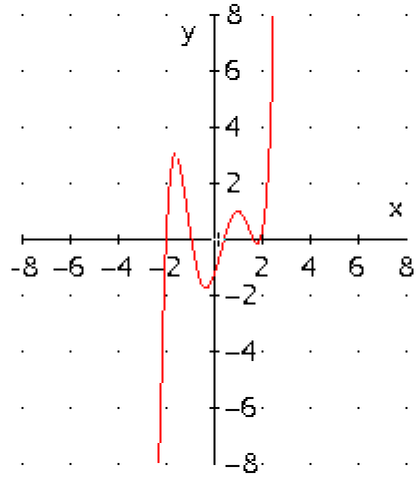
#1: Hinweise zum Sekantenverfahren von hbm

#2: --

#3: Berechnen Sie eine Nullstelle mit der regula falsi.

#4: $$f(x) := \frac{1}{2}\cdot(x - 2)\cdot(x - 1.5)\cdot(x - 0.5)\cdot(x + 1)\cdot(x + 2) + \frac{1}{4}$$



#5: --

#6: Sekantenformel:

#7: $$newx3(x1, x2) := x1 - \frac{f(x1)\cdot(x2 - x1)}{f(x2) - f(x1)}$$

#8: --

#9: Ich vermute, dass ungefähr bei −2 eine Nullstelle liegt.

#10: x1 := −2.2

#11: x2 := −1.8

#12: SIGN(f(x1)) = −1

#13: SIGN(f(x2)) = 1

#14: Die Vorzeichen von f(x1) und f(x2) sind also verschieden. F(x1) ist negativ, f(x2) ist positiv.

#15: x3 := newx3(x1, x2)

#16: Zur Kontrolle approximiere ich den Wert von x3, definiere ihn aber nicht als Dezimalzahl. Und ich berechne den Funktionswert, denn der soll sich ja an null nähern.

```
#17:  [x3, f(x3)]

#18:                        [-1.939321631, 1.191849525]

#19:  ----------------------------------------

#20:  Ich sehe, dass f(x3) positiv ist. Da f(x1) negativ war und noch
        ist, kommen nur x1 und x3 in Frage. Zur Sicherheitprüfe ich das:

#21:                     SIGN(f(x1)) ≠ SIGN(f(x3)) = true

#22:                     SIGN(f(x2)) ≠ SIGN(f(x3)) = false

#23:  Also war die Wahl von x1 und x3 richtig:

#24:  x4 ≔ newx3(x1, x3)

#25:  [x4, f(x4)]

#26:                       [-1.991304112, 0.3996304761]

#27:  ----------------------------------------

#28:  F(x4) ist immer noch positiv:

#29:                     SIGN(f(x1)) ≠ SIGN(f(x4)) = true

#30:  x5 ≔ newx3(x1, x4)

#31:  [x5, f(x5)]

#32:                       [-2.007390481, 0.1188076944]

#33:  ----------------------------------------

#34:                     SIGN(f(x1)) ≠ SIGN(f(x5)) = true

#35:  x6 ≔ newx3(x1, x5)

#36:  [x6, f(x6)]

#37:                       [-2.012056994, 0.03404057072]

#38:  ----------------------------------------

#39:                     SIGN(f(x1)) ≠ SIGN(f(x6)) = true

#40:  x7 ≔ newx3(x1, x6)

#41:  [x7, f(x7)]

#42:                       [-2.01338459, 0.009649619997]

#43:  ----------------------------------------

#44:                     SIGN(f(x1)) ≠ SIGN(f(x7)) = true
```

```
#45:  x8 ≔ newx3(x1, x7)

#46:  [x8, f(x8)]

#47:                    [-2.013760172, 0.002727123931]

#48:  ----------------------------------------

#49:                   SIGN(f(x1)) ≠ SIGN(f(x8)) = true

#50:  x9 ≔ newx3(x1, x8)

#51:  [x9, f(x9)]

#52:                    [-2.013866256, 0.0007700635326]

#53:  ----------------------------------------

#54:  Jetzt reicht es mir.

#55:  Man sieht, dass in diesem Fall die Annäherung nur von rechts
        erfolgt.

#56:  Kontrolle mit Derive:

#57:  NSOLVE(f(x) = 0, x, -2.5, -1.8)

#58:                           x = -2.013907987

#59:  ----------------------------------------
```