CONSTRUCTION AND SOLUTION OF AN INVERSE PROBLEM
POSED BY “PIEKARA’S CHAIR”

A Thesis by

Daniel Callahan

M.F.A., Wichita State University, 1998
B.A., Lakeland College, 1994

Submitted to the Department of Mathematics & Statistics
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements of the degree of
Master of Science

May 2008
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
CONSTRUCTION AND SOLUTION OF AN INVERSE PROBLEM
POSED BY “PIEKARA’S CHAIR”

I have examined the final copy of this thesis for form and content, and recommend that it be accepted in partial fulfillment of the requirement for the degree of Master of Science with a major in Mathematics.

_____________________________________
Elizabeth Behrman, Committee Chair

We have read this thesis and recommend its acceptance:

_____________________________________
Kirk Lancaster, Committee Member

_____________________________________
Niall Shanks, Committee Member
DEDICATION

To my entire family, especially Valorie and Deirdre
You’ll do best by filling your minds and meditating on things true, noble, reputable, authentic, compelling, gracious – the best, not the worst; the beautiful, not the ugly; things to praise, not things to curse. Do that, and God, who makes everything work together, will work you into his most excellent harmonies.

Philippians 4:8-9 ("The Message")
ACKNOWLEDGMENTS

This thesis could not have been written without the direct and indirect support of many. Their contributions have been critical even when they seemed small, so in I will resort to an alphabetical listing: Dr. Johnson Agbo, Adam Anthony, Dr. Wally Axmann (without whose help this thesis would not have been possible), Dr. Elizabeth Behrman (my stalwart thesis director), Dr. Stephen Brady, Katherine Earles, Dr. Alan Elcrat, Karl Elder, Tina Graves, Holly Haynes, Palmer Haynes, Dr. Viktor Isakov, Dr. Thalia Jeffres, Dr. Kirk Lancaster, Dr. A. J. Mandt, Patric Mitchell, Sue Neal, Dr. Phil Parker, Dr. “Jack” Qian, Steve Redpath, Steven Rokiski, Dr. Niall Shanks, Kim Tarver, Joan Wagner, Pastor Don Waite, Pastor Bobby Williford, Michele Wolfe, and the faculty and administration of Sterling College.

The article “Piekar's Chair: Mechanical Model for Atomic Energy Levels” by Zofia Golab-Meyer presents a model of the real-world problem of determining classical energy states suitable for high school or undergraduate college students. This thesis combines this idea with simple geometry to demonstrate how “Piekar's Chair” may be posed and then solved as inverse problems in $\mathbb{R}^2$. 
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Definitions</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Axioms</td>
<td>3</td>
</tr>
<tr>
<td>2. CONSTRUCTING THE FORWARD PROBLEM</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Equilateral triangle</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Algorithm for constructing the general N-gon</td>
<td>7</td>
</tr>
<tr>
<td>3. SOLVING THE INVERSE PROBLEM I: USING THE KEY ROW</td>
<td>8</td>
</tr>
<tr>
<td>3.1 Equilateral Triangle</td>
<td>8</td>
</tr>
<tr>
<td>3.2 The Pathological Triangle</td>
<td>12</td>
</tr>
<tr>
<td>3.3 General 3-gon</td>
<td>14</td>
</tr>
<tr>
<td>3.4 Algorithm for the general N-gon using the key row</td>
<td>17</td>
</tr>
<tr>
<td>4. SOLVING THE INVERSE PROBLEM II: WITHOUT KEY ROW</td>
<td>19</td>
</tr>
<tr>
<td>4.1 S1 known</td>
<td>19</td>
</tr>
<tr>
<td>4.2 S1 not known</td>
<td>19</td>
</tr>
<tr>
<td>4.4 Algorithm for the general N-gon</td>
<td>26</td>
</tr>
<tr>
<td>5. OUTLINE FOR THE SQUARE</td>
<td>27</td>
</tr>
<tr>
<td>6. SUMMARY</td>
<td>30</td>
</tr>
<tr>
<td>7. CODE</td>
<td>32</td>
</tr>
<tr>
<td>7.1 bisection.m</td>
<td>32</td>
</tr>
<tr>
<td>7.2 check_square.m</td>
<td>34</td>
</tr>
<tr>
<td>7.3 check_triangle.m</td>
<td>35</td>
</tr>
<tr>
<td>7.4 checker.m</td>
<td>36</td>
</tr>
<tr>
<td>7.5 equilateral.m</td>
<td>37</td>
</tr>
<tr>
<td>7.6 onerow.m</td>
<td>39</td>
</tr>
<tr>
<td>7.7 solver.m</td>
<td>42</td>
</tr>
<tr>
<td>7.8 square.m</td>
<td>43</td>
</tr>
<tr>
<td>7.9 tworow.m</td>
<td>45</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>46</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. An equilateral triangle</td>
<td>3</td>
</tr>
<tr>
<td>2. The path of our perturbation</td>
<td>5</td>
</tr>
<tr>
<td>3. Perturbation of an equilateral triangle</td>
<td>6</td>
</tr>
<tr>
<td>4. Our example of a pathological Triangle</td>
<td>13</td>
</tr>
<tr>
<td>5. Our example of a general 3-gon</td>
<td>15</td>
</tr>
<tr>
<td>6. The Law of Cosines</td>
<td>21</td>
</tr>
<tr>
<td>7. The perturbation of a pathological triangle</td>
<td>26</td>
</tr>
<tr>
<td>8. A square with a CM at (-0.1, 0.2)</td>
<td>27</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

1.1 Background

Arkadiusz Piekara [1] developed a pedagogical model to help his physics students understand “the notions of potential energy in a uniform gravitational field, the center of gravity (center of mass), and the stability of the state” via a common, household object: a chair. However, in the classroom, such a demonstration was easier for Golub-Meyer to achieve with a table.

The idea is this: if the chair is flat on its back, then it has achieved its lowest energy state (its ground state). If it is on its side, the center of mass is higher, and the system has greater potential energy (or, it is on a higher energy level). This process is repeated until we have an energy state higher than all others (the chair is standing on its legs).

However, all of these energy states are stable: we presume that the chair won't be moved, knocked over, or fall to bits via entropy. But the chair also possesses unstable energy states, such as balancing the chair on one of its legs; of course, the chair falls to one of its stable states. The energy released by this fall is determined by the unstable state from which it fell and the stable state to which it landed – the difference in energy between these states is recorded as emission spectra. A set of emission spectra over all possible changes in energy produces an energy level pattern, which we shall call an Energy Level Diagram.

In her paper on Piekara's Chair, Zofia Golab-Meyer reproduces a cartoon where a furniture collector displays an Energy Level Diagram to a fellow collector and says, “Look at this energy level pattern. That is my new piece of furniture. Can you guess what it is?” The cartoon is meant to illustrate the difficulty that all physics students face in studying objects they cannot directly observe. However, it also suggests an inverse problem based on Piekara's Chair: given a suitable domain of objects in real space which we “drop” in order to produce an Energy Level Diagram, under what conditions could the object be reconstructed?

There are many interesting ways in which to formulate such a problem, not the least of which would be the behavior of objects in \( \mathbb{R}^3 \) (or even \( \mathbb{R}^4 \)). However, as Golab-Meyer
notes, the best possible intuitive picture begins by working in \( \mathbb{R}^2 \), so that is where we shall begin. We shall also “drop” n-gons, since they are the simplest objects in Euclidean space to study.

1.2 Definitions

1. Let \( n \) equal the number of sides of an n-gon in \( \mathbb{R}^2 \) where \( n \in \mathbb{N} \) such that \( 2 \leq n < \infty \). When \( n=2 \), we have the trivial case of the line. We will investigate the case where \( n=3 \), the triangle.

2. Calculation of equilibria

   a. \( n \) unstable equilibria (U1, ..., UN) are calculated by their Euclidean distance from the center of mass (CM) to each of the apexes of the n-gon.

   b. \( n \) stable equilibria (S1, ..., SN) are calculated by the shortest Euclidean distance from the CM to each side of the n-gon.

   c. The difference between an unstable equilibrium and its adjacent stable equilibria provide \( 2n \) emission spectra. While the spectrum of an unperturbed CM is graphed in \( \mathbb{R} \), the collection of these spectra obtained from a perturbed CM is graphed on an Energy Level Diagram (ELD) in \( \mathbb{R} \times \mathbb{R} \).

3. Labeling of equilibria: we label an arbitrary line segment from the CM to an apex as U1 and proceed to label counterclockwise. (We sometimes refer to this apex as U1 since this slight abuse of definition is unambiguous.) See Figure 1.
1.3 Axioms

1. All n-gons “fall” onto a “surface” in \( \mathbb{R}^2 \); more formally, a line with a slope of 0. Wlog, we consider this to be the x-axis.

2. All n-gons have “mass” and “volume”; that is, we exclude n-gon's with content zero.

3. The center of mass (CM) is located within the convex hull of an n-gon.

Axioms 1-3 guarantee that we need not consider an “exotic” n-gon – Piekara's Chair is analogous to a real world object, so our own choice of n-gon should fulfill the same role.

Figure 1: An equilateral triangle

*Our example of an equilateral triangle with stable and unstable equilibria calculated from the center of mass (CM). Note that the equilibria create interior triangles with angles \( a_1, \ldots, a_6 \).*
4. Uniqueness of an n-gon is determined by its equilibria (i.e., not by rotation, translation, etc.).

Furthermore, we say that if \( a, b \in \{ \text{the set of all } n-\text{gons} : n \in \mathbb{N}, 2 \leq n < \infty \} \) have identical ELD, then \( a \sim b \); thus there exists an equivalence relation between \( a \) and \( b \).

5. We know the ELD for each n-gon under consideration. Axioms 4-5 guarantees existence and uniqueness of the n-gon’s under scrutiny.
CHAPTER 2
CONSTRUCTING THE FORWARD PROBLEM

2.1 Equilateral triangle

We begin with an equilateral triangle with a CM which satisfy our five axioms implemented in MATLAB (equilateral.m). The CM, located within the interior of the triangle, is perturbed via the scheme in Figure 2: the CM moves from its original position to an arbitrary apex, along an adjacent side until it is at the midpoint of the segment, and back to its original position. (We shall see in section 3.3 that all but one component of this perturbation scheme is arbitrary.)

![Figure 2: The path of our perturbation.](image)

Notice that the perturbation moves the CM from its original position to U1 in four steps inclusive; likewise, the CM moves from U1 along S1 and from that point back to its original position in three distinct steps each. Or, if we wish to have T perturbations from one sub-destination to another, we will have a total of $3T-2$ perturbations. (In this example, $T=4$.)

At each step, we calculate the new distances from the CM to each of the stable and unstable equilibria. The differences between relevant stable and unstable equilibria (e.g., U1–S1 and U1–S3 but not U1–S2) give us an ELD. For the sake of convenience, this can be plotted (see Figure 3) or expressed as a table.

---

1 All MATLAB scripts have been developed in Release 14 with Service Pack 1 and can be found in Chapter 7.
The code generates the forward problem:

\[
\text{EDU}>> \ [es] = \text{equilateral}( \ [0 \ 0], 4 )
\]

Figure 3: Perturbation of an equilateral triangle.

Row \( n \) of the data above corresponds to step \( n, 1 \leq n \leq 3T-2 \). At step 4, we reach the apex \( U1 \) where we achieve a minimum and maximum. At step 7, the CM is perturbed until it reaches the midpoint of the segment as indicated in Figure 2. At step 10, we have returned to the original position of the CM; hence, the value at step 10 is identical to that of step 1.

The code generates the forward problem:
Thus, from the above we note that:

- Rows 1-4 cover the perturbation from the CM to the apex
- Rows 4-7 measure from the apex to the midpoint of the side length
- Rows 7-10 measure from the midpoint back to the CM
- Rows 1 and 10 are identical.

2.2 Algorithm for constructing the general N-gon

1. Identify the equilibria and label them as in Figure 1.

2. Starting at the original position of the CM, measure the equilibria as in Definition 2. A matrix forms with columns similar to the above, *mutatis mutandis*, with $3T-2$ rows and $2n$ columns.

3. Perturb the CM via the chosen scheme and measure again. Create a row for each perturbation. Continue until the perturbation scheme is completed.

We define the row of information which develops when a perturbation scheme causes the CM to coincide with an apex as the *key row*. We shall also see in 3.3 that all but one element of our perturbation scheme has no bearing on the solution of the inverse problem.

A general, non-pathological triangle is implemented in `triangle.m`. Implementations of a sample 4-gon is included in Chapter 7 under `square.m`.

Thus, from the above we note that:

- Rows 1-4 cover the perturbation from the CM to the apex
- Rows 4 -7 measure from the apex to the midpoint of the side length
- Rows 7-10 measure from the midpoint back to the CM
- Rows 1 and 10 are identical.
CHAPTER 3

SOLVING THE INVERSE PROBLEM I: USING THE KEY ROW

3.1 Equilateral Triangle

Upon being presented with the matrix of energy spectra, \( es \), our first task is to determine which column in the matrix of emission spectra belongs to \( U1-S1 \), etc.

For the equilateral triangle, we employ the following algorithm:

1. Notice that:

\[
U1 + U2 + U3 = U1 + U2 + U3
\]

\[
U1 + (-S1) + U2 + (-S2) + U3 + (-S3) = U1 + (-S3) + U2 + (-S1) + U3 + (-S2)
\]

\[
(U1 - S1) + (U2 - S2) + (U3 - S3) = (U1 - S3) + (U2 - S1) + (U3 - S2)
\]

That is, the sum of three of our equilibria must equal the sum of the remaining equilibria, and these equilibria are contained in the columns calculated in step 2 of the forward problem; we call this the \textit{equilibria equality}. Taking the integers 1, ..., 6, we choose three and add the sums of these columns; we do the same to the remaining three. Taking the Euclidean norm of the difference between each combination of sums, we obtain a minimum value. This minimum indicates that the sum of the columns on the left-hand side of the equality must be either \( U1-S1 \), \( U2-S2 \), or \( U3-S3 \) and that the columns on the right-hand side of the equality must be either \( U1-S3 \), \( U2-S1 \), or \( U3-S2 \).

2. Note that the key row of the matrix \( es \) is also the row which contains the maximum value of \( es \). The columns which will be labeled \( U1-S1 \) and \( U1-S3 \) will contain 0 at the key row, due to our choice of perturbation. The CM now coincides with the apex \( U1 \), and the distance from \( U1 \) to either of its adjacent sides is now 0, so its energy emission is 0.
By the above, we have distinctly labeled two columns.

3. Similarly, the columns containing U3–S3 and U2–S1 are equal and the value of the side length of the equilateral triangle (since S1 and S3 are zero). Thus four columns are distinctly labeled.

4. Since only columns U2–S2 and U3–S2 remain, we obtain their labeling by exhaustion.

We further notice that the maximum value of $es$ is a side length of the triangle; since our triangle is equilateral, it is the side length. This algorithm is implemented with the MATLAB script `check_triangle.m`:

```matlab
EDU>> [solution] = check_triangle(es)
sol =
   1   2   3
   4   5   6
```

The output states that the sum of the first three columns is equal to the sum of the last three columns. By the above, we are able to label each column as U1-S1, U2-S2, etc.

Thus we determine what information each column of $es$ represents. Since our method of perturbation has produced only one row with a maximum value (the key row), the order in which the row information is presented can be arbitrary (i.e., if the row ordering is scrambled before we receive it, it is of no consequence).

We now wish to relabel the columns of $es$:

i. $L_1 = U_1-S_1$
ii. $L_2 = U_2-S_2$
iii. $L_3 = U_3-S_3$
iv. $L_4 = U_1-S_3$
v. $L_5 = U_2-S_1$
vi. $L_6 = U_3-S_2$

If we define $\vec{u}=[U1 \ S3 \ U3 \ S2 \ U2 \ S1]^T$ and $\vec{L}=[L1 \ L2 \ L3 \ L4 \ L5 \ L6]^T$, then we may write $\vec{L} = \vec{U} \ \vec{u}$ where:
Now, if $\bar{U}$ were invertible, the solution to the inverse problem would be trivial. However, it is not; the reduced row echelon form of $\bar{U}$ shows us that $\bar{U}$ has rank 5:

$$
\bar{U} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & -1 & 1 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 \\
1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & 0 \\
0 & 0 & 1 & -1 & 0 & 0
\end{bmatrix}
$$

This is equivalent to the statement that if we knew one measurement from the CM to any apex or to any side that we could construct a matrix of rank 6 which would allow us to determine the remaining five measurements. We accomplish this by assuming that we know or can find $S_1$. This allows us to write $U_1, U_2, U_3, S_2, \text{ and } S_3$:

i. $L_1 = U_1 - S_1 \Rightarrow U_1 = L_1 + S_1$

ii. $L_2 = U_2 - S_2 \Rightarrow U_2 = L_1 + L_2 + L_3 - L_4 - L_6 + S_1$

iii. $L_3 = U_3 - S_3 \Rightarrow U_3 = L_1 + L_3 - L_4 + S_1$

iv. $L_4 = U_1 - S_3 \Rightarrow S_3 = L_1 - L_4 + S_1$

v. $L_5 = U_2 - S_1 \Rightarrow L_1 + L_2 + L_3 - L_4 - L_6 + S_1$

vi. $L_6 = U_3 - S_2 \Rightarrow S_2 = L_1 + L_3 - L_4 - L_6 + S_1$

Thus we have that $\bar{u} = A \bar{L} + s \bar{1}$ where $\bar{L}$ is defined above, $s = S_1$, $\bar{1}$ is a 6x1 vector of 1's, and $A$ is defined as:
A = \[
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & -1 & 0 & 0 \\
1 & 0 & 1 & -1 & 0 & 0 \\
1 & 0 & 1 & -1 & 0 & -1 \\
1 & 1 & 1 & -1 & 0 & -1 \\
1 & 1 & 1 & -1 & -1 & -1 \\
\end{pmatrix}
\]

While $A$ is rank 6 and invertible, we note that inverting $A$ will not offer a solution to the problem; however, this is not an obstacle.

From this system of equations we can determine the distance from CM to each of the stable and unstable equilibria provided that we know $S_1$; since the key row of the matrix $es$ occurs when $S_1 = 0$, we have solved the inverse problem for the equilateral triangle.

As an example in MATLAB, we take an equilateral triangle with vertices at $(1,0), (-1/2, \sqrt{3}/2), (-1/2, -\sqrt{3}/2)$ and a CM at $(-1/3, -1/5)$. Clearly, it has a side length of $\sqrt{3}$. Then:

```matlab
EDU>> [es] = equilateral([-1/3,-1/5], 4)
es =
0.5084    0.9123    0.1931    0.8548    0.2391    0.5199
0.3389    0.5603    0.6251    0.5699    0.6115    0.3430
0.1695    0.3530    1.1596    0.2849    1.1286    0.2685
0.0    0.2321    1.7321    0    1.7321    0.2321
0.2887    0.1934    1.3573    0.0387    1.4434    0.3573
0.5774    0.1547    1.0275    0.0774    1.1547    0.5275
0.8660    0.1160    0.7500    0.1160    0.8660    0.7500
0.6902    0.2950    0.5572    0.3057    0.5706    0.6661
0.5790    0.5670    0.3695    0.5600    0.3682    0.5874
0.5084    0.9123    0.1931    0.8548    0.2391    0.5199
EDU>> A*es(4,:)
ans =
0
0
1.7321
1.5000
1.7321
0
```

11
Thus from the apex U1, we see from the key row that the distance to S1 and S3 are 0, as expected. The distance from U1 to U2 and from U1 to U3 is $\sqrt{3}$. The distance from U1 to the nearest point on the opposite side of the triangle is $3/2$. Further work using either the Law of Cosines or perturbation of the CM to apex U2 will determine all three sides.

3.2 The Pathological Triangle

We pause to consider the case we label as “pathological” (see Figure 4). To produce this triangle, we modify equilateral.m by removing the following code from the “Defaults” section:

```matlab
L = linspace(0,2*pi,4);
xv = cos(L)';
yv = sin(L)';
```

replacing it with:

```matlab
xv = [1 -1 0 1];
yv = [0 3 0 0];
```

We may rename the MATLAB function as pathological.m for simplicity's sake, but because it is so similar to equilateral.m, it is not included in Chapter 7. We generate our energy spectrum with and attempt to solve this case with our previous method.
Figure 4: Our example of a pathological Triangle.

With a CM located at (-0.4, 1.7), we violate an assumption that the CM can be measured to each side of the triangle at a right angle. Note that a CM located at (x,y) where the CM is not on the boundary and x \geq 0 does not violate our assumptions; however, we have not assumed that we can guarantee the location of the CM once it is located within the object's convex hull (by Axiom 3).
Our method calculates two side lengths accurately (1.0000, 3.6056), but the distance from the apex $U_1$, while accurate, does not form a right angle to the opposite side of the triangle. However, the information is sufficient to reconstruct the triangle.

3.3 General 3-gon

We consider a non-pathological triangle that is neither equilateral nor pathological. To produce this triangle, we modify `equilateral.m` by removing the following code from the “Defaults” section:

```matlab
L = linspace(0,2*pi,4);
```

replacing it with:

```matlab
L = [0 1.7 4 2*pi ];
```

We may rename the MATLAB function as `triangle.m` for simplicity’s sake, but because it is so similar to `equilateral.m`, it is not included in Chapter 7.
The triangle has side lengths of 1.5026, 1.8186, 1.8255 on its right, left, and bottom sides, respectively. Starting with a CM at the origin, and perturbing it a total of five times in each line segment, we produce an ELD of:

\[
\text{EDU}>> \text{es} = \text{triangle}( \begin{bmatrix} 0 & 0 \end{bmatrix}, 5 )
\]

\[
es = 
\begin{bmatrix}
  0.3400 & 0.5915 & 0.5839 & 0.5839 & 0.3400 & 0.5915 \\
  0.2550 & 0.4136 & 0.8666 & 0.4379 & 0.5666 & 0.5308 \\
  0.1700 & 0.2869 & 1.1717 & 0.2919 & 0.8443 & 0.4923 \\
  0.0850 & 0.1982 & 1.4906 & 0.1460 & 1.1601 & 0.4678 \\
   0 & 0.1363 & 1.8186 & 0 & 1.5026 & 0.4523 \\
  0.1878 & 0.1193 & 1.5789 & 0.0164 & 1.3147 & 0.5548 \\
  0.3756 & 0.1022 & 1.3572 & 0.0328 & 1.1269 & 0.6754 \\
  0.5635 & 0.0852 & 1.1553 & 0.0492 & 0.9391 & 0.8157 \\
  0.7513 & 0.0681 & 0.9742 & 0.0655 & 0.7513 & 0.9768 \\
  0.6042 & 0.1547 & 0.8766 & 0.1508 & 0.6042 & 0.8805 \\
  0.4906 & 0.2747 & 0.7790 & 0.2696 & 0.4906 & 0.7842 \\
  0.4047 & 0.4225 & 0.6814 & 0.4161 & 0.4047 & 0.6878 \\
  0.3400 & 0.5915 & 0.5839 & 0.5839 & 0.3400 & 0.5915 
\end{bmatrix}
\]

Since the fifth row contains the largest value of the matrix \( es \), we label row 5 as the key row. We then check to see if our algorithm for labeling the columns holds:

Figure 5: Our example of a general 3-gon.
We can see by examination of the triangle.m code that this labeling is correct. However, had we not had the forward problem on hand, we would need a procedure to label each column. Row five contains the maximum value of $es$, so it is the key row. Since the perturbation scheme sends the CM to U1 and since columns 1 and 4 are 0 at the key row, they must contain U1. Given that Column 1 must be either U1-S1, U2-S2 or U3-S3, we label it U1-S1 and column 4 as U1-S3.

Since $S1=0$ at the key row, \( U2-S1 > U2-S2 \). Similarly, since $S3=0$ at the key row, \( U3-S3 > U3-S2 \). Finally, $S2$ is the height of the triangle at the key row, so it is reasonable to assume that \( U2-S2, U3-S2 \) are close to 0. This leads us to label column 2 as \( U2-S2 \), column 3 as \( U3-S3 \), column 5 as \( U2-S1 \), and column 6 as \( U3-S2 \). We then employ \( \vec{u} = A \vec{L} + s \vec{1} \).

It follows that the solution to the inverse problem can be determined via MATLAB:

```matlab
EDU>> A*es(5,:)
ans =
    0
    0
  1.8186
  1.3663
  1.5026
    0
```

The distance from U1 to U2 is 1.5026 and from U1 to U3 is 1.8186, as determined in the forward problem. The distance from (1,0) to the line segment from U2 to U3 is:

$$
\frac{\|(\sin(4) - \sin(1.7))(\cos(4) - 1) - (\sin(4) - 0)(\cos(4) - \cos(1.7))\|}{\sqrt{(\cos(4) - \cos(1.7))^2 + (\sin(4) - \sin(1.7))^2}} = 1.36627467843725
$$

which our solution reasonably approximates.
Because our perturbation scheme sends the CM to an apex of an arbitrary 3-gon and because of the information obtained when the CM is at the apex that the choice of perturbation scheme is arbitrary as long as at least one step of the scheme includes this feature. For practical purposes, we simply choose one perturbation scheme among many that shares this feature based on its ease of implementation and use.

3.4 Algorithm for the general \( N \)-gon using the key row

1. Using the fact that:

\[
U_1 + U_2 + \cdots + U_N = U_1 + U_2 + \cdots + U_N
\]

\[
U_1 + (-S_1) + U_2 + (-S_2) + \cdots + U_N + (-S_N) = U_1 + (-S_N) + U_2 + (-S_1) + \cdots + U_N + (-S(N-1))
\]

\[
(U_1 - S_1) + (U_2 - S_2) + \cdots + (U_N - S_N) = (U_1 - S_N) + (U_2 - S_1) + \cdots + (U_N - S(N-1))
\]

we see that the sum of \( n/2 \) of our equilibria must equal the sum of the remaining columns, calculated in step 2 of the forward problem. We call this the equilibria equality. Taking the Euclidean norm of the differences of the columns, we determine a minimum value. (If we were to perform these calculations by hand, we would search for an equality. However, due to round-off error, we must search for a minimum.) This minimum indicates that the sum of the columns on the left-hand side of the equality must be one of \( (U_1 - S_1), (U_2 - S_2), \ldots, (U_N - S_N) \) and that the columns on the right-hand side of the equality must be one of \( (U_1 - S_N), (U_2 - S_1), \ldots, (U_N - S(N-1)) \).

2. The key row of the matrix \( es \) contains the maximum value of \( es \). To see this, suppose that there exists a row other than the key row that contains the maximum value of \( es \). It follows that the maximum of this row exceeds the maximum of the key row. Recall that the maximum in the key row is obtained by the CM coinciding with an apex of the triangle; at this row, the CM coincides with the apex \( U_1 \), and the distance from \( U_1 \) to either of its adjacent sides is 0. Since the maximum value of \( es \) exceeds this, the CM must be outside of the convex hull of the triangle. This violates axiom 3, a contradiction. Thus, by the above, we have distinctly labeled two columns.

3. By a process of elimination, we continue labeling pairs of columns, one on each side of the equality, based on the number of sides of the \( n \)-gon. Since this is known (it is always
one half of the count of the columns of \( es \), it is a matter of computation to label \( n-2 \) sides. (Like the “middlegame” of chess or certain aspects of non-zero sum game theory, this step requires art as well as mathematical skill as \( n \) grows large.)

4. Since only two columns remain, we obtain their labeling by exhaustion.

Since our method of perturbation has produced a matrix \( es \) with only one row with a maximum value (the key row), the order in which the row information is presented can be arbitrary. We construct the vectors \( \bar{u}, \bar{L} \) (where \( \bar{L} \) is the transpose of the key row) and the matrices \( \bar{A}, \bar{1} \) as above, *mutatis mutandis*; that is, \( \bar{A} \) is an \( n \times n \) matrix whose entries are either -1, 0, or 1, and \( \bar{1} \) is an \( n \times 1 \) vector of 1's. Thus we obtain \( \bar{u} = \bar{A} \bar{L} + s \bar{1} \). Since the key row of the matrix \( es \) occurs when \( S1=0 \), we have solved the inverse problem for the general \( n \)-gon.
4.1 S1 known

Since $A$ is constant, if S1 is known for arbitrary row $k$ in $es$, we calculate
\[ \vec{u} = A \vec{L} + s \vec{I} \]
via the general algorithm in 3.3 where $\vec{L}$ is the transpose of the row $k$. Then the 1st, 5th and 3rd components of $\vec{u}$ are the distances to the apexes labeled U1, U2, and U3 respectively; the 6th, 4th, and 2nd components of $\vec{u}$ are the distances to the sides perpendicular to the line segments labeled S1, S2, and S3, respectively.

4.2 S1 not known

We begin with an equilateral triangle such that the CM is not on the boundary. By Figure 1, we note that each angle $a_i$ can be written as:

\[
\cos(a_1) = \frac{S1}{U1}, \quad \cos(a_2) = \frac{S1}{U2}, \quad \cos(a_3) = \frac{S2}{U2}, \quad \cos(a_4) = \frac{S2}{U3}, \quad \cos(a_5) = \frac{S3}{U3}, \quad \cos(a_6) = \frac{S3}{U1}
\]

By our previous work, we have identified the columns of $es$. We now test the permutation of all columns by recreating the side lengths and angles of the triangles using $a_1, ..., a_6$, the Law of Cosines, and the Law of Sines. (This algorithm is implemented in onerow.m).

1. We employ check_triangle.m as in Section 3.1 This identifies three columns that represent U1–S1, U2–S2, or U3–S3, leaving the remaining to be U1–S3, U2–S1, or U3–S2. However, we do not assume that we have received our data in the correct columnar order. To test our data, we permute the left-hand side of the equality and test the results against each permutation of the right-hand side. That is, if we have identified columns 1, 2, 3 as belonging to U1–S1, U2–S2, or U3–S3, we must test the permutations

\[
\begin{array}{ccc}
3 & 2 & 1 \\
3 & 1 & 2 \\
2 & 3 & 1 \\
\end{array}
\]
against all permutations of columns 4, 5, 6:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

2. Section 3.1 demonstrated how our measurements from the CM to all sides and apexes can be determined with only the measurement of the distance from the CM to one side; specifically, to the side labeled S1. Using solver.m, we determine an approximation of S1 for each of our 36 possible columnar combinations using the bisection method for locating the root of a nonlinear function. Following our work in section 3.1, we convert our measurements:

- \( U_1 = L_1 + S_1 \)
- \( U_2 = L_1 + L_2 + L_3 - L_4 - L_6 + S_1 \)
- \( U_3 = L_1 + L_3 - L_4 + S_1 \)
- \( S_3 = L_1 - L_4 + S_1 \)
- \( S_2 = L_1 + L_3 - L_4 - L_6 + S_1 \)

where \( L_n \) refers to the value in the \( n^{th} \) column of \( es \). We then test values ranging from \( 0 + \epsilon \) to \( 3 + \epsilon \) in place of \( S_1 \) by calculating \( \sum_{i=1}^{6} a_i - 2\pi \) using the bisection method where

\[
\arccos\left(\frac{S_1}{L_1 + S_1}\right) = a_1
\]

\[
\arccos\left(\frac{S_1}{L_1 + L_2 + L_3 - L_4 - L_6 + S_1}\right) = a_2
\]

\[
\arccos\left(\frac{L_1 + L_3 - L_4 - L_6 + S_1}{L_1 + L_2 + L_3 - L_4 - L_6 + S_1}\right) = a_3
\]
\[
\begin{align*}
\arccos\left(\frac{L1 + L3 - L4 - L6 + S1}{L1 + L3 - L4 + S1}\right) &= a_4 \\
\arccos\left(\frac{L1 - L4 + S1}{L1 + L3 - L4 + S1}\right) &= a_5 \\
\arccos\left(\frac{L1 - L4 + S1}{L1 + S1}\right) &= a_6 \\
\end{align*}
\]

We obtain one approximation for S1 for each of our 36 columnar combinations.

3. Since this is clearly an expensive algorithm, we use \texttt{checker.m} to check if \(a_i\), \(i = 1, \ldots, 6\) is less than \(\pi/2\) radians. If not, then we move to the next combination.

4. We verify our results with the Law of Cosines by generating an approximate solution based on our calculation of S1 and each \(a_i\). Now the Law of Cosines states that, given a triangle of the form in Figure 7, that

\[
a = \sqrt{b^2 + c^2 - 2bc \cos \alpha}
\]

and similarly for the other two sides.

![Figure 6: The Law of Cosines](image)

*The Law of Cosines relates side a to sides b and c as well as angle \(\alpha\). This follows for all sides as well as angles. (Graphic by Pmx, released into the public domain.)*
We estimate the length of all three sides and then employ the Law of Cosines a second
time using the form $A = \arccos \frac{b^2 + c^2 - a^2}{2bc}$, estimating all three interior angles. If an
angle is determined to be complex-valued, then we move to the next combination.
Finally, since the Law of Cosines also states that $a^2 = b^2 + c^2 - 2bc \cos \alpha$, it follows
that:

$$a^2 + b^2 + c^2 = (b^2 + c^2 - 2bc \cos \alpha) + (a^2 + c^2 - 2ac \cos \beta) + (a^2 + c^2 - 2ac \cos \gamma)$$
or:

$$a^2 + b^2 + c^2 - (b^2 + c^2 - 2bc \cos \alpha) - (a^2 + c^2 - 2ac \cos \beta) - (a^2 + c^2 - 2ac \cos \gamma) = 0$$

However, since we are estimating sides and angles, and because of the round-off error
inherent in numerical computation, it is highly unlikely that the RHS will be 0 in the
actual calculation. Instead, we compute the LHS and take its 2-norm. If the norm for the
triangle is less than a given standard norm, we proceed.

5. We verify our results with the Law of Sines, which states that $\frac{a}{\sin A} = \frac{b}{\sin B} = \frac{c}{\sin C}$.

We note that this law implies that:

$$\frac{a}{\sin A} - \frac{b}{\sin B} = 0 \quad , \quad \frac{a}{\sin A} - \frac{c}{\sin C} = 0 \quad \text{and} \quad \frac{c}{\sin C} - \frac{b}{\sin B} = 0$$

By the above, we note that the RHS of each equation is unlikely to be 0. Instead, we add the grand total of each
RHS. If this grand total is negative, we move to the next combination. If this grand total
is less than 1, we consider this combination to be a possible solution to the inverse
problem.

By following the above in the case of an equilateral triangle, we obtain the following (truncated)
output:

```
EDU>> [es] = equilateral([0 0], 5);
EDU>> T = onerow (es(1,:))
```

22
Recall that the first, third, and fifth column are the triangle's side lengths and that the second, fourth, and sixth columns are the triangle's interior angles. The seventh column contains the norm of the sum of the sides obtained via the Law of Cosines, and the eighth columns contains the sum derived from the Law of Sines. We notice that we have multiple instances of what appears to be the same solution, although this is due to the fact that our data is truncated to four decimal places. Clearly, however, we have obtained the correct solution.

By following the above in the case of an nonequilateral triangle, we obtain the following:

EDU>> [ES] = triangle([0 0], 5);
EDU>> T = onerow (ES(1,:))

T =

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.6253</td>
<td>1.2627</td>
<td>1.9186</td>
<td>0.9394</td>
<td>1.6253</td>
<td>0.9394</td>
<td>0.2737</td>
<td>-0.4213</td>
</tr>
<tr>
<td>1.6311</td>
<td>1.2597</td>
<td>1.9164</td>
<td>0.9373</td>
<td>1.6225</td>
<td>0.9446</td>
<td>0.2648</td>
<td>-0.4192</td>
</tr>
<tr>
<td>1.6225</td>
<td>1.2597</td>
<td>1.9164</td>
<td>0.9466</td>
<td>1.6311</td>
<td>0.9373</td>
<td>0.2686</td>
<td>-0.4108</td>
</tr>
<tr>
<td>1.6283</td>
<td>1.2566</td>
<td>1.9141</td>
<td>0.9425</td>
<td>1.6283</td>
<td>0.9425</td>
<td>0.2595</td>
<td>-0.4088</td>
</tr>
<tr>
<td>1.8186</td>
<td>1.1500</td>
<td>1.8255</td>
<td>0.8500</td>
<td>1.5026</td>
<td>1.1416</td>
<td>0.2866</td>
<td>-0.3606</td>
</tr>
<tr>
<td>1.8234</td>
<td>1.1469</td>
<td>1.8234</td>
<td>0.8478</td>
<td>1.5000</td>
<td>1.1469</td>
<td>0.2952</td>
<td>-0.3588</td>
</tr>
<tr>
<td>1.8208</td>
<td>1.1447</td>
<td>1.8208</td>
<td>0.8522</td>
<td>1.5051</td>
<td>1.1447</td>
<td>0.2817</td>
<td>-0.3511</td>
</tr>
<tr>
<td>1.8255</td>
<td>1.1416</td>
<td>1.8186</td>
<td>0.8500</td>
<td>1.5026</td>
<td>1.1500</td>
<td>0.2903</td>
<td>-0.3494</td>
</tr>
<tr>
<td>1.7300</td>
<td>1.0538</td>
<td>1.7300</td>
<td>1.0340</td>
<td>1.7103</td>
<td>1.0538</td>
<td>0.0012</td>
<td>-0.0245</td>
</tr>
<tr>
<td>1.7268</td>
<td>1.0516</td>
<td>1.7276</td>
<td>1.0392</td>
<td>1.7152</td>
<td>1.0508</td>
<td>0.0004</td>
<td>-0.0153</td>
</tr>
<tr>
<td>1.7276</td>
<td>1.0508</td>
<td>1.7268</td>
<td>1.0392</td>
<td>1.7152</td>
<td>1.0516</td>
<td>0.0004</td>
<td>-0.0144</td>
</tr>
<tr>
<td>1.5026</td>
<td>1.1500</td>
<td>1.8255</td>
<td>1.1416</td>
<td>1.8186</td>
<td>0.8500</td>
<td>0.2903</td>
<td>-0.0112</td>
</tr>
<tr>
<td>1.9164</td>
<td>0.9446</td>
<td>1.6311</td>
<td>0.9373</td>
<td>1.6253</td>
<td>1.2597</td>
<td>0.2686</td>
<td>-0.0084</td>
</tr>
<tr>
<td>1.7243</td>
<td>1.0486</td>
<td>1.7243</td>
<td>1.0444</td>
<td>1.7202</td>
<td>1.0486</td>
<td>0.0001</td>
<td>-0.0051</td>
</tr>
<tr>
<td>1.7152</td>
<td>1.0516</td>
<td>1.7276</td>
<td>1.0508</td>
<td>1.7268</td>
<td>1.0392</td>
<td>0.0004</td>
<td>-0.0009</td>
</tr>
<tr>
<td>1.7202</td>
<td>1.0486</td>
<td>1.7243</td>
<td>1.0486</td>
<td>1.7243</td>
<td>1.0444</td>
<td>0.0001</td>
<td>-0.0000</td>
</tr>
<tr>
<td>1.7103</td>
<td>1.0538</td>
<td>1.7300</td>
<td>1.0538</td>
<td>1.7300</td>
<td>1.0340</td>
<td>0.0012</td>
<td>-0.0000</td>
</tr>
<tr>
<td>1.9186</td>
<td>0.9394</td>
<td>1.6253</td>
<td>0.9394</td>
<td>1.6253</td>
<td>1.2627</td>
<td>0.2737</td>
<td>-0.0000</td>
</tr>
<tr>
<td>1.5051</td>
<td>1.1447</td>
<td>1.8208</td>
<td>1.1447</td>
<td>1.8208</td>
<td>0.8522</td>
<td>0.2817</td>
<td>0.0000</td>
</tr>
<tr>
<td>1.5000</td>
<td>1.1469</td>
<td>1.8234</td>
<td>1.1469</td>
<td>1.8234</td>
<td>0.8478</td>
<td>0.2952</td>
<td>0.0000</td>
</tr>
<tr>
<td>1.9141</td>
<td>0.9425</td>
<td>1.6283</td>
<td>0.9425</td>
<td>1.6283</td>
<td>1.2566</td>
<td>0.2595</td>
<td>0.0000</td>
</tr>
<tr>
<td>1.7152</td>
<td>1.0508</td>
<td>1.7268</td>
<td>1.0516</td>
<td>1.7276</td>
<td>1.0392</td>
<td>0.0004</td>
<td>0.0009</td>
</tr>
<tr>
<td>1.7124</td>
<td>1.0444</td>
<td>1.7202</td>
<td>1.0486</td>
<td>1.7243</td>
<td>1.0486</td>
<td>0.0001</td>
<td>0.0051</td>
</tr>
<tr>
<td>1.9164</td>
<td>0.9373</td>
<td>1.6225</td>
<td>0.9446</td>
<td>1.6311</td>
<td>1.2597</td>
<td>0.2648</td>
<td>0.0084</td>
</tr>
<tr>
<td>1.5026</td>
<td>1.1416</td>
<td>1.8186</td>
<td>1.1500</td>
<td>1.8255</td>
<td>0.8500</td>
<td>0.2866</td>
<td>0.0112</td>
</tr>
</tbody>
</table>
We notice a pattern in subsequent rows: the top two rows present two permutations of the same data. Rows three through five present three permutations of the same data. This is not inconsistent with the method in which we obtained the data, but it does present a larger problem than that of the equilateral triangle.

We can fix both problems by examining a second row of data obtained from the forward problem via the following procedure (implemented in `tworow.m`):

1. We take the norm of the difference between the eighth column of row $i$ and the eighth column of row $i+1$, where $i$ is the row count of $es$.

2. If the norm of the difference is less than a given $\epsilon$, then we save the first row. We then display all rows that have met this criteria.

In practice, we obtain:

```plaintext
EDU>> T = tworow(ES(1,:), ES(2,:))
```

```plaintext
T =

1.8255  1.1416  1.8186  0.8500  1.5026  1.1500  0.2903  -0.3494
1.5026  1.1500  1.8255  1.1416  1.8186  0.8500  0.2903  -0.0112
1.8186  0.8500  1.5026  1.1500  1.8255  1.1416  0.2903  0.3606
```

By inspection, we see that our data reconstructs one triangle: the triangle generated by `triangle.m` In essence, we obtain rows of results that are either inaccurate due to round-off error or are triangles that, by Axiom 4, are within the equivalence class of solutions. However, either norm previously discussed appears to be unique for each triangle, allowing us to identify the correct solution with two rows of data.
4.3 The Pathological Triangle

Our forward problem assumes that each stable equilibria is the line segment connecting the CM orthogonally to a side of the triangle. This is not always the case; for the pathological triangle in figure 5, we may place the CM at \((-0.5, 2)\) and "lose" S3. We observe this in the energy spectrum:

\[
\text{EDU}>> \text{[es] = pathological([ -.5, 2], 10 )}
\]

\[
es = \\
\begin{array}{cccccc}
2.3613 & 0.9599 & 0.0616 & 0.5000 & 0.9794 & 1.9034 \\
2.0990 & 1.1463 & 0.0310 & 0.4444 & 1.2690 & 1.5628 \\
1.8366 & 1.3338 & 0.0089 & 0.3889 & 1.5597 & 1.2307 \\
1.5742 & 1.5220 & 0 & 0.3333 & 1.8512 & 0.9117 \\
1.3118 & 1.7107 & 0.0124 & 0.2778 & 2.1431 & 0.6141 \\
1.0495 & 1.8996 & 0.0604 & 0.2222 & 2.4353 & 0.3520 \\
0.7871 & 2.0887 & 0.1667 & 0.1667 & 2.7277 & 0.1482 \\
0.5247 & 2.2780 & 0.3568 & 0.1111 & 3.0202 & 0.0282 \\
0.2624 & 2.4674 & 0.6402 & 0.0556 & 3.3128 & 0.0016 \\
0 & 2.6569 & 1.0000 & 0 & 3.6056 & 0.0513 \\
0.2003 & 2.5093 & 0.7377 & 0.0336 & 3.4052 & 0.0084 \\
0.4006 & 2.3617 & 0.5129 & 0.0673 & 3.2049 & 0.0029 \\
0.6009 & 2.2141 & 0.3333 & 0.1009 & 3.0046 & 0.0428 \\
0.8012 & 2.0665 & 0.2011 & 0.1346 & 2.8043 & 0.1299 \\
1.0015 & 1.9188 & 0.1111 & 0.1682 & 2.6040 & 0.2593 \\
1.2019 & 1.7712 & 0.0541 & 0.2019 & 2.4037 & 0.4216 \\
1.4022 & 1.6236 & 0.0210 & 0.2355 & 2.2034 & 0.6079 \\
1.6025 & 1.4760 & 0.0046 & 0.2691 & 2.0031 & 0.8109 \\
1.8028 & 1.3284 & 0 & 0.3028 & 1.8028 & 1.0257 \\
1.8645 & 1.2866 & 0.0010 & 0.3243 & 1.7104 & 1.1173 \\
1.9263 & 1.2449 & 0.0038 & 0.3460 & 1.6182 & 1.2109 \\
1.9882 & 1.2034 & 0.0083 & 0.3678 & 1.5261 & 1.3060 \\
2.0502 & 1.1621 & 0.0143 & 0.3896 & 1.4342 & 1.4027 \\
2.1123 & 1.1210 & 0.0216 & 0.4116 & 1.3426 & 1.5007 \\
2.1745 & 1.0802 & 0.0301 & 0.4336 & 1.2513 & 1.5999 \\
2.2367 & 1.0397 & 0.0396 & 0.4557 & 1.1602 & 1.7001 \\
2.2990 & 0.9996 & 0.0501 & 0.4778 & 1.0696 & 1.8013 \\
2.3613 & 0.9599 & 0.0616 & 0.5000 & 0.9794 & 1.9034 \\
\end{array}
\]

Instead a single row containing zeros, we now have three (as can also be seen in Figure 8). However, our forward problem generates S3 regardless, a neat example of “garbage in, garbage out”.
Since we cannot choose to solve the inverse problem for a CM at an arbitrary point within the interior, we instead choose a perturbation scheme that sends the CM to an apex, reducing the problem to the algorithm described in section 3.2.

### 4.4 Algorithm for the general N-gon

With the labeling of each column of \( es \) and the correct computation of \( L_i \), the algorithm follows from the above *mutatis mutandis*, provided that our triangle is not pathological.

---

*Figure 7: The perturbation of a pathological triangle.*

*With several local minima, the solution to our inverse problem now becomes more difficult to solve with the methods previously outlined.*
CHAPTER 5

OUTLINE FOR THE SQUARE

Following the general algorithm, we construct a square with a side length of 1 as below:

This is implemented in `square.m` and is demonstrated below, taking note of the key row.

Figure 8: A square with a CM at (-0.1, 0.2).

*The + symbols follow its perturbation scheme from its original location at apex U1. We then continue to label the apexes counterclockwise (U2 at the top, etc.).*

This is implemented in `square.m` and is demonstrated below, taking note of the key row.
We assume an accurate transmission of data; hence, there is no need to verify whether rows or columns have been transposed. We relabel the columns of \( es \):

i. \( \text{L1} = \text{U1} - \text{S1} \Rightarrow \text{U1} = \text{L1} + \text{S1} \)

ii. \( \text{L2} = \text{U2} - \text{S2} \Rightarrow \text{U2} = \text{L1} + \text{L2} + \text{L3} + \text{L4} - \text{L5} - \text{L7} - \text{L8} + \text{S1} \)

iii. \( \text{L3} = \text{U3} - \text{S3} \Rightarrow \text{U3} = \text{L1} + \text{L3} + \text{L4} - \text{L5} - \text{L8} + \text{S1} \)

iv. \( \text{L4} = \text{U4} - \text{S4} \Rightarrow \text{U4} = \text{L1} + \text{L4} - \text{L5} + \text{S1} \)

v. \( \text{L5} = \text{U1} - \text{S4} \Rightarrow \text{S4} = -\text{L2} - \text{L3} - \text{L4} + \text{L6} + \text{L7} + \text{L8} + \text{S1} \)

vi. \( \text{L6} = \text{U2} - \text{S1} \Rightarrow \text{L6} = -\text{L1} - \text{L2} - \text{L3} - \text{L4} + \text{L5} + \text{L6} + \text{L7} + \text{L8} + \text{S1} \)

vii. \( \text{L7} = \text{U3} - \text{S2} \Rightarrow \text{S2} = -\text{L2} + \text{L6} + \text{S1} \)

viii. \( \text{L8} = \text{U4} - \text{S3} \Rightarrow \text{S3} = -\text{L2} - \text{L3} + \text{L6} + \text{L7} + \text{S1} \)

That is, each column can be determined as a function of \( S1 \). Equivalently, we have that

\[
\bar{u} = A \bar{L} + s \bar{I}
\]

where, \( \bar{u} = \begin{bmatrix} \text{U1} & \text{U2} & \text{U3} & \text{U4} & \text{S1} & \text{S2} & \text{S3} & \text{S4} \end{bmatrix}^T \)

\[
A = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & -1 & 0 & -1 & -1 \\
1 & 0 & 1 & 1 & -1 & 0 & 0 & -1 \\
1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\
-1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\
0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & -1 & -1 & 0 & 0 & 1 & 1 & 0 \\
0 & -1 & -1 & -1 & 0 & 1 & 1 & 1
\end{bmatrix}
\]

\[
\bar{L} = \begin{bmatrix} \text{L1} & \text{L2} & \text{L3} & \text{L4} & \text{L5} & \text{L6} & \text{L7} & \text{L8} \end{bmatrix}^T \quad s = \text{S1} \quad \bar{I} \text{ is a 8x1 vector of 1's.}
\]

From this system of equations we can determine the distance from CM to each of the stable and unstable equilibria provided that we know \( S1 \); the key row of the matrix \( es \) occurs when \( S1 = 0 \).
Our square clearly has a side length of $\sqrt{2}$. Then using our algorithm above, we find:

```
EDU>> A*es(7,:)'
ans =
    0
    1.4142
    2.0000
    1.4142
    0
    1.4142
    1.4142
    0
```

That is, the distance from the CM to U2 and to U4 is $\sqrt{2}$. The distance from the CM to U3 is 2, which uniquely identifies the square, thus solving the inverse problem.
CHAPTER 6

SUMMARY

In practice, a single row of data from an energy spectra produced from any triangle can be analyzed via our final algorithm, implemented in section 4.2. A non-pathological triangle will be completely determined, while a pathological triangle will not.

If we may arbitrarily perturb the CM of a triangle, the triangle will be completely determined, as implemented in part 3.

Future research in this area should begin in \( \mathbb{R}^3 \) and \( \mathbb{R}^4 \). We note that in \( \mathbb{R}^2 \) that an object may fall on an apex or a side; in \( \mathbb{R}^3 \), an object may fall on an apex, a side or a face. This raises the question as to whether we may define objects in \( \mathbb{R}^4 \) such that they may land on an apex, side, face or its solid. And if so, would this model have applications outside of pure mathematics?

While still in \( \mathbb{R}^2 \), this model could also be expanded to ask the question: what potential energy state was the object in before its fall? We might also introduce fragility into the model, making the object breakable when dropped from certain heights.

We may also consider regular or irregular n-gons where n is sufficiently large such that the objects under consideration behave as if \( n \to \infty \). One example would be a regular n-gon where \( n = 10^6 \) such that contact at certain angles between the object and the “floor” onto which it falls causes the object to roll to one side or the other, depending on the location of the CM. Not only would this allow us to calculate acceleration and velocity, but if the floor was assumed to have a positive friction coefficient, then we could calculate the distance of the object as well. Both of the above ideas could also be posed in \( \mathbb{R}^3 \).

A more advanced problem would lie in the nonlinear analysis of pathological n-gons in \( \mathbb{R}^N, N \geq 2 \).

If we abandon the real world application of Piekara's Chair, this problem may have intriguing points when:
1. Axiom 3 is relaxed (allowing the CM to be located outside of the convex hull of the object).

2. Definition 1 is revised such that the objects exist an arbitrary field of dimension 2 or higher.
function y = bisection ( f, a, b, TOL )
%BISECTION   bisection method for locating the root of a nonlinear function
%     calling sequences:
%             y = bisection ( 'f', a, b, TOL )
%             bisection ( 'f', a, b, TOL )
%     inputs:
%             f       string containing name of m-file defining function
%                     whose root is to be located
%             a,b     left and right endpoints, respectively, of interval
%                     known to contain root of f
%             TOL     absolute error convergence tolerance
%                     (iterations will be performed until the size of
%                     enclosing interval is smaller than 2*TOL)
%     output:
%             y       approximate value of root
%     NOTE:
%             if BISECTION is called with no output arguments, the iteration
%             number, the current enclosing interval and the current
%             approximation to the root are displayed
%     (c) Brian Bradie

sfa = sign(feval(f,a));
Nmax = floor ( log((b-a)/TOL) / log(2.0) ) + 1;

for i = 1 : Nmax
    p = ( a + b ) / 2.0;
    sfp = sign(feval(f,p));
    if ( nargout == 0 )
        disp ( sprintf ( '	%3d 	 (%.10f,%.10f) 	 %.10f 
', i, a, b, p ) )
    end
end
if ( (b-a)<2*TOL | sfp == 0 )
    if ( nargout == 1 )
        y = p;
    end
    return
elseif ( sfa * sfp < 0 )
    b = p;
else
    a = p;
    sfa = sfp;
end
end
function [soln] = check_square( es )

% CHECK_SQUARE

% calling sequences:
% [soln] = check_square( es )
% check_square( es )

% inputs:
% es  Energy Spectra

% outputs:
% [soln]  Solution

% (c) 2008 Daniel Callahan, released under the MIT license

% Determine the number of rows and columns in [es]
size_es1 = size(es,1);
size_es2 = size(es,2);

% If we have K columns, we choose K/2 of them to test
test = nchoosek(1:size_es2,size_es2/2);

size_test = size(test, 1);
half = size_test/2;
solution = zeros(2, size_es2/2);
k = size_test;
stdnorm = 1;

% We test our columns, chosen above, against the columns we did not choose.
for i = 1:half
    lhs = es(:, test(i,1))+es(:, test(i,2))+es(:, test(i,3))+es(:, test(i,4));
    rhs = es(:, test(k,1))+es(:, test(k,2))+es(:, test(k,3))+es(:, test(k,4));
    total = lhs - rhs;
    result = norm(total,2);
    if result < stdnorm
        soln(1,:) = test(i,:);
        soln(2,:) = test(k,:);
        stdnorm = result;
    end
    k = k-1;
end

% key_row = size_es1 - (size_es1+2)/3 +1
% sidelength = max(es(key_row,:))
% sidelength = max(max(es))
function [soln] = check_triangle( es )
%CHECK_TRIANGLE
% calling sequences:
% [solution] = check_triangle( es )
%
% inputs:
% es Energy Spectra
%
% outputs:
% [soln] Our solution
%
% (c) 2008 Daniel Callahan, released under the MIT license

% Determine the number of rows and columns in [es]
size_es1 = size(es,1);
size_es2 = size(es,2);

% If we have K columns, we choose K/2 of them to test
test = nchoosek(1:size_es2,size_es2/2);

size_test = size(test, 1);
half = size_test/2;
solution = zeros(2, size_es2/2);
k = size_test;
stdnorm = 1;

% We test our three columns, chosen above, against the columns we did not
% choose.
for i = 1:half
    lhs = es(:, test(i,1))+es(:, test(i,2))+es(:, test(i,3));
    rhs = es(:, test(k,1))+es(:, test(k,2))+es(:, test(k,3));
    total = lhs - rhs;
    result = norm(total,2);
    if result < stdnorm
        solution(1,:) = test(i,:);
        solution(2,:) = test(k,:);
        stdnorm = result;
    end
    k = k-1;
end
function result = checker(a,b,c,d,e,s)
%CHECKER
%
% calling sequences:
% result = checker(a,b,c,d,e,s)
% checker(a,b,c,d,e,s)
%
% inputs:
%
% a       es(1)
% b       es(1)+es(2)+es(3)-es(4)-es(6)
% c       es(1)+es(3)-es(4)-es(6)
% d       es(1)+es(3)-es(4)
% e       es(1)-es(4)
% s       solver( es )
%
% outputs:
%
% result  1 if each angle measures < pi/2
% 0 otherwise
%
% (c) 2008 Daniel Callahan, released under the MIT license

result = 1;

if acos(s/(a+s)) >= pi/2
    result = 0;
end

if acos(s/(b+s)) >= pi/2
    result = 0;
end

if acos((c+s)/(b+s)) >= pi/2
    result = 0;
end

if acos((c+s)/(d+s)) >= pi/2
    result = 0;
end

if acos((e+s)/(d+s)) >= pi/2
    result = 0;
end

if acos((e+s)/(a+s)) >= pi/2
    result = 0;
end
7.5 equilateral.m

function [es] = equilateral( cm, T )
%EQUILATERAL generates an equilateral triangle with a center of mass
%
% calling sequences:
%    [es] = equilateral( cm, T )
%    equilateral( cm, T )
%
% inputs:
%    cm     coordinate of the center of mass
%    T      the number of "steps" that the CM will take in each
%            segment of its permutation (three segments total, with
%            a grand total of 3T-2 steps)
%
% outputs:
%    [es]   the energy spectra
%
% (c) 2008 Daniel Callahan, released under the MIT license

%%% Defaults
% The apices of the triangle
L = linspace(0,2*pi,4);
xv = cos(L)';
yv = sin(L)';

% The midpoint on the upper segment of the triangle
xh = (xv(1)+xv(2))/2;
yh = (yv(1)+yv(2))/2;

%%% Perturbation of cm
% See also perturbationdemo.m
p = zeros(2, 3*T-2);
t = linspace(0,1,T);
for i = 1:T
    p(1, i) = t(i)*xv(1)+(1-t(i))*cm(1);
    p(2, i) = t(i)*yv(1)+(1-t(i))*cm(2);
end
for i = 2:T
    p(1, i+T-1) = t(i)*xh+(1-t(i))*xv(1);
    p(2, i+T-1) = t(i)*yh+(1-t(i))*yv(1);
end
for i = 2:T
    p(1, i+2*T-2) = t(i)*cm(1)+(1-t(i))*xh;
    p(2, i+2*T-2) = t(i)*cm(2)+(1-t(i))*yh;
end
Calculations

```matlab
es = zeros(3*T-2, 7);
for i = 1:3*T-2
    % Unstable equilibria obtained by measuring the Euclidean distance from
    % the perturbed CM to the apex
    u1 = sqrt((xv(1)-p(1,i))^2+(yv(1)-p(2,i))^2);
    u2 = sqrt((xv(2)-p(1,i))^2+(yv(2)-p(2,i))^2);
    u3 = sqrt((xv(3)-p(1,i))^2+(yv(3)-p(2,i))^2);

    % Stable equilibria obtained by the formula for the shortest distance
    % from a point to a line
    s1a = abs((yv(2)-yv(1))*(xv(1)-p(1,i))-(yv(1)-p(2,i))*(xv(2)-xv(1)));
    s1b = sqrt((yv(2)-yv(1))^2+(xv(2)-xv(1))^2);
    s1 = s1a/s1b;
    s2a = abs((yv(3)-yv(2))*(xv(2)-p(1,i))-(yv(2)-p(2,i))*(xv(3)-xv(2)));
    s2b = sqrt((yv(3)-yv(2))^2+(xv(3)-xv(2))^2);
    s2 = s2a/s2b;
    s3a = abs((yv(1)-yv(3))*(xv(3)-p(1,i))-(yv(3)-p(2,i))*(xv(1)-xv(3)));
    s3b = sqrt((yv(1)-yv(3))^2+(xv(1)-xv(3))^2);
    s3 = s3a/s3b;

    % Energy spectra
    es(i, 1) = i;
    es(i, 2) = u1 - s1;
    es(i, 3) = u2 - s2;
    es(i, 4) = u3 - s3;
    es(i, 5) = u1 - s3;
    es(i, 6) = u2 - s1;
    es(i, 7) = u3 - s2;
end
```

Plotting

```matlab
hold on
plot(es(:,1), es(:,2), 'bo-', es(:,1), es(:,3), 'ro-', es(:,4), 'go-')
plot(es(:,1), es(:,5), 'b+-', es(:,1), es(:,6), 'r+-', es(:,1), es(:,7), 'g+-')
hold off
axis([1 3*T-2 0 max(max(es(:,2:7)))])
es = es(:,2:7);
```
function T = onerow ( es_row )

% calling sequences:
%   T = onerow ( es_row )
%   onerow ( es_row )
%
% inputs:
%   es_row  row of data collected from the forward problem
%%
% outputs:
%   T       results
%
% (c) 2008 Daniel Callahan, released under the MIT license

%% Defaults
V = zeros(6,1);
T = zeros(12,7);
side = zeros(1,3);
angle = zeros(1,3);

stdnorm = 10;
A = [ 1 0 0 0 0 0; ... 
    1,0,0,-1,0,0; ... 
    1,0,1,-1,0,0; ... 
    1,0,1,-1,0,-1; ... 
    1,1,1,-1,0,-1; ... 
    1,1,1,-1,-1,-1];
t = 1;
solution = check_triangle( es_row );
lhs = perms(solution(1,:));
rhs = perms(solution(2,:));
size_p = size(lhs,1);

%%
for i = 1:size_p
    for j = 1:size_p
        % our test row
        es = [ es_row(lhs(i,1)), es_row(lhs(i,2)), es_row(lhs(i,3)),...
                es_row(rhs(j,1)), es_row(rhs(j,2)), es_row(rhs(j,3)) ];

        a = es(1);
b = es(1)+es(2)+es(3)-es(4)-es(6);
c = es(1)+es(3)-es(4)-es(6);
d = es(1)+es(3)-es(4);
e = es(1)-es(4);

s = solver(a,b,c,d,e);
% Test to see if any of the angles are greater than pi/2 radians
if checker(a,b,c,d,e,s) == 0
    continue
end

% Test via Law of Cosines
a1 = acos(s/(a+s));
a2 = acos(s/(b+s));
a3 = acos((c+s)/(b+s));
a4 = acos((c+s)/(d+s));
a5 = acos((e+s)/(d+s));
a6 = acos((e+s)/(a+s));

V = A*es'+s*ones(6,1); % solution

% V(1) = U1, V(5) = U2, V(3) = U3
side(1) = sqrt( V(1)^2 + V(5)^2 - 2*V(1)*V(5)*cos(a1+a2));
side(2) = sqrt( V(5)^2 + V(3)^2 - 2*V(5)*V(3)*cos(a3+a4));
side(3) = sqrt( V(1)^2 + V(3)^2 - 2*V(1)*V(3)*cos(a5+a6));

angle(1) = acos( (side(1)^2 + side(3)^2 - side(2)^2) /
                (2*side(1)*side(3)) );
angle(2) = acos( (side(1)^2 + side(2)^2 - side(3)^2) /
                (2*side(1)*side(2)) );
angle(3) = acos( (side(2)^2 + side(3)^2 - side(1)^2) /
                (2*side(2)*side(3)) );

% checking for complex angles
not = 0;
for k = 1:3
    if isreal( angle(k) ) == 0
        not = 1;
    end
end

if not == 1
    continue
end

% taking sides
sides = side(1)^2 + side(2)^2 + side(3)^2 + ...
       -(side(2)^2 + side(3)^2 - 2*side(2)*side(3)*cos(angle(1))) + ...
       -(side(1)^2 + side(3)^2 - 2*side(1)*side(3)*cos(angle(2))) + ...
       -(side(1)^2 + side(2)^2 - 2*side(1)*side(2)*cos(angle(3)));
normed_sides = norm(sides);

if normed_sides < stdnorm
    T(t,:) = [ side(1), angle(1),...
                side(2), angle(2),...
                side(3), angle(3),... ]
end
normed_sides];
    t = t+1;
    stdnorm = normed_sides;
end
end
end

%%% setup
TT = zeros(3,8);
i = 1;

%%% test via Law of Sines
for q = 1:size(T,1)
    ls1 = T(q,3)/T(q,2);
    ls2 = T(q,1)/T(q,6);
    ls3 = T(q,5)/T(q,4);

    L1 = ls1 - ls2;
    L2 = ls1 - ls3;
    L3 = ls2 - ls3;

    LL = L1+L2+L3;

    if abs(LL) < 1
        TT(i,1:7) = T(q,:);
        TT(i,8) = LL;
        i = i+1;
    end
end

T = sortrows(TT,8);
function y = solver(a,b,c,d,e)
%SOLVER
% calling sequences:
% y = solver(a,b,c,d,e)
% solver(a,b,c,d,e)
%
% inputs:
% a       es(1)
% b       es(1)+es(2)+es(3)-es(4)-es(6)
% c       es(1)+es(3)-es(4)-es(6)
% d       es(1)+es(3)-es(4)
% e       es(1)-es(4)
%
% outputs:
% y       approximation of S1
%
% (c) 2008 Daniel Callahan, released under the MIT license

K = 101;
s = linspace(50*eps,3+50*eps,K);
S = 0;
exact = 0;

for i = 1:K-1
    F = @(x) acos(x/(a+x))+...  
         acos(x/(b+x))+...  
         acos((c+x)/(b+x))+... 
         acos((c+x)/(d+x))+...  
         acos((e+x)/(d+x))+... 
         acos((e+x)/(a+x))-2*pi;

    if ( isreal(F(s(i))) == 0 | isreal(F(s(i+1))) == 0 )
        continue
    end

    if F(s(i)) > 0
        if F(s(i+1)) < 0
            S = bisection(F, s(i), s(i+1), 10*eps);
        end
    end

    y = S;
end
function [es] = square( cm, T )
%SQUARE generates a square with a center of mass
% calling sequences:
% [es] = square( cm, T )
% square( cm, T )
% inputs:
% cm coordinate of the center of mass
% T the number of "steps" that the CM will take in each segment of its permutation (three segments total, with a grand total of 3T-2 steps)
% outputs:
% es the energy spectra
%
% (c) 2008 Daniel Callahan, released under the MIT license

%% Defaults
L = linspace(0,2.*pi,5);
xv = cos(L)';
yv = sin(L)';

% The midpoint on the upper segment of the square
xh = (xv(1)+xv(2))/2;
yh = (yv(1)+yv(2))/2;

%% Perturbation of cm
p = zeros(2, 3*T-2);
t = linspace(0,1,T);
for i = 1:T
    p(1, i) = t(i)*xv(1)+(1-t(i))*cm(1);
    p(2, i) = t(i)*yv(1)+(1-t(i))*cm(2);
end
for i = 2:T
    p(1, i+T-1) = t(i)*xh+(1-t(i))*xv(1);
    p(2, i+T-1) = t(i)*yh+(1-t(i))*yv(1);
end
for i = 2:T
    p(1, i+2*T-2) = t(i)*cm(1)+(1-t(i))*xh;
    p(2, i+2*T-2) = t(i)*cm(2)+(1-t(i))*yh;
end

%% Calculations
es = zeros(3*T-2, 6);
for i = 1:3*T-2
    % Unstable equilibria
    u1 = sqrt((xv(1)-p(1,i))^2+(yv(1)-p(2,i))^2);
    u2 = sqrt((xv(2)-p(1,i))^2+(yv(2)-p(2,i))^2);
    u3 = sqrt((xv(3)-p(1,i))^2+(yv(3)-p(2,i))^2);
    u4 = sqrt((xv(4)-p(1,i))^2+(yv(4)-p(2,i))^2);

    % Stable equilibria
    s1a = abs((yv(2)-yv(1))*(xv(1)-p(1,i))-(yv(1)-p(2,i))*(xv(2)-xv(1)));
    s1b = sqrt((yv(2)-yv(1))^2+(xv(2)-xv(1))^2);
    s1 = s1a/s1b;
    s2a = abs((yv(3)-yv(2))*(xv(2)-p(1,i))-(yv(2)-p(2,i))*(xv(3)-xv(2)));
    s2b = sqrt((yv(3)-yv(2))^2+(xv(3)-xv(2))^2);
    s2 = s2a/s2b;
    s3a = abs((yv(4)-yv(3))*(xv(3)-p(1,i))-(yv(3)-p(2,i))*(xv(4)-xv(3)));
    s3b = sqrt((yv(4)-yv(3))^2+(xv(4)-xv(3))^2);
    s3 = s3a/s3b;
    s4a = abs((yv(1)-yv(4))*(xv(4)-p(1,i))-(yv(4)-p(2,i))*(xv(1)-xv(4)));
    s4b = sqrt((yv(1)-yv(4))^2+(xv(1)-xv(4))^2);
    s4 = s4a/s4b;

    % Energy spectra
    es(i, 1) = i;
    es(i, 2) = u1 - s1;
    es(i, 3) = u2 - s2;
    es(i, 4) = u3 - s3;
    es(i, 5) = u4 - s4;
    es(i, 6) = u1 - s4;
    es(i, 7) = u2 - s1;
    es(i, 8) = u3 - s2;
    es(i, 9) = u4 - s3;
end

% Plotting
hold on
plot(es(:,1), es(:,2), 'bo-'); plot(es(:,1), es(:,3), 'bo-'); plot(es(:,3), es(:,4), 'bo-'); plot(es(:,1), es(:,5), 'ko-'); plot(es(:,1), es(:,6), 'ko-'); plot(es(:,3), es(:,7), 'ko-'); plot(es(:,1), es(:,8), 'ro-'); plot(es(:,1), es(:,9), 'ro-');
hold off
axis([1 3*T-2 0 max(max(es(:,2:9)))]);
es = es(:,2:9);
function TT = tworow(es_row1, es_row2)
%TWOROW
%
% calling sequences:
% TT = tworow ( es_row1, es_row2 )
% tworow ( es_row1, es_row2 )
%
% inputs:
% es_row1     row of data collected from the forward problem
% es_row2     row of data, distinct from es_row1
%
% outputs:
% TT          results
%
% (c) 2008 Daniel Callahan, released under the MIT license

T1 = onerow(es_row1);
T2 = onerow(es_row2);
TT = zeros(1,8);

size_T1 = size(T1,1);
size_T2 = size(T2,1);

k = 1;

for i = 1:size_T1
    for j = 1:size_T2
        if norm (T1(i,7) - T2(j,7)) < 1e5*eps
            TT(k,:) = T1(i,:);
            k = k+1;
        end
    end
end
end
REFERENCES
LIST OF REFERENCES