

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- `limit()`

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

- Bilden von Ableitungen- `differentiate()`

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

- Lösen von Gleichungen- `solve()`:

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

- Berechnen numerischer Approximationen- `float()`

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

- Plotten einer Funktion- `plotting`

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

Symbolisches Rechnen

- Symbolisch Integrieren- `integrate()`

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

- Numerisch Integrieren

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

- Faktorisieren- `factor()`

```
factor(<expr>)
```

- Sortieren- `collect()`

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

- Partialbruchzerlegung- `partial_fraction()`

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

- (vollständiges) Vereinfachen- `simplify_full()`

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

- Vereinfachen mit radicals- `simplify_radical()`

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

AGLA

- Matrix eingeben- `matrix()`

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

- Vektor eingeben- `vector()`

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

- LGS lösen

```
A\b
```

- Matrixoperationen

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

- Matrix invertieren- `inverse()`

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

- Substituieren- `subs()`

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

Etwas Programmieren

- Listen (geordnet)- `list()`

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

- Tuple- `tuple()`

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

- (Nicht-mathematische) Funktionen- `def`

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

- Einzelige Schleifen- `for`

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

Zahlentheorie

- Teiler- `divisors()`

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

- Anzahl Teiler- `number_of_divisors()`

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

- Primzahl-Überprüfung- `is_prime()`

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