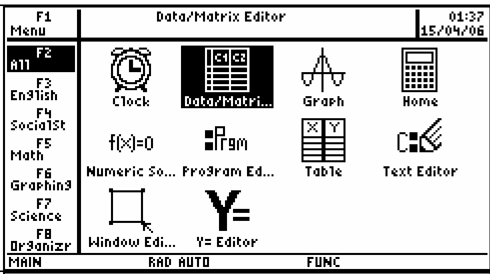
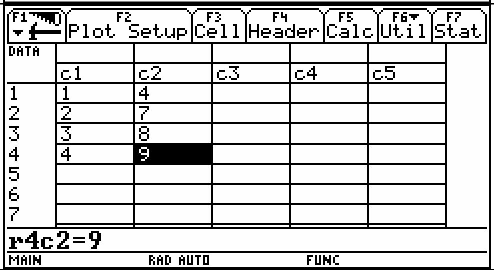
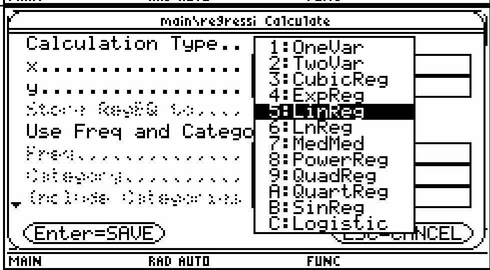
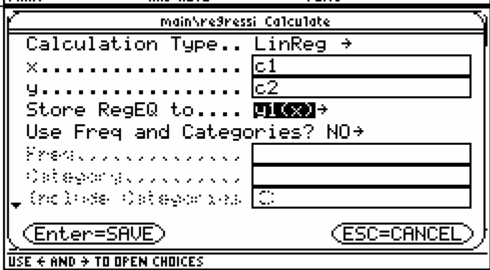
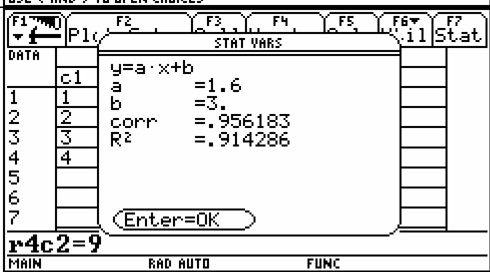
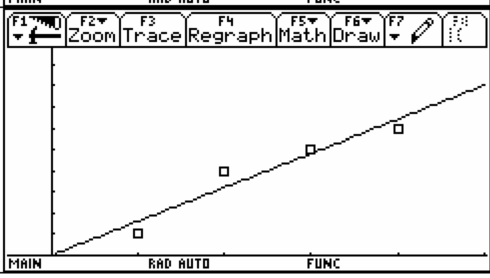
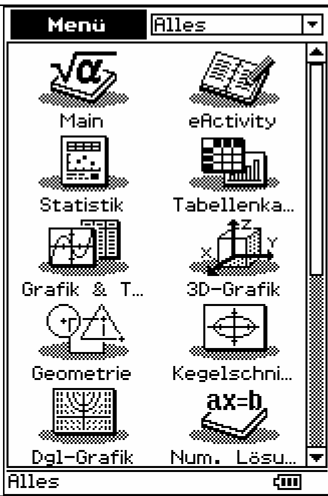
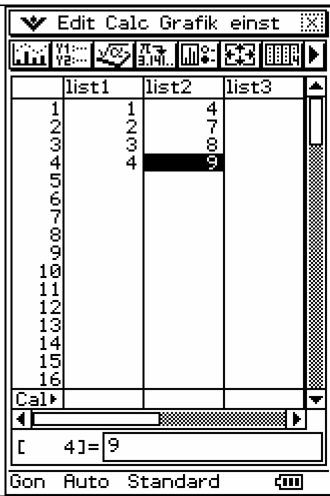
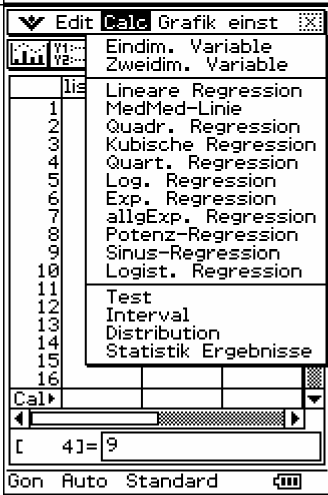
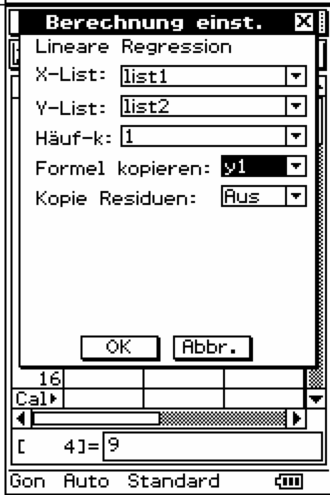
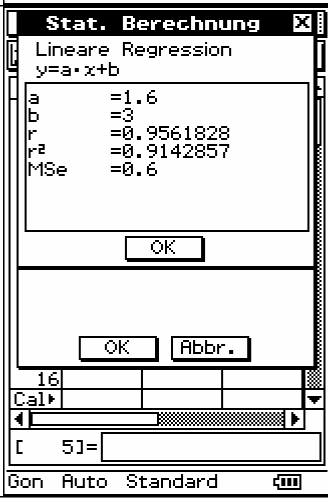
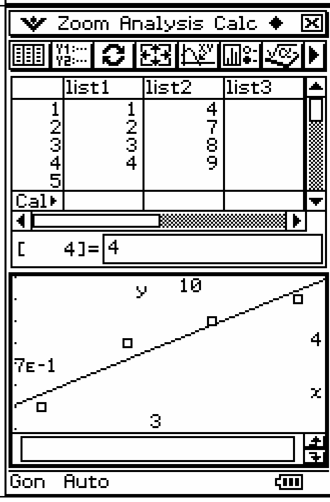
Regressionen mit CA-Systemen

TI-Voyage 200:

Aufruf der Applikation Data-Matrix-Editor	
Eingabe der Messwerte	
Aufruf von <F5> Calc und Auswahl der linearen Regression	
Zuweisung der x-Werte (c1) und y-Werte (c2); ggf. Speichern der Regressionsfunktion in Y1	
Die ENTER-Taste liefert die gewünschten Regressionsparameter	
Ggf. kann man sich nun die Punkte und die Regressionskurve noch mit <GRAPH> zeichnen lassen (ggf. Zeichenbereich anpassen)	

Classpad:

Statistik-Modul im Hauptmenü auswählen Messwerte in die Spalten eintragen		
Calc – Lineare Regression aufrufen list1 und list2 für die x-y-Werte eintragen; ggf. y1 als Funktionsvariable eintragen		
„OK“ liefert die Parameter der Regressionskurve. Das Symbol bzw. die „OK“-Taste zeichnet die Regressionskurve samt Messpunkten		

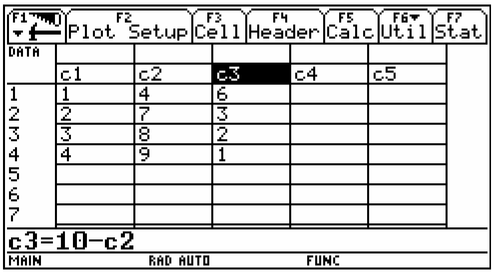
Approximation des exponenziell begrenzten Wachstums

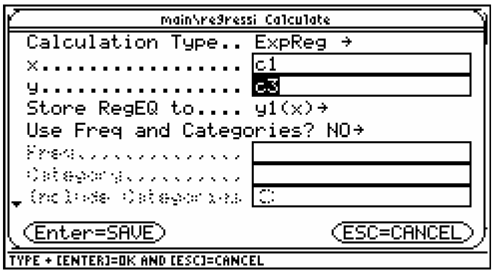
In diesem Fall muss bei den Kleinrechnern eine entsprechende Transformation durchgeführt werden. Hierzu gehört, dass der vermeintliche Grenzwert „erraten“ ode r sonst wie bestimmt wird, z.B. aus dem Aufgabenkontext.

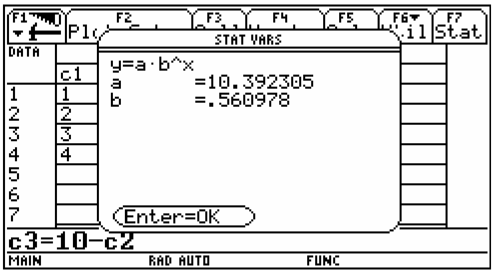
TI-Voyage:

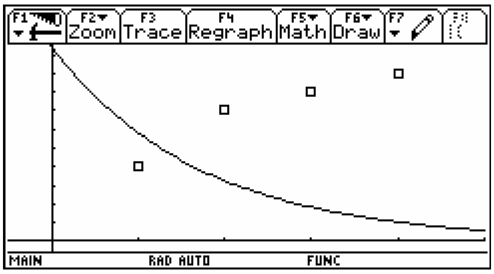
Eingabe der Formel „10 -c2“ in der Spaltenüberschrift von c3 (10 entspricht dem vermuteten Grenzwert, 10-c2 bewirkt, dass in jeder Zelle von Spalte c3 die Differenz von 10 und dem entsprechenden Zellenwert der Spalte c2 eingetragen wird.

Danach wird die exponenzielle Regression zwischen c1 und c3 bestimmt.

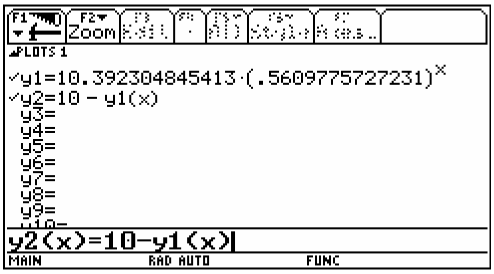


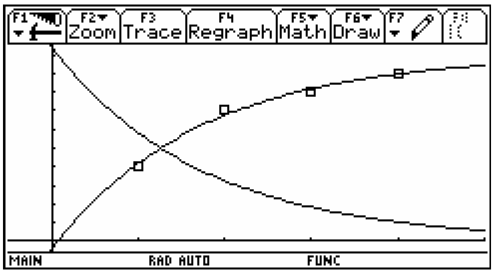






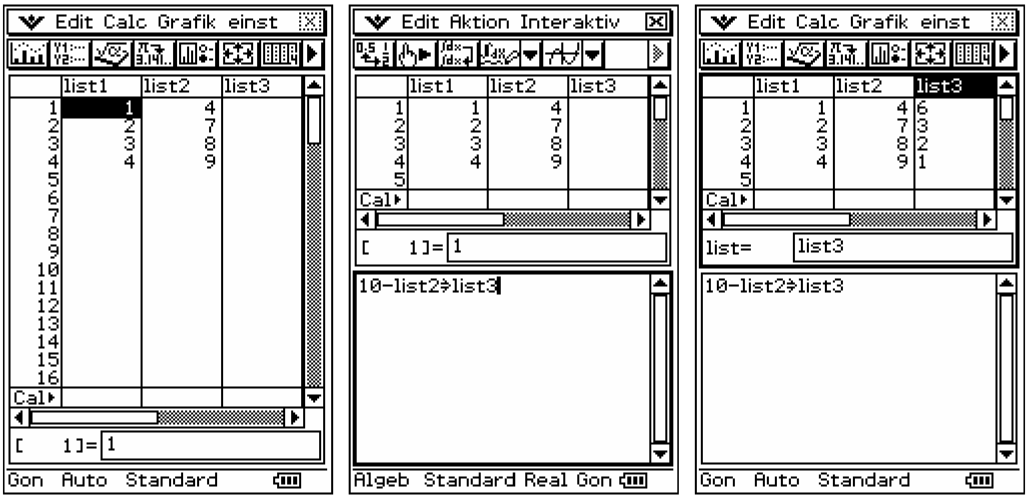
Nun wird die Regressionsfunktion analog zurücktransformiert (in y2 gespeichert):



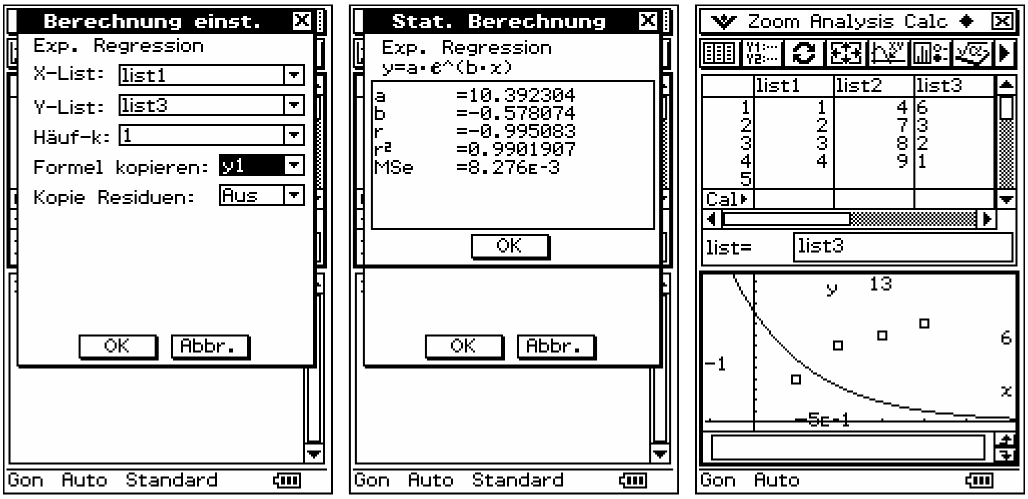


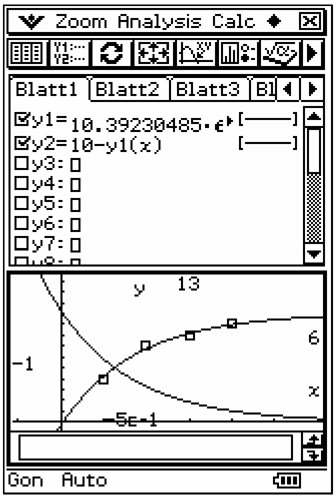
Classpad:

Nach Eingabe der Messwerte in list1 und list2 umschalten auf das Main-Menü und dort die Differenzwerte „10-list2“ in list3 speichern. Erneutes Umschalten auf den Data-Matrix-Editor bewirkt, dass nun die Spaltenwerte in list3 angezeigt werden.



Nun kann die exponenzielle Regression auf die Spalten list1 und list3 angewandt werden und die Regressionskurve (sowie anschließend auch die rücktransformierte Kurve) gezeichnet werden:





Maple:

Das Statistics-Paket liefert einige Regressions-Anpassungen:

ExponentialFit	fit an exponential function to data
Fit	fit a model function to data
LinearFit	fit a linear model function to data
LogarithmicFit	fit a logarithmic function to data
NonlinearFit	fit a nonlinear model function to data
OneWayANOVA	generate a one-way ANOVA table
PolynomialFit	fit a polynomial to data
PowerFit	fit a power function to data

Dabei ist der Fit-Befehl universell, liefert aber leider keine Korrelationskoeffizienten:

```
> restart:
with(Statistics):
X := Vector([1,2,3,4]);
Y := Vector([4,7,8,9]);
```

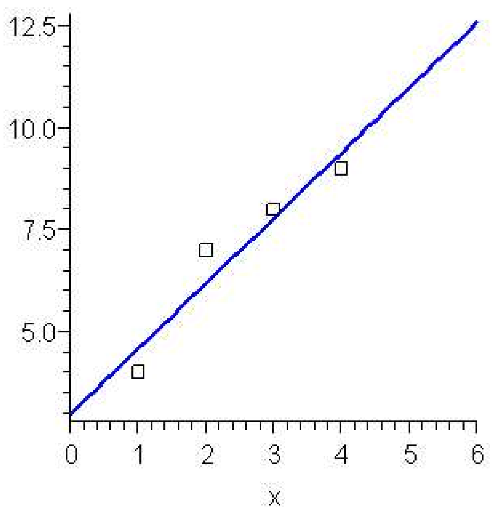
$$X := \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \quad Y := \begin{bmatrix} 4 \\ 7 \\ 8 \\ 9 \end{bmatrix}$$

Der Fit-Befehl des Pakets Statistics ist universell - man kann sehr viele Funktionstypen zur Anpassung angeben.

```
> Reg:=Fit(a*x+b,X,Y,x);
```

$$Reg := 1.60000000000000052\,x + 2.99999999999999986$$

```
> p1:=plot(Reg,x=0..6,thickness=2,color=blue):
p2:=pointplot({seq( [X[i],Y[i]],
i=1..nops(X)+1)},symbol=box,symbolsize=15,thickness=3):
with(plots):display({p1,p2});
```



Mit der Option "output=solutionmodule" lassen sich die Ergebnisse ausgeben - leider fehlt der Korrelationskoeffizient!

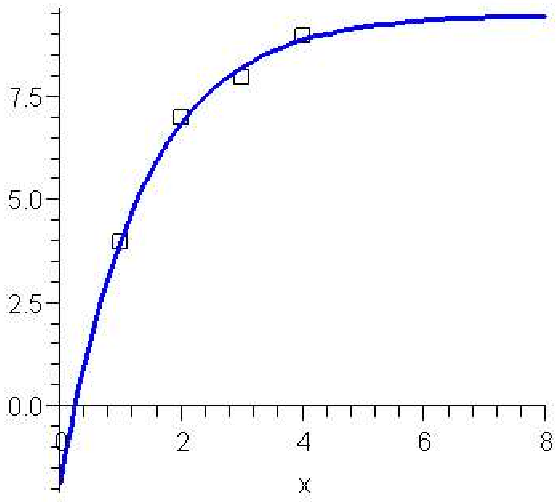
```
> m:=Fit(a*x+b,X,Y,x,output=solutionmodule);
m:-Results();
```

["residualmeansquare" = 0.60000000000000
, "residualsumofsquares" = 1.20000000000000416
, "residualstandarddeviation" = 0.77459666924148
, "degreesoffreedom" = 2
, "parametervalues" = [1.60000000000000052, 2.99999999999999866]
, "parametervector" = [1.60000000000000052, 2.99999999999999866]
, "leastsquaresfunction" = 1.60000000000000052 x + 2.99999999999999866
, "standarderrors" = [[0.346410161513776182 , 0.948683298050515876]] ,
"confidenceintervals" = [[0.1095173729498 .. 3.0904826270502], [− 1.0818547820248 .. 7.0818547820248]] ,
"residuals" = [[− .600000000000001310 , 0.800000000000001042 , 0.200000000000000842 , − .400000000000000744]] , "leverages" = [[0.700000000000000954 , 0.300000000000000264 , 0.299999999999999932 , 0.699999999999999732]]
, "variancecovariancematrix" = [[0.120000000000000494 , − .300000000000001320], [− .300000000000001320 , 0.900000000000003908]]]

Regressionskurve für begrenztes Wachstum:
Beachten Sie, dass vor dem Parameter "c" das Minuszeichen stehen muss:

```
> with(Statistics):Reg:=Fit(a+b*exp(-c*x),X,Y,x);
p1:=plot(Reg,x=0..8,thickness=2,color=blue):
p2:=pointplot({seq( [X[i],Y[i]],
i=1..4)},symbol=box,symbolsize=15,thickness=3):
with(plots):display({p1,p2});
```

$$Reg := 9.49492748881018578 - 11.3555417648639594\, e^{(-0.730771884780653580\, x)}$$



Zweite Methode: Regression mit dem stats-Paket

```
> restart:
with(stats):
X := [1,2,3,4];
Y := [4,7,8,9];
```

$$X := [1, 2, 3, 4] \quad Y := [4, 7, 8, 9]$$

Der fit-Befehl des Pakets stats erlaubt auch vielfältige Regressionen, allerdings nur solche in linearen Parametern

```
> fit[leastsquare[[x,y],y=a*x+b, {a,b}]]([X,Y]);
Reg:=rhs(%):
```

$$y = \frac{8}{5}x + 3$$

Dafür liefert das Paket stats auch den Korrelationskoeffizienten:

```
> describe[linearcorrelation](X,Y);
evalf(%);
```

$$\frac{4}{35}\sqrt{70}$$

0
.9561828875

Quadratische Anpassung

```
> fit[leastsquare[[x,y],y=a+b*x+c*x^2, {a,b,c}]]([X,Y]);
p1:=plot(rhs(%),x=0..5,thickness=2,color=blue):
p2:=pointplot([seq( [X[i],Y[i]],
i=1..4)],symbol=box,symbolsize=15,thickness=3):
with(plots):display({p1,p2});
```

$$y = \frac{1}{2} + \frac{41}{10}x - \frac{1}{2}x^2$$