

```
#
# Best Fitting a curve of the form  $y = Cx$  to data points
#  $(x_1, y_1), \dots, (x_n, y_n)$ 
#
```

```
bestFitLineThroughOrigin := proc(pts)
local n, dn, nm, i, C :
n := nops(pts) :
dn := 0 :
nm := 0 :
for i from 1 to n do
nm := nm + pts[i][2]·pts[i][1] :
dn := dn + (pts[i][1])2 :
od:
C :=  $\frac{nm}{dn}$  :
x → C·x;
end
```

```
proc(pts) (1)
```

```
    local n, dn, nm, i, C;
    n := nops(pts);
    dn := 0;
    nm := 0;
    for i to n do nm := nm + pts[i][2]*pts[i][1]; dn := dn + pts[i][1]^2 end do;
    C := nm / dn;
    x → C * x
```

```
end proc
```

```
pts := [[1, 3], [2, -1], [4, 3], [7, 12]];
[[1, 3], [2, -1], [4, 3], [7, 12]] (2)
```

```
with(plots);
```

```
[animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d,
conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, densityplot,
display, dualaxisplot, fieldplot, fieldplot3d, gradplot, gradplot3d, implicitplot, implicitplot3d,
inequal, interactive, interactiveparams, intersectplot, listcontplot, listcontplot3d,
listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, multiple, odeplot, pareto,
plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d,
polyhedra_supported, polyhedraplot, rootlocus, semilogplot, setcolors, setoptions,
setoptions3d, shadebetween, spacecurve, sparsematrixplot, surfdata, textplot, textplot3d,
tubeplot] (3)
```

```
plot1 := plot(pts, symbol = circle, style = point) :
f := bestFitLineThroughOrigin(pts);
```

$$x \rightarrow C x$$

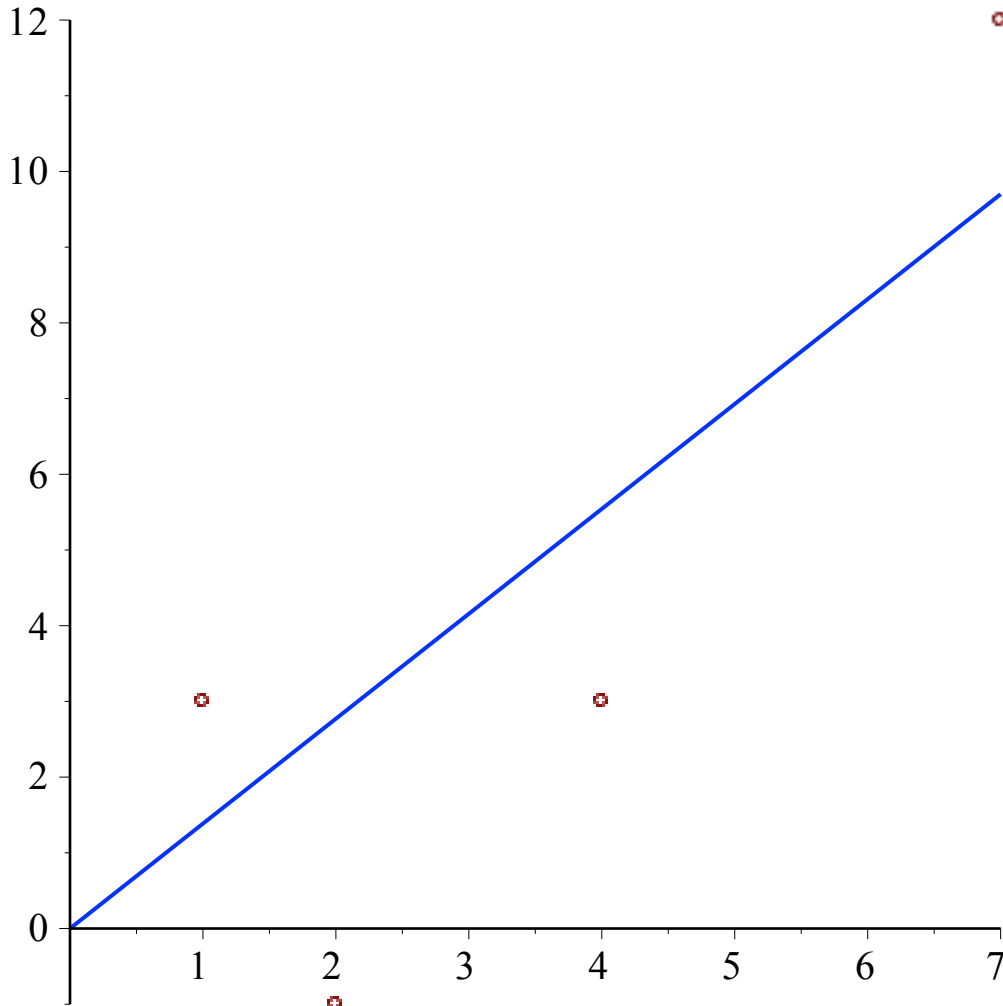
(4)

$f(x);$

$$\frac{97}{70} x$$

(5)

```
plot2 := plot(f, 0..7, colour = blue) :
display([plot1, plot2]);
```



```
sumsqdev := proc(pts, f)
local n, S, i :
n := nops(pts) :
S := 0 :
for i from 1 to n do
  S := S + (f(pts[i][1]) - pts[i][2])2 :
od :
evalf(S);
end;
```

proc(pts, f)

(6)

local n, S, i;

n := nops(pts);

```

S := 0;
for i to n do S := S + (f(pts[i][1]) - pts[i][2])^2 end do;
evalf(S)
end proc

```

```
sumsqdev(pts,f);
```

28.58571429 (7)

```
evalf(f(x));
```

1.385714286 x (8)

```
sumsqdev(pts, x → 1.2 · x);
```

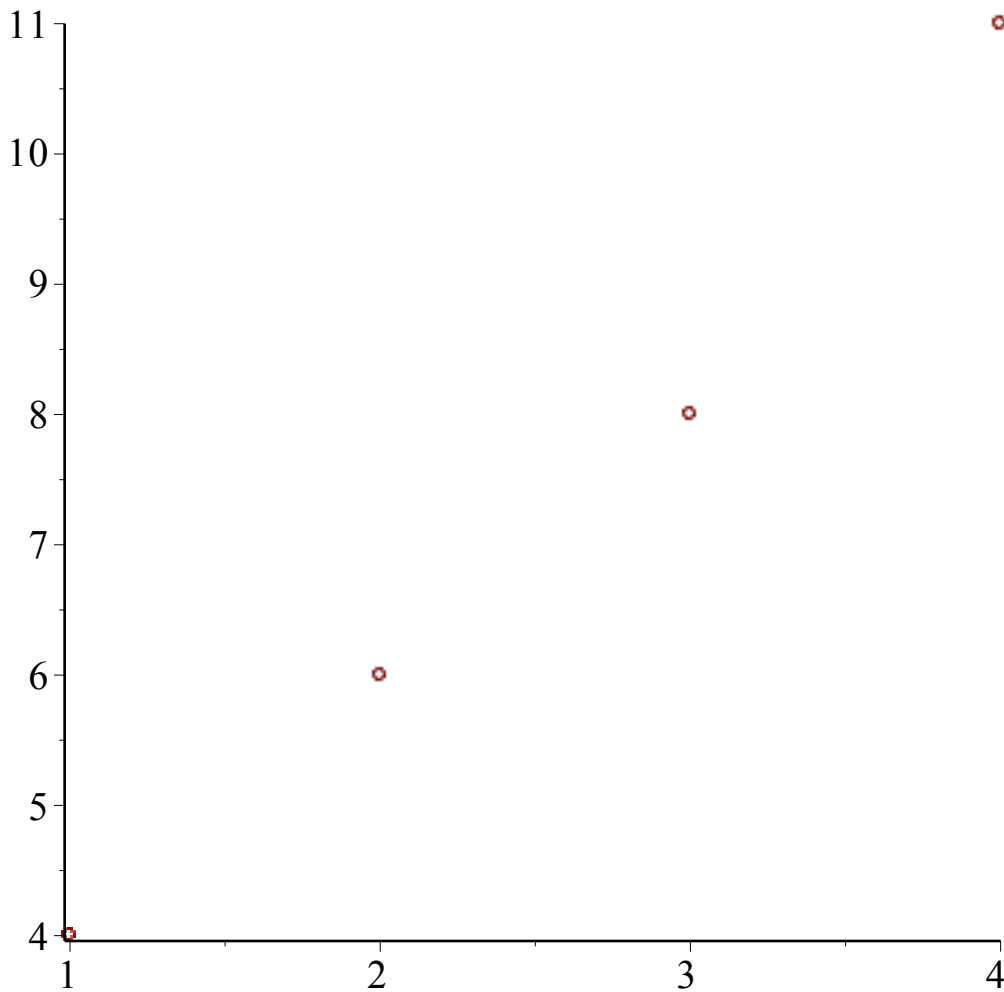
31.00 (9)

```
pts := [[1, 4], [2, 6], [3, 8], [4, 11]];
```

```
plot1 := plot(pts, symbol = circle, style = point) :
```

```
display(plot1);
```

[[1, 4], [2, 6], [3, 8], [4, 11]]

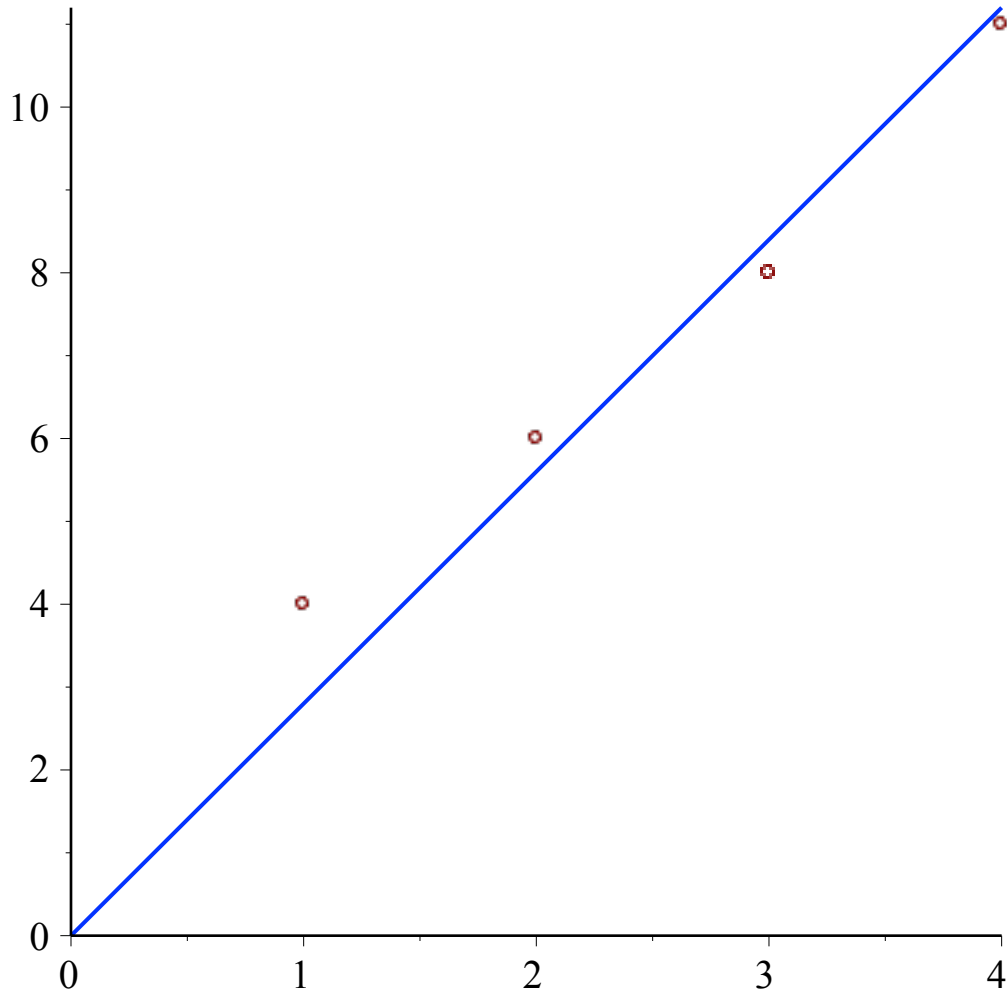


```
f := bestFitLineThroughOrigin(pts);
```

$x \rightarrow Cx$ (10)

$$f(x); \quad \frac{14}{5}x \quad (11)$$

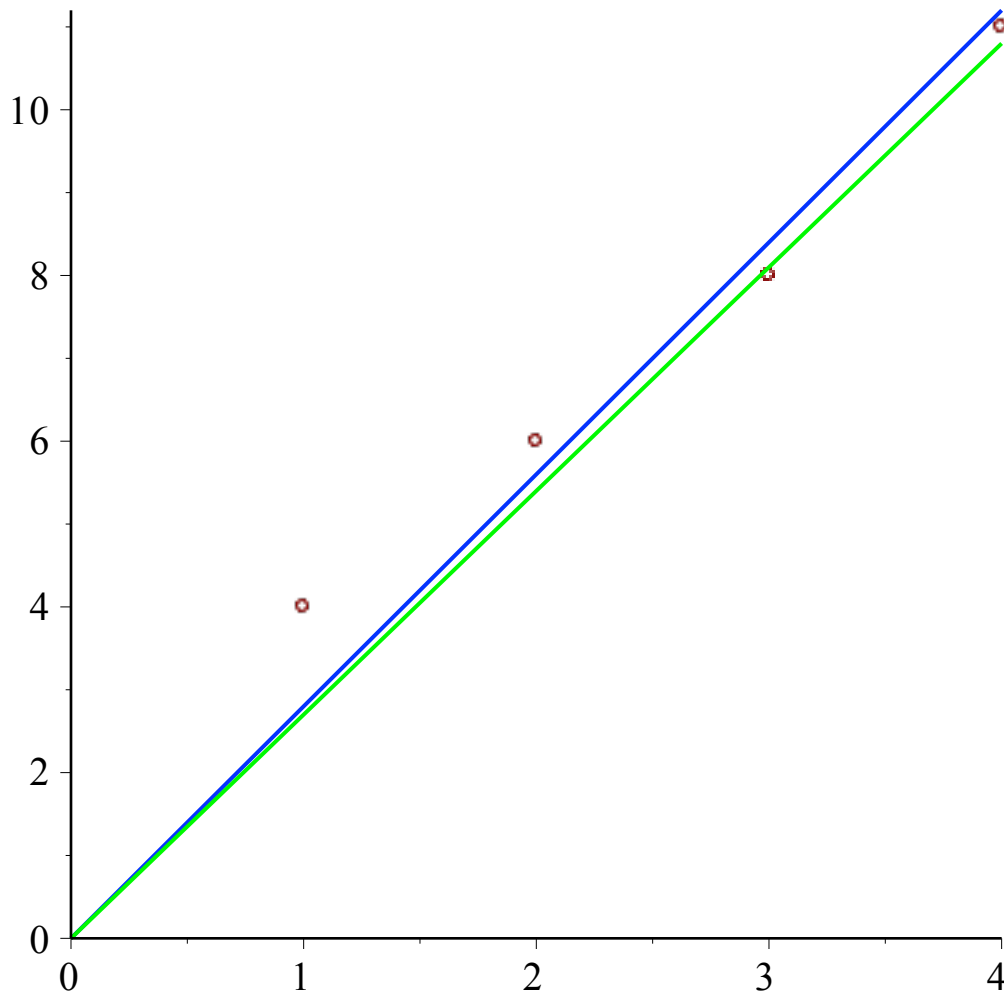
```
plot2 := plot(f, 0..4, colour = blue) :
display([plot1, plot2]);
```



$$\text{sumsqdev}(pts, f); \quad 1.800000000 \quad (12)$$

$$\text{sumsqdev}(pts, x \rightarrow 2.81 \cdot x); \quad 1.8030 \quad (13)$$

```
plot3 := plot(x → 2.7 · x, 0..4, colour = green) :
display([plot1, plot2, plot3]);
```



```
pts := [[0,0], [1, 1.2], [2, 3.8], [3, 8], [4, 16.5]];
      [[0,0], [1, 1.2], [2, 3.8], [3, 8], [4, 16.5]]
```

(14)

```
plot1 := plot(pts, symbol = circle, style = point) :
```

```
bestFitCxSquared := proc(pts)
```

```
  local n, dn, nm, i, C :
```

```
  n := nops(pts) :
```

```
  dn := 0 :
```

```
  nm := 0 :
```

```
  for i from 1 to n do
```

```
    nm := nm + pts[i][2] · (pts[i][1])2 :
```

```
    dn := dn + (pts[i][1])4 :
```

```
  od :
```

```
  C :=  $\frac{nm}{dn}$  :
```

```
  x → C · x2 ;
```

```
  end
```

```
proc(pts)
```

```
  local n, dn, nm, i, C ;
```

(15)

```

n := nops(pts);
dn := 0;
nm := 0;
for i to n do nm := nm + pts[i][2]*pts[i][1]^2; dn := dn + pts[i][1]^4 end do;
C := nm / dn;
x → C * x^2
end proc
f := bestFitCxSquared(pts);

```

$$x \rightarrow C x^2 \quad (16)$$

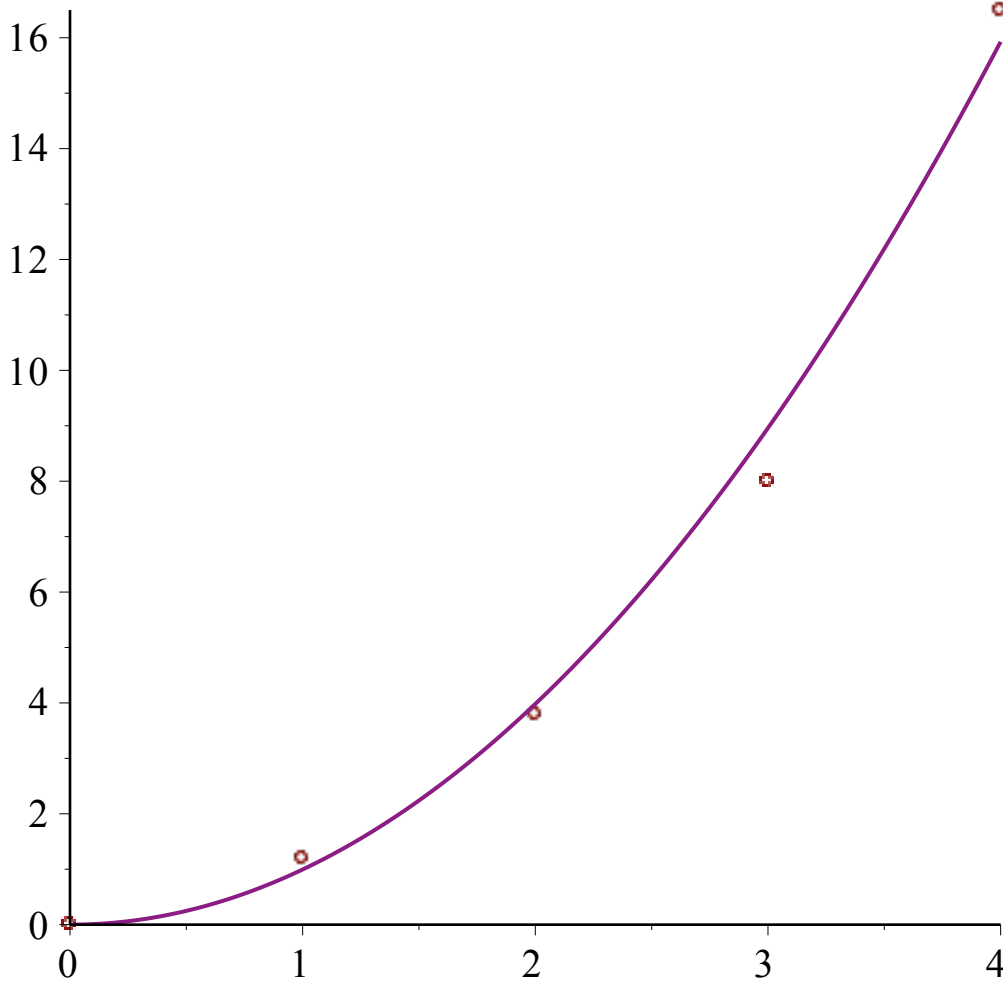
```
evalf(f(x));
```

$$0.9954802260 x^2 \quad (17)$$

```

plot2 := plot(f, 0..4, colour = purple) :
display([plot1, plot2]);

```



```
sumsqdev(pts, f);
```

$$1.322768357 \quad (18)$$

```
sumsqdev(pts, x → 0.99 · x^2);
```

