Approssimazione della soluzione di un sistema lineare

col metodo di Gauss-Seidel

Enrico Bertolazzi

Carica le librerie

```
> initialize ;
  with(LinearAlgebra) :
  with(plots):
```

initialize

```
Warning, the name changecoords has been redefined
```

Definisce la procedura Normi

```
> Normi := proc(v)
    local i ;
    return max(seq(abs(v[i]),i=1..Dimension(v))) ;
  end proc :
```

Definisce la procedura GS

```
> GS := proc(A, b, xs, n, epsi)
    local i, j, k, resj, x, dm, bf, rlist ;
    rlist := [] ;
    dm    := Dimension(b) ;
    x     := xs ;
    for k from 1 to n do
      bf := 0 ;
      for i from 1 to dm do
        resj := b[i] - add(evalf(A[i,j]*x[j]),j=1..dm) ;
        x[i] := evalf(x[i] + resj/A[i,i]) ;
        if bf < abs(resj) then bf := abs(resj) ; end if ;
      end do ;
      rlist := [ op(rlist), [k,bf] ] ;
      if bf < epsi then break end if ;
    end do ;
    return rlist, x ;
  end proc :
```

Esempio d'uso

```
> # definisce la matrice
  A := <<2,0,-2,1>|<0,2,0,-1>|<-1,0,1,0>|<-3,-1,0,1>> ;
```

$$A := \begin{bmatrix} 2 & 0 & -1 & -3 \\ 0 & 2 & 0 & -1 \\ -2 & 0 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{bmatrix}$$

```
> # definisce il termine noto per avere soluzione <1,2,3,4>
  b := A.<1,2,3,4> ;
```

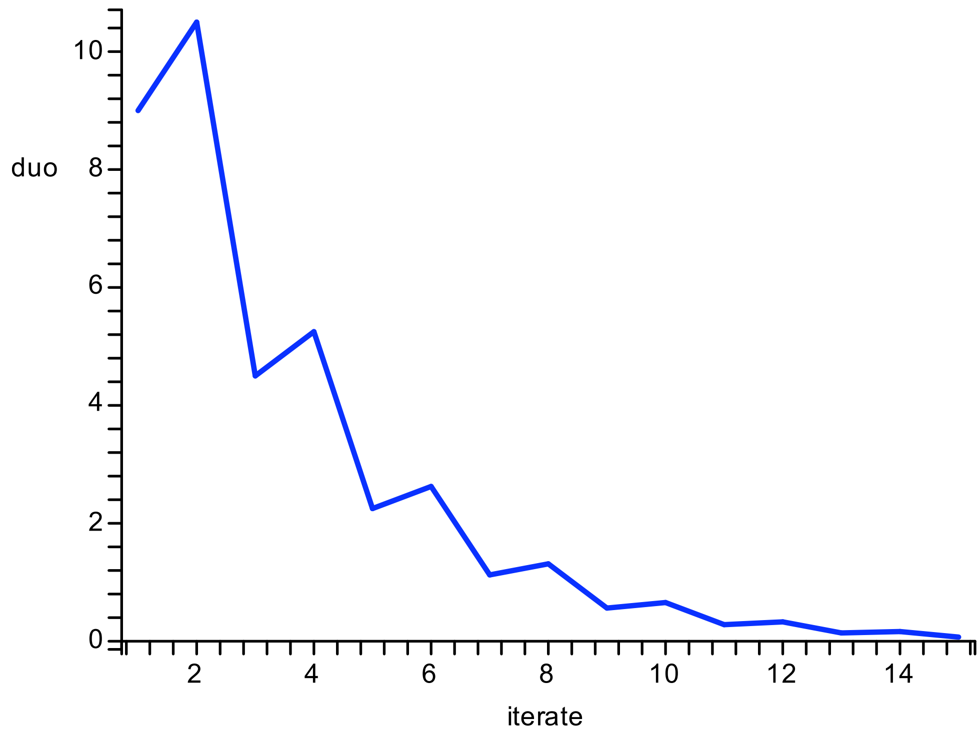
$$b := \begin{bmatrix} -13 \\ 0 \\ 1 \\ 3 \end{bmatrix}$$

```
> # approssima la soluzione con Jacobi
  rlist, sol := GS(A, b, <0,3,2,1>, 15,1e-6) :
```

```
> # stampa la soluzione
  sol ;
```

$$\begin{bmatrix} 1.039062500 \\ 2.011718750 \\ 3.078125000 \\ 3.972656250 \end{bmatrix}$$

```
> # disegna la norma del residuo
  plot(rlist,style=line,thickness=2,color=blue,
       labels=["iterate","residuo"]);
```


duo
10
8
6
4
2
0
2
4
6
8
10
12
14
iterate
>