Einführung in Sage

Zusammenfassung Einheit 01

Hinweis: Textbausteine mit `<name>` weisen darauf hin, das anstatt des Ausdrucks eine passende Variable eingefügt werden muss.

Kurvendiskussion

- Deklarieren von Variablen mit

```
var('<varname>')
```

- Definieren von Variablen

```
<varname>=<value>
```

- Definieren von (mathematischen) Funktionen

```
<functionname>(<arguments>) = <expr>
```

- Grenzwertbestimmung:

```
<expr>.limit(x=<a>, dir='<plus|minus>')
```

- Bilden von Ableitungen

```
<expr>.differentiate(<variable>)
```

- Lösen von Gleichungen

```
solve( f(x)==0, x)
```

- Berechnen numerischer Approximationen

```
float(<expr>)
```

- Plotten einer Funktion

```
plot(<function>,(<lowerbound>,<upperbound>))
```

Symboliches Rechnen

- symbolisch Integrieren

```
integrate(<expr>,<variable>)
```

- numerisch Integrieren

```
integrate(<expr>,<variable>,<lower>,<upper>)
```

- faktorisieren

```
expand(<expr>)
```

- sortieren

```
<expr>.collect(x)
```

- partialbruchzerlegung

```
<expr>.partial_fraction()
```

- (vollständiges) Vereinfachen

```
<expr>.full_simplify()
```

- Vereinfachen mit radicals

```
<expr>.radical_simplify()
```

AGLA

- Matrix eingeben

```
matrix([ [z1s1,z1s2],[z2s1,z2s2] ])
```

- Vektor eingebe

```
vector([a,b,c])
```

- LGS lösen

```
A\b
```

- Matrixoperationen

```
A+B, A-B, A*B
```

- Matrix invertieren

```
A^(-1); A.inverse()
```

- Substitutieren

```
<expr>.subs(<variable>=<subs>)
```

Etwas Programmieren

- Listen (geordnet)

```
[a,b,c,..]
```

- Tuple

```
(a,b,c,..)
```

- (Nicht-mathematische) Funktionen

```
def <function>(<argument>): <befehle> return
     <rueckgabewert>
```

- Einzeilige Schleifen

```
[<expr(var)> for <var> in <range|liste> if <
    expr>]
```

Zahlentheorie

- Teiler

```
divisors(<number>)
```

- Anzahl Teiler

```
number_of_divisors(<number>)
```

- Primzahl-Überprüfung

```
is_prime(<number>)
```