea {\alpha, \text{alpha :real;}}
\def \aTriangle {0.5 * \text{sqr}(a) * \text{tangentTheta;}}
\def \aArea {\aTriangle + (a * (b - (a * \text{tangentTheta})))}
\def \alpha {\text{arctan}(b/a);}
if \text{tangentTheta} <> 0.0 then
begin
  \def \bTriangle {abs((0.5 * \text{sqr}(b))/\text{tangentTheta);}}
  \def \bArea {\bTriangle + abs((b * (a - (b/\text{tangentTheta})))}}
end;
if (angle = 0.0) then (It's a rectangle.)
  \def \WhatType {4}
else if (angle = 90.0) then (It's still a rectangle.)
  \def \WhatType {5}
else if \text{theta} <= \text{alpha} then
begin
  if (\aTriangle > \text{area}) then
    \def \WhatType {1}
  else if ((\aArea > \text{area}) and (\bArea > \text{area})) then
    \def \WhatType {2}
  else \def \WhatType {3};
end
else
begin
  if (\bTriangle > \text{area}) then
    \def \WhatType {1}
  else if (\bArea > \text{area}) then
    \def \WhatType {6}
  else \def \WhatType {7};
end;
\end{verbatim}

\textbf{procedure TypeOne;}
This procedure will calculate the mass moment of inertia of a triangular prism.

\begin{verbatim}

\def \aArea {\alpha, \text{alpha :real;}}
\def \aTriangle {0.5 * \text{sqr}(a) * \text{tangentTheta;}}
\def \aArea {\aTriangle + (a * (b - (a * \text{tangentTheta})))}
\def \alpha {\text{arctan}(b/a);}
if \text{tangentTheta} <> 0.0 then
begin
  \def \bTriangle {abs((0.5 * \text{sqr}(b))/\text{tangentTheta);}}
  \def \bArea {\bTriangle + abs((b * (a - (b/\text{tangentTheta})))}}
end;
if (angle = 0.0) then (It's a rectangle.)
  \def \WhatType {4}
else if (angle = 90.0) then (It's still a rectangle.)
  \def \WhatType {5}
else if \text{theta} <= \text{alpha} then
begin
  if (\aTriangle > \text{area}) then
    \def \WhatType {1}
  else if ((\aArea > \text{area}) and (\bArea > \text{area})) then
    \def \WhatType {2}
  else \def \WhatType {3};
end
else
begin
  if (\bTriangle > \text{area}) then
    \def \WhatType {1}
  else if (\bArea > \text{area}) then
    \def \WhatType {6}
  else \def \WhatType {7};
end;
\end{verbatim}

\textbf{procedure TypeTwo;}
This procedure will calculate the mass moment of inertia of a volume which can be broken down into one triangular and one rectangular prism.

\begin{verbatim}

\def \aArea {\alpha, \text{alpha :real;}}
\def \aTriangle {0.5 * \text{sqr}(a) * \text{tangentTheta;}}
\def \aArea {\aTriangle + (a * (b - (a * \text{tangentTheta})))}
\def \alpha {\text{arctan}(b/a);}
if \text{tangentTheta} <> 0.0 then
begin
  \def \bTriangle {abs((0.5 * \text{sqr}(b))/\text{tangentTheta);}}
  \def \bArea {\bTriangle + abs((b * (a - (b/\text{tangentTheta})))}}
end;
if (angle = 0.0) then (It's a rectangle.)
  \def \WhatType {4}
else if (angle = 90.0) then (It's still a rectangle.)
  \def \WhatType {5}
else if \text{theta} <= \text{alpha} then
begin
  if (\aTriangle > \text{area}) then
    \def \WhatType {1}
  else if ((\aArea > \text{area}) and (\bArea > \text{area})) then
    \def \WhatType {2}
  else \def \WhatType {3};
end
else
begin
  if (\bTriangle > \text{area}) then
    \def \WhatType {1}
  else if (\bArea > \text{area}) then
    \def \WhatType {6}
  else \def \WhatType {7};
end;
\end{verbatim}

\textbf{procedure TypeOne;}
This procedure will calculate the mass moment of inertia of a triangular prism.

\begin{verbatim}

\def \aArea {\alpha, \text{alpha :real;}}
\def \aTriangle {0.5 * \text{sqr}(a) * \text{tangentTheta;}}
\def \aArea {\aTriangle + (a * (b - (a * \text{tangentTheta})))}
\def \alpha {\text{arctan}(b/a);}
if \text{tangentTheta} <> 0.0 then
begin
  \def \bTriangle {abs((0.5 * \text{sqr}(b))/\text{tangentTheta);}}
  \def \bArea {\bTriangle + abs((b * (a - (b/\text{tangentTheta})))}}
end;
if (angle = 0.0) then (It's a rectangle.)
  \def \WhatType {4}
else if (angle = 90.0) then (It's still a rectangle.)
  \def \WhatType {5}
else if \text{theta} <= \text{alpha} then
begin
  if (\aTriangle > \text{area}) then
    \def \WhatType {1}
  else if ((\aArea > \text{area}) and (\bArea > \text{area})) then
    \def \WhatType {2}
  else \def \WhatType {3};
end
else
begin
  if (\bTriangle > \text{area}) then
    \def \WhatType {1}
  else if (\bArea > \text{area}) then
    \def \WhatType {6}
  else \def \WhatType {7};
end;
\end{verbatim}

\textbf{procedure TypeTwo;}
This procedure will calculate the mass moment of inertia of a volume which can be broken down into one triangular and one rectangular prism.

\begin{verbatim}

\def \aArea {\alpha, \text{alpha :real;}}
\def \aTriangle {0.5 * \text{sqr}(a) * \text{tangentTheta;}}
\def \aArea {\aTriangle + (a * (b - (a * \text{tangentTheta})))}
\def \alpha {\text{arctan}(b/a);}
if \text{tangentTheta} <> 0.0 then
begin
  \def \bTriangle {abs((0.5 * \text{sqr}(b))/\text{tangentTheta);}}
  \def \bArea {\bTriangle + abs((b * (a - (b/\text{tangentTheta})))}}
end;
if (angle = 0.0) then (It's a rectangle.)
  \def \WhatType {4}
else if (angle = 90.0) then (It's still a rectangle.)
  \def \WhatType {5}
else if \text{theta} <= \text{alpha} then
begin
  if (\aTriangle > \text{area}) then
    \def \WhatType {1}
  else if ((\aArea > \text{area}) and (\bArea > \text{area})) then
    \def \WhatType {2}
  else \def \WhatType {3};
end
else
begin
  if (\bTriangle > \text{area}) then
    \def \WhatType {1}
  else if (\bArea > \text{area}) then
    \def \WhatType {6}
  else \def \WhatType {7};
end;
\end{verbatim}
CounterRotate(xbar,ybar,gx,gy);
momentTri := IBarTri(massTri,a,j) + (massTri * (sqr(xbar) + sqr(ybar)));
x := 0;
y := h/2 - b/2;
CounterRotate(x3,y3,x,y);
momentRect := IBarRect(massRect,a,h) + (massRect * (sqr(x3) + sqr(y3)));
moment := momentTri + momentRect;

Procedure TypeThree;
(This Procedure will calculate the mass moment of inertia of a volume which) (must be broken down into one triangular and two rectangular prisms. )
var
    h, j, k, m,
    massTri, (mass of triangle)
    massRect1, (mass of rectangle below triangle)
    massRect2, (mass of rectangle next to triangle)
    momentTri,
    momentRect1,
    momentRect2,
    tri, (the triangular area not filled with fluid)
    x1, x2, x3, x4, x5, x6,
    y1, y2, y3, y4, y5, y6,
    moment1, moment2:real;
begin
    tri := (a * b) - area;
    j := sqrt((2 * tri)/tangentTheta);
    h := j * tangentTheta;
    k := a - j;
    m := b - h;
    massTri := 0.5 * h * j * c * rho;
    massRect1 := a * m * c * rho;
    massRect2 := h * k * c * rho;
    x2 := k + j/3 - a/2;
    y2 := b/2 - (2 * h/3);
    CounterRotate(x1,y1,x2,y2);
    momentTri := IBarTri(massTri,h,j) + (massTri * (sqr(x1) + sqr(y1)));
    x5 := 0;
    y5 := m/2 - b/2;
    CounterRotate(x6,y6,x5,y5);
    momentRect1 := IBarRect(massRect1,a,m)+(massRect1 * (sqr(x6) + sqr(y6)));
    x4 := k/2 - a/2;
    y4 := b/2 - h/2;
    CounterRotate(x3,y3,x4,y4);
    momentRect2 := IBarRect(massRect2,h,k)+(massRect2 * (sqr(x3) + sqr(y3)));
    moment := momentTri + momentRect1 + momentRect2;
end;

Procedure TypeFour;
var
    c4:real;
begin
    c4 := percent * b;
    moment := IBarRect(mass,a,c4) + (mass * sqr((b/2) * (percent - 1)));
end;

Procedure TypeFive;
var
    c5:real;
begin
    c5 := percent * a;
    moment := IBarRect(mass,c5,b) + (mass * (sqr((a/2) * (percent - 1))));
end;
Procedure TypeSix;
(This Procedure will calculate the mass moment of inertia of a volume which)
(can be broken down into one triangular and one rectangular prism.)
var
area2,  (Area of Rect)
areaTriangle,  (Height of Rect)
massTri,  (Mass of triangular prism)
massRect,  (Mass of Rect. prism)
momentTri,  (Moment of inertia of triangular prism rel. to origin)
momentRect,  (Moment of inertia of Rect. prism rel. to origin)
j, gx, gy, xbar, ybar,
x, y, x3, y3:real;
begin
areaTriangle := 0.5 * sqr(b) * tangentBeta;
h := (area - areaTriangle)/b;
j := b * tangentBeta;
massTri := areaTriangle * c * rho;
massRect := (area - areaTriangle) * c * rho;
gx := h + j/3 - a/2;
gy := b/3 - b/2;
CounterRotate(xbar,ybar,gx,gy);
momentTri := IBarTri(massTri,b,j) + (massTri * (sqr(xbar) + sqr(ybar)));
momentRect := IBarRect(massRect,b,h) + (massRect * (sqr(x3) + sqr(y3)));
moment := momentTri + momentRect;
end;

Procedure TypeSeven;
(This Procedure will calculate the mass moment of inertia of a volume which)
(must be broken down into one triangular and two rectangular prisms.)
var
h, j, k, m,
massTri,  (mass of triangle)
massRect1,  (mass of rectangle below triangle)
massRect2,  (mass of rectangle next to triangle)
momentTri,
momentRect1,
momentRect2,
tri,  (the triangular area not filled with fluid)
x1, x2, x3, x4, x5, x6,
y1, y2, y3, y4, y5, y6,
moment1, moment2:real;
begin
tri := (a * b) - area;
j := sqrt((2 * tri)/tangentBeta);
h := j * tangentBeta;
k := a - h;
m := a - h;
massTri := 0.5 * h * j * c * rho;
massRect1 := b * m * c * rho;
massRect2 := h * k * c * rho;
x2 := a/2 - (2 * h/3);
y2 := b/2 - (2 * j/3);
CounterRotate(x1,y1,x2,y2);
momentTri := IBarTri(massTri,h,j)+1massTri * (sqr(x1) + sqr(y1)));
x5 := m/2 - a/2;
y5 := 0;
CounterRotate(x6,y6,x5,y5);
momentRect1 := IBarRect(massRect1,b,m)+(massRect1 * (sqr(x6) + sqr(y6)));
x4 := a/2 - h/2;
y4 := k/2 - b/2;
CounterRotate(x3,y3,x4,y4);
momentRect2 := IBarRect(massRect2,h,k)+(massRect2 * (sqr(x3) + sqr(y3)));
moment := momentTri + momentRect1 + momentRect2;
ClearScreen;
writeln ('Enter dimensions for rectangular tank:');
readln (a,b,c);
writeln ('Enter water level as a percentage of "b":');
readln (percent);
while ((percent )= 1) or (percent (= 0)) do
begin
    writeln('0.0 ( Water Level < 1.0. Try again.');
    readln(percent);
end;
writeln('Enter angle theta increment.');
readln(increment);
writeln('Enter output file name.');
readln(outFileName);
angle := 0.0;
area := a * b * percent;
mass := a * percent * b * c;
rewrite(outFile,outFileName);
writeln(outFile,’A = ',a:8:4);
writeln(outFile,’B = ',b:B:4);
writeln(outFile,’C = ',c:B:4);
writeln(outFile,’Percent full = ',percent:B:4);
writeln(outFile,’Type Theta Moment’);
while (angle (= 90.0) do
begin
    moment := 0.0;
    beta := ((90 - angle)/360)*(2 * pi);
    theta := (angle/360)*(2 * pi);
    tangentBeta := sin(beta)/cos(beta);
    tangentTheta := sin(theta)/cos(theta);
    form := WhatType;
case (form) of
        1:TypeOne;
        2:TypeTwo;
        3:TypeThree;
        4:TypeFour;
        5:TypeFive;
        6:TypeSix;
        7:TypeSeven;
end;
writeln(outFile,form, ’ ,angle:6:2, ’ ,moment:B:4);
angle := angle ÷ increment;
end;
close(outFile);
End. (*END MAIN PROGRAM*)