1. On peut trouver un projeté orthogonal en résolvant un système ... ou en calculant un produit scalaire (ce qui est tout de même bien plus simple !)
2. $\dfrac{R_M R_{M'}}{MM'} = \dfrac{R_M R_{M'}}{Q_M Q_{M'}} \dfrac{Q_M Q_{M'}}{P_M P_{M'}} \dfrac{P_M P_{M'}}{MM'} = |\cos b||\cos a||\cos c| = k < 1.$
   $\forall n,\ |x_{n+1} - x_n| = |\varphi(x_n) - \varphi(x_{n-1})| \leq k|x_n - x_{n-1}| \leq ... \leq k^n|x_1 - x_0|$ et $0 \leq k < 1$ donc la série $\sum(x_{n+1} - x_n)$ est absolument convergente, d'où la convergence de la suite (x_n).
   φ est continue (car affine) donc $\lim x_n$ en est un point fixe (et une fonction affine a au plus un point fixe).

O19-C03

```
> restart;
> with(LinearAlgebra):
> proj:=proc(i,j,m)
  i+(DotProduct(m-i,j-i)/DotProduct(j-i,j-i))*(j-i)
  end;
```

$proj := \mathbf{proc}(i, j, m)$

$i +$

$LinearAlgebra\text{:-}DotProduct(m - i, j - i) * (j - i) \,/\, LinearAlgebra\text{:-}DotProduct(j - i, j - i)$

end proc

```
> i:=<1,0>:j:=<4,0>:m:=<2,1>:proj(i,j,m);
```

$$\begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

```
> RM:=proc(x)
  local m,p,q,r;
  m:=<x,0>;
  p:=proj(a,c,m);
  q:=proj(a,b,p);
  r:=proj(b,c,q);
  r[1]
  end;
```

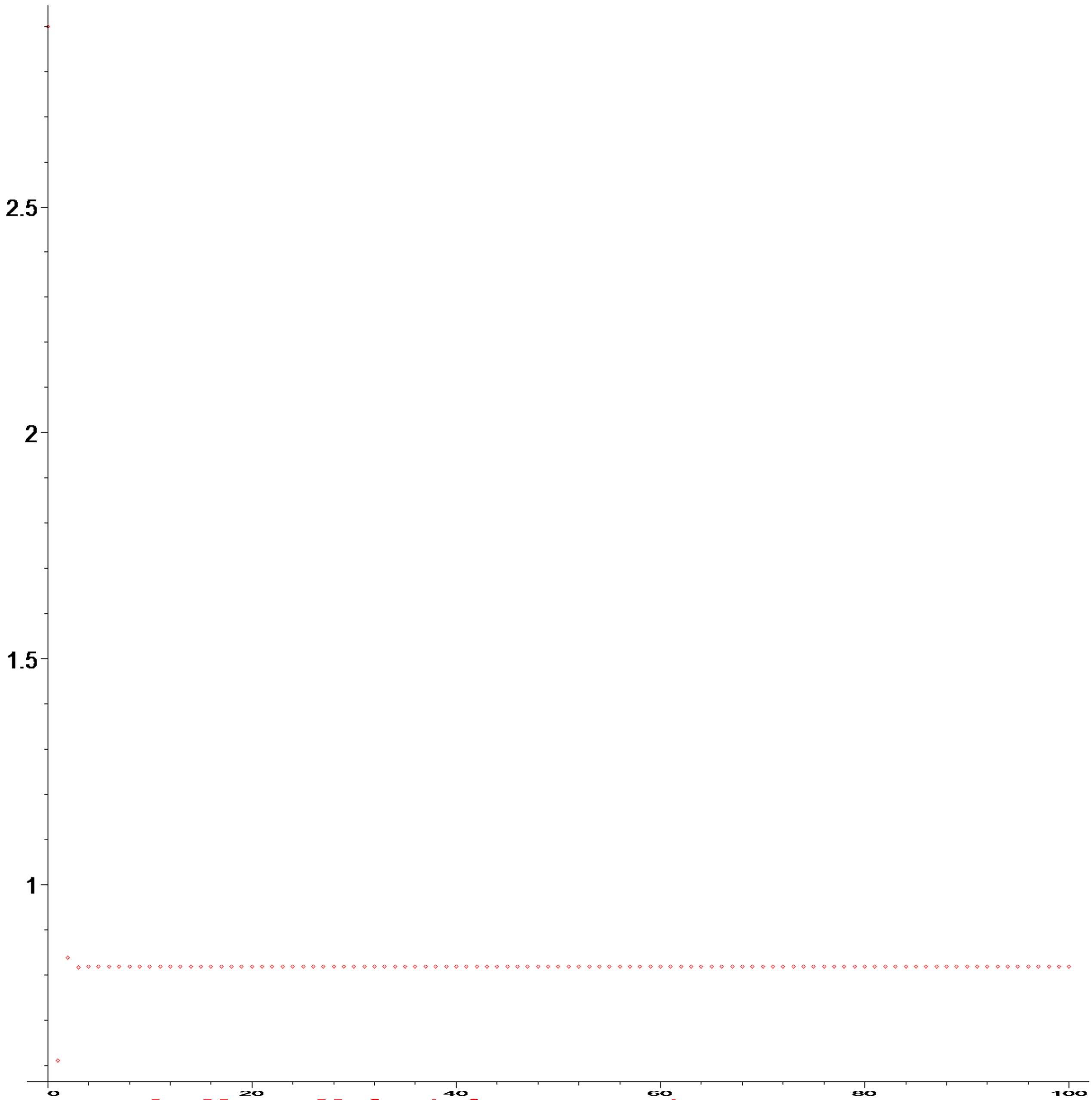
$RM := \mathbf{proc}(x)$

$\mathbf{local}\ m, p, q, r;$

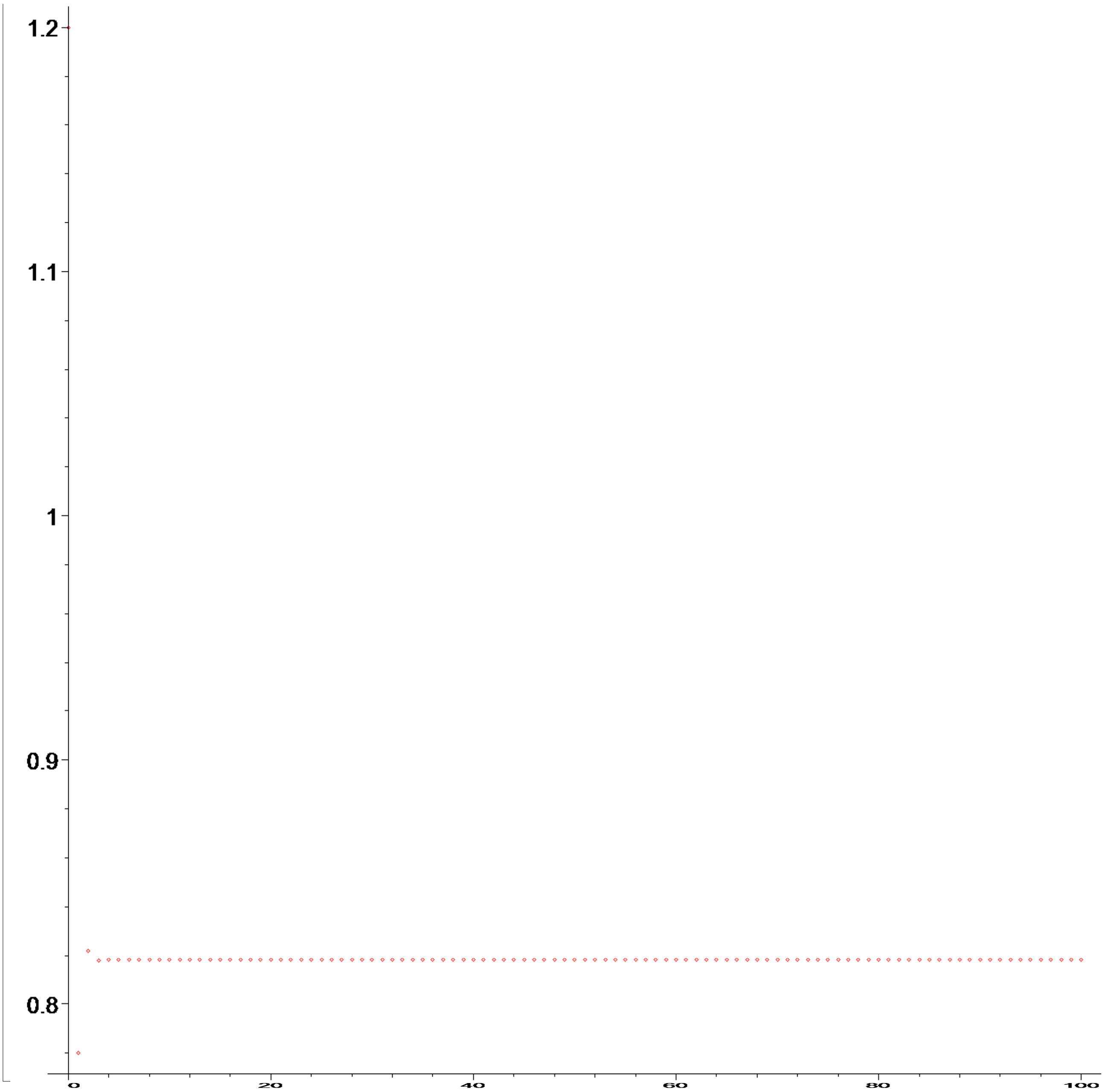
$m := \langle x, 0 \rangle;\ p := \mathrm{proj}(a, c, m);\ q := \mathrm{proj}(a, b, p);\ r := \mathrm{proj}(b, c, q);\ r[1]$

end proc

```
> a:=<1,2>:b:=<0,0>:c:=<3,0>:
> N:=100:l:=[[0,2.9]]:for k from 1 to N do
  x:=RM(l[k][2]);l:=[op(l),[k,x]] od:
> plot(l,style=point);
```




```
> N:=100:l:=[[0,1.2]]:for k from 1 to N do
  x:=RM(l[k][2]);l:=[op(l),[k,x]] od:
> plot(l,style=point);
```



1.2
1.1
1
0.9
0.8
0
20
40
60
80
100

>