

```
> restart:
```

Aufgabe 5

Rekursive Programmierung

a) Definieren Sie eine Funktion $A(n)$ zur Konstruktion quadratischer Matrizen der Dimension n mit der Eigenschaft: $A_{i,i} = i, i = 1 \dots n$ (Hauptdiagonalelemente), die untere und obere Nebendiagonale sei konstant 1 und der Rest der Matrix gleich 0.

Hinweis: Verwenden Sie das Kommando [Matrix](#) und erzeugen Sie die Matricelemente mit Hilfe einer anonymen Funktion.

```
> A:=proc(n::nonnegint)
    description "Funktion fuer die Generierung einer nxn Matrix mit
    1 auf der Hauptdiagonale, ansonsten 0":
    uses LinearAlgebra:
    Matrix(n, (i,j) -> if i=j then i elif i-1=j or j-1=i then 1
    else 0 end if):
end proc;
```

```
A := proc(n::nonnegint)
```

```
description
```

```
"Funktion fuer die Generierung einer nxn Matrix mit 1 auf der Hauptdiagonale, ansonsten 0";
```

```
Matrix(n, (i,j) -> if i=j then i elif i - 1 = j or j - 1 = i then 1 else 0 end if)
```

```
end proc
```

```
> matrix_A := A(4);
```

$$matrix_A := \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 3 & 1 \\ 0 & 0 & 1 & 4 \end{bmatrix}$$

(2)

b) Die Laplace-Entwicklung ([Pierre-Simon Laplace](#)) der Determinante von $A(n)$ bzgl. der letzten Reihe zeigt, dass $\det A(n)$ durch folgende Rekursion berechnet werden kann:

$$\det A(n) = n \det A(n-1) - \det A(n-2)$$

Schreiben Sie mit Hilfe dieser Rekursion eine Prozedur zur Berechnung von $\det A(n)$.

```
> detA:=proc(n::nonnegint)
    description "Funktion fuer die rekursive Berechnung der
    Determinante einer Matrix.":
```

```

    if n < 2 then
        1
    else
        n*detA(n-1)-detA(n-2)
    end if:
end proc;

```

```
> detA(4);
```

```
detA := proc(n:nonnegint)
```

```
    description "Funktion fuer die rekursive Berechnung der Determinante einer Matrix.";

```

```
    if n < 2 then 1 else n*detA(n - 1) - detA(n - 2) end if

```

```
end proc
```

7

(3)

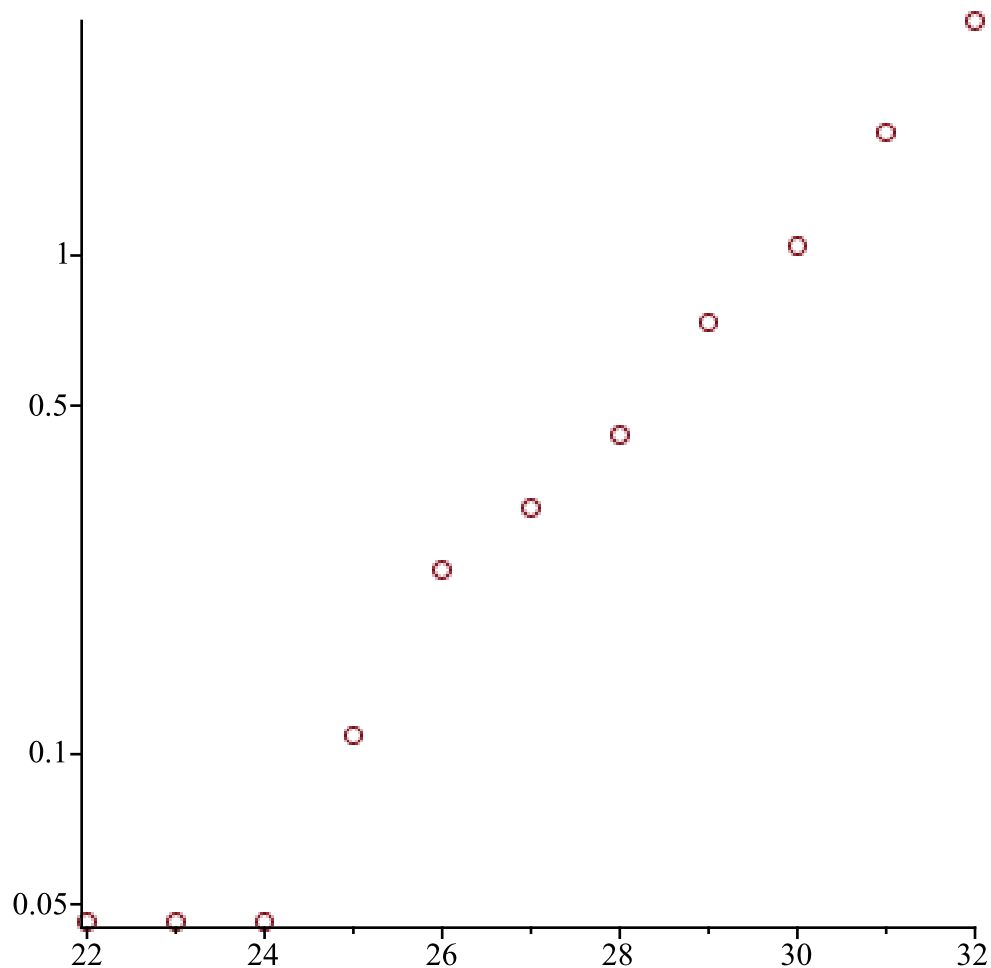
c) Berechnen Sie die Determinanten $\det A(n)$, $n = 1 \dots 32$, und messen Sie die benötigten CPU-Zeiten. Plotten Sie die Zeiten für $i = 22 \dots 32$ in ein semi-logarithmisches Diagramm mit logarithmischer Skalierung in y -Richtung.

Hinweis: Nutzen Sie [time](#) für die Zeitmessung, [seq](#) für die Erzeugung der Liste mit den Zeitwerten und [logplot](#) aus dem Paket [plots](#) für die Darstellung der Datenpunkte (style=point).

```

> times1 := seq([i,time(detA(i))], i=1..32):
with(plots):
log_plot_1 := logplot([times1[22..32]], style=point,symbol=
circle,symbolsize=15);

```



d)

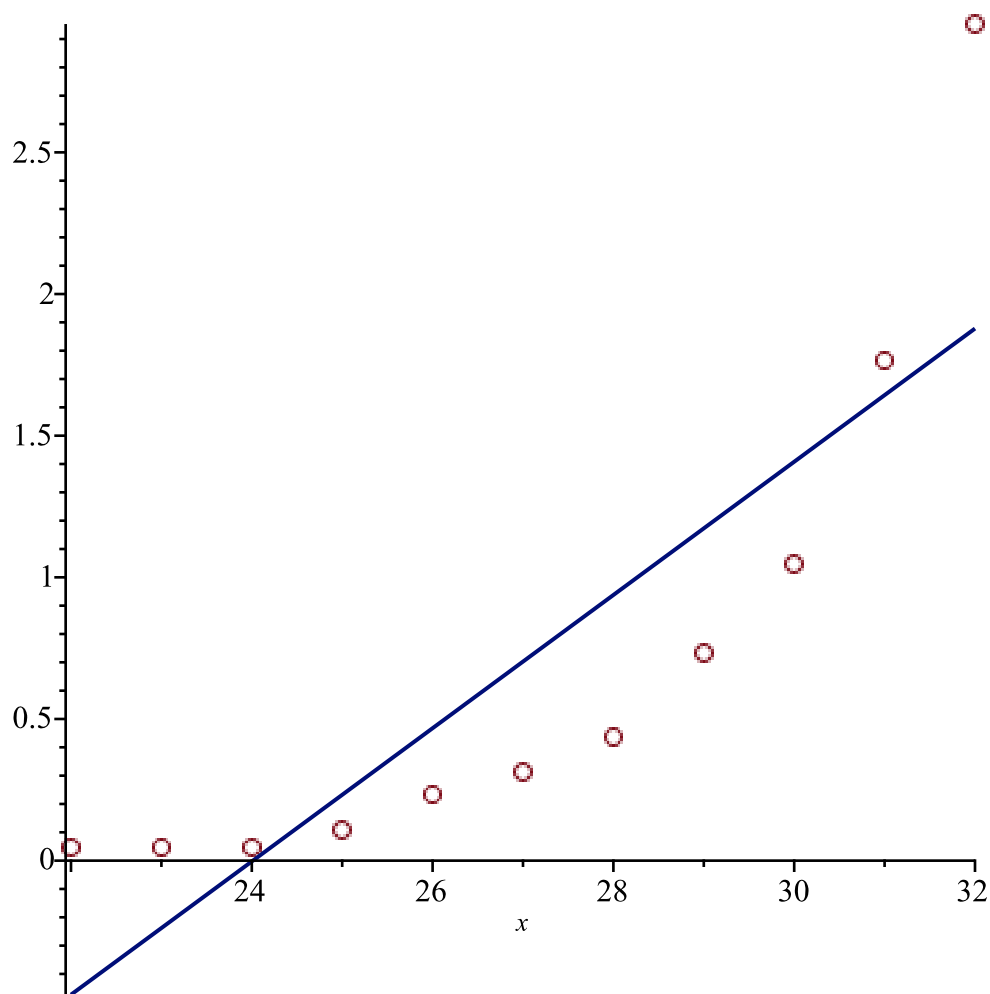
i) Berechnen Sie mit der Funktion [LeastSquares](#) aus dem [CurveFitting](#)-Paket die Ausgleichsgerade für die logarithmierten Zeitwerte von $n = 22 \dots 32$ und plotten Sie die Datenpunkte und die Ausgleichsgerade in ein Schaubild.

Hinweis: Kapitel 1.8 des Skripts zeigt ein Beispiel zur Anwendung der Funktion `LeastSquares`.

```
> with(CurveFitting):
g:=unapply(LeastSquares([times1[22..32]],x),x);
p1:=plot([times1[22..32]], style=point,symbol=circle,symbolsize=
15):
p2:=plot(g(x),x=22..32):
display(p1, p2);

$$g := x \mapsto -5.64589090909090352 + 0.235127272727272546 \cdot x$$

```



ii) Lösen Sie die Aufgabe i) mit der Maplet-Oberfläche [Interactive](#) aus dem Paket CurveFitting.

```
> with(CurveFitting):  
Interactive([times1[22..32]]);
```

e) Bestimmen Sie mit Hilfe der Ausgleichsgeraden die CPU-Zeit, die Ihr Computer für die Berechnung von $\det A(100)$ benötigen würde.

Hinweis: Ein Jahr hat etwa $365.25 \cdot 24 \cdot 3600$ Sekunden.

```
> g(100);  
17.8668363636364
```

(4)

Das Ergebnis ist wahrscheinlich eher unrealistisch klein, da die Punkte eher nach einer e-Funktion aussehen, als dass diese einer Gerade zugeordnet werden könnten. Man sieht vorallem die stärker werdende Steigung bei den Punkten 30 und größer.

f) Erweitern Sie die Prozedur unter b) mit der *option remember* und messen Sie wiederum die benötigte CPU-Zeit für die Berechnung von $\det A(100)$.

```

> detA_rem:=proc(n::nonnegint)
  description "Funktion fuer die rekursive Berechnung der
  Determinante einer Matrix.":
  option remember;
  if n < 2 then
    1
  else
    n*detA(n-1)-detA(n-2)
  end if;
end proc;
detA_rem := proc(n::nonnegint)
  option remember;
  description "Funktion fuer die rekursive Berechnung der Determinante einer Matrix.";
  if n < 2 then 1 else n*detA(n - 1) - detA(n - 2) end if
end proc

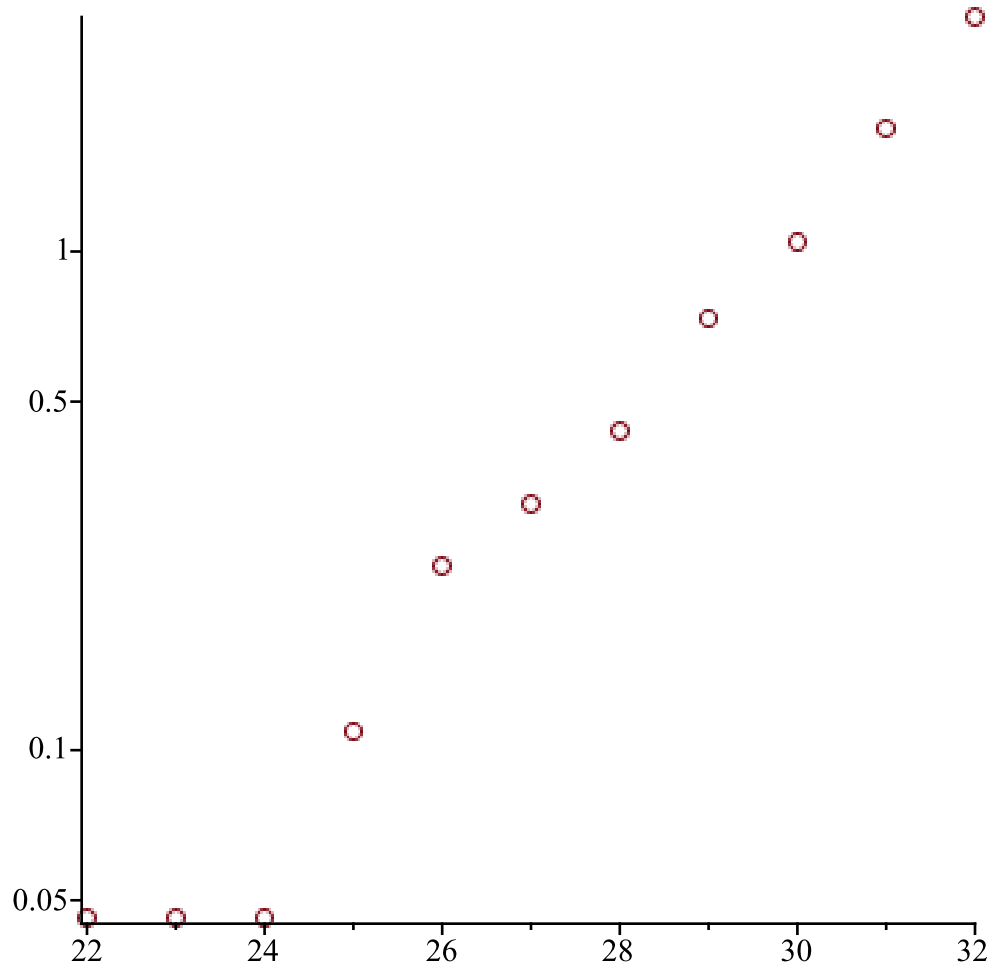
```

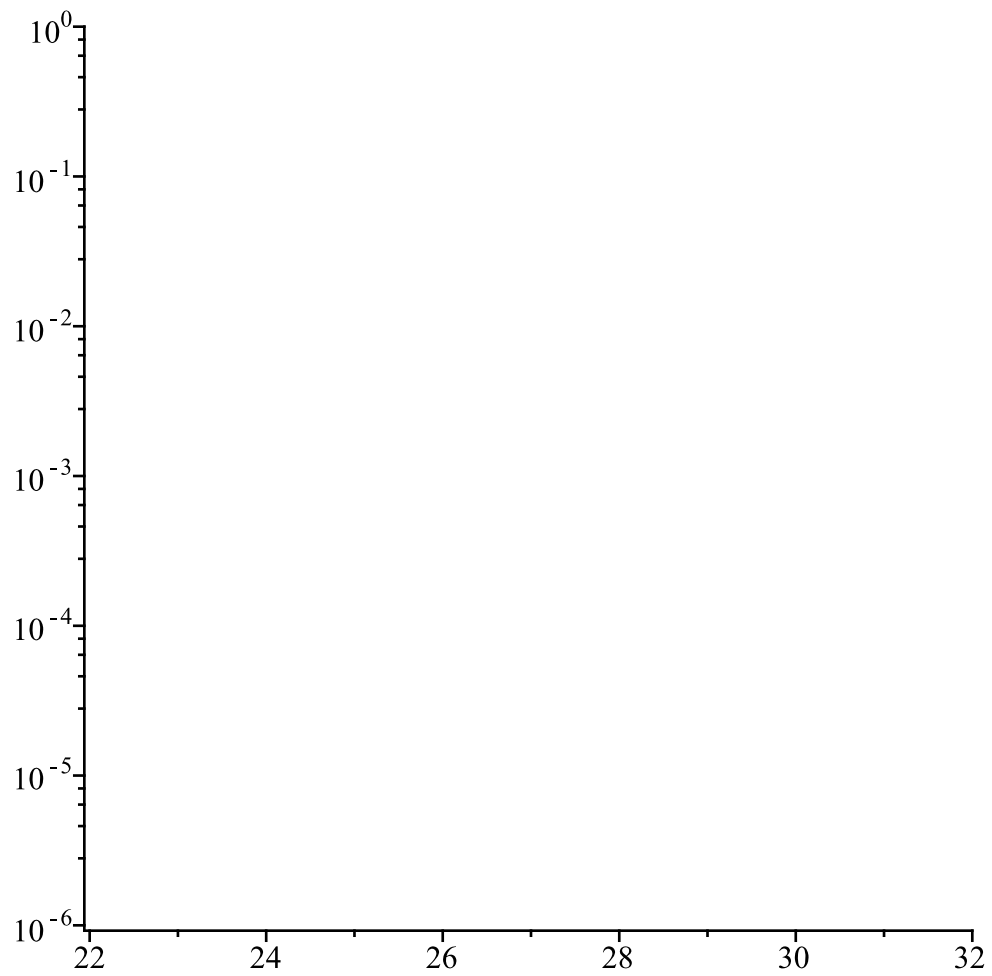
```

> times2 := seq([i,time(detA_rem(i))], i=1..32):
with(plots):
log_plot_1;
lop_plot_2 := logplot([times2[22..32]], style=point,symbol=
circle,symbolsize=15);

```

(5)





Man sieht, dass die Punkte gar nicht erst angezeigt werden, da die rekursive Funktion mit der remeber option anscheinend sehr sehr schnell ist, allerdings nur im zweiten Durchlauf.