

Кроссплатформенный программный комплекс изображения графиков функций (C++ и Fortran)

Евгений Алексеев, Денис Лутошкин,
Вячеслав Стародумов
Вятский Государственный Университет
ealekseev@gmail.com

Постановка задачи

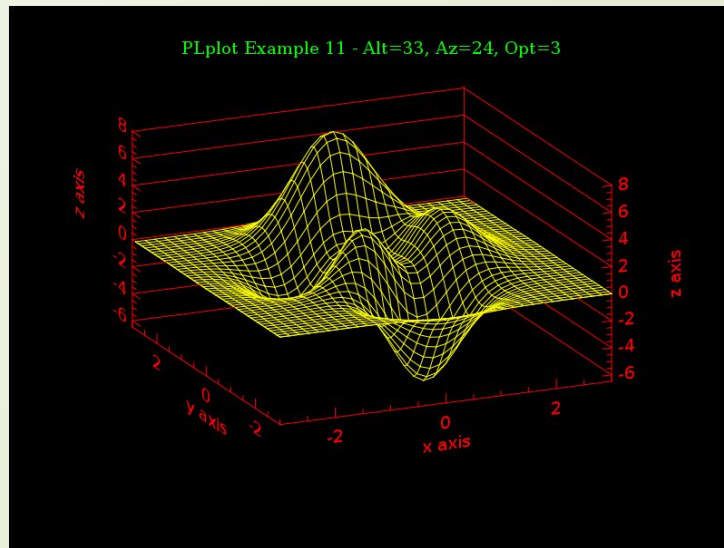
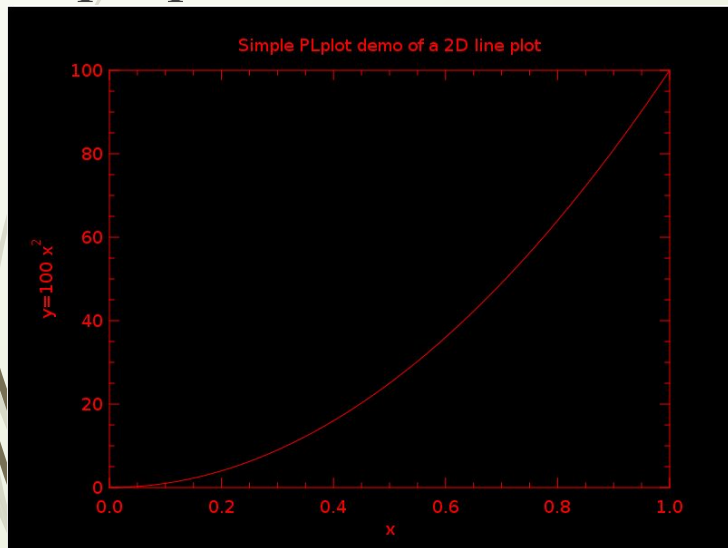
Разработать универсальные программные комплексы построения графиков на языках программирования Fortran и C++

Аналоги графических библиотек и приложений

- PLPlot - <http://plplot.sourceforge.net/>
- PGPlot - <http://www.astro.caltech.edu/~tjp/pgplot/>
- Dislin - <http://www.mps.mpg.de/dislin>
- GNUPlot – <http://www.gnuplot.info/>

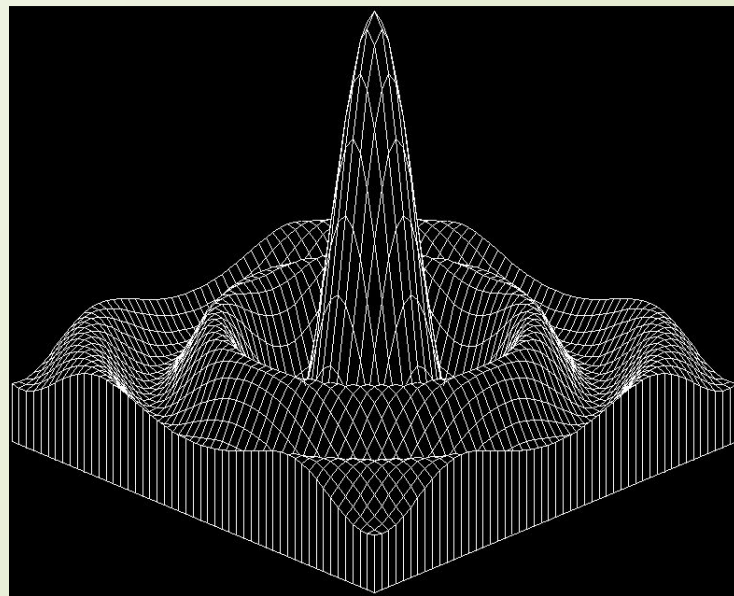
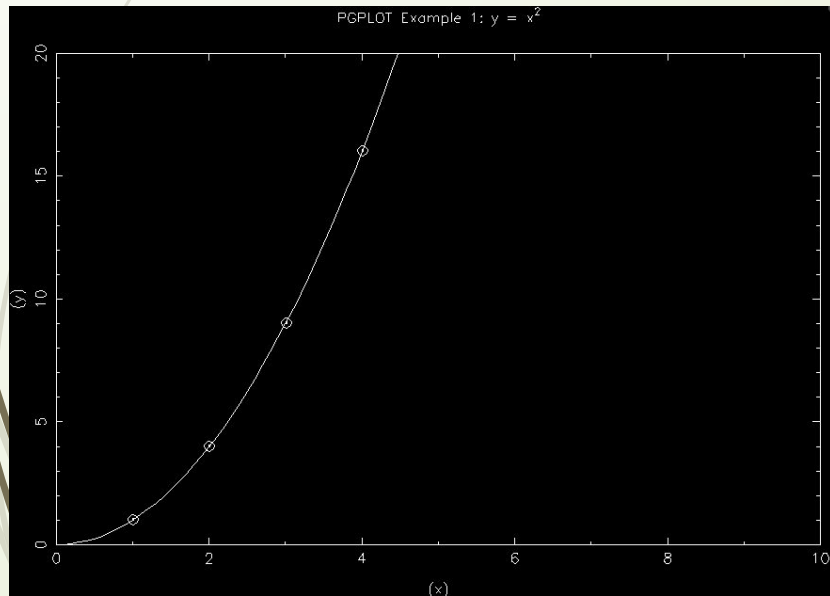
PLPlot

- PLPlot - набор подпрограмм, которые часто используются для создания научных графиков. Это свободное программное обеспечение под лицензией LGPL



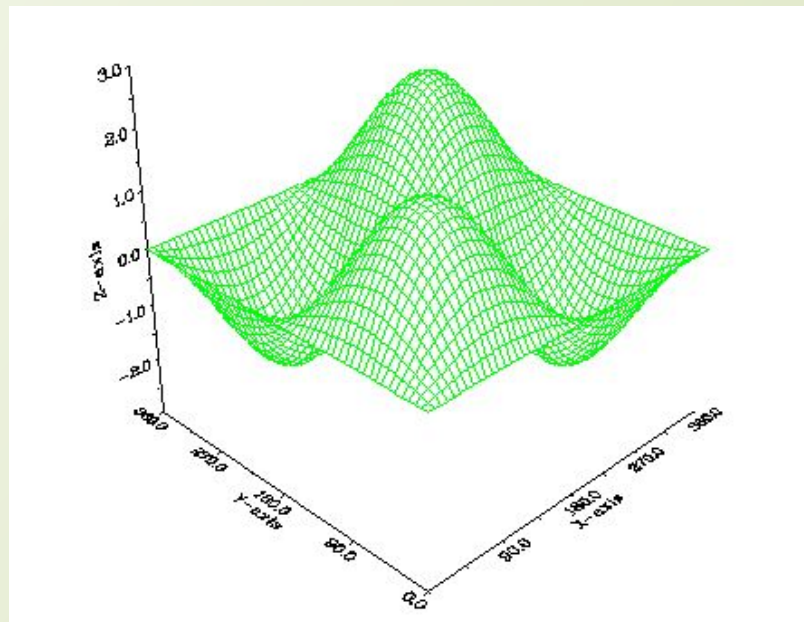
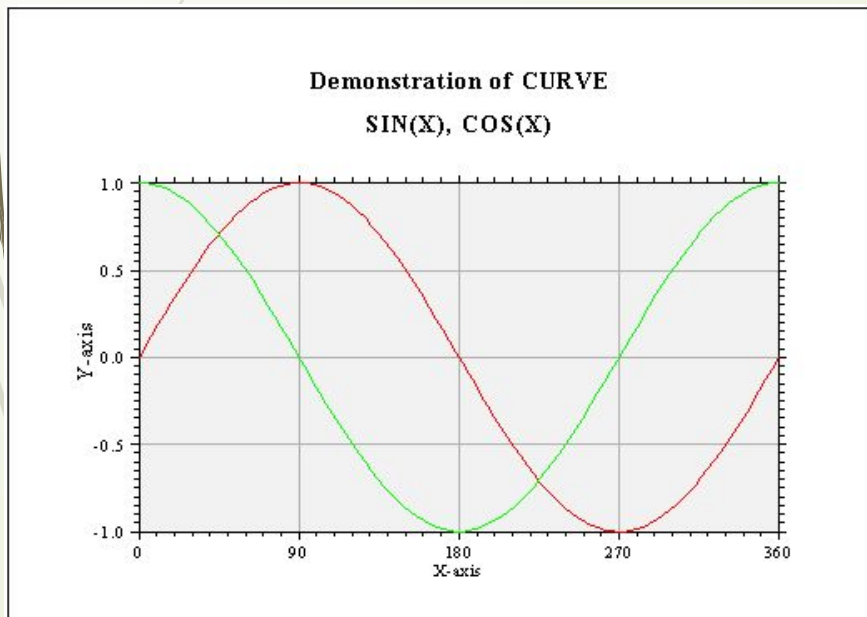
PGPlot

- PGPlot - набор подпрограмм для языка программирования Fortran для изображения простых научных графиков



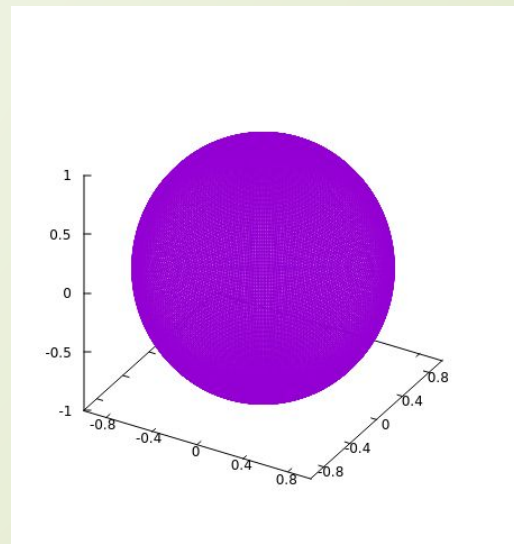
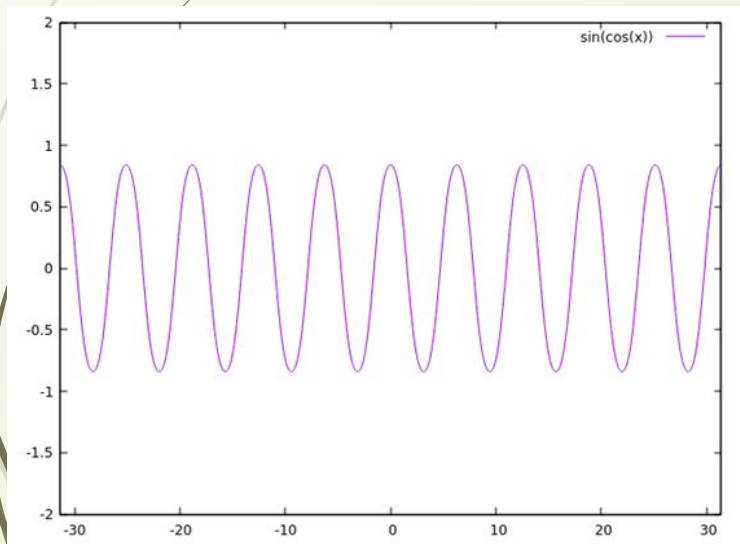
Dislin

- Dislin - высокоуровневая кроссплатформенная библиотека визуализации пользовательских данных.



GNUPlot

- GNUPlot – одна из самых распространённых свободных кроссплатформенных программ для графического вывода результатов.



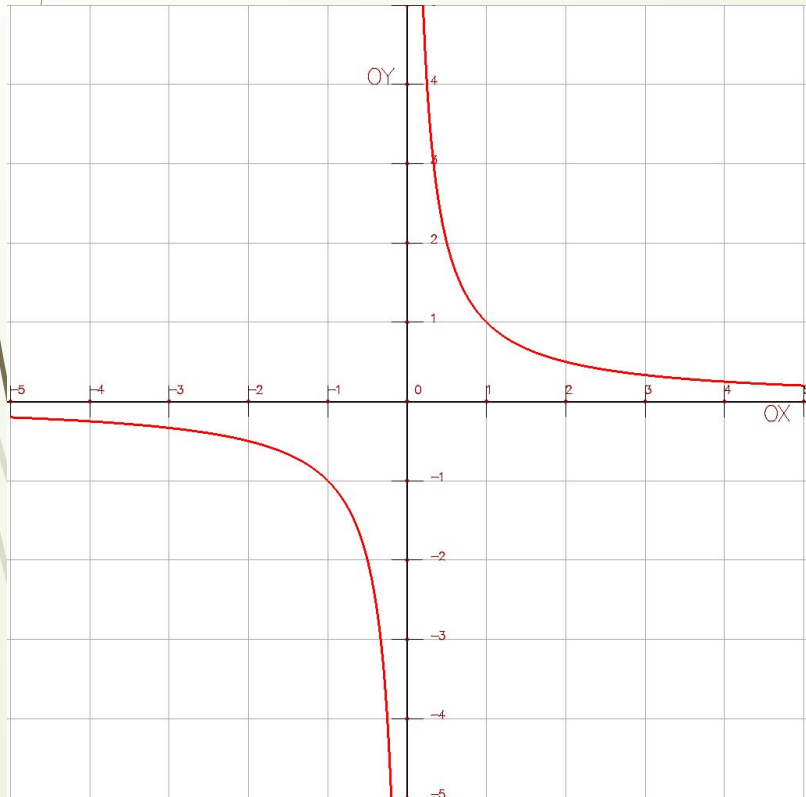
Библиотеки Plotter

- Были разработаны свободные и кроссплатформенные библиотеки Plotter построения двух- и трёхмерных графиков.
- Для использования библиотеки достаточно вызвать главную функцию `Draw()` (C++) и `DrawGraph()` (Fortran).

Возможности разработанных библиотек

- Пользователь может вращать, смещать камеру по осям, масштабировать графики.
- Библиотеки могут строить несколько графиков на одном изображении.
- Перед началом работы пользователь может изменить некоторые настройки отображения графика.
- Учитывается соотношение сторон окна.
- Существует возможность отображения легенды.

Пример построения графика одной функции



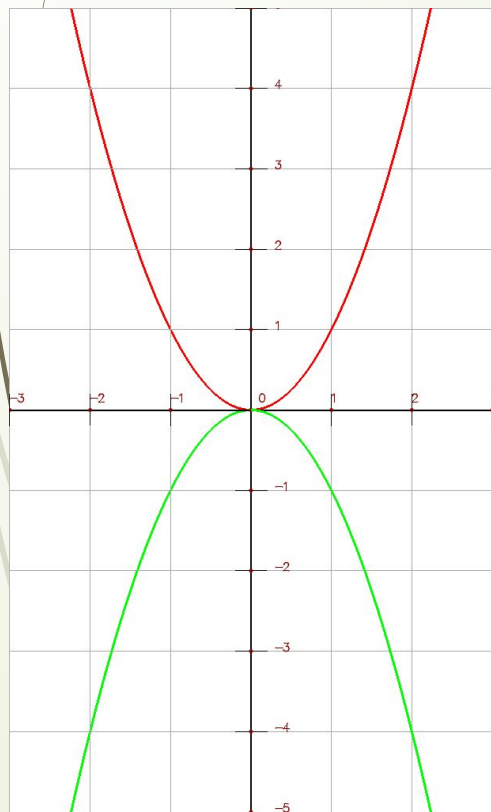
Fortran:

```
DrawGraph(i_window_width,i_window_height,i_diapason_x_start,i_diapason_x_end,i_function)
```

C++:

```
Draw(int i_window_width, int i_window_height,  
double i_diapason_x_start, double i_diapason_x_end,  
double(**i_function)(double))
```

Пример построения графиков нескольких функций



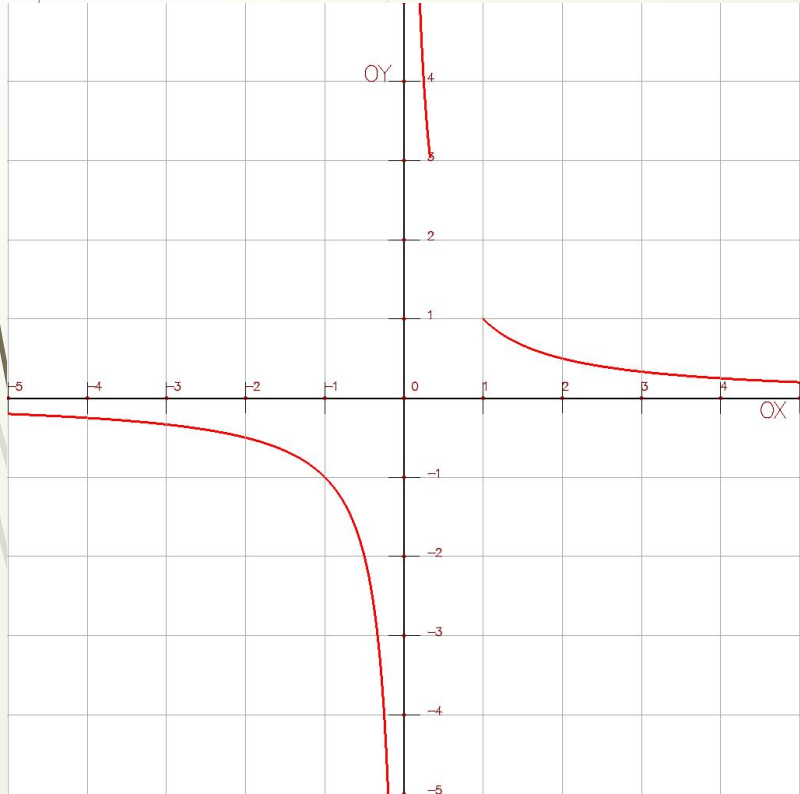
Fortran:

```
DrawGraph(i_window_width,i_window_height,  
i_diapason_x_start,i_diapason_x_end,i_functions,i_functions_number)
```

C++

```
void Draw(int i_window_width, int i_window_height, double  
i_diapason_x_start, double i_diapason_x_end,  
double(**i_functions)(double), int i_functions_number)
```

Пример построения графика функции с пользовательскими интервалами исключения



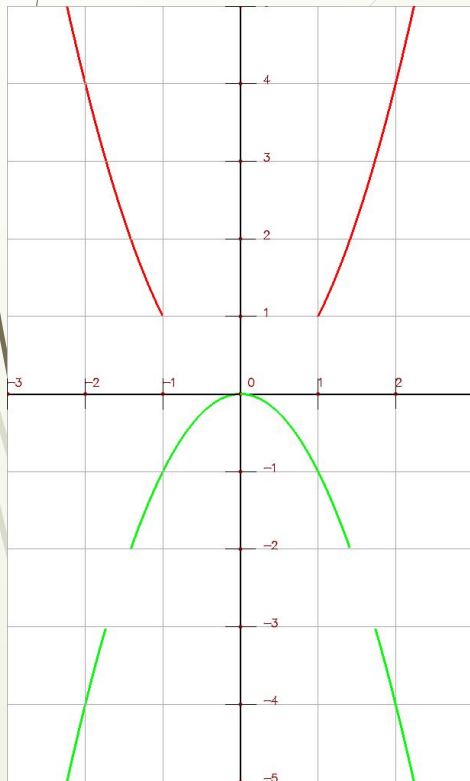
Fortran:

```
DrawGraph(i_window_width,i_window_height,  
i_diapason_x_start, i_diapason_x_end, i_function,  
i_exclude_intervals,i_exclude_pairs_number)
```

C++:

```
void Draw(int i_window_width, int i_window_height,  
double** i_points_xyz, int i_point_number, int  
i_axis_number, double** i_exclude_intervals, int  
i_exclude_pairs_number)
```

Пример построения нескольких графиков функций с пользовательскими интервалами исключения



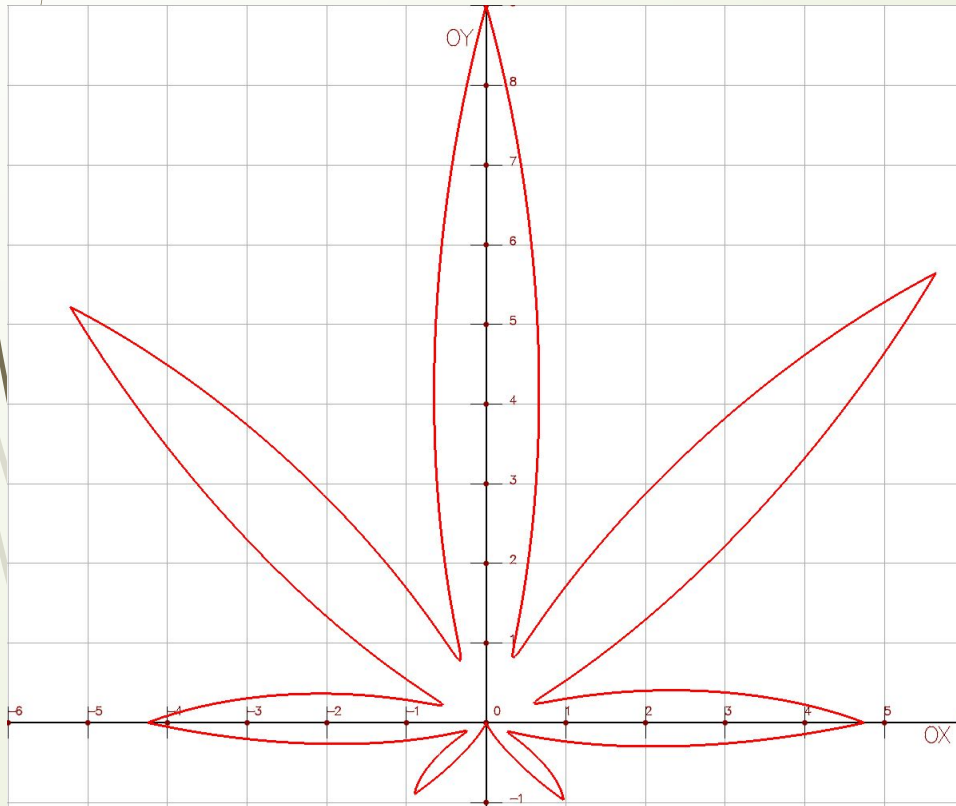
Fortran:

```
DrawGraph(i_window_width, i_window_height, i_diapason_x_start,  
i_diapason_x_end,  
i_functions,i_functions_number,i_exclude_intervals,i_exclude_pairs_num  
ber)
```

C++:

```
void Draw(int i_window_width, int i_window_height, double***  
i_points_xyz, int i_point_number, int i_axis_number, int i_set_number,  
bool**  
i_perfocards)
```

Пример построения графика по точкам



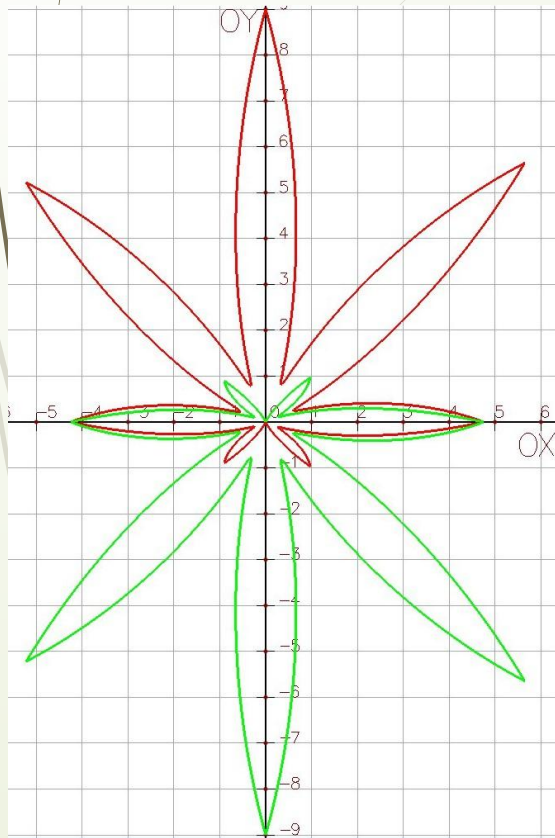
Fortran:

```
DrawGraph(i_window_width,i_window_height,i_points_xyz,i_point_number)
```

C++:

```
void Draw(int i_window_width, int i_window_height, double **i_points_xyz, int i_point_number, int i_axis_number)
```

Пример построения нескольких графиков по точкам



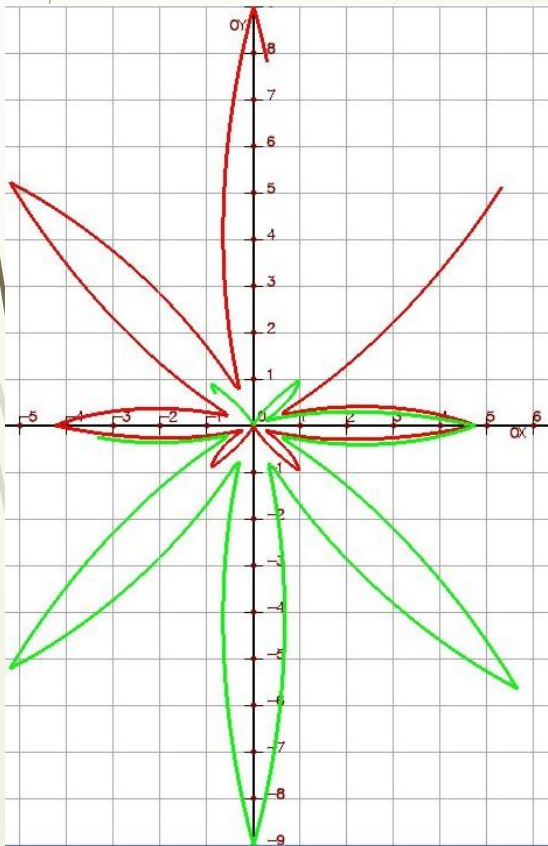
Fortran:

```
DrawGraph(i_window_width,i_window_height,i_points_xyz,  
i_point_number,i_set_number);
```

C++:

```
void Draw(int i_window_width, int i_window_height, double***  
i_points_xyz, int i_point_number, int i_axis_number, int  
i_set_number)
```

Пример построения нескольких графиков по точкам с пользовательскими точками ИСКЛЮЧЕНИЯ



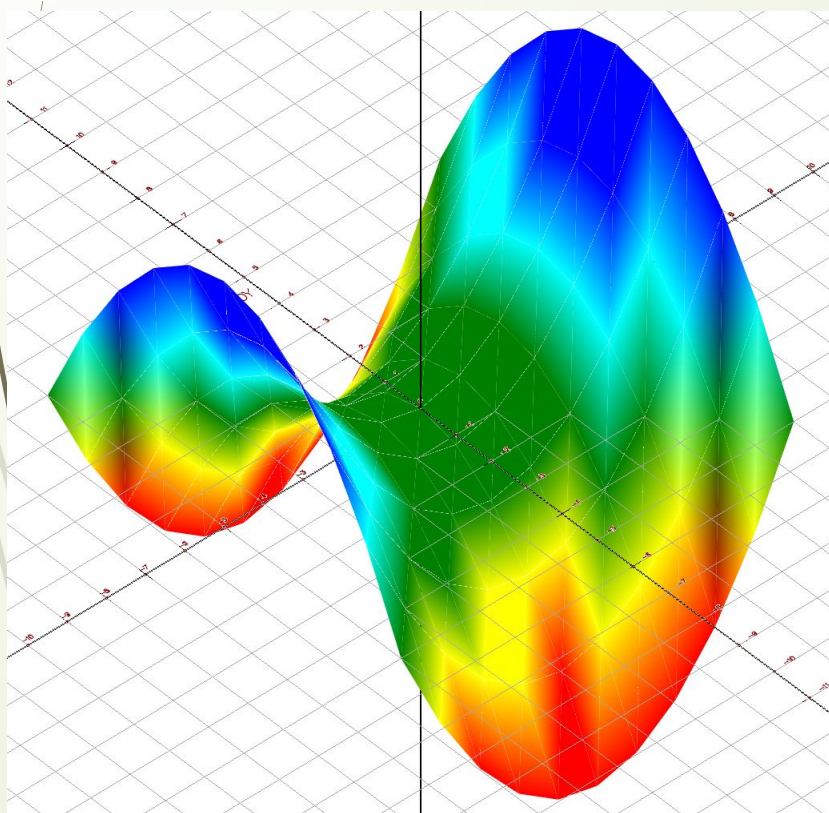
Fortran:

```
DrawGraph(i_window_width,  
i_window_height,i_points_xyz,i_point_number,i_set_number,i_perfo  
cards);
```

C++:

```
void Draw(int i_window_width, int i_window_height, double***  
i_points_xyz, int i_point_number, int i_axis_number, int i_set_number,  
bool** i_perfo cards)
```


Пример построения поверхности



Fortran:


```
DrawGraph(i_window_width,i_window_height,i_diapason_x_start,i_diapason_x_end,i_diapason_y_start,i_diapason_y_end,i_surface);
```

C++:

```
void Draw(int i_window_width, int i_window_height, double i_diapason_x_start, double i_diapason_x_end, double i_diapason_y_start, double i_diapason_y_end, double(**i_surface_function)(double, double))
```

Заключение

- Библиотека Plotter используется в учебной деятельности ВятГУ при обучении бакалавров и магистрантов направления «Прикладная математика и информатика».
- Данный программный комплекс можно использовать в научной деятельности: при решении инженерных задач, задач вычислительной математики, при разработке вычислительных приложений на языках C++ и Fortran.
- Библиотеки можно загрузить по адресу: github.com/LibPlotter



Спасибо за внимание!
Готов ответить на ваши
вопросы.