Adaptive Systeme - Hausaufgabe 1

Henry Fock

Für reelle Konstanten a, b mit $0.25 \leq a, b \leq 0.75 \in \mathbb{R}$ und $a + b = 1$ lautet die zu optimierende Zielfunktion

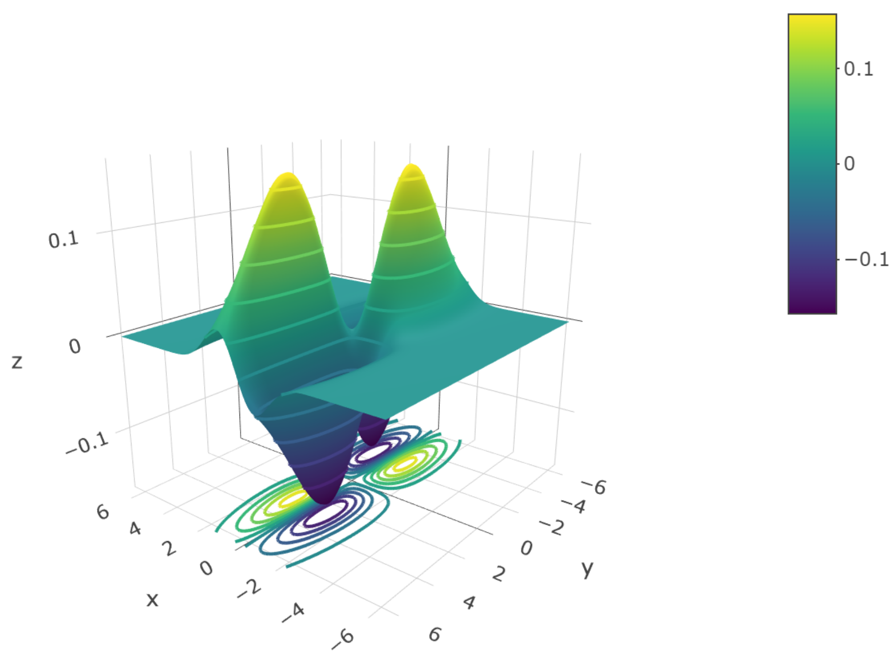
$$\begin{aligned} f &: \mathbb{R}^2 \to \mathbb{R} \\ (x, y)^T &\mapsto f(x, y) = \sin(ax)\sin(by)\exp\left(-\left(a^2x^2 + b^2y^2\right)\right) \end{aligned}$$

Aufgabe 1

Visualisieren Sie die Funktion mittels Matlab, Octave oder Gnuplot im Bereich

$$U = \left\{(x, y)^T \in \mathbb{R}^2 | \text{ mit } \left\|(x, y)^T\right\|_\infty < 2\pi\right\}$$

```
f <- expression(sin(a*x)*sin(b*y)*exp(-(a^2 * x^2 + b^2 * y^2)))
f.func <- function(x,y,a = 0.75) {
  b <- 1 - a
  return(eval(f))
}
```



Aufgabe 2

Berechnen Sie den Gradientenvektor ∇f und stellen Sie die Bedingungsgleichungen für die Extremstellen von f auf. Die globalen Maxima und Minima werden innerhalb des Bereichs U angenommen, da f auf Grund der dämpfenden Exponentialfunktion sehr schnell abfällt.

Antwort:

$$\begin{aligned}\frac{\partial f}{\partial x} &= a\sin(by)\exp\left(-a^2x^2 - b^2y^2\right)(\cos(ax) - \sin(ax)2ax)\\ \frac{\partial f}{\partial y} &= b\sin(ax)\exp\left(-a^2x^2 - b^2y^2\right)(\cos(by) - \sin(by)2by)\end{aligned}$$

Die Bedingungsgleichungen lauten:

$$\begin{aligned}0 &= \frac{\partial f}{\partial x}\\ 0 &= \frac{\partial f}{\partial y}\end{aligned}$$

Aufgabe 3

Die Bedingungsgleichungen aus Teil 2 lassen sich durch geeignete Umformung in eine einzelne äquivalente eindimensionale Gleichung $h_f(u)$ überführen, aus deren Nullstellen u_* sich auch die von $\nabla f(x, y)$ ergeben. Wie lautet die Gleichung h_f?

Antwort: Da die Gleichungen von ∇f symmetrisch aufgebaut sind, werden alle Vorkommnisse von ax und by mit u substituiert:

$$\begin{aligned}0 &= a\sin(u)\exp\left(-2u^2\right)(\cos(u) - \sin(u)2u)\\ 0 &= b\sin(u)\exp\left(-2u^2\right)(\cos(u) - \sin(u)2u)\end{aligned}$$

Addieren der beiden Gleichungen ergibt:

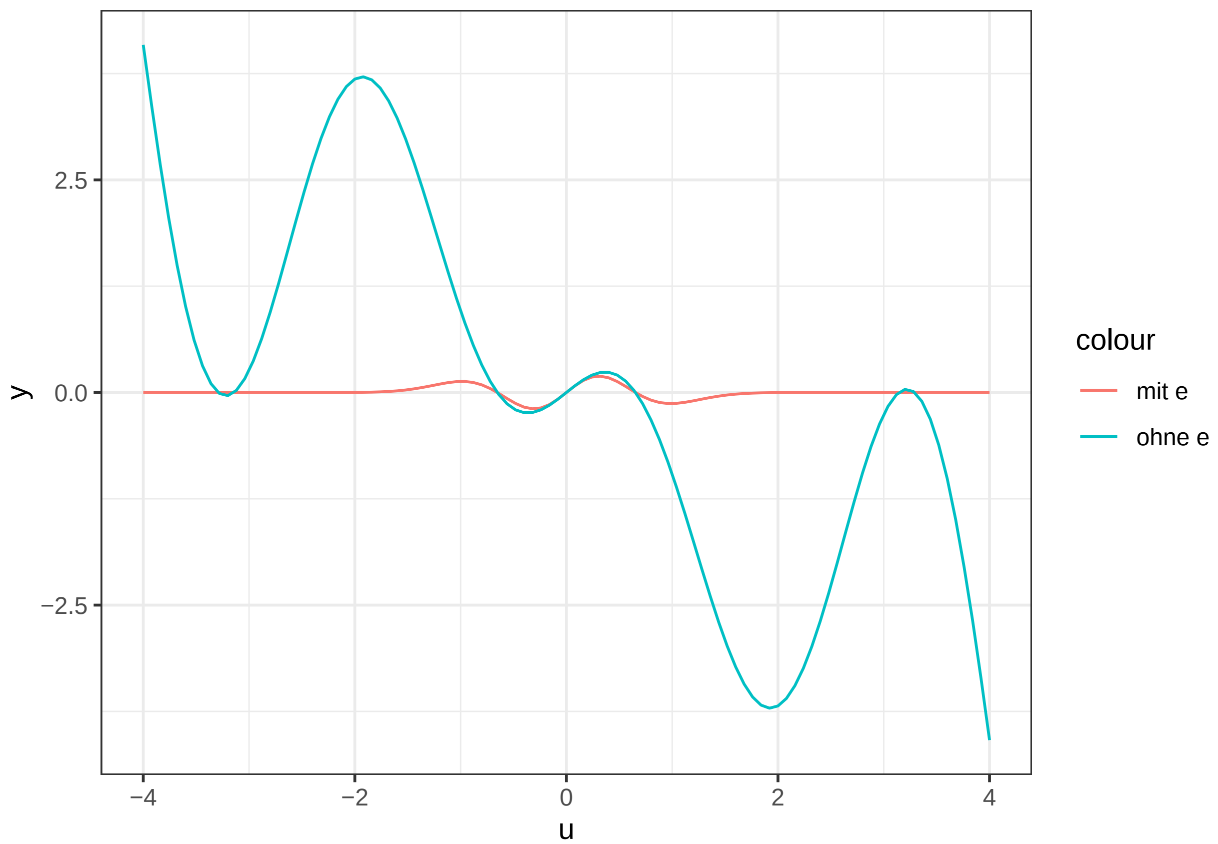
$$\begin{aligned}0 &= (a+b)\left(\sin(u)\exp\left(-2u^2\right)(\cos(u) - \sin(u)2u)\right)\\ &= \sin(u)\exp\left(-2u^2\right)(\cos(u) - \sin(u)2u)\end{aligned}$$

Da $a + b = 1$ ist, ergibt sich als Gleichung für h_f:

$$h_f(u) = \sin(u)\exp\left(-2u^2\right)(\cos(u) - \sin(u)2u)$$

Aufgabe 4

Visualisieren Sie die Funktion $h_f(u)$ auf dem Intervall $|u| < 4$ einmal mit und einmal ohne den dämpfenden Exponentialfaktor



Aufgabe 5

Bestimmen Sie die globalen Extrema von $f(x, y)$ auf 10 Stellen genau mit einem geeigneten Iterationsverfahren

Antwort: Implementieren des Newton-Verfahrens:

```
find.zeros <- function(start.vec, expr, namevec, precision = 1e-10, max.iter = 20) {
  grads <- deriv(expr, namevec, function.arg = T)

  xk <- start.vec
  xkp <- start.vec
  delta <- Inf
  k <- 0

  while(delta > precision && k < max.iter) {
    grad.calc <- do.call(grads, as.list(xk))
    gradient <- t(attr(grad.calc, "gradient"))
    xkp <- xk - grad.calc / gradient

    delta <- norm(xk - xkp)
    xk <- xkp
    k <- k+1
  }
```

```
  return(xkp)
}
```

Suche Nullstellen der eindimensionalen Funktion $h_f(u)$:

```
h <- expression(sin(u)*exp(-2*u^2)*(cos(u)-sin(u)*2*u))
start.vec <- matrix(c(0.5))
h.zero <- find.zeros(start.vec, h, c("u"))[[1]]
h.zero
```

```
## [1] 0.6532711871
```

Die Nullstellen von $h_f(u)$ befinden sich an $u_* = 0.6532711871$ und -0.6532711871, da die Funktion punktsymmetrisch ist. Eine weitere (triviale) Nullstelle befindet sich an $u_* = 0$.

Als nächstes muss die Substitution aufgelöst werden, um Werte für x und y zu erhalten. Es gilt $x_* = \frac{u_*}{a}$ und $y_* = \frac{u_*}{b}$. Daraus ergeben sich die Nullstellen von ∇f.

Aufgabe 6

Zeigen Sie, ob es sich bei den Extrema um Minima oder Maxima handelt, mittels der Auswertung der Hessematrix H_f.

Antwort: Wähle für $a = 0.75$ und $b = 0.25$. Berechne Nullstellen für x un y als:

$$x_0 = \frac{\pm 0.6532711871}{0.75} = \pm 0.8710282495$$
$$y_0 = \frac{\pm 0.6532711871}{0.25} = \pm 2.6130847484$$

Die Hessematrizen ergeben sich aus der Kombination der positiven und negativen Nullstellen, sowie der Stelle $(0,0)^T$.

$$H_f(0.8710282495, 2.6130847484) = \begin{pmatrix} -0.416574819 & -6.9388939039 \times 10^{-18} \\ -6.9388939039 \times 10^{-18} & -0.046286091 \end{pmatrix}$$
$$H_f(-0.8710282495, 2.6130847484) = \begin{pmatrix} 0.416574819 & -6.9388939039 \times 10^{-18} \\ -6.9388939039 \times 10^{-18} & 0.046286091 \end{pmatrix}$$
$$H_f(0.8710282495, -2.6130847484) = \begin{pmatrix} 0.416574819 & -6.9388939039 \times 10^{-18} \\ -6.9388939039 \times 10^{-18} & 0.046286091 \end{pmatrix}$$
$$H_f(-0.8710282495, -2.6130847484) = \begin{pmatrix} -0.416574819 & -6.9388939039 \times 10^{-18} \\ -6.9388939039 \times 10^{-18} & -0.046286091 \end{pmatrix}$$
$$H_f(0,0) = \begin{pmatrix} 0 & 0.1875 \\ 0.1875 & 0 \end{pmatrix}$$

Als nächstes werden die Abschnittsdeterminanten berechnet und die definitheit bestimmt. Diese sind wie folgt:

```
## Nullstelle:
##              [,1]
## [1,] 0.8710282495
## [2,] 2.6130847484
##
## Hessian:
##                  [,1]              [,2]
## [1,] -4.165748190e-01 -6.938893904e-18
## [2,] -6.938893904e-18 -4.628609100e-02
##
## Abschnittsdeterminanten:
## [1] -0.41657481903  0.01928161998
##
## negativ definit!
## ----------
## Nullstelle:
##               [,1]
## [1,] -0.8710282495
## [2,]  2.6130847484
##
## Hessian:
##                  [,1]              [,2]
## [1,]  4.165748190e-01 -6.938893904e-18
## [2,] -6.938893904e-18  4.628609100e-02
##
## Abschnittsdeterminanten:
## [1] 0.41657481903 0.01928161998
##
## positiv definit!
## ----------
## Nullstelle:
##               [,1]
## [1,]  0.8710282495
## [2,] -2.6130847484
##
## Hessian:
##                  [,1]              [,2]
## [1,]  4.165748190e-01 -6.938893904e-18
## [2,] -6.938893904e-18  4.628609100e-02
##
## Abschnittsdeterminanten:
## [1] 0.41657481903 0.01928161998
##
## positiv definit!
## ----------
## Nullstelle:
##               [,1]
## [1,] -0.8710282495
## [2,] -2.6130847484
##
## Hessian:
##                  [,1]              [,2]
## [1,] -4.165748190e-01 -6.938893904e-18
## [2,] -6.938893904e-18 -4.628609100e-02
```

```
##
## Abschnittsdeterminanten:
## [1] -0.41657481903  0.01928161998
##
## negativ definit!
## ----------
## Nullstelle:
##      [,1]
## [1,]    0
## [2,]    0
##
## Hessian:
##        [,1]   [,2]
## [1,] 0.0000 0.1875
## [2,] 0.1875 0.0000
##
## Abschnittsdeterminanten:
## [1]  0.00000000 -0.03515625
##
## indefinit!
## ----------
```

Optionale Lösung von $f(x, y)$ mittels Newton-Verfahren für mehrdimensionale Gleichungen

Implementierung der mehrdimensionalen Optimierung mittels Newton-Verfahren:

```
find.extrema <- function(start.vec, grads, precision = 1e-10, max.iter = 20, rate = 0.5) {
  xk <- start.vec
  xkp <- start.vec
  delta <- Inf
  k <- 0

  while(delta > precision && k < max.iter) {
    grad.calc <- do.call(grads, as.list(xkp))
    grad <- t(attr(grad.calc, "grad"))
    hesse <- matrix(attr(grad.calc, "hessian"), nrow = length(start.vec))

    hesse.inv <- solve(hesse)

    # Updaterate musste verringert werden, da die Schritte zu groß für die Funktion wurden
    xkp <- xk - rate * (hesse.inv %*% grad)
    delta <- norm(xk - xkp)
    xk <- xkp
    k <- k+1
  }

  return(round(xkp, digits = 10))
}
```

Implementiere Funktion zum überprüfen der Hessematrix:

```
check.hesse <- function(extrema.vec, grads) {
  grad.calc <- do.call(grads, as.list(extrema.vec))
```

```
  hesse <- matrix(attr(grad.calc, "hessian"), nrow = length(extrema.vec))

  section.det <- vector("double", nrow(hesse))
  for (i in seq_len(nrow(hesse))) {
    section <- round(as.matrix(hesse[1:i, 1:i]), digits = 10)
    section.det[i] <- det(section)
  }

  definit.pos <- all(section.det > 0)
  definit.neg <- all(section.det[seq(1, length(section.det), 2)] < 0) &&
    all(section.det[seq(2, length(section.det), 2)] > 0)

  if (definit.pos)
    return("positiv definit")
  else if(definit.neg)
    return("negativ definit")
  else
    return("indefinit")
}
```

Festlegen der Parameter:

```
f <- expression(sin(a*x)*sin(b*y)*exp(-(a^2 * x^2 + b^2 * y^2)))
a <- 0.75
b <- 1 - a
vars <- c("x", "y")
start.vec <- matrix(c(1,1))
```

Suchen eines Extrema:

```
f.gradients <- deriv3(f, vars, function.arg = T)
(f.extrema <- find.extrema(start.vec, f.gradients, rate = 0.5))
```

```
##              [,1]
## [1,] 0.8710280877
## [2,] 2.6130838131
```

```
(max_min <- check.hesse(f.extrema, f.gradients))
```

```
## [1] "negativ definit"
```

Extrema wurde an $x = 0.8710280877$ und $y = 2.6130838131$ gefunden. Die Hessematrix ist an der gefundenen Stelle negativ definit.