

Project #3B:  
Solving Systems of Linear Equations

Henry Song  
May 19, 2021  
MA375  
Dr. L. Aquino

## 1. Gaussian Elimination Code:

The snippet shown below is the method I created for gaussian elimination in Python. The script is based off of the pseudocode in Topic 7, Part 2 lecture notes (from Epperson). I modified the method so that it uses an augmented matrix rather than separate  $A$  and  $b$  matrices as it seems more natural to use.

```
def gauss(a, b):
    aug_matrix = create_augmented_matrix(a,b) #creates augmented matrix to use for elimination
    n = len(aug_matrix) #number of columns + 1. Since gaussian elimination uses nxn matrices, this is also size of matrix + 1
    (augmented matrix)

    #forward elimination
    for i in range(0, n):
        a_m = abs(aug_matrix[i][i]) # sets a_m to value on diagonal

        # searches for largest value of column i
        p = i #counter for searching column i
        for j in range(i + 1, n):
            if abs(aug_matrix[j][i]) > a_m:
                a_m = abs(aug_matrix[j][i]) #replaces a_m with larger variable
                p = j #set p as new index of larger value

        # swaps row i with row p
        for k in range(i, n+1):
            hold_value = aug_matrix[p][k]
            aug_matrix[p][k] = aug_matrix[i][k]
            aug_matrix[i][k] = hold_value

        # startt elimination for column i
        for j in range(i + 1, n):
            m = -aug_matrix[j][i]/aug_matrix[i][i] # calculates multiplied to zero out entry
            for k in range(i, n+1):
                if i == j: aug_matrix[j][k] = 0
                else: aug_matrix[j][k] += m * aug_matrix[i][k]

    #backward substitution
    x = [0 for i in range(n)] #blank matrix for x (solution) matrix
    for i in range(n-1, -1, -1): # works backwards from last matrix row to first matrix row
        x[i] = (aug_matrix[i][n]/aug_matrix[i][i]) #solves x for row n-1
        for k in range(i-1, -1, -1): aug_matrix[k][n] -= aug_matrix[k][i] * x[i]

    return x
```

## 2a. Matrix Equation A:

Using the given matrices for matrix equation A and the gaussian elimination method created, I was able to achieve the solution:

$$[0 \ 1 \ 2 \ 3 \ 4]^T$$

```
henry@Henrys-MacBook-Pro project3 % python3 GaussianElimination.py
MA375 - Project #3B
Henry Song

1. Use matrix equation A from Project 3B
2. Use matrix equation B from Project 3B
3. Use custom matrix equation
4. Use randomly generated matrix
5. Quits program
Please select an option: 1

A =
| 14   14   -9    3   -5 |
| 14   52  -15    2  -32 |
| -9  -15   36   -5   16 |
| 3    2   -5   47   49 |
| -5  -32   16   49   79 |

B =
| -15.0000 |
| -100.0000 |
| 106.0000 |
| 329.0000 |
| 463.0000 |

Option 1: x = | -0.0000  1.0000  2.0000  3.0000  4.0000 |

1. Use matrix equation A from Project 3B
2. Use matrix equation B from Project 3B
3. Use custom matrix equation
4. Use randomly generated matrix
5. Quits program
Please select an option: █
```

## 2b. Matrix Equation B:

Using the given matrices for matrix equation A and the gaussian elimination method created, I was able to achieve the solution:

$$[0 \ 1 \ 2 \ 3 \ 4]^T$$

```
henry@Henrys-MacBook-Pro project3 % python3 GaussianElimination.py
MA375 - Project #3B
Henry Song

1. Use matrix equation A from Project 3B
2. Use matrix equation B from Project 3B
3. Use custom matrix equation
4. Use randomly generated matrix
5. Quits program
Please select an option: 2

A =
| 9      3      2      0      7 |
| 7      6      9      6      4 |
| 2      7      7      8      2 |
| 0      9      7      2      2 |
| 7      3      6      4      3 |

B =
| 35.0000 |
| 58.0000 |
| 53.0000 |
| 37.0000 |
| 39.0000 |

Option 2: x = | -0.0000  1.0000  2.0000  3.0000  4.0000 |

1. Use matrix equation A from Project 3B
2. Use matrix equation B from Project 3B
3. Use custom matrix equation
4. Use randomly generated matrix
5. Quits program
Please select an option: █
```

### 3. Solving other matrices:

These matrices are generated using a manually entered size and filled with randomly generated integers. I apologize in advance for the tiny pictures, especially as the matrix gets larger. Normal resolution pictures are also included as stand-alone images in submission folder.

<p><b>2x2 matrix:</b></p> <pre>1. Use matrix equation A from Project 3B 2. Use matrix equation B from Project 3B 3. Use custom matrix equation 4. Use randomly generated matrix 5. Quits program Please select an option: 4  Enter size of desired random matrix: 2  A =   228   743     527   174    B =   760.0000     591.0000    Option 4: x =   0.8721  0.7553  </pre>	<p><b>3x3 matrix:</b></p> <pre>1. Use matrix equation A from Project 3B 2. Use matrix equation B from Project 3B 3. Use custom matrix equation 4. Use randomly generated matrix 5. Quits program Please select an option: 4  Enter size of desired random matrix: 3  A =   579   345   430     473   863   215     398   748   723    B =   671.0000     384.0000     864.0000    Option 4: x =   0.4451 -0.0480  0.9997  </pre>
<p><b>4x4 matrix:</b></p> <pre>1. Use matrix equation A from Project 3B 2. Use matrix equation B from Project 3B 3. Use custom matrix equation 4. Use randomly generated matrix 5. Quits program Please select an option: 4  Enter size of desired random matrix: 4  A =   249   299   806   556     586   873   272   907     794   158   85   787     677   787   498   387    B =   574.0000     321.0000     52.0000      518.0000    Option 4: x =   0.0549  0.2315  0.6851 -0.1098  </pre>	<p><b>5x5 matrix:</b></p> <pre>1. Use matrix equation A from Project 3B 2. Use matrix equation B from Project 3B 3. Use custom matrix equation 4. Use randomly generated matrix 5. Quits program Please select an option: 4  Enter size of desired random matrix: 5  A =   457   889   258   653   958     928   839   651   680   410     675   476   593   95   498     387   465   619   666   440     884   831   178   727   188    B =   901.0000     638.0000     912.0000     418.0000     554.0000    Option 4: x =   2.4044 -2.6581 -0.9485  0.5566  2.1362  </pre>

### 6x6 matrix:

```
1. Use matrix equation A from Project 3B
2. Use matrix equation B from Project 3B
3. Use custom matrix equation
4. Use randomly generated matrix
5. Quits program
Please select an option: 4
```

Enter size of desired random matrix: 6

```
A =
| 389 217 681 362 502 90 |
| 352 559 941 443 177 482 |
| 540 807 299 233 787 681 |
| 709 780 689 959 450 526 |
| 183 911 261 547 242 511 |
| 717 667 355 624 282 903 |
```

```
B =
| 587.0000 |
| 645.0000 |
| 627.0000 |
| 949.0000 |
| 111.0000 |
| 306.0000 |
```

Option 4: x = | 5.2366 3.3675 0.4267 -2.7490 -2.2463 -3.8731 |

### 7x7 matrix:

```
1. Use matrix equation A from Project 3B
2. Use matrix equation B from Project 3B
3. Use custom matrix equation
4. Use randomly generated matrix
5. Quits program
Please select an option: 4
```

Enter size of desired random matrix: 7

```
A =
| 672 854 687 791 448 372 346 |
| 413 251 457 653 268 707 916 |
| 304 835 615 895 486 933 445 |
| 808 647 845 452 697 273 548 |
| 98 22 800 329 289 42 536 |
| 430 822 637 847 560 924 633 |
| 535 461 944 643 371 357 619 |
```

```
B =
| 632.0000 |
| 376.0000 |
| 834.0000 |
| 963.0000 |
| 24.0000 |
| 692.0000 |
| 230.0000 |
```

Option 4: x = | 1.9965 -4.4272 -0.8400 3.0589 4.2427 1.1111 -3.1370 |

### 8x8 matrix:

Enter size of desired random matrix: 8

```
A =
| 102 179 423 47 562 724 912 38 |
| 767 990 274 802 293 944 308 147 |
| 67 884 675 575 422 375 127 443 |
| 847 857 36 972 720 139 894 148 |
| 764 95 927 493 863 45 273 508 |
| 43 945 442 901 437 258 95 928 |
| 44 525 375 151 518 499 592 209 |
| 289 436 363 179 101 53 913 472 |
```

```
B =
| 690.0000 |
| 116.0000 |
| 265.0000 |
| 487.0000 |
| 646.0000 |
| 46.0000 |
| 243.0000 |
| 561.0000 |
```

Option 4: x = | -0.3081 -0.6327 0.9470 0.9693 -0.3999 -0.1071 0.7802 -0.5458 |

### 9x9 matrix:

Enter size of desired random matrix: 9

```
A =
| 138 747 657 809 83 230 526 235 690 |
| 395 653 206 233 881 104 559 598 255 |
| 997 634 383 787 688 292 208 329 807 |
| 672 69 399 943 685 322 174 312 722 |
| 749 609 503 484 946 582 791 84 879 |
| 389 591 276 568 255 56 705 783 447 |
| 397 627 148 407 23 499 73 697 192 |
| 703 666 568 30 801 122 817 455 441 |
| 2 527 827 936 618 821 100 595 37 |
```

```
B =
| 642.0000 |
| 615.0000 |
| 636.0000 |
| 544.0000 |
| 237.0000 |
| 610.0000 |
| 113.0000 |
| 0.0000 |
| 120.0000 |
```

Option 4: x = | -0.6231 0.5739 -0.7194 0.7238 0.4995 -0.8009 -0.2375 0.1449 0.6086 |

### 10x10 matrix:

Enter size of desired random matrix: 10

```
A =
| 420 351 561 24 589 972 780 101 994 179 |
| 237 767 105 409 937 901 728 182 28 344 |
| 688 243 20 498 805 841 298 648 938 491 |
| 180 365 989 96 85 342 942 899 22 332 |
| 969 862 235 914 688 348 485 358 969 427 |
| 238 655 867 872 432 217 735 740 595 956 |
| 131 248 996 415 204 697 779 406 667 694 |
| 482 465 630 242 348 254 284 386 827 176 |
| 316 24 359 104 776 902 700 663 221 318 |
| 4 201 234 283 781 523 42 908 442 31 |
```

```
B =
| 658.0000 |
| 777.0000 |
| 46.0000 |
| 671.0000 |
| 442.0000 |
| 578.0000 |
| 226.0000 |
| 439.0000 |
| 686.0000 |
| 946.0000 |
```

Option 4: x = | -0.7291 -0.0566 0.3297 0.2538 1.3576 -0.7862 0.9592 0.0769 0.1921 -1.0571 |

### 10x10 matrix:

Enter size of desired random matrix: 10

```
A =
| 989 830 140 172 978 164 153 443 686 685 |
| 127 503 915 538 333 794 557 446 539 293 |
| 724 147 700 80 300 724 463 88 801 764 |
| 12 932 986 737 759 226 237 87 784 237 |
| 753 624 724 942 934 429 79 249 224 465 |
| 341 472 422 687 424 7 326 686 278 573 |
| 531 612 809 343 541 48 688 208 760 491 |
| 335 938 202 617 851 622 950 618 319 211 |
| 472 866 403 518 844 269 751 891 993 833 |
| 305 31 929 548 23 109 363 915 855 961 |
```

```
B =
| 915.0000 |
| 585.0000 |
| 518.0000 |
| 662.0000 |
| 423.0000 |
| 514.0000 |
| 835.0000 |
| 376.0000 |
| 114.0000 |
| 571.0000 |
```

Option 4: x = | 6.2506 4.3139 -3.2894 5.7250 -9.3137 0.2945 -1.1414 -0.6807 5.2419 -5.0086 |

**Source Code:**

- Song\_Henry\_Project3.zip
  - Project Report.pdf
  - GaussianElimination.py
    - \*Note: This file contains both the gaussian method and the driver scripts. To use, simply run 'python3 GaussianElimination.py' in local directory.
  - /matrix\_calculation\_images
    - 2x2.png
    - 3x3.png
    - 4x4.png
    - 5x5.png
    - 6x6.png
    - 7x7.png
    - 8x8.png
    - 9x9.png
    - 10x10\_1.png
    - 10x10\_2.png
- GitHub
  - <https://github.com/henrysong671/MA375-Project2>