

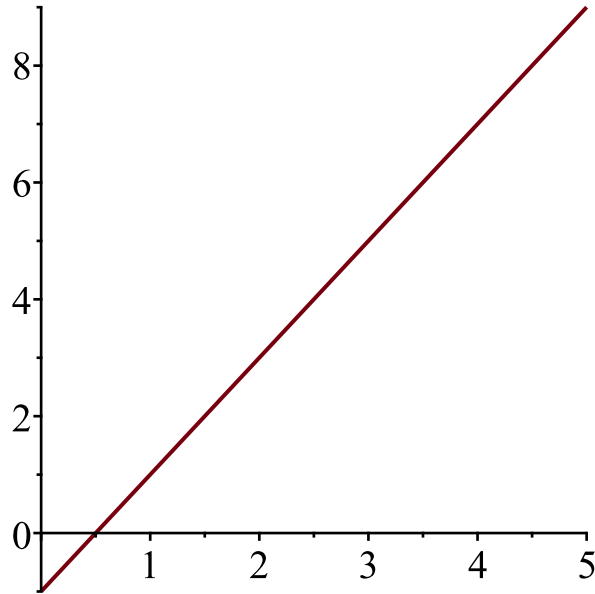
Визуализация наборов точечных данных

Если явное выражение для функциональной зависимости неизвестно, а в наличии имеется только дискретный набор точек $[x_i, y_i]$, то система

Maple позволяет отобразить этот набор с помощью команды **plot**, используя следующий формат команды: `plot(v1, v2)`.

Этот формат неудобен, поскольку абсциссы и ординаты точек заносятся в разные аргументы команды. Аргумент `v1` должен содержать абсциссы точек. Их ординаты должны быть занесены в `v2`. С точки зрения синтаксиса `v1` и `v2` являются списками или же векторами.

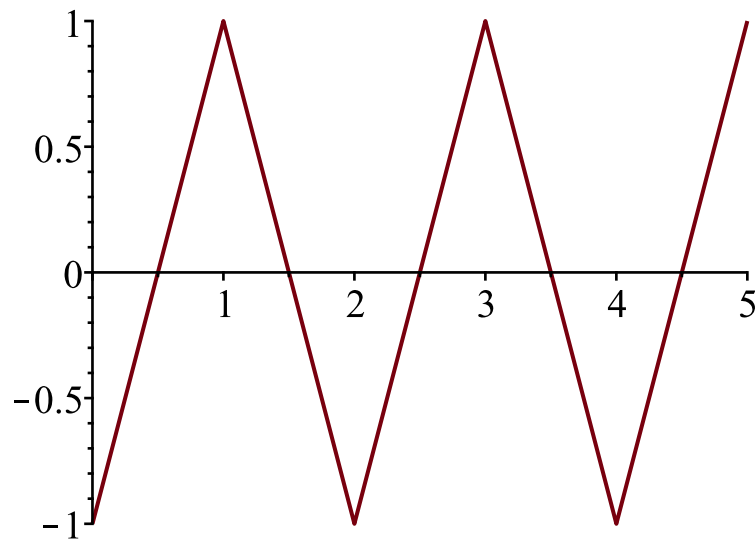
```
> plot([0,1,2,3,4,5], [-1,1,3,5,7,9]);
```



По умолчанию все точки соединяются отрезками. Если эти точки лежат на одной прямой, то возникает иллюзия аналитически заданной функции, а не ломаной.

Суть процесса рисования удобно проиллюстрировать на примере "пилообразной" ломаной линии.

```
> plot([0,1,2,3,4,5], [-1,1,-1,1,-1,1]);
```



Списки с координатами лучше выносить из команды **plot**. В этом случае работать с ними станет удобнее, в том числе и программным образом.

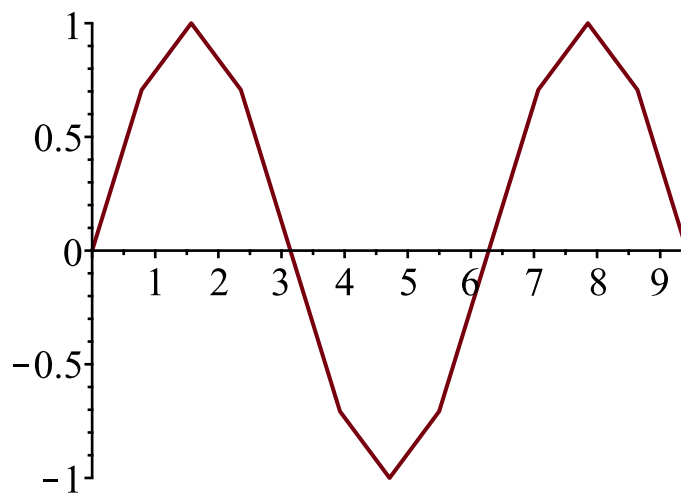
```
> Xp := [seq(Pi/4*j, j=0..12)];
```

$$Xp := \left[0, \frac{1}{4} \pi, \frac{1}{2} \pi, \frac{3}{4} \pi, \pi, \frac{5}{4} \pi, \frac{3}{2} \pi, \frac{7}{4} \pi, 2 \pi, \frac{9}{4} \pi, \frac{5}{2} \pi, \frac{11}{4} \pi, 3 \pi \right]$$

```
> Yp:=map(z->sin(z),Xp);
```

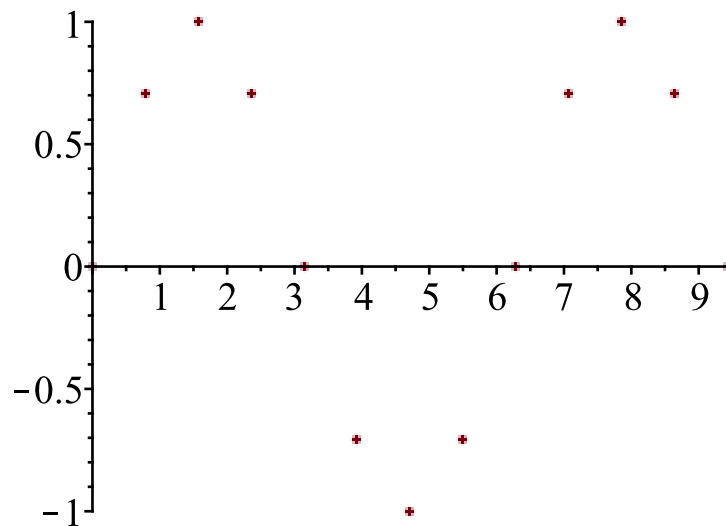
```
Yp := [0, 1/2 * sqrt(2), 1, 1/2 * sqrt(2), 0, -1/2 * sqrt(2), -1, -1/2 * sqrt(2), 0, 1/2 * sqrt(2), 1, 1/2 * sqrt(2), 0]
```

```
> plot(Xp,Yp);
```



Если мы хотим, чтобы точечные данные отображались в виде набора точек, то после обязательных параметров следует добавить опцию *style*.

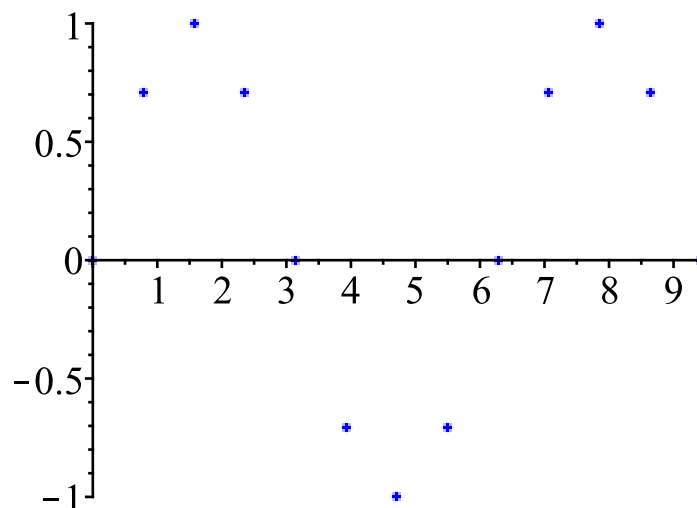
```
> plot(Xp,Yp,style=point);
```



Работают и все другие, рассматривавшиеся ранее, опции команды **plot**.

Например, точки можно окрасить в другой цвет.

```
> plot(Xp,Yp,style=point,color=blue);
```



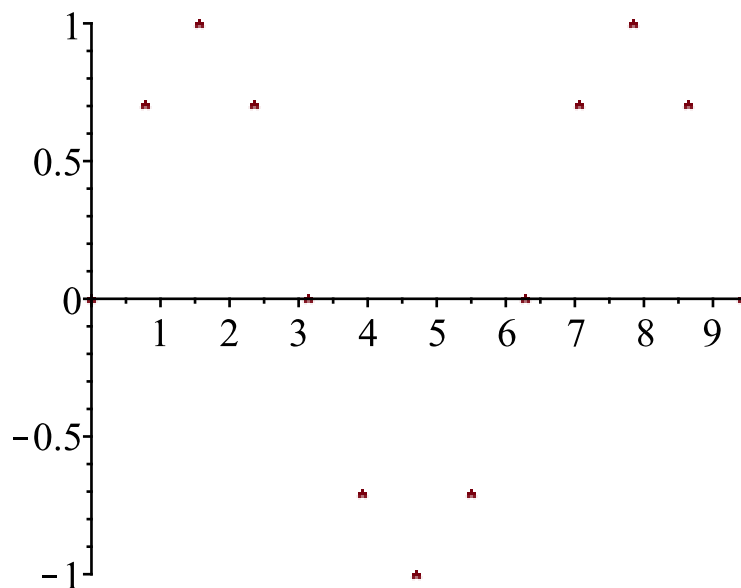
При отображении наборов точечных данных особую роль играет опция *symbol*. С её помощью можно указать графический образ, которым будут выводиться точки на плоскости. Допустимые значения выбираются из приводимой ниже таблицы:

Символы для отображения точек

<i>asterisk</i>	<i>diamond</i>
<i>box</i>	<i>point</i>
<i>circle</i>	<i>solidbox</i>
<i>cross</i>	<i>solidcircle</i>
<i>diagonalcross</i>	<i>soliddiamond</i>

По умолчанию используется опция *diamond*.

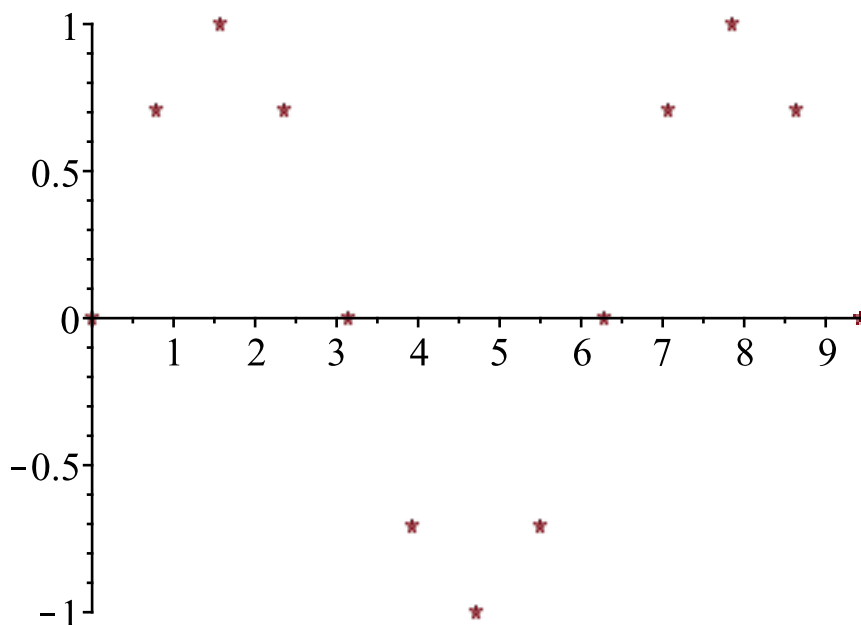
```
> plot(Xp, Yp, style=point, symbol=asterisk);
```



Размер графических образов можно задавать при помощи опции *symbolsize*. Он выражается в пунктах и по умолчанию равен 10.

Увеличим "звёздочки" из предыдущего примера.

```
> plot(Xp, Yp, style=point, symbol=asterisk, symbolsize=14);
```



С помощью наборов точек можно построить объекты, которые,

вообще говоря, не являются графиками явно заданных функций.

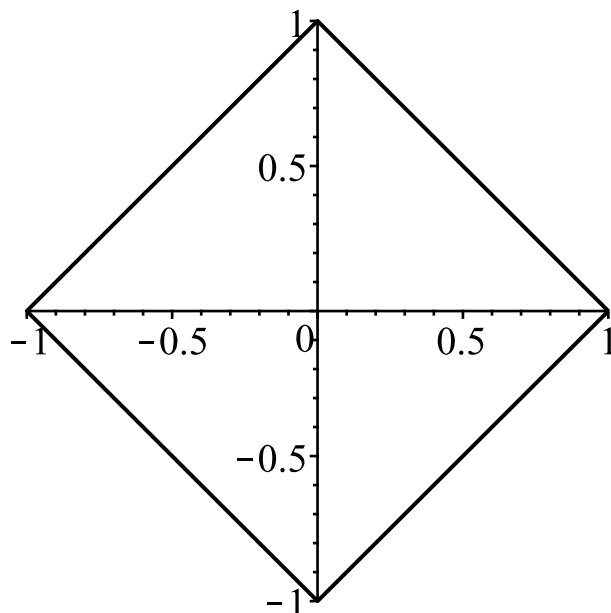
```
> SX:=[1,0,-1,0,1];  
SY:=[0,1,0,-1,0];
```

```
SX := [1, 0, -1, 0, 1]
```

```
SY := [0, 1, 0, -1, 0]
```

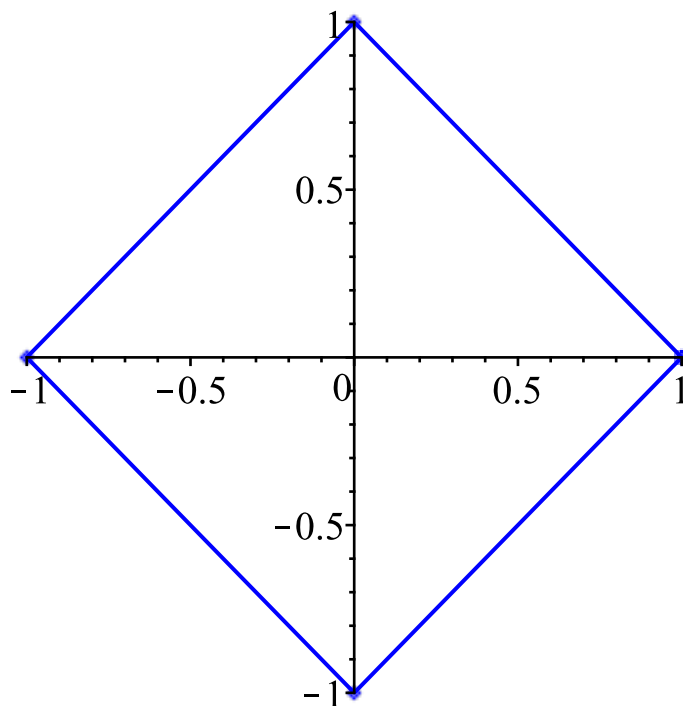
Если необходимо построить квадрат, то достаточно указать координаты его вершин. Для построения замкнутой ломаной координаты первой вершины следует продублировать в конце списка.

```
> plot(SX,SY,color=black);
```



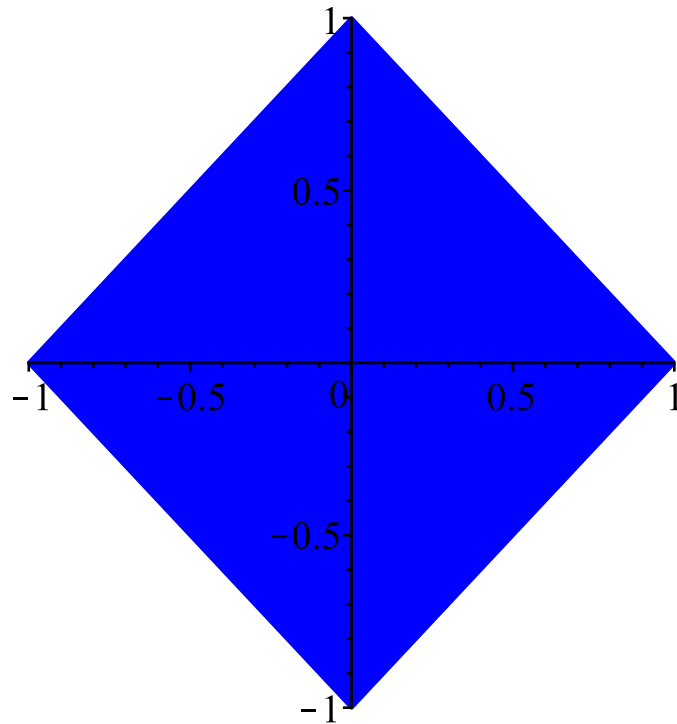
Когда необходимо отобразить и сами точки, и соединяющие их линии проще всего воспользоваться опцией *style*. Значение *pointline* является комбинацией стилей *point* и *line*.

```
> plot(SX,SY,style=pointline,color=blue,symbolsize=12);
```



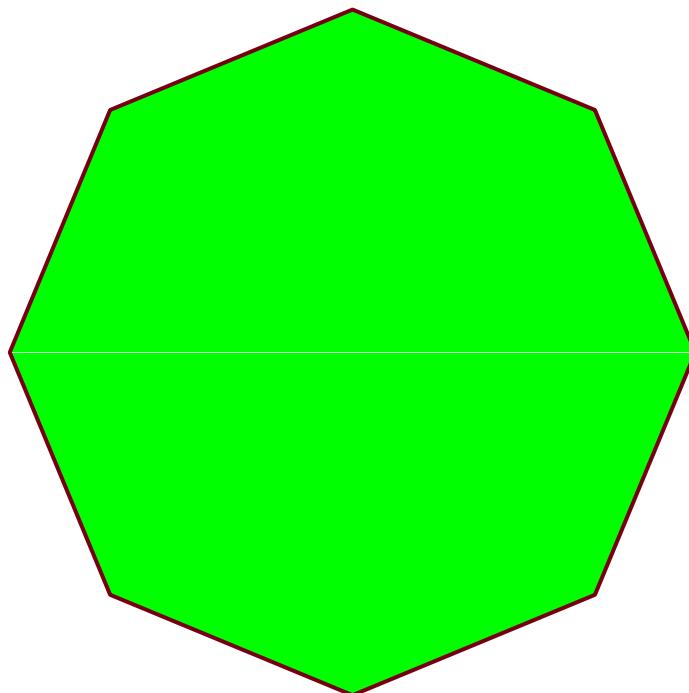
Если же вам очень хочется закрасить фигуру, то в некоторых случаях помогает добавление опции *filled*.

```
> plot(SX,SY,filled=true,color=blue);
```



Достоинством системы является возможность задавать координаты точек программным путём. Вот так, например, можно нарисовать правильный n -угольник.

```
> n:=8;
  Xpol:= [seq(cos(2*Pi/n*j), j=0..n)];
  Ypol:= [seq(sin(2*Pi/n*j), j=0..n)];
                                     n := 8
  Xpol := [1, 1/2 * sqrt(2), 0, -1/2 * sqrt(2), -1, -1/2 * sqrt(2), 0, 1/2 * sqrt(2), 1]
  Ypol := [0, 1/2 * sqrt(2), 1, 1/2 * sqrt(2), 0, -1/2 * sqrt(2), -1, -1/2 * sqrt(2), 0]
> plot(Xpol, Ypol, scaling=constrained, axes=none,
       filled=[color=green]);
```



В соответствии с синтаксисом процедуры **plot**, координаты точек могут задаваться не только списками, но и векторами. Maple поддерживает два родственных типа данных: *vector* и *Vector*.

Выясним, какой из них поддерживается процедурой **plot**.

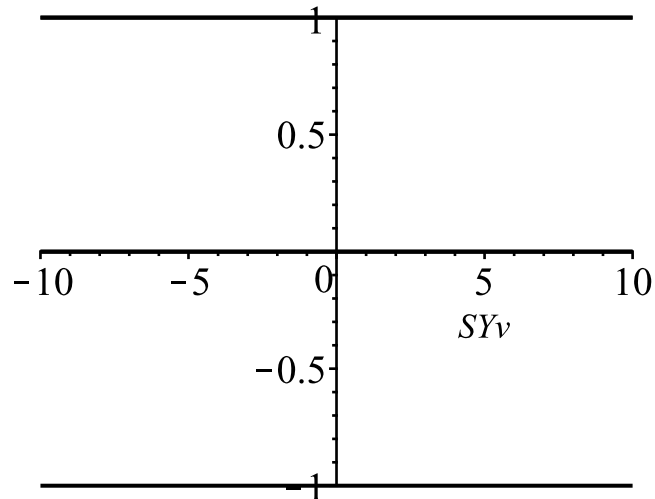
```
> SXv:=convert(SX,vector);  
SYv:=convert(SY,vector);
```

```
SXv := [ 1 0 -1 0 1 ]
```

```
SYv := [ 0 1 0 -1 0 ]
```

Попробуем ещё раз нарисовать квадрат

```
> plot(SXv,SYv,color=black);
```



Ошибки не возникло, но результат неожиданный!

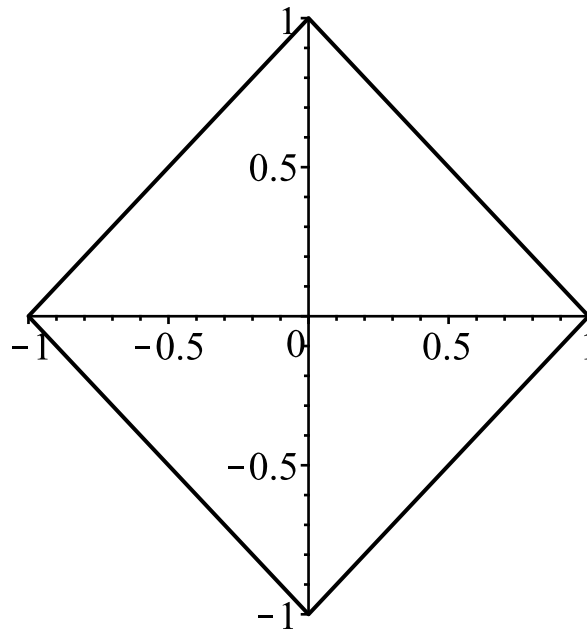
Повторим эту операцию с другим типом данных.

```
> SXV:=convert(SX,Vector);  
SYV:=convert(SY,Vector);
```

```
SXV := [ 1  
0  
-1  
0  
1 ]
```

```
SYV := [ 0  
1  
0  
-1  
0 ]
```

```
> plot(SXV,SYV,color=black);
```



Вывод: *не путайте типы данных!*

Объекты типа *vector* основаны на структурах данных *table*.

В то же время в основе типа *Vector* лежит структура *rtable*.

На практике координаты x_i и y_i удобно группировать. Команда

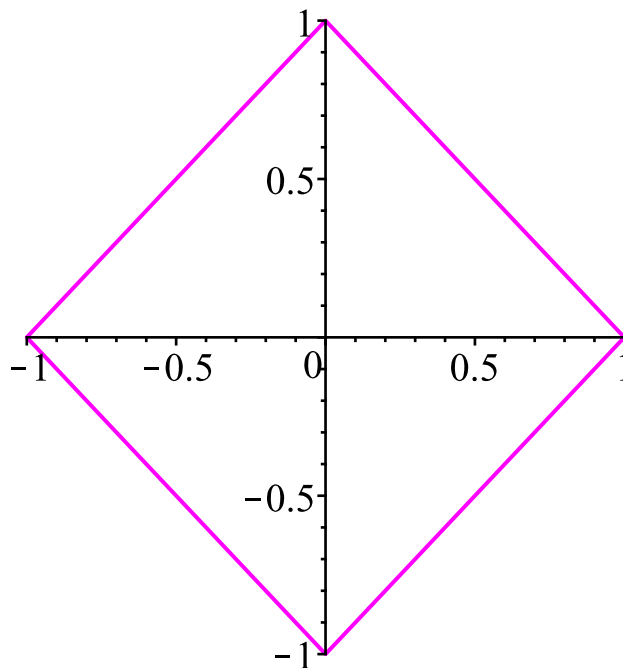
plot позволяет задавать координаты точек в виде вложенного списка вида $[[x_1, y_1], [x_2, y_2], [x_3, y_3] \dots [x_n, y_n]]$

Вот как можно изобразить предыдущий квадрат.

```
> vlist:=[[1,0],[0,1],[-1,0],[0,-1],[1,0]];
```

```
      vlist := [[1, 0], [0, 1], [-1, 0], [0, -1], [1, 0]]
```

```
> plot(vlist,color=magenta);
```

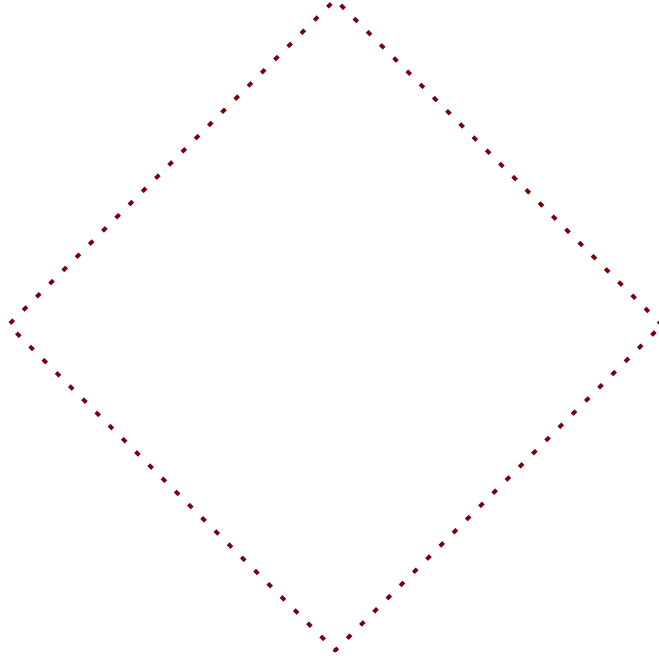


Координаты точек иногда задаются в виде матрицы, то есть объекта типа *Matrix*. Эта матрица должна содержать ровно 2 столбца. В первом столбце следует разместить значения x_i , а во втором $-y_i$.

```
> M2col:=Matrix(nops(vlist),2,vlist);
```

$$M2col := \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \\ 1 & 0 \end{bmatrix}$$

```
> plot(M2col, linestyle=dot, axes=None);
```



На координаты точек каких-либо существенных ограничений не налагается, поэтому отрезки ломаной, соединяющей различные точки, могут пересекаться.

```
> Vlist := [[-1, 0], [1, 0], [1, 1], [-1, 1], [-1, 0], [1, 1], [0, 3/2], [-1, 1], [1, 0]];
```

$$Vlist := \left[[-1, 0], [1, 0], [1, 1], [-1, 1], [-1, 0], [1, 1], \left[0, \frac{3}{2} \right], [-1, 1], [1, 0] \right]$$

```
> plot(Vlist, color=gray, axes=None, scaling=constrained);
```

