Bahria University
Karachi Campus

Course Code: *CEN - 221*
Couse: *Computer Architecture & Organization*
Term: *Fall 2021*
Class: *BSE – 3(B)*

"Semester Project's Report"

Submitted By:

S.No.	*Name*	*Enrollment*
1.	*Mohammad Abdul Nazif Malhi*	*02-131192-002*
2.	*Noor us Sabah*	*02-131192-038*

Submitted To:
Course Instructor: *Engr. Syed Samar Yazdani*
Lab Instructor: *Engr. Mohammad Rehan Big*

Remarks: _____ ***Date:*** _____ ***Signature:*** _____.

Contents

Abstract 2

Introduction 4

Code 5

Output 36

1. 2D Equation 36
2. 3D Equation 37

Gauss Elimination Calculator in MIPS MARS

Abstract

The equation solver allows to solve equations with an unknown with calculation steps: linear equation, quadratic equation, logarithmic equation, differential equation. An **equation** is an algebraic equality involving one or more unknowns. Solving an equation is the same as determining that unknown or unknowns. The unknown is also called a variable. This **equation calculator** can solve equations with an unknown, the calculator can solve **equations with variables on both sides** and also **equations with parentheses**:

1. **Solving linear equation**
2. **Solving quadratic equation**
3. **Solving cubic equation**

For equation solving, Solve and Reduce functions, which contain a broad range of methods for all kinds of algebra, from basic linear and quadratic equations to multivariate nonlinear systems. Linear algebra methods such as ***Gaussian elimination*** are used, with optimizations to increase speed and reliability. Other operations rely on theorems and algorithms from number theory, abstract algebra and other advanced fields to compute results. These methods are carefully designed and chosen to enable to solve the greatest variety of problems while also minimizing computation time.

Although such methods are useful for direct solutions, it is also important for the system to understand how a human would solve the same problem. As a result, separate algorithms to show algebraic operations step by step using classic techniques that are easy for humans to recognize and follow. This includes elimination method

Introduction

This Mips Mars calculator will help you to solve a system of linear equations using Gauss-Jordan elimination. Using this mips Mars, you will receive a detailed step-by-step solution to your problem, which will help you understand the algorithm how to solve system of linear equations by Gauss-Jordan elimination.
The Gaussian elimination algorithm (also called Gauss-Jordan, or pivot method) makes it possible to find the solutions of a system of linear equations, and to determine the inverse of a matrix.

The algorithm works on the rows of the matrix, by exchanging or multiplying the rows between them (up to a factor).

At each step, the algorithm aims to introduce into the matrix, on the elements outside the diagonal, zero values.
Tool to apply the Gaussian elimination method and get the row reduced echelon form, with steps, details, inverse matrix and vector solution.

Code

```
.data
.align 2
varword: .word main , c1 , c2

wlcm: .asciiz "\n_____\n  ****Equation Solving By Guass Jordan
Elimination****\n_____"
cases: .asciiz  "\n\n   1.  ax1 + bx2 = c\n   2.  ax1 + bx2 + cx3 = d"
msg: .asciiz "\n\n   Plz select correct code for equation solving : "
again: .asciiz "\n\n\n    Do you want to solve another equation ?\n\tEnter (1) for 'YES' & (0) for
'NO' : "
desh: .asciiz "\n_____\n"

#_____
_____
#_____
_____
#_____
_____
#_____
_____
myArray:  .space 96
M11: .asciiz "\tEnter M11  --->  "
M12: .asciiz "\tEnter M12  --->  "
M13: .asciiz "\tEnter M13  --->  "
M21: .asciiz "\tEnter M21  --->  "
M22: .asciiz "\tEnter M22  --->  "
M23: .asciiz "\tEnter M23  --->  "
M31: .asciiz "\tEnter M31  --->  "
M32: .asciiz "\tEnter M32  --->  "
M33: .asciiz "\tEnter M33  --->  "
X1:  .asciiz "\tEnter  X1  --->  "
X2:  .asciiz "\tEnter  X2  --->  "
X3:  .asciiz "\tEnter  X3  --->  "

#_____
_____
space: .asciiz "  "
```

```
newLine: .asciiz  "\n\t"
tab: "\t"
b: .asciiz  " | "
s: .asciiz "  "

formatm: .asciiz "\n\n\n Plz enter your input for the equation in following format:\n\tm11  m12  |
x1\n\tm21  m22  |  x2\n\nLet, start entering the entities.....\n"
format: .asciiz " Plz enter your input for the equation in following format:\n\tm11  m12  m13  |
x1\n\tm21  m22  m23  |  x2\n\tm31  m32  m33  |  x3\n\nLet, start entering the entities.....\n"

augmented: .asciiz "\n\n    The augmented form of matrix is :"



step_1:.asciiz "\n\n\t<---- Step : 1 ---->\nmaking  pivot of 1st  row - 1st  column\n"
step_2:.asciiz "\n\n\t<---- Step : 2 ---->\neliminating the elements below the pivot\n"
step_3:.asciiz "\n\n\t<---- Step : 3 ---->\n making  pivot  of  2nd  row - 2nd column\n"
step_4:.asciiz "\n\n\t<---- Step : 4 ---->\neliminating the elements below the pivot\n"
step_5:.asciiz "\n\n\t<---- Step : 5 ---->\n making  pivot  of  3rd row - 3rd element\n"
step_6:.asciiz "\n\n\t<---- Step : 6 ---->\n eliminating the elements below the pivot\n"



skip: .asciiz "As the pivot to be set is already 1, \NSo that we will skip this step and move to the
next sstep"

solution_msg: .asciiz "\n\n\t\tSolution\n  The augmneted matrix says that :\n"
X1_sol: .asciiz "\n   1 -----> x1 = "
X2_sol: .asciiz "\n   2 -----> x2 = "
X3_sol: .asciiz "\n   3 -----> x3 = "

no_sol: .asciiz
infinite_sol: .asciiz

#_____________________________________________________________________________________________
_____________________________
#_____________________________________________________________________________________________
_____________________________
#_____________________________________________________________________________________________
_____________________________
#_____________________________________________________________________________________________
_____________________________
```

```
thank: .asciiz "\n\n_____thankyou_____ "

#_____
_____
#_____
_____
#_____
_____
.text
main:
la     $a0, wlcm
li     $v0, 4
syscall

operation:
la     $a0, cases
li     $v0, 4
syscall

correct:
la     $a0, msg
li     $v0, 4
syscall

li $v0 , 5
syscall

blt $v0 , 1 , correct
bgt $v0 , 2 , correct

la $a1 , varword
sll $t0 , $v0 , 2

add $t1, $a1 , $t0
lw $t2 , 0($t1)
jr $t2



#_____
_____
#_____
_____
```

```
#_____
_____
#_____
_____
c1:
#_____
_____
#_____
_____
addi $t0 , $zero , 0   #indexing
#_____
_____
#_____
_____

la     $a0, formatm
li     $v0, 4
syscall

la $a0, M11
li $v0, 4
syscall

li $v0, 6
syscall
mfc1 $v0 , $f0
swc1 $f0 , myArray($t0)

#_____
_____
#_____
_____

la $a0, M12
li $v0, 4
syscall

addi $t0 , $t0 , 4
li $v0, 6
syscall
mfc1 $v0 , $f0

swc1 $f0 , myArray($t0)
```

```
#_____
_____
#_____
_____
la $a0, X1
li $v0, 4
syscall

addi $t0 , $t0 , 4
li $v0, 6
syscall
mfc1 $v0 , $f0

swc1 $f0 , myArray($t0)
#_____
_____
#_____
_____
la $a0, M21
li $v0, 4
syscall

addi $t0 , $t0 , 4
li $v0, 6
syscall
mfc1 $v0 , $f0

swc1 $f0 , myArray($t0)
#_____
_____
#_____
_____
la $a0, M22
li $v0, 4
syscall

addi $t0 , $t0 , 4
li $v0, 6
syscall
mfc1 $v0 , $f0

swc1 $f0 , myArray($t0)
```

```
#_____
_____
#_____
_____
la $a0, X2
li $v0, 4
syscall

addi $t0 , $t0 , 4
li $v0, 6
syscall
mfc1 $v0 , $f0

swc1 $f0 , myArray($t0)
#_____
_____
#_____
_____

la  $a0, augmented
li  $v0, 4
syscall

addi $t9 , $zero , 0  #flag for step
j printingm



step1m:

la $a0, step_1
li $v0,4
syscall


addi $t9 , $t9 , 1
j guassian_pivotm


#_____
_____
#_____
_____
```

```
step2m:
la $a0, step_2
li $v0,4
syscall

addi $t3 , $zero , 0
addi $t3 , $t3 , 12

lwc1 $f1, myArray($t3)
mov.s $f0 , $f1

sub.s  $f1 , $f1 , $f1
swc1 $f1 , myArray($t3)
#_____________________________________________________________________________
_____________________________
#_____________________________________________________________________________
_____________________________
addi $t3 , $zero , 0
addi $t3 , $t3 , 4

lwc1 $f1, myArray($t3)
mov.s $f2 , $f1


#_____________________________________________________________________________
_____________________________
addi $t3 , $zero , 0
addi $t3 , $t3 , 8

lwc1 $f1, myArray($t3)
mov.s $f3 , $f1

#_____________________________________________________________________________
_____________________________

mul.s $f5 , $f2 , $f0
mul.s $f6 , $f3 , $f0

#_____________________________________________________________________________
_____________________________
#_____________________________________________________________________________
_____________________________
```

```
addi $t3 , $zero , 0
addi $t3 , $t3 , 16

lwc1 $f1, myArray($t3)
sub.s  $f1 , $f1 , $f5
swc1 $f1 , myArray($t3)


addi $t3 , $zero , 0
addi $t3 , $t3 , 20

lwc1 $f1, myArray($t3)
sub.s  $f1 , $f1 , $f6
swc1 $f1 , myArray($t3)

#_____________________________________________________________________________________
_____________________________

addi $t9 , $t9 , 1
j printingm

#_____________________________________________________________________________________
_____________________________
#_____________________________________________________________________________________
_____________________________


step3m:

la $a0, step_3
li $v0,4
syscall


addi $t9 , $t9 , 1
j guassian_pivotm


#_____________________________________________________________________________________
_____________________________
#_____________________________________________________________________________________
_____________________________
```

step4m:

la $a0, step_4
li $v0,4
syscall

addi $t3 , $zero , 0
addi $t3 , $t3 , 4

lwc1 $f1, myArray($t3)
mov.s $f0 , $f1

sub.s $f1 , $f1 , $f1
swc1 $f1 , myArray($t3)
#_____
_____
#_____
_____

#_____
_____
addi $t3 , $zero , 0
addi $t3 , $t3 , 20

lwc1 $f1, myArray($t3)
mov.s $f2 , $f1

#_____
_____

mul.s $f5 , $f2 , $f0

#_____
_____
#_____
_____

addi $t3 , $zero , 0
addi $t3 , $t3 , 8

lwc1 $f1, myArray($t3)

```
sub.s  $f1 , $f1 , $f5
swc1 $f1 , myArray($t3)


addi $t9 , $t9 , 1
j printingm


#_____________________________________________________________________________
_____________________________
#_____________________________________________________________________________
_____________________________
guassian_pivotm:
beq $t9 , 1 , steponem
beq $t9 , 3 , stepthreem

steponem:
addi $t3 , $zero , 0
j reductionm


stepthreem:
addi $t3 , $zero , 0
addi $t3 , $t3 , 16
j reductionm


reductionm:
lwc1 $f1, myArray($t3)
mov.s $f0 , $f1

div.s $f1 , $f1 , $f1
swc1 $f1 , myArray($t3)

#_____________________________________________
addi $t3,$t3,4
lwc1 $f1, myArray($t3)

div.s $f1 , $f1 , $f0
swc1 $f1 , myArray($t3)

beq  $t9 , 3 , printingm
#_____________________________________________
```

```
addi $t3,$t3,4
lwc1 $f1, myArray($t3)

div.s $f1 , $f1 , $f0
swc1 $f1 , myArray($t3)

beq $t9 , 1 , printingm
#_____________________________________________________________________________
_____________________________
#_____________________________________________________________________________
_____________________________

#_____________________________________________________________________________
_____________________________
#_____________________________________________________________________________
_____________________________
printingm:
addi $t1 , $zero , 0   #loop counter
addi $t0 , $zero , 0   #indexing
addi $t2 , $zero , 5  #upper bound for 3*3 matrix

la $a0, newLine
li $v0,4
syscall


print_loopm:
bgt $t1 , $t2 , next_stepm

lwc1 $f13, myArray($t0)
li $v0,2
mov.s $f12,$f13
syscall


la $a0, s
li $v0,4
syscall

addi $t1, $t1, 1
```

```
addi $t0, $t0, 4


beq $t1,3,new_linem

beq $t1,2,blockerm
beq $t1,5,blockerm

j print_loopm

#_____________________________________________________________________________
_____________________________
#_____________________________________________________________________________
_____________________________
new_linem:
la $a0, newLine
li $v0,4
syscall
j print_loopm

#_____________________________________________________________________________
_____________________________
#_____________________________________________________________________________
_____________________________
blockerm:
la $a0 , b
li $v0,4
syscall
j print_loopm

#_____________________________________________________________________________
_____________________________
#_____________________________________________________________________________
_____________________________
next_stepm:
beq $t9,0,step1m
beq $t9,1,step2m
beq $t9,2,step3m
beq $t9,3,step4m
beq $t9,6,solutionm
```

```
#_____
_____
#_____
_____
solutionm:
la $a0, desh
li $v0,4
syscall


la $a0, solution_msg
li $v0,4
syscall

la $a0, X1_sol
li $v0,4
syscall

addi $s0,$s0,8


lwc1 $f13, myArray($s0)
li $v0,2
mov.s $f12,$f13
syscall

la $a0, X2_sol
li $v0,4
syscall

addi $s0,$s0,12

lwc1 $f13, myArray($s0)
li $v0,2
mov.s $f12,$f13
syscall

#_____
_____
#_____
_____
```

```
la     $a0, desh
li     $v0, 4
syscall

j again_perform



#_____
_____
#_____
_____
#_____
_____
#_____
_____
c2:
la     $a0, desh
li     $v0, 4
syscall



#_____
#_____
addi $t0 , $zero , 0   #indexing
#_____
#_____


la $a0, M11
li $v0, 4
syscall

li $v0, 6
syscall
mfc1 $v0 , $f0
swc1 $f0 , myArray($t0)

#_____
```

```
#_____________

la $a0, M12
li $v0, 4
syscall

addi $t0 , $t0 , 4
li $v0, 6
syscall
mfc1 $v0 , $f0

swc1 $f0 , myArray($t0)

#_____________
#_____________

la $a0, M13
li $v0, 4
syscall

addi $t0 , $t0 , 4
li $v0, 6
syscall
mfc1 $v0 , $f0

swc1 $f0 , myArray($t0)
#_____________
#_____________
la $a0, X1
li $v0, 4
syscall

addi $t0 , $t0 , 4
li $v0, 6
syscall
mfc1 $v0 , $f0

swc1 $f0 , myArray($t0)
#_____________
#_____________
la $a0, M21
li $v0, 4
syscall
```

```
addi $t0 , $t0 , 4
li $v0, 6
syscall
mfc1 $v0 , $f0

swc1 $f0 , myArray($t0)
#_____________
#_____________
la $a0, M22
li $v0, 4
syscall

addi $t0 , $t0 , 4
li $v0, 6
syscall
mfc1 $v0 , $f0

swc1 $f0 , myArray($t0)
#_____________
#_____________
la $a0, M23
li $v0, 4
syscall

addi $t0 , $t0 , 4
li $v0, 6
syscall
mfc1 $v0 , $f0

swc1 $f0 , myArray($t0)
#_____________
#_____________
la $a0, X2
li $v0, 4
syscall

addi $t0 , $t0 , 4
li $v0, 6
syscall
mfc1 $v0 , $f0

swc1 $f0 , myArray($t0)
```

```
#_____________
#_____________
la $a0, M31
li $v0, 4
syscall

addi $t0 , $t0 , 4
li $v0, 6
syscall
mfc1 $v0 , $f0

swc1 $f0 , myArray($t0)
#_____________
#_____________
la $a0, M32
li $v0, 4
syscall

addi $t0 , $t0 , 4
li $v0, 6
syscall
mfc1 $v0 , $f0

swc1 $f0 , myArray($t0)
#_____________
#_____________
la $a0, M33
li $v0, 4
syscall

addi $t0 , $t0 , 4
li $v0, 6
syscall
mfc1 $v0 , $f0

swc1 $f0 , myArray($t0)
#_____________
#_____________
la $a0, X3
li $v0, 4
syscall

addi $t0 , $t0 , 4
```

```
li $v0, 6
syscall
mfc1 $v0 , $f0

swc1 $f0 , myArray($t0)
#_____________
#_____________

la  $a0, augmented
li  $v0, 4
syscall


addi $t9 , $t9 , 0 #flag for step
j printing


#_____________
#_____________
#_____________
#_____________
step1:

la $a0, step_1
li $v0,4
syscall


addi $t9 , $t9 , 1
j guassian_pivot


#_____________
#_____________
#_____________
#_____________

step2:
la $a0, step_2
li $v0,4
syscall

addi $t9 , $t9 , 1
```

```
step2_1st:
addi $t8 , $zero , 0
addi $t8 , $t8 , 1 #flag for elimating

addi $t3 , $zero , 0
addi $t3 , $t3 ,  16

addi $t4 , $zero , 0
addi $t4 , $t4 ,  4

addi $t5 , $zero , 0
addi $t5 , $t5 ,  20

j elimination

step2_2nd:
addi $t8 , $t8 , 1 #flag for elimating


addi $t3 , $zero , 0
addi $t3 , $t3 ,  32

addi $t4 , $zero , 0
addi $t4 , $t4 ,  4

addi $t5 , $zero , 0
addi $t5 , $t5 ,  36

j elimination

#_____________________________________________________________________________
_____________________________
#_____________________________________________________________________________
_____________________________
#_____________________________________________________________________________
_____________________________


#_____________________
#_____________________
step3:
la $a0, step_3
```

```
li $v0,4
syscall


addi $t9 , $t9 , 1
j guassian_pivot


#_____________
#_____________
step4:
la $a0, step_4
li $v0,4
syscall

addi $t9 , $t9 , 1
addi $t8 , $zero , 0

step4_1st:
addi $t8 , $t8 , 1
addi $t3 , $zero , 0
addi $t3 , $t3 ,  4

addi $t4 , $zero , 0
addi $t4 , $t4 ,  24

addi $t5 , $zero , 0
addi $t5 , $t5 ,  8

j elimination

step4_2nd:
addi $t8 , $t8 , 1 #flag for elimating

addi $t8 , $t8 , 1
addi $t3 , $zero , 0
addi $t3 , $t3 ,  36

addi $t4 , $zero , 0
addi $t4 , $t4 ,  24

addi $t5 , $zero , 0
addi $t5 , $t5 ,  40
```

```
j elimination


#_________________
#_________________
step5:
la $a0, step_5
li $v0,4
syscall

addi $t9 , $t9 , 1
j guassian_pivot

#_________________
#_________________
step6:
la $a0, step_6
li $v0,4
syscall


addi $t9 , $t9 , 1
addi $t8 , $zero , 0

step6_1st:
addi $t8 , $t8 , 1

addi $t3 , $zero , 0
addi $t3 , $t3 ,  8

addi $t4 , $zero , 0
addi $t4 , $t4 ,  44

addi $t5 , $zero , 0
addi $t5 , $t5 ,  12

j elimination

step6_2nd:
addi $t8 , $t8 , 1

addi $t3 , $zero , 0
```

```
addi $t3 , $t3 ,  24

addi $t4 , $zero , 0
addi $t4 , $t4 ,  44

addi $t5 , $zero , 0
addi $t5 , $t5 ,  28


j elimination

#_____________
#_____________
#_____________
#_____________
#_____________
#_____________
guassian_pivot:
beq $t9 , 1 , stepone
beq $t9 , 3 , stepthree
beq $t9 , 5 , stepfive

stepone:
addi $t3 , $zero , 0
j reduction


stepthree:
addi $t3 , $zero , 0
addi $t3 , $t3 , 20
j reduction

stepfive:
addi $t3 , $zero , 0
addi $t3 , $t3 , 40
j reduction


reduction:
lwc1 $f1, myArray($t3)
mov.s $f0 , $f1

div.s $f1 , $f1 , $f1
```

```
swc1 $f1 , myArray($t3)

#_____
addi $t3,$t3,4
lwc1 $f1, myArray($t3)

div.s $f1 , $f1 , $f0
swc1 $f1 , myArray($t3)

beq  $t9 , 5 , printing
#_____
addi $t3,$t3,4
lwc1 $f1, myArray($t3)

div.s $f1 , $f1 , $f0
swc1 $f1 , myArray($t3)

beq  $t9 , 3 , printing
#_____
addi $t3,$t3,4
lwc1 $f1, myArray($t3)

div.s $f1 , $f1 , $f0
swc1 $f1 , myArray($t3)

beq $t9 , 1 , printing

#_____________
#_____________
#_____________
#_____________


elimination:
beq $t9 , 2 , eliminating1
beq $t9 , 4 , eliminating2
beq $t9 , 6 , eliminating3


eliminating1:

lwc1 $f1, myArray($t3)
mov.s $f0 , $f1
```

```
sub.s  $f1 , $f1 , $f1
swc1 $f1 , myArray($t3)
#_____
_____
#_____
_____


lwc1 $f1, myArray($t4)
mov.s $f2 , $f1


#_____
_____
addi $t4 , $t4 , 4

lwc1 $f1, myArray($t4)
mov.s $f3 , $f1

#_____
_____
addi $t4 , $t4 , 4

lwc1 $f1, myArray($t4)
mov.s $f4 , $f1

#_____
_____

mul.s $f5 , $f2 , $f0
mul.s $f6 , $f3 , $f0
mul.s $f7 , $f4 , $f0

#_____
_____
#_____
_____


lwc1 $f1, myArray($t5)
sub.s  $f1 , $f1 , $f5
swc1 $f1 , myArray($t5)
```

```
addi $t5 , $t5 , 4

lwc1 $f1, myArray($t5)
sub.s  $f1 , $f1 , $f6
swc1 $f1 , myArray($t5)


addi $t5 , $t5 , 4

lwc1 $f1, myArray($t5)
sub.s  $f1 , $f1 , $f7
swc1 $f1 , myArray($t5)


beq  $t8 , 1 , step2_2nd
beq  $t8 , 2 , printing



eliminating2:
lwc1 $f1, myArray($t3)
mov.s $f0 , $f1

sub.s  $f1 , $f1 , $f1
swc1 $f1 , myArray($t3)
#_____________________________________________________________________________
_____________________________
#_____________________________________________________________________________
_____________________________


lwc1 $f1, myArray($t4)
mov.s $f2 , $f1

#_____________________________________________________________________________
_____________________________
addi $t4 , $t4 , 4

lwc1 $f1, myArray($t4)
mov.s $f3 , $f1
```

```
#_____________________________________________________________________________
_____________________________

mul.s $f5 , $f2 , $f0
mul.s $f6 , $f3 , $f0

#_____________________________________________________________________________
_____________________________
#_____________________________________________________________________________
_____________________________


lwc1 $f1, myArray($t5)
sub.s  $f1 , $f1 , $f5
swc1 $f1 , myArray($t5)


addi $t5 , $t5 , 4

lwc1 $f1, myArray($t5)
sub.s  $f1 , $f1 , $f6
swc1 $f1 , myArray($t5)


beq  $t8 , 1 , step4_2nd
beq  $t8 , 2 , printing

eliminating3:
lwc1 $f1, myArray($t3)
mov.s $f0 , $f1

sub.s  $f1 , $f1 , $f1
swc1 $f1 , myArray($t3)
#_____________________________________________________________________________
_____________________________
#_____________________________________________________________________________
_____________________________


lwc1 $f1, myArray($t4)
mov.s $f2 , $f1
```

```
#_____________________________________________________________________________
_____________________________

mul.s $f5 , $f2 , $f0


#_____________________________________________________________________________
_____________________________
#_____________________________________________________________________________
_____________________________


lwc1 $f1, myArray($t5)
sub.s  $f1 , $f1 , $f5
swc1 $f1 , myArray($t5)

beq  $t8 , 1 , step6_2nd
beq  $t8 , 2 , printing




skipping:
la     $a0, skip
li     $v0, 4
syscall

addi $t9 , $t9 , 1

beq $t9,1,step2
beq $t9,3,step4
beq $t9,5,step6


#_________________
#_________________


#_________________
#_________________
```

```
printing:
addi $t1 , $zero , 0   #loop counter
addi $t0 , $zero , 0   #indexing
addi $t2 , $zero , 11  #upper bound for 3*3 matrix

la $a0, newLine
li $v0,4
syscall


print_loop:
bgt $t1 , $t2 , next_step

lwc1 $f13, myArray($t0)
li $v0,2
mov.s $f12,$f13
syscall


la $a0, s
li $v0,4
syscall

addi $t1, $t1, 1
addi $t0, $t0, 4


beq $t1,4,new_line
beq $t1,8,new_line

beq $t1,3,blocker
beq $t1,7,blocker
beq $t1,11,blocker

j print_loop

#_____________
#_____________
new_line:
la $a0, newLine
li $v0,4
syscall
j print_loop
```

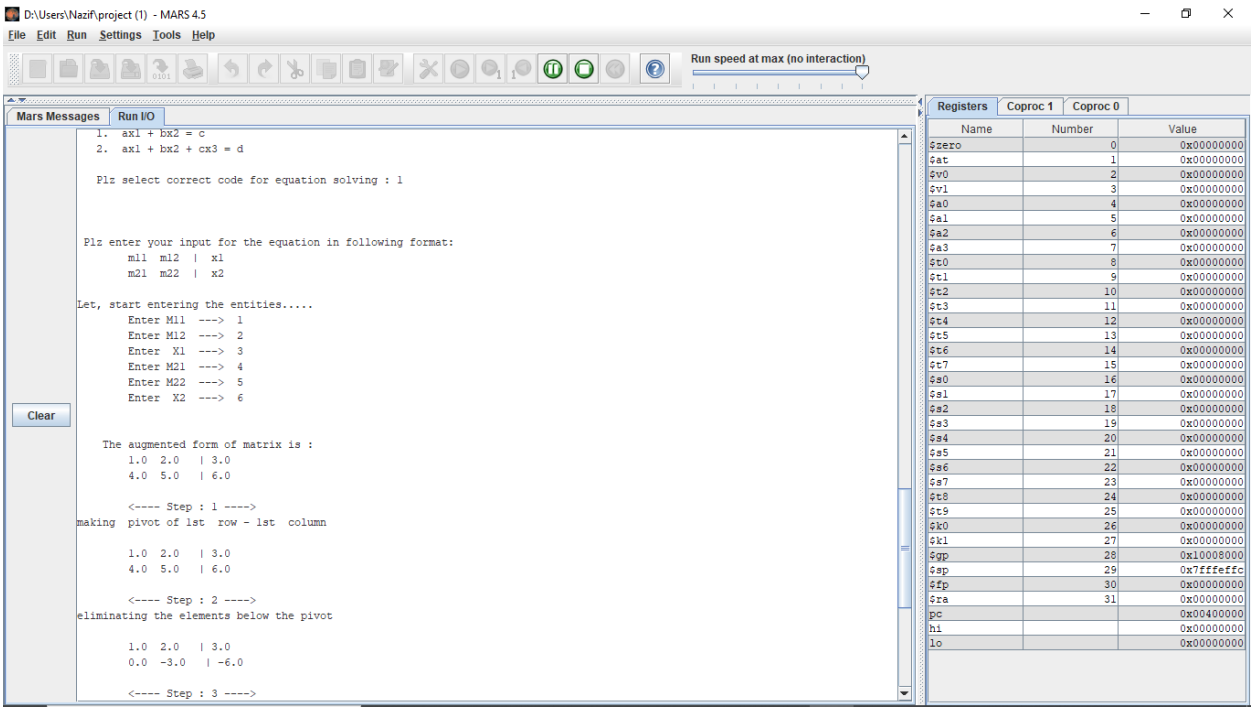
```
#_________________
#_________________
blocker:
la $a0 , b
li $v0,4
syscall
j print_loop

#_________________
#_________________
next_step:
beq $t9,0,step1
beq $t9,1,step2
beq $t9,2,step3
beq $t9,3,step4
beq $t9,4,step5
beq $t9,5,step6
beq $t9,6,solution


#_________________
#_________________
solution:
la $a0, solution_msg
li $v0,4
syscall

la $a0, X1_sol
li $v0,4
syscall

addi $s0,$s0,12


lwc1 $f13, myArray($s0)
li $v0,2
mov.s $f12,$f13
syscall

la $a0, X2_sol
li $v0,4
syscall
```

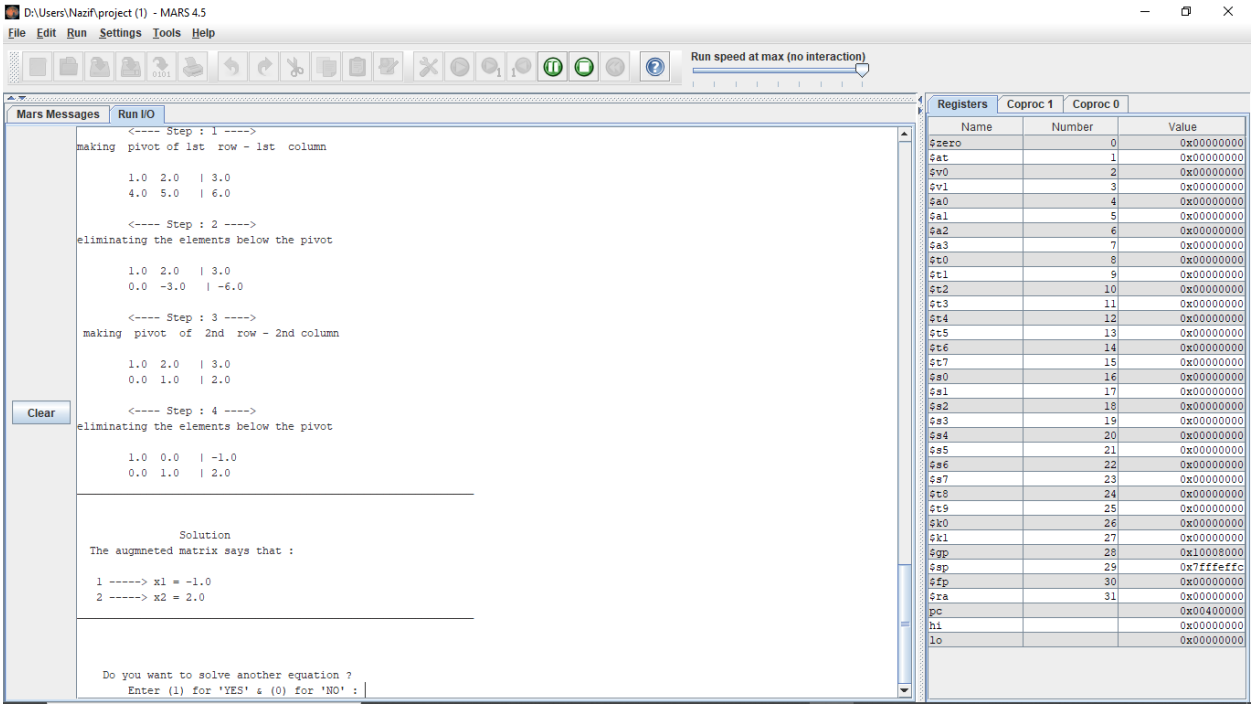
```
addi $s0,$s0,16

lwc1 $f13, myArray($s0)
li $v0,2
mov.s $f12,$f13
syscall


la $a0, X3_sol
li $v0,4
syscall

addi $s0,$s0,16

lwc1 $f13, myArray($s0)
li $v0,2
mov.s $f12,$f13
syscall

#_________________
#_________________


j again_perform


#_____________________________________________________________________________________
_____________________________
#_____________________________________________________________________________________
_____________________________
#_____________________________________________________________________________________
_____________________________
again_perform:
la     $a0, again
li     $v0, 4
syscall

li $v0 , 5
syscall
move $t7 , $v0
beqz $t7 , exit
beq $t7 , 1 , operation
```

```
bgt $t7 , 1 , again_perform
blt $t7 , 0 , again_perform


#_____________________________________________________________________________
_____________________________
#_____________________________________________________________________________
_____________________________
#_____________________________________________________________________________
_____________________________
exit:
la $a0, thank
li $v0,4
syscall

li $v0 , 10
syscall


#_____________________________________________________________________________
_____________________________
#_____________________________________________________________________________
_____________________________
#_____________________________________________________________________________
_____________________________
```

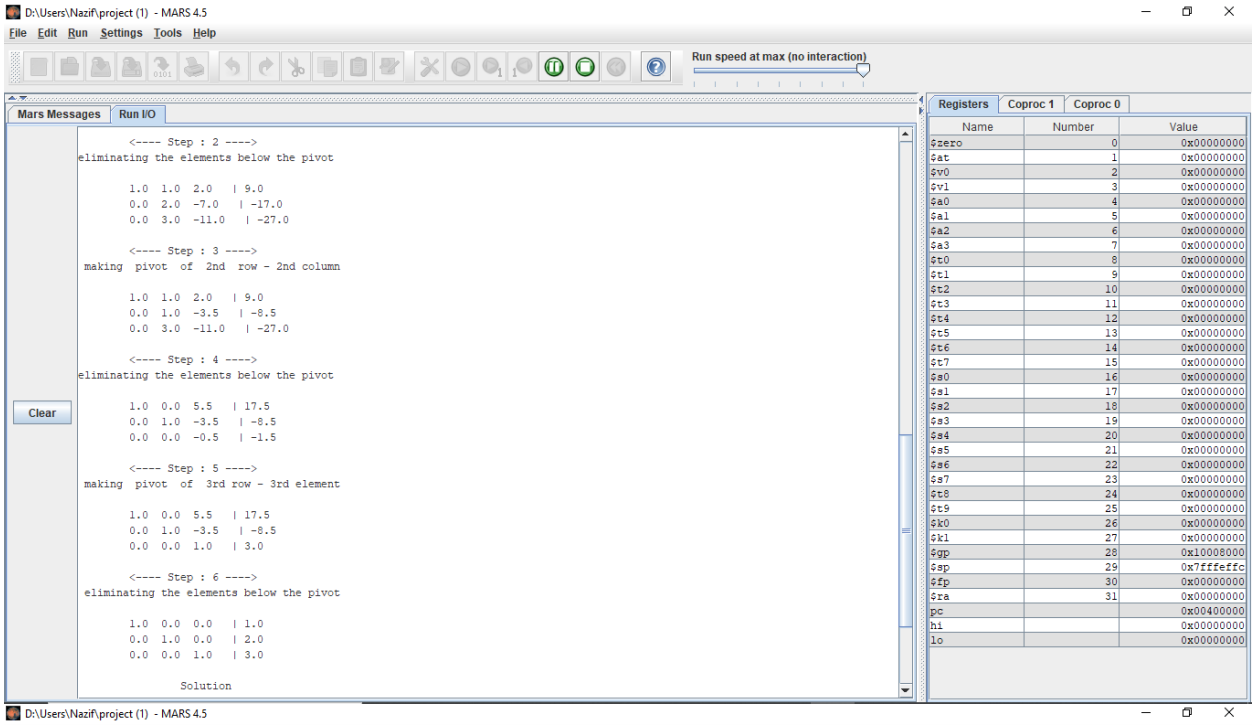
Output

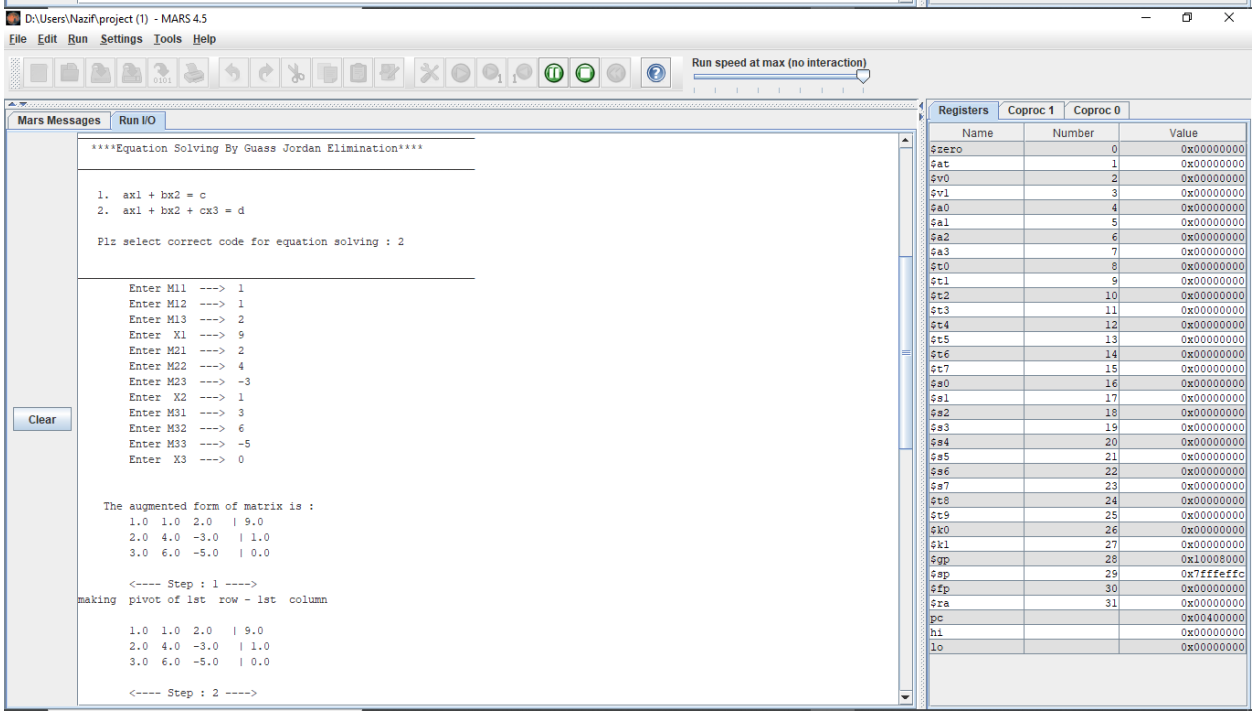
1. 2D Equation

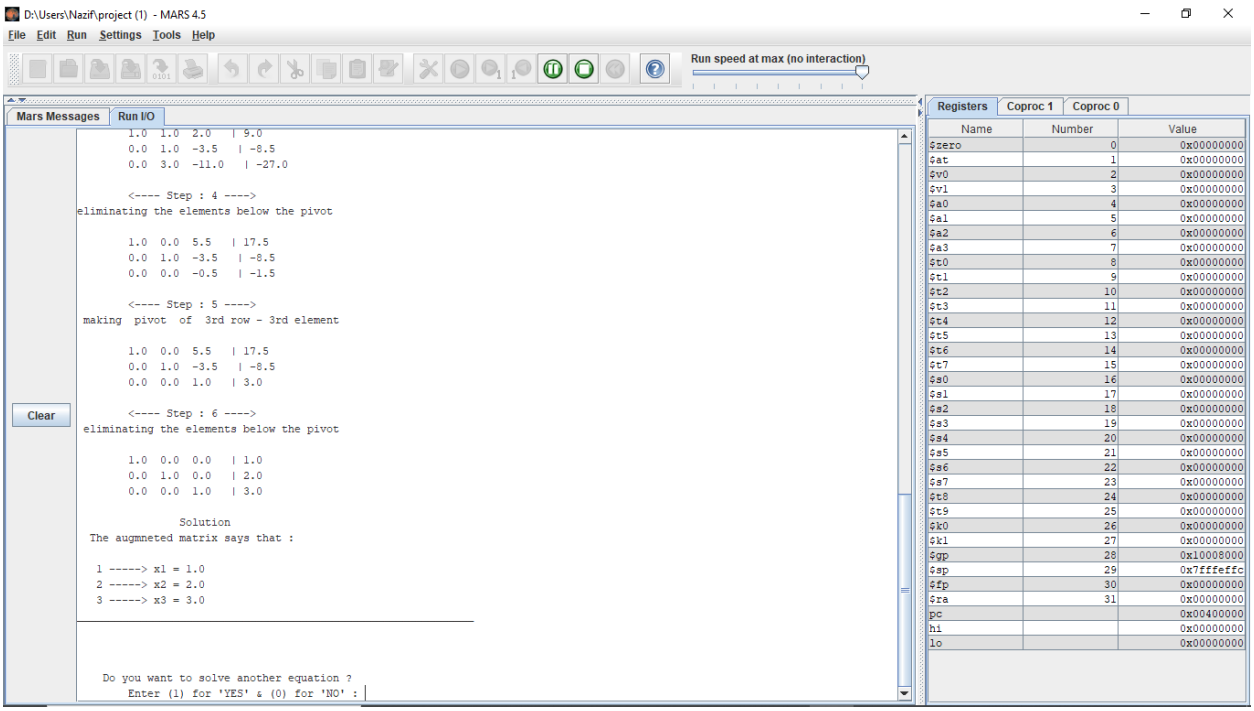




2. 3D Equation






D:\Users\Nazif\project (1) - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Mars Messages
Run I/O
1.0 1.0 2.0 | 9.0
0.0 1.0 -3.5 | -8.5
0.0 3.0 -11.0 | -27.0
<---- Step : 4 ---->
eliminating the elements below the pivot
1.0 0.0 5.5 | 17.5
0.0 1.0 -3.5 | -8.5
0.0 0.0 -0.5 | -1.5
<---- Step : 5 ---->
making pivot of 3rd row - 3rd element
1.0 0.0 5.5 | 17.5
0.0 1.0 -3.5 | -8.5
0.0 0.0 1.0 | 3.0
Clear
<---- Step : 6 ---->
eliminating the elements below the pivot
1.0 0.0 0.0 | 1.0
0.0 1.0 0.0 | 2.0
0.0 0.0 1.0 | 3.0
Solution
The augmneted matrix says that :
1 -----> x1 = 1.0
2 -----> x2 = 2.0
3 -----> x3 = 3.0
Do you want to solve another equation ?
Enter (1) for 'YES' & (0) for 'NO' :
Registers
Coproc 1
Coproc 0
Name Number Value
$zero 0 0x00000000
$at 1 0x00000000
$v0 2 0x00000000
$v1 3 0x00000000
$a0 4 0x00000000
$a1 5 0x00000000
$a2 6 0x00000000
$a3 7 0x00000000
$t0 8 0x00000000
$t1 9 0x00000000
$t2 10 0x00000000
$t3 11 0x00000000
$t4 12 0x00000000
$t5 13 0x00000000
$t6 14 0x00000000
$t7 15 0x00000000
$s0 16 0x00000000
$s1 17 0x00000000
$s2 18 0x00000000
$s3 19 0x00000000
$s4 20 0x00000000
$s5 21 0x00000000
$s6 22 0x00000000
$s7 23 0x00000000
$t8 24 0x00000000
$t9 25 0x00000000
$k0 26 0x00000000
$k1 27 0x00000000
$gp 28 0x10008000
$sp 29 0x7fffeffc
$fp 30 0x00000000
$ra 31 0x00000000
pc 0x00400000
hi 0x00000000
lo 0x00000000