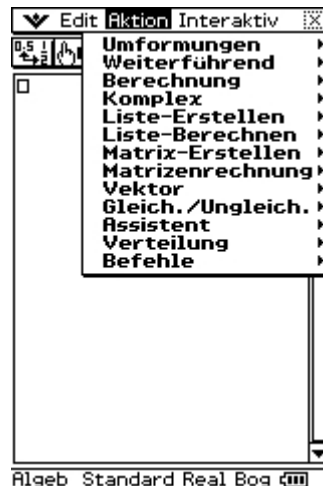


Die wichtigsten Befehle für den ClassPad (v.3.03)

Die wichtigsten Befehle für den ClassPad sind über das Menü **Aktion** zu erreichen:



1. Analysis

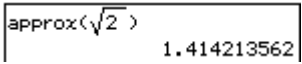
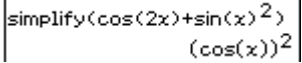
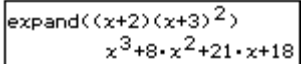
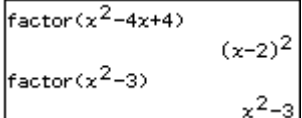
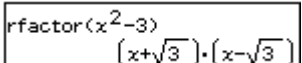
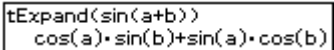
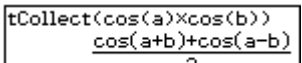
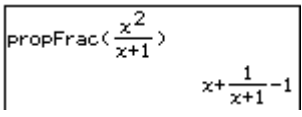
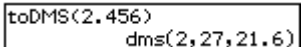
Verwendung des Untermenüs **Befehle**

<p>Define Funktion: Erstellt eine anwenderdefinierte Funktion Syntax: Define <i>Funktionsname</i>(<i>Variable</i>) = <i>Term</i></p>	<pre>Define f(x,t)=(x+t)e^{-t*x} done</pre>
---	---

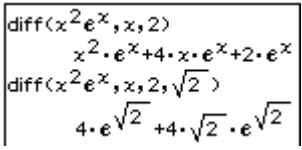
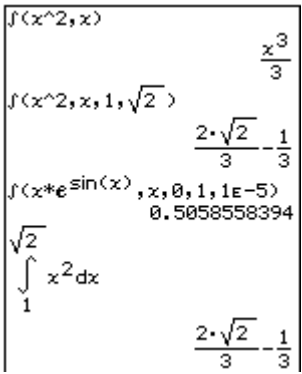
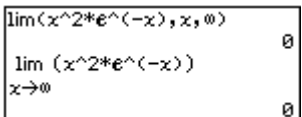
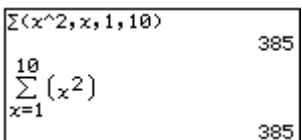
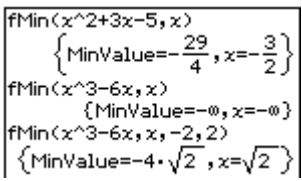
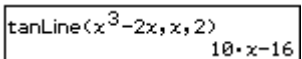
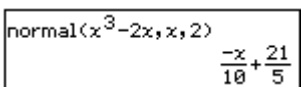
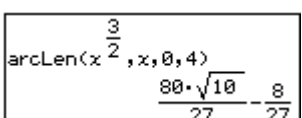
Verwendung des Untermenüs **Gleich./Ungleich.**

<p>solve Funktion: Liefert die Lösung einer Gleichung oder Ungleichung oder eines Gleichungssystems. Syntax: solve (<i>Gleichung</i> , <i>Variable</i>) solve (<i>Gleichung</i> , <i>Variable</i> , <i>Schätzwert</i>) solve ({<i>Gleichung</i>, <i>Gleichung</i>}, {<i>Variable</i>, <i>Variable</i>})</p> <p>Bem: Die Eingabe eines Schätzwertes für die Lösung erzwingt ein numerisches Lösungsverfahren.</p> <p>Gleichungssysteme können auch über das 2D-Keyboard eingegeben werden.</p>	<pre>solve(e^{0.5x+2}=15,x) {x=2*ln(5)+2*ln(3)-4} solve(e^{0.5x+2}=15,x,1) {x=1.416100402}</pre> <pre>solve({3x+4y=5,2x-3y=-8},{x,y}) {x=-1,y=2}</pre>
<p>dSolve Funktion: Löst gewöhnliche Differenzialgleichungen erster, zweiter und dritter Ordnung. Syntax: dSolve (<i>Gleichung</i> , <i>unabhängige Variable</i> , <i>abhängige Variable</i> , [<i>Anfangsbedingung(en)</i>])</p>	<pre>dSolve(y'=-0.2(y-30),x,y) {y=-20*e^{-x/5}+30}</pre>
<p>rSolve Funktion: Liefert die explizite Formel einer Zahlenfolge, die rekursiv definiert ist. Syntax: rSolve (<i>Rekursionsgleichung</i> , <i>Anfangsbedingung</i>)</p>	<pre>rSolve(a_{n+1}=0.5a_n-1,a₀=1) {a_n=3*(1/2)ⁿ-2}</pre>

Verwendung des Untermenüs **Umformungen**

<p>approx Funktion: Transformiert einen Term in eine approximative numerische Darstellung Syntax: <code>approx(numerischer Ausdruck)</code></p>	
<p>Simplify Funktion: Vereinfacht einen Term Syntax: <code>simplify (Term)</code></p>	
<p>expand Funktion: Multipliziert einen Term aus Syntax: <code>expand (Term)</code></p>	
<p>factor Funktion: Ergibt die rationalen Faktoren eines Terms. Syntax: <code>factor(Term)</code></p>	
<p>rFactor Funktion: Ergibt die Faktoren eines Terms bis hin zu dessen Wurzeln, sofern vorhanden. Syntax: <code>rfactor(Term)</code></p>	
<p>tExpand Funktion: Verwendet die Additionstheoreme, um eine trigonometrische Funktion zu zerlegen. Syntax: <code>tExpand(trig.Term)</code></p>	
<p>tCollect Funktion: Verwendet die Produkt- und Summenformeln, um das Produkt trigonometrischer Terme in Summenform zu transformieren. Syntax: <code>tCollect(trig.Term)</code></p>	
<p>propFrac Funktion: Zerlegt eine gebrochen rationalen Term in seinen ganzrationalen Anteil und seinen echt gebrochenrationalen Anteil. Syntax: <code>propFrac (Term)</code></p>	
<p>toDMS Funktion: Transformiert eine Altgraddarstellung in die gleichwertige Darstellung des Altgrad/Minuten/Sekunden-Formats. Syntax: <code>toDMS (Dezimalzahl)</code> Bem: <code>toDMS</code> wird oft verwendet, um Stundenangaben in Stunden/Minuten/Sekunden zu verwandeln.</p>	

Verwendung des Untermenüs **Berechnungen**

<p>diff Funktion: Berechnet die Ableitung eines Terms hinsichtlich einer bestimmten Variablen. Syntax: <code>diff(Term, Variable, Ordnung)</code> <code>diff(Term, Variable, Ordnung, Stelle)</code></p>	
<p>∫ Funktion: Berechnet das Integral für einen Term bezüglich einer bestimmten Variablen. Syntax: <code>∫ (Term, Variable)</code> <code>∫ (Term, Variable, untere Grenze, obere Grenze, tol)</code> <ul style="list-style-type: none"> • „tol“ ist der zulässige Fehler bei einer num. Integration. • Dieser Befehl liefert einen approximativen Integralwert, wenn für „tol“ eine Fehlerschranke vorgegeben wird. Bem: Integrale (und Differentiale) können auch mit dem 2D-Keyboard eingegeben werden.</p>	
<p>lim Funktion: Bestimmt den Grenzwert eines Terms. Syntax: <code>lim(Term, Variable, Punkt)</code> Bem: Grenzwerte können auch mit dem 2D-Keyboard eingegeben werden.</p>	
<p>Σ Funktion: Interpretiert die Variable in einem Term als diskrete Variable innerhalb eines Lauf-Bereichs und berechnet danach die Summe. Syntax: <code>Σ(Term, Variable, Anfangswert, Endwert)</code> Bem: Summen können auch mit dem 2D-Keyboard eingegeben werden.</p>	
<p>fMin / fMax Funktion: Liefert den globalen Extrempunkt (Minimum bzw. Maximum) einer Funktion in einem bestimmten Intervall. Syntax: <code>fMin(Term, Variable)</code> <code>fMin(Term, Variable, Anfangswert, Endwert)</code></p>	
<p>tanLine Funktion: Liefert den Term für die Tangente an einen bestimmten Punkt der Kurve. Syntax: <code>tanLine (Term, Variable, Stelle)</code></p>	
<p>normal Funktion: Liefert den Term für die Normale auf einem bestimmten Punkt der Kurve. Syntax: <code>normal (Term, Variable, Stelle)</code></p>	
<p>arcLen Funktion: Berechnet die Bogenlänge einer Kurve von einem Anfangswert bis zu einem Endwert bezüglich der vorgegebenen Variablen. Syntax: <code>arcLen (Term, Variable, Startwert, Endwert)</code></p>	



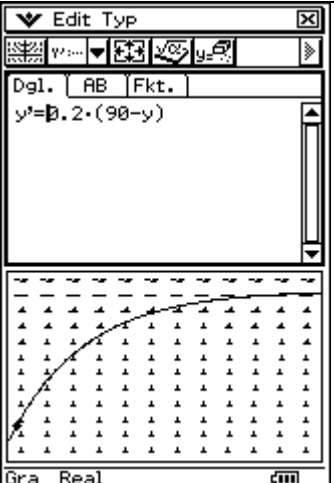
Verwendung des Untermenüs **Liste-Erstellen**

<p>Seq Funktion: Erzeugt eine Liste in Abhängigkeit von einem numerischen Folgenterm. Syntax: <code>seq (Term, Variable, Startwert, Endwert [,Schrittweite])</code></p>	<pre>seq(x^2+x,x,1,5) {2,6,12,20,30}</pre>
<p>sortA / sortD Funktion: Sortiert die Elemente der Liste in ansteigender bzw. absteigender Reihenfolge. Syntax: <code>sortA (List) bzw. sortD (List)</code></p>	<pre>sortA({2,6,1,3,9,5,17}) {1,2,3,5,6,9,17}</pre>

Verwendung des Untermenüs **Liste-Berechnen**

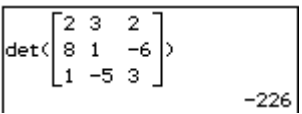
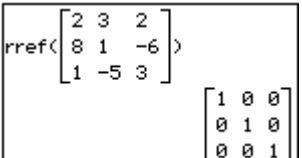
<p>sum Funktion: Liefert die Summe der Elemente in einer Liste. Syntax: <code>sum(Liste)</code></p>	<pre>seq(x^2,x,1,5)≠list1 {1,4,9,16,25} sum(list1) 55</pre>
<p>cuml Funktion: Liefert die kumulativen Summen der Elemente in einer Liste. Syntax: <code>cuml (List)</code></p>	<pre>seq(x^2,x,1,5)≠list1 {1,4,9,16,25} cuml(list1) {1,5,14,30,55}</pre>
<p>Δlist Funktion: Liefert eine Liste, deren Elemente den Differenzen zwischen zwei benachbarten Elementen einer anderen Liste entsprechen.</p>	<pre>seq(x^2,x,1,5)≠list1 {1,4,9,16,25} Δlist(list1) {3,5,7,9}</pre>
<p>sequence Funktion: Liefert ein Polynom des niedrigsten Grades, das der durch die eingegebene Liste ausgedrückten Zahlenfolge (mit Indexbereich ab 1,2,...) entspricht. Wenn zwei Listen vorhanden sind, dann liefert dieser Befehl ein Polynom, das jedem Element der ersten Liste das entsprechende Element der zweiten Liste zuordnet (Polynom durch gegebene Punktepaare).</p>	<pre>sequence({0,3,8,15,24}) x^2-1 sequence({1,5,9},{6,2,-1}) x^2 - 19·x + 229 32 - 16 + 32</pre>

Richtungsfelder von linearen DGL 1.Ordnung

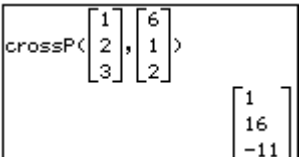
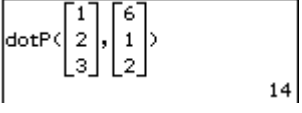
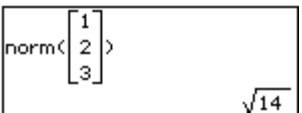
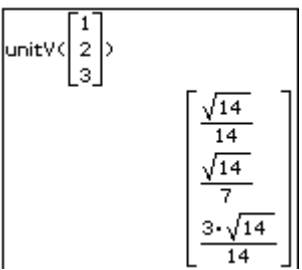
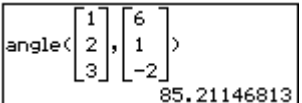
<p>Menü / Dgl-Grafik Die DGL wird unter der Registerkarte Dgl. eingegeben, die Anfangsbedingungen werden unter der Registerkarte AB festgelegt. Mit dem Icon  kommt man ins Grafik-Fenster. Unter  können die Fensterabmessungen und die Schrittweite für die Richtungspfeile festgelegt werden.</p>	
---	---

2. Analytische Geometrie

Verwendung des Untermenüs **Matrizenrechnung**




<p>det Funktion: Liefert die Determinante einer quadratischen Matrix. Syntax: <code>det (Matrix)</code></p>	
<p>rref Funktion: Liefert die reduzierte Zeilenstufenform von einer Matrix (diagonalisierte Form). Syntax: <code>rref (Matrix)</code></p>	

Verwendung des Untermenüs **Vektor**

<p>crossP Funktion: Liefert das Kreuz-Produkt zweier Vektoren. Syntax: <code>crossP (Vektor₁, Vektor₂)</code></p>	
<p>dotP Funktion: Liefert das Skalar-Produkt zweier Vektoren. Syntax: <code>dotP (Vektor₁, Vektor₂)</code></p>	
<p>norm Funktion: Liefert die Euklidische Norm (Länge) eines Vektors. Syntax: <code>norm (Vektor)</code></p>	
<p>unitV Funktion: Normiert einen Vektor. Syntax: <code>unitV (Vektor)</code></p>	
<p>angle Funktion: Liefert den von zwei Vektoren gebildeten Winkel. Syntax: <code>angle (Vektor₁, Vektor₂)</code> Bem: sinnvolle Nutzung nur im Dezimal-Mode</p>	

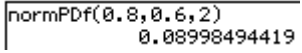
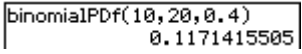
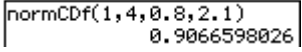
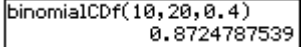
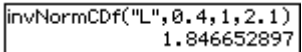
3. Stochastik

Kombinatorik

n-Fakultät Syntax: Zahl !	
Permutationen Ziehen ohne Zurücklegen unter Beachtung der Reihenfolge Syntax: nPr(Auswahl, Gesamtheit)	
Kombinationen Ziehen mit einem Griff Syntax: nCr(Auswahl, Gesamtheit)	

Verwendung des Untermenüs **Verteilungen**

Normalverteilung / Binomialverteilung

normPDF bzw. binomialPDF P(X=x) Funktion: Der Befehl berechnet die Wahrscheinlichkeitsdichte p einer Normal- bzw. Binomialverteilung an einer bestimmten Stelle x. Syntax: normPDF(x, σ , μ) bzw. binomialPDF(x, Anzahl, Trefferwkt.) Bem: Die berechnete Wahrscheinlichkeitsdichte wird als numerischer Wert ausgegeben und in der System-Variable prob gespeichert.	 
normCDF binomialCDF P(X ≤ oberer Wert) Funktion: Mit dem Befehl kann die Intervallwahrscheinlichkeit einer Normal- bzw. Binomialverteilung berechnet werden. Syntax: normCDF(unterer Wert, oberer Wert, σ , μ) bzw. binomialCDF(x, Anzahl, Trefferwkt.) Bem: Die berechnete Wahrscheinlichkeit wird als numerischer Wert ausgegeben und in der System-Variable prob gespeichert.	 
invNormCDF x_γ mit P(X ≤ x_γ) = γ Funktion: Die Umkehrfunktion der N(μ, σ^2)-Verteilungsfunktion dient zur Berechnung der Intervallgrenze x_γ , so dass $\gamma = P(X ≤ x_\gamma)$ Syntax: invNormCDF(p , σ , μ) Bem: Die berechnete Intervallgrenze wird in der Systemvariablen x_1 Inv gespeichert.	
invBinomialCDF x_γ mit P(X ≤ x_γ) ≥ γ Funktion: Der invBinomialCDF-Befehl berechnet die kleinste Zahl x_γ , so dass $P(X ≤ x_\gamma) ≥ \gamma$ Syntax: invBinomialCDF(p,n,P) Bem: Die berechnete Intervallgrenze wird in der Systemvariablen xInv gespeichert.	