

GMAT

- TEAM 15



TEAM MEMBERS AND ROLES

- NARAYANA SREEHITHA REDDY - TEAM MANAGER
- MALLA SRIHARI - LANGUAGE GURU
- UPPALA SEHOURIEY - LANGUAGE GURU
- AMBATI SANJAY KRISHNA - EXAMPLE CREATOR
- POCHETTI RASHMITHA - EXAMPLE CREATOR
- LIKITHA LANGARU - EXAMPLE CREATOR
- K SATYA SANDESH - EXAMPLE CREATOR

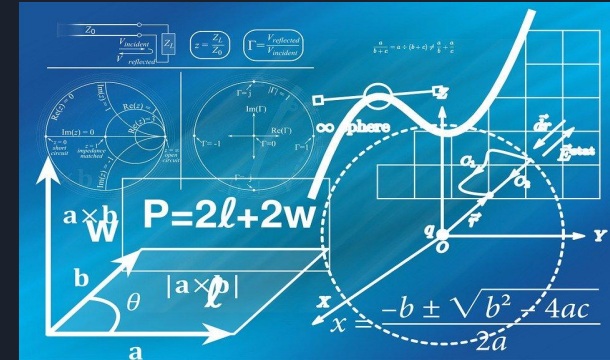
INTRODUCTION


gmat Language Overview:

gmat language is designed specially for 2D geometry problem-solving(extension to general problem solving). This language's main focus is to solve complex geometry problems with ease. Enhanced data types, advanced mathematical operations and functions are specifically designed to suit the needs of the mathematicians. Besides that it supports all the aspects of general purpose programming languages.

Name and Origin of our programming Language:

The name of this programming language is termed as 'gmat' since it is created to solve the complex geometry('g') problems in mathematics('mat'). The origin of developing this specific language is the complexity of the mathematics involved in solving geometric problems.



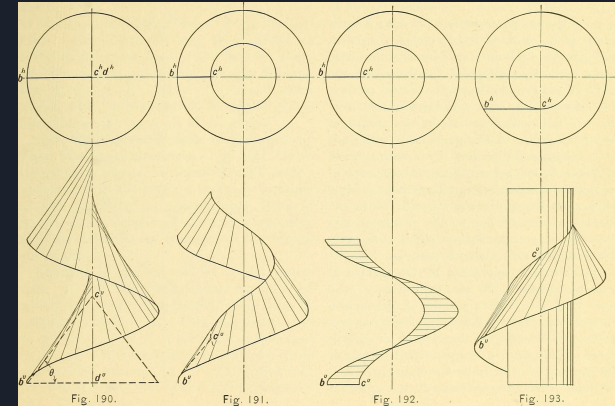


gmat' is an efficient, powerful and yet easy to use programming language from general-purpose to high-level use. The features of this language have been chosen based on the complexity of topics, their importance and frequency of their use in programming.

The General Goal of gmat Language: It is an expressive and versatile language. Its syntax aims to be concise so that users can easily understand how to use it. It is a user-friendly language. Functional programming is introduced which makes it more computational friendly. Its main purpose is to solve algorithms for solving 2D geometry problems including conics with an ease. It is also helpful in solving and finding roots of polynomial equations.

SPECIAL FEATURES

- DYNAMIC BINDING OF '_num' AND '_str' DATATYPES
- SOLVING POLYNOMIAL EQUATIONS
- SOLVING 2D GEOMETRY
- SOLVING CONICS
- PLOTTING THE GRAPHS AND FIGURES





`_num` and `_str`

The `_num` and `_str` data types in GMAT are specially designed for avoiding the overflows.

The `_num` data type can hold any type of real number. It can store very small numbers to even very large numbers, decimal values with any desired precision. The notion of bringing this data type into the language is that the user or programmer need not worry about the data type to be used in case of dealing with some sort of large numbers or decimal values with a precision.

Coming to the `_str` data type, it can hold a single character to multiple characters, what we call a string. As a bunch of single characters makes a string, there is no specific difference between single character and string in GMAT. The notion of this data type is very same that of `_num`, that it is user friendly.

`_num` `var = 10`, `var = 15.5`, `var = 290998422807909986754` are all valid
`_str` `char = 'A'`, `char = "A"`, `string = "This is a string"` are all valid

SOLVING POLYNOMIAL EQUATIONS

gmat can also be used in mathematical and computing fields. It is a tool for solving complex polynomial equations.

Example:

```
add_library :+ io.LIB
add_library :+ geo.LIB
main :: _num
```

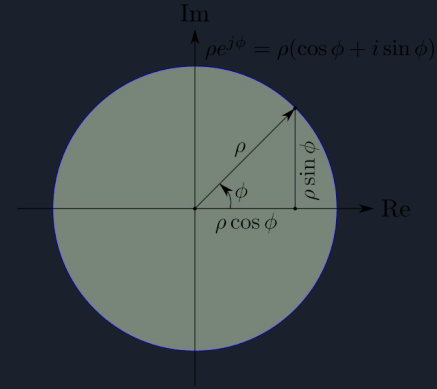
```
_num a,b,c
_poly Eq : 2
write("input coefficients of the quadratic equation:")
read(a,b,c)
Eq = [a,b,c]
_num x,y
(x,y) = solve(Eq)
write("The roots of the given quadratic equation are "x", "y".)
return 0
```

- Solving polynomial equations is done with the help of inbuilt functions in gmat like solve, getcoeff ...etc.
- These functions are implemented with sophisticated algorithms to give precise outputs, lessening the burden on programmer. An exception is raised when roots are complex .

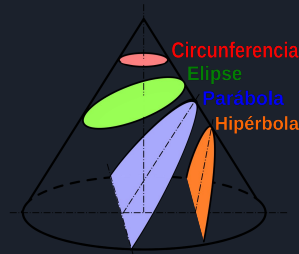
SOLVING 2D GEOMETRY

Example:

```
add_library :+ io.LIB
add_library :+ geo.LIB
main :: _num
    _num x,r,c
    write("enter the number x: ")
    read((0.3)x)
    write("number entered is: ",x,"\n")
    write("enter the radius r: ")
    read(r)
    write("radius entered: ",(0.3)r,"\n")
    c = getcircumference(r)
    write("circumference of the formed circle: ",(0.5)c,"\n")
    return 0
```



- Solving geometric data, shapes and figures like conics, polygons and finding areas under graph is done using data types like `_conic`, `_polygon`.
- `geo.LIB` library contains all the necessary tools for performing above mentioned programming.
- Problems like finding the convex hull, nearest neighbour points...etc related to computational geometry are made inbuilt in `gmat`.

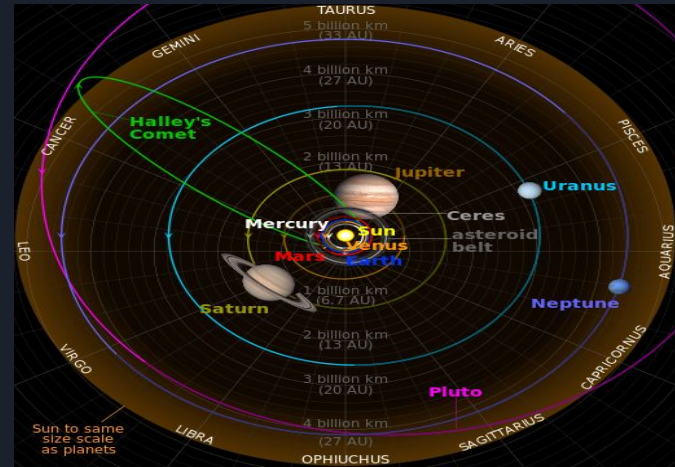


SOLVING CONICS

Example:

```
add_library:+ io.LIB
add_library:+ geo.LIB
main::_num
  _num a,h,b,f,g,c,x,y
  write("Enter the coefficients of the conic in order:")
  read(a,h,b,f,g,c)
  _conic variable
  Variable.inject[a,h,b,f,g,c]
  write(variable.latusrectum,"\n")
  (x,y) = variable.centre
  write("(","x","y")")
  return 0
```

- Solving conic sections are important in astronomical physics
- The reflective properties of conic sections are used in radio-telescopes and searchlights. gmat makes it easy to know and see the variation of properties of conics with their shape.

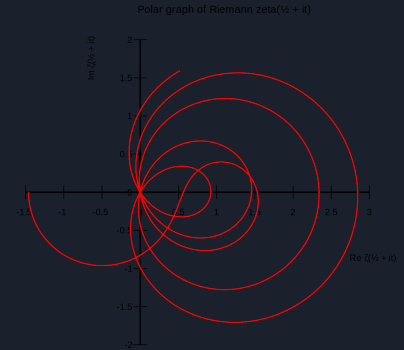


PLOTTING THE GRAPHS AND FIGURES

Plotting gives us good visualization in solving 2D geometry problems. Gmat can plot both data and 2D figures.

Example:

```
add_library :+ io.LIB
add_library :+ plotG.LIB
main::_num
    _polygon T
    _num x1,x2,x3,y1,y2,y3
    write("Enter the coordinates of the triangle:")
    read(x1,y1,x2,y2,x3,y3)
    T=[x1,y1;x2,y2;x3,y3]
    while: Area(T) > 10
        Plotgraph(T)
        write(Area(T))
        T=Medialtriangle(T)
    return 0
```



The idea behind this inclusion of plots in gmat is that, the user can actually experience the trajectories of the conics or the equations they are dealing with. Still some problems in geometry are not solved, because of their complexity in trajectories. User gets visual understanding of what he is dealing with, which in fact produces productivity, and even gets easier to identify bugs if any.



OOP - Object Oriented Programming

As we know that today's problems in real life need a pragmatic way of approach, we introduced this Object Oriented Programming concept in GMAT. It allows our software to split into bite sized problems and then we can deal one object at a time.

Main benefits we can get through this OOP aspect in our GMAT are

1. Modularity for easy troubleshooting.
2. Inheritance allows us to reuse the code.
3. Flexibility through polymorphism
4. Effective problem solving.



CONCLUSION

Thus, gmat language has many special features which can be implemented with ease. It can be useful from general purpose programming to real time projects. The inbuilt libraries and the syntax are user friendly which motivates the user. The plotter is one of the main feature of this language. This language is Object driven, so it supports both the procedural programming and Object oriented programming and functional program.

Future Extensions: At present, gmat language is limited to 2D. But, this idea of notion can be extended to 3D solving along with 3D plotting, which provides user with better understanding, analysis along with greater visualization .