

Fläche(ninhalt) einer Wiese

Ein Rand der Wiese wird durch einen Fluss mit folgender Funktionsvorschrift gegeben

- `f := x --> x^2/4+2`

$$x \rightarrow \frac{x^2}{4} + 2$$

Genauer gesagt, soll der (blaue) Graph der Funktion f im Bereich $x=0$ bis $x=4$ den Fluss-Verlauf beschreiben

- `Fluss := plot::Function2d(f, x=0..4, LineColor = RGB::Blue)`

$$\text{plot::Function2d}\left(x \rightarrow \frac{x^2}{4} + 2, x = 0..4\right)$$

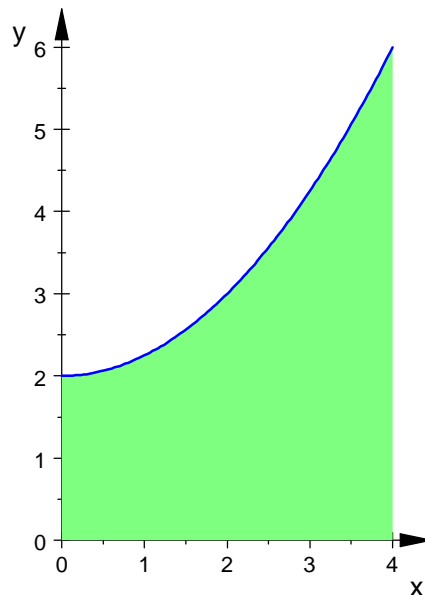
Die Wiese ist die (grün) schraffierte Fläche, die vom Fluss [und den dort schon genannten Grenzen $x=0$ bzw. $x=4$] sowie der x -Achse ($y=0$) berandet wird:

- `Wiese := plot::Hatch(Fluss, 0, FillColor = RGB::Green, Scaling=Constrained)`

$$\text{plot::Hatch}\left(\text{"Function2d}(x \rightarrow 1/4*x^2 + 2, x = 0..4)", 0, -\infty ..\infty\right)$$

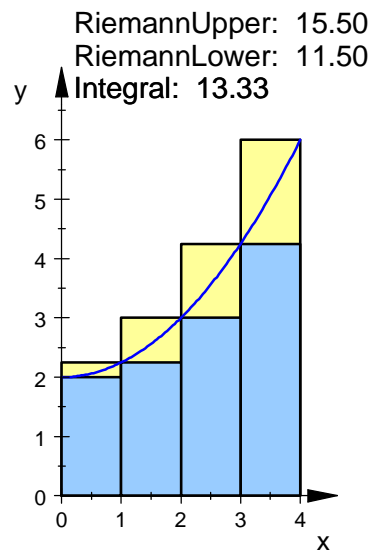
Jetzt werden Fluss und Wiese gezeichnet (*Nachträgliche Anmerkungen: 1. sollte es „und“ statt „uns“ heißen; 2. ist die grüne Fläche in Wirklichkeit schraffiert; in der aus MuPAD richtig exportierten rtf-Datei war sie noch schraffiert, erst bei meiner offenbar unzulänglichen Umwandlung in ein pdf-Dokument wurde die Fläche voll dargestellt*)

- `plot (Wiese, Fluss)`



Jetzt werden mit 4 gleichbreiten Rechtecken Werte für die Unter- bzw. Obersumme bestimmt. Leider wird auch schon automatisch das später erst gewünschte Integral angezeigt

- `plot (student::plotRiemann (f(x), x=0..4, 4), Scaling = Constrained)`



Jetzt soll die Rechnung allgemein für n Streifen durchgeführt werden, wobei n eine natürliche Zahl sein soll. MuPAD nennt für die angegebene Untersumme S_u direkt einen geschlossenen Ausdruck!

- `assume (n, Type::PosInt)`

$$\mathbb{Z} \cap (0, \infty)$$

- `Su := sum (4/n*f(i*4/n) , i=0..n-1)`

$$\frac{8 \cdot (5 \cdot n^2 - 3 \cdot n + 1)}{3 \cdot n^2}$$

Ebenso gibt's eine Summenformel für die Obersumme S_o mit n gleichbreiten Streifen:

- `So := sum (4/n*f(i*4/n) , i=1..n)`

$$\frac{40 \cdot n^2 + 24 \cdot n + 8}{3 \cdot n^2}$$

MuPAD berechnet auch die Grenzwerte von Unter- und Obersumme für n gegen Unendlich:

- `limit (Su, n = infinity)`

$$\frac{40}{3}$$

- `limit (So, n = infinity)`

$$\frac{40}{3}$$

Da beide Grenzwerte übereinstimmen, dürfte dies der wahre Wert der Fläche sein und wird als Integral definiert. Natürlich kann MuPad auch das Integral direkt berechnen und erhält den gleichen Wert:

- `int(f(x), x=0..4)`

Wegen dem automatischen Zusammenfassen von $n/4 * f(i*4/n)$ hinterm Summenzeichen ist hingegen die Angabe der Untersumme nach Riemann (trotz richtigem Ergebnis) ohne Nachrechnen kaum verständlich

- `Untersumme0 := student::riemann (f(x), x=0..4, 4, Left)`

$$\sum_{i3=0}^3 \left(\frac{i3^2}{4} + 2 \right)$$

- `float (Untersumme0)`

11.5

Erinnert werden soll daran, dass die vorstehenden 3 Seiten allein durch folgende Texte und Eingaben erzeugt wurden:

Fläche(ninhalt) einer Wiese

Ein Rand der Wiese wird durch einen Fluss mit folgender Funktionsvorschrift gegeben

- `f := x --> x^2/4+2`

Genauer gesagt, soll der (blaue) Graph der Funktion f im Bereich x=0 bis x=4 den Fluss-Verlauf beschreiben

- `Fluss := plot::Function2d(f, x=0..4, LineColor = RGB::Blue)`

Die Wiese ist die (grün) schraffierte Fläche, die vom Fluss [und den dort schon genannten Grenzen x=0 bzw. x=4] sowie der x-Achse (y=0) berandet wird:

- `Wiese := plot::Hatch(Fluss, 0, FillColor = RGB::Green, Scaling=Constrained)`

Jetzt werden Fluss und Wiese gezeichnet

- `plot (Wiese, Fluss)`

Jetzt werden mit 4 gleichbreiten Rechtecken Werte für die Unter- bzw. Obersumme bestimmt. Leider wird auch schon automatisch das später erst gewünschte Integral angezeigt

- `plot (student::plotRiemann (f(x), x=0..4, 4), Scaling = Constrained)`

Jetzt soll die Rechnung allgemein für n Streifen durchgeführt werden, wobei n eine natürliche Zahl sein soll.

MuPAD nennt für die angegebene Untersumme Su direkt einen geschlossenen Ausdruck!

- `assume (n, Type::PosInt)`
- `Su := sum (4/n*f(i*4/n) , i=0..n-1)`

Ebenso gibt's eine Summenformel für die Obersumme So mit n gleichbreiten Streifen:

- `So := sum (4/n*f(i*4/n) , i=1..n)`

MuPAD berechnet auch die Grenzwerte von Unter- und Obersumme für n gegen Unendlich:

- `limit (Su, n = infinity)`
- `limit (So, n = infinity)`

Da beide Grenzwerte übereinstimmen, dürfte dies der wahre Wert der Fläche sein und wird als Integral definiert.

Natürlich kann MuPad auch das Integral direkt berechnen und erhält den gleichen Wert:

- `int(f(x), x=0..4)`

Wegen dem automatischen Zusammenfassen von $n/4 * f(i*4/n)$ hinterm Summenzeichen ist hingegen die Angabe der Untersumme nach Riemann (trotz richtigem Ergebnis) ohne Nachrechnen kaum verständlich

- `Untersumme0 := student::riemann (f(x), x=0..4, 4, Left)`
- `float (Untersumme0)`

Die blauen Ausgaben und Graphen auf den vorangegangenen Seiten wurden dazu von MuPAD erzeugt.

Wie auch bei allen anderen CAS-Programme muss man natürlich die Syntax beachten. Eine dem menschlichen Leser verständliche Abkürzung mit ".." für verschiedene Terme wird naturgemäß leider nicht verstanden (auch wenn ".." z.B. beim Integrationsbereich $x=0..4$ als "bis" gedeutet wird).

- `Untersumme1 := 4/n * (f(0*4/n)+f(1*4/n)+f(2*4/n)+...+f((n-1)*4/n)`
`Untersumme1 := 4/n * (f(0*4/n)+f(1*4/n)+f(2*4/n)+...+`
^ Error: '::' expected [line 1, col 52]

Zu allen Befehlen gibt's in MuPAD umfangreiche Hilfe mit kopierbaren Beispielen. So führt z.B.

- `?hatch`

zum abgebildeten Hilfefenster (mit gelb unterlegten Verweisen):

plot::Hatch - eine schraffierte Fläche

plot::Hatch(f) schraffiert die Fläche zwischen der Funktion f und der x-Achse.

plot::Hatch(f, base) schraffiert die Fläche zwischen der Funktion f und der horizontalen Linie $y = \text{base}$.

plot::Hatch(f, g) schraffiert die Fläche zwischen den beiden Funktionen f und g.

plot::Hatch(c) schraffiert die von der Kurve c umschlossene Fläche .

Elementkonstruktor(en):

```

plot::Hatch(f1, f2 <, x1..x2> <, a = amin .. amax> <, attr, ...>)
plot::Hatch(f1 <, base> <, x1..x2> <, a = amin .. amax> <, attr, ...>)
plot::Hatch(c <, a = amin .. amax> <, attr, ...>)

```

Parameter:

f1, f2 — die die Schraffur begrenzende(n) Funktion(en): Objekte vom Typ `plot::Function2d`. Entsprechen den Attributen `Function1`, `Function2`.

c — die die Schraffur begrenzende Kurve: eine parametrisierte Kurve vom Typ `plot::Curve2d`. Entspricht dem Attribut `Function1`.

base — die die Schraffur begrenzende Grundlinie: ein reeller numerischer Wert oder ein arithmetischer Ausdruck im Animationsparameter a.. Entspricht dem Attribut `Baseline`.

x1..x2 — ein Bereich auf der x-Achse, der die Schraffur nach links und rechts begrenzt: numerische reelle Werte oder arithmetische Ausdrücke des Animationsparameters a. Entsprechen den Attributen `IMin`, `IMax`, `IRange`.

a — der Animationsparameter: ein Bezeichner oder ein indizierter Bezeichner. Entspricht dem Attribut `ParameterName`.

amin .. amax — der Animationsbereich: amin, amax müssen reelle numerische Werte sein. Entspricht dem Attribut `ParameterRange`.

attr, ... — Optionen, also Gleichungen `Attribut = Wert`, wobei `Attribut` eines der unten angegebenen oder ein „Hint“ ist. Abschnitt 7 erläutert die Einzelheiten.

Verwandte Domains: `plot::Curve2d`, `plot::Function2d`

Verwandte Funktionen: `plot`, `plot::copy`

Zu den **Beispielen**

Seite 120

Dieses pdf-Dokument ist Teil einer Besprechung verschiedener CAS-Programme (hier von MuPAD 3.1 Pro) in meinem Webangebot www.r-krell.de