Lösen von Gleichungssystemen und symbolische Gleichungen in R

Kolja Hopfmann
Betreuer: Eugen Betke

Universität Hamburg
Informatik
PIR 2016

08/06/2016

informatik
die zukunft

Inhaltsverzeichnis

Definition

„In Computeralgebrasystemen(CAS) bedeutet der Ausdruck symbolische Mathematik, dass Operationen und Kalkulationen von mathematischen Ausdrücken mit Variablen auf Computern ausgeführt werden.“ [Planetmath.org]

- CAS: Programm für Verarbeitung symbolischer Ausdrücke.
- Variable: Symbol für Wert aus einer Wertemenge.

Symbolische Manipulationen

- Lösung eines Gleichungssystems(GLS) bilden.
- Matrix-Operationen.
- Arithmetische Operationen auf Polynomen.
- Vereinfachen von komplizierten symbolischen Ausdrücken in eine Standardform.
- Wertzuweisung der Variablen und Berechnung.
- Ableiten von symbolischen Ausdrücken.
- Bilden von Integralen.

Symbolische Manipulation in R[1]

R als CAS?

Symbolische Manipulationen in R[2]

- Wir behandeln folgende Pakete zur symbolischen Mathematik:
  - Solve
  - mpoly
  - Simplr
  - D
  - Integrate

Schnittstellen zu Algebrasystemen

- Ryacas [CRAN:Ryacas]
  - Yacas: alleinstehendes opensource-CAS.
- RSymPy[Sympy.org]
  - SymPy: CAS Library unter Python.

Inhaltsverzeichnis

Vorstellung

Build-in package in R

- Funktionen:
  - Lösen von Gleichungssystemen in Matrix-form.
  - Invertieren einer Matrix.
- Syntax:
  - solve(a,b)
  - Mit a als Matrix und b als Matrix/Vektor
  - Ist b nicht gegeben, so wird die Matrix a invertiert.

Beispiele[1]

- Lösen eines Gleichungsystems.

$$1x + 2y = 5$$
$$3x + 4y = 6$$

$$\begin{bmatrix} 1 & 2 & 5 \\ 3 & 4 & 6 \end{bmatrix}$$

```
1  >A = matrix(c(1,2,3,4),nrow=2,ncol=2,byrow=T)
2  >B  = c(5,6)
3  >solve(A,B)
4  [1] -4.0  4.5
```

Beispiele[2]

- Invertieren einer Matrix.

$$A \times A^{-1} = I$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \times A^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

```
1  >A = matrix(c(1,2,3,4),nrow=2,ncol=2,byrow=T)
2  >solve(A)
3    [,1] [,2]
4 [1,] -2.0  1.0
5 [2,]  1.5 -0.5
```

Beispiele[3]

- Fehlermeldung bei nichtlösbarem GLS.

$$1x + 2y = 5$$
$$0x + 0y = 4$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 4 \end{bmatrix}$$

```
1  >A = matrix(c(1,2,0,0),nrow=2,ncol=2,byrow=T)
2  >solve(A,c(3,4))
3  Error in solve.default(A, c(3, 4))
```

Vor-/Nachteile

- Vorteile:
  - Bequemliche Lösung zum behandeln von GLS und Invertieren.
  - Fehlerminimierung.
- Nachteile:
  - Keine Angabe von Lösungsmengen.
  - Bei großen GLS wird die Ausgabe unübersichtlich.

Inhaltsverzeichnis

Vorstellung

Package in R unter 3.2.5.

- Funktionen:
  - Einführung von mp als Objekt für Polynome.
  - Rechenoperationen zum Bearbeiten der Polynome.
  - Umwandeln der Polynome in Funktionen.

mp

Eingabe eines String der zu einem Polynom übersetzt wird.

- Syntax:
  - mp(string)

```
1  > f = mp("x^3+x^2+x+1")
2  >f
3  x^3 + x^2 + x + 1
```

mp Arithmetik

- Rechenoperationen auf mp Objekten:

```
>f-f
0
>f+f
2 x^3 + 2 x^2 + 2 x + 2
>f*f
x^6 + 2 x^5 + 3 x^4 + 4 x^3 + 3 x^2 + 2 x + 1
```

- Abfragen auf Gleichheit:

```
>f==f
[1] TRUE
>f == mp("x")
[1] FALSE
```

Listen

Möglichkeit mp Objekte in Listen zu verwalten.

- Syntax:
  - mpolyList(...)
  - Übergabe von mp Objekten als Parameter.

```
1 >f = mp("x^3+x^2+x+1")
2 >g = mp("x^2")
3 >h = mp("2y")
4 >k = mpolyList(f,g,h)
5 >k
6 x^3 + x^2 + x + 1
7 x^2
8 2 y
```

Listen-Arithmetik[1]

Rechenoperationen auf mpListen:

- Wie mit Vektoren in R
- Das i-te Element der 1. Liste wird mit dem i-ten Element der 2. Liste verknüpft.

```
1 >k+k
2 2 x^3 + 2 x^2 + 2 x + 2
3 2 x^2
4 4 y
5 >k-k
6 0
7 0
8 0
```

Listen-Arithmetik[2]

Rechenoperationen auf mpListen:

- Wie mit Vektoren in R
- Das i-te Element der 1. Liste wird mit dem i-ten Element der 2. Liste verknüpft.

```
1 >k*k
2 x^6 + 2 x^5 + 3 x^4 + 4 x^3 + 3 x^2 + 2 x + 1
3 x^4
4 4 y^2
```

as.function

Umwandlung von Polynomen zu Funktionen.

- Syntax:
  - as.function(x)
  - x als mp Objekt.

$$f(x) = x^2$$
$$f(0) = 0$$
$$f(2) = 4$$

```
1 >f = as.function(mp(x^2))
2 f(.) with . = x
3 >f(0)
4 [1] 0
5 >f(2)
6 [1] 4
```

deriv

Ableiten einer ganzrationalen Funktion:

- Syntax:
  - deriv(mp,x)
  - mp als beliebiges mp Objekt
  - x als abzuleitende Variable.

$$f(x) = x^3 + x^2 + x + 1$$
$$f'(x) = 3x^2 + 2x + 1$$

```
1 >f = mp("x^3+x^2+x+1")
2 >deriv(f, 'x')
3 3 x^2 + 2 x + 1
```

weitere Funktionen

Zur Untersuchung von Polynomen und Funktionen:

- totaldeg(x)
  - Gibt den Grad des Polynoms an.
- monomials(x)
  - Gibt die Mononome des Polynoms aus.

```
1 >f = mp("x^3+x^2+x+1")
2 >totaldeg(f)
3 [1] 3
4 >monomials(f)
5 x^3
6 x^2
7 x
8 1
```

Vor-/Nachteile

- Vorteile:
  - Großer Umfang an Funktionen bezüglich grundlegender Analysis
  - Praktisches einsetzen und Auslesen von Funktionen.
  - Eigene Exemplare für Polynome und Funktionen.
- Nachteile:
  - Behandelt nur ganzrationale Funktionen.

Vorstellung

Paket in R zur Vereinfachung von Polynomen.

- Syntax:
  - simplifyq(expression)
  - expression als zu vereinfachender Ausdruck.

Beispiele

- Koeffizient mit 0: direkte Ausgabe von 0.
- Vereinfachen von Trigonometrischen Ausdrücken.
- Bilden einer Standardform.

```
1 >simplifyq((a+b)*0)
2 [1] 0
3 >simplifyq(sin(x)^2+cos(x)^2)
4 [1] 1
5 simplifyq(a+a+a+b+b+b+c+c+c)
6 [1] 3 * c + (3 * b + 3 * a)
```

Vor-/Nachteile

- Vorteile:
  - Überführung von großen Polynomen in eine Standardform.
  - Trigonometrischen Funktionen und Exponentiale Funktionen.
- Nachteile:
  - simplifyq als einzige Operation.
  - Keine Verknüpfung mit mpoly.

Vorstellung

Built-in function in R

- Deklarieren von Ausdrücken
- Ableiten der Ausdrücke
- Syntax:
  - f = expression(x1, 'x2')
  - x1 als Eingabe für den Ausdruck.
  - x2 als Parameter für die abhängige Variable.
  - D(f,' x')
  - f als Polynom in Form eines Ausdrucks.
  - 'x' als abzuleitende Variable.

Beispiele[1]

- Ableiten einer ganzrationalen Funktion:

$$f(x) = x^3 + x^2 + x + 1$$
$$f'(x) = 3x^2 + 2x + 1$$

```
1 >fx = expression("x^3+x^2+x+1", 'x')
2 >fx
3 fx = expression("x^3+x^2+x+1, 'x')
```

Beispiele[2]

- Ableiten einer trigonometrischen Funktion:

$$h(x) = sin(x^2)$$
$$h'(x) = 2x * cos(x^2)$$

```
>D(fx, 'x')
3 * x^2 + 2 * x + 1
>hx = expression(sin(x^2),'x')
>D(hx,'x')
cos(x^2) * (2 * x)
```

Beispiele[3]

- Ableiten einer exponential Funktion:

$$g(x) = e^x$$
$$g'(x) = e^x$$

```
1 >gx = expression(e^x,'x')
2 >D(gx,'x')
3 e^x * log(e)
```

Vor-/Nachteile

- Vorteile:
  - Built-in
  - Exponentialfunktionen und trigonometrische Funktionen.
- Nachteile:
  - Ableiten und Arithmetik als einzige Funktion.
  - Integrieren nicht möglich, weitere Pakete vonnöten.

Vorstellung

Paket in R zur Bestimmung von Integralen eindimensionaler Funktionen.

- Benutzt die Gauß-Quadratur als Annäherungsverfahren. [math.ethz.ch]
- (vereinfachte) Syntax:
  - integrate(f, upper, lower)
  - f als Eingabe für eine Funktion.
  - upper und lower als Grenzen des gewünschten Intervalls.
  - Unbegrenzte Intervalle möglich mit dem Parameter int/-inf.

Beispiele

```
>integrate(dnorm, -1.96, 1.96)
>integrate(dnorm, -Inf, Inf)
## a slowly-convergent integral
>f<- function(x) {1/((x+1)*sqrt(x))}
>integrate(f, lower = 0, upper = Inf)
## don't do this if you really want the integral from
    0 to Inf
integrate(integrand, lower = 0, upper = 10)
integrate(integrand, lower = 0, upper = 100000)
integrate(integrand, lower = 0, upper = 1000000, stop.
    on.error = FALSE)
## fails on many systems
integrate(dnorm, 0, 20000)
integrate(dnorm, 0, Inf)    ## works
```

[math.ethz.ch]

Vor-/Nachteile

- Vorteile:
  - Effiziente Lösung von komplexen Integralen.
  - Möglichkeit zur bestimmung von unbegrenzten Integralen.
- Nachteile:
  - Hohe Fehlerquote bei enorm großen Intervallen.

Inhaltsverzeichnis

Vorstellung

Yacas: Yet Another Computer Algebra System.

- Open-Source unter GPL(GNU).
- Eingebettet in eigener Programmiersprache.
- Funktionen zum Plotten über Gnuplot.
- Ryacas als Interface/Schnittstelle zu Yacas in R.
  - Kompletter funktionaler Zugriff auf Yacas als CAS in der R-Umgebung.

Beispiele[1]

R expressions:

- Call an Yacas durch eine R expression
- Achtung: Rückgabewert als Yacas-Objekt.

```
1 > f = yacas(expression(Factor(x^3+x^2+x+1))) #Yacas-
    Funktionsaufruf.
2 > f
3 expression((x^2 + 1) * (x + 1))
4 > Eval(f,list(x=1))
5 [1] 4
6 > Eval(f,list(x=0))
7 [1] 1
```

Beispiele[2]

Sym objects:

- Sym als Klasse zur Bestimmung von Symbolen.
- Ermöglicht Umgang als Funktion.

```
x = Sym("x")
f = function(x){(x^3+x^2+x+1)}
> f(0)
[1] 1
> f(1)
[1] 4
> f(42)
[1] 75895
> g = yacas(deriv(f(x), x)) #Yacas-Funktionsaufruf.
> g
expression(3 * x^2 + 2 * x + 1)
```

Beispiele[3]

integrate:

- Integrieren von Funktionen.
- Trigonometrisch:

$$f(x) = -x^2 + 1$$

$$\int_{-1}^{1} f(x)\,\mathrm{d}x = 1,3\overline{33}$$

```
1 > f = function(x)-x^2+1
2 > integrate(f,-1,1)
3 1.333333 with absolute error < 1.5e-14
```

Beispiele[4]

integrate:

- Integrieren von Funktionen.
- Ganzrational:

$$g(x) = sin(x)$$

$$\int_0^{\pi} g(x)\,\mathrm{d}x = 2$$

```
1 > g = function(x)sin(x)
2 > g
3 function(x)sin(x)
4 > integrate(g,0,pi)
5 2 with absolute error < 2.2e-14
```

Pretty Printing

Funktion in Ryacas zum formatieren:

- ASCII
- LaTeX

```
> h = expression(x + x^2/2 + x^3/6 + 1)
> PrettyForm(h)

      2     3
     x     x
x + -- + -- + 1
     2     6
>TeXForm(h)
expression("$x + \frac{x ^{2}}{2}  + \frac{x ^{3}}{6}
     + 1$")
```

Pretty Printing

```
1 > h = expression(x + x^2/2 + x^3/6 + 1)
2 >TeXForm(h)
3 expression("$x + \frac{x ^{2}}{2}  + \frac{x ^{3}}{6}
      + 1$")
```

Übersetzt in LaTeX:

- $x + \frac{x^2}{2} + \frac{x^3}{6} + 1$

Vorstellung

SymPy: CAS Python-Library

- Open-Source unter GPL(GNU).
- Direktes Ansprechen der Python Umgebung.
- Syntax in Python.

Inhaltsverzeichnis

Zusammenfassung

- Wir haben behandelt:
  - Lösen von Gleichungssystemen mit Solve
  - Rechenoperationen auf Polynommen und Funktionen mit mpoly.
  - Vereinfachung von Polynomen mit simplr
  - Ableiten von Funktionen mit deriv.
  - Bilden von Integralen mit Integrate.
  - Verfügbare CAS Schnittstellen in R mit weiteren Funktionen.

Symbolische Manipulationen

1. Lösung eines Gleichungssystems bilden. [Solve] ✓

Symbolische Manipulationen

1. Lösung eines Gleichungssystems bilden. [Solve] ✓
2. Matrix-Operationen [Matrix-Datentyp] ✓

Symbolische Manipulationen

1. Lösung eines Gleichungssystems bilden. [Solve] ✓
2. Matrix-Operationen [Matrix-Datentyp] ✓
3. Arithmetische Operationen auf Polynomen. [mpoly] ✓

Symbolische Manipulationen

1. Lösung eines Gleichungssystems bilden. [Solve] ✓
2. Matrix-Operationen [Matrix-Datentyp] ✓
3. Arithmetische Operationen auf Polynomen. [mpoly] ✓
4. Vereinfachen von komplizierten symbolischen Ausdrücken in eine Standardform. [simplr] ✓

Symbolische Manipulationen

1. Lösung eines Gleichungssystems bilden. [Solve] ✓
2. Matrix-Operationen [Matrix-Datentyp] ✓
3. Arithmetische Operationen auf Polynomen. [mpoly] ✓
4. Vereinfachen von komplizierten symbolischen Ausdrücken in eine Standardform. [simplr] ✓
5. Wertzuweisung der Variablen und Berechnung. [mpoly] ✓

Symbolische Manipulationen

1. Lösung eines Gleichungssystems bilden. [Solve] ✓
2. Matrix-Operationen [Matrix-Datentyp] ✓
3. Arithmetische Operationen auf Polynomen. [mpoly] ✓
4. Vereinfachen von komplizierten symbolischen Ausdrücken in eine Standardform. [simplr] ✓
5. Wertzuweisung der Variablen und Berechnung. [mpoly] ✓
6. Ableiten von symbolischen Ausdrücken. [D] ✓

Symbolische Manipulationen

1. Lösung eines Gleichungssystems bilden. [Solve] ✓
2. Matrix-Operationen [Matrix-Datentyp] ✓
3. Arithmetische Operationen auf Polynomen. [mpoly] ✓
4. Vereinfachen von komplizierten symbolischen Ausdrücken in eine Standardform. [simplr] ✓
5. Wertzuweisung der Variablen und Berechnung. [mpoly] ✓
6. Ableiten von symbolischen Ausdrücken. [D] ✓
7. Bilden von Integralen [Integrate] ✓

Fazit

- R bietet in Kombination mit CRAN Funktionen für grundlegende symbolische Mathematik.
  - Zusammenarbeit der Pakete ist nicht vorhanden.
  - Kein kontinuierlicher workflow.
- Ryacas und rSympy bieten CAS Schnittstelle an.
  - größerer Funktionsumfang.
  - Zusammenhang innerhalb des Computeralgebrasystems.

R als CAS?

- R als primitives Computeralgebrasystem.
- Entwickelt als System zur Datenanalyse und Statistik.
- Interpretierte Programmiersprache
  - R ist schwach in hoher numerischer Berechnung.

R als CAS?

- R als ausgereiftes CAS unzureichend.
- Daher: Interface zu existierenden CAS.

Danke für Ihre Aufmerksamkeit!

Inhaltsverzeichnis

Quellen[1]

[Sympy.org] Sympy, http://www.sympy.org/en/index.html

[CRAN:Ryacas] Ryacas, https://cran.r-project.org/web/packages/Ryacas/index.html, http://www.r-bloggers.com/using-r-as-a-computer-algebra-system-with-ryacas/

[Planetmath.org] Planetmath, http://planetmath.org/

[EconBS:D] Differentiation in R http://www.econometricsbysimulation.com/2012/08/symbolic-differentiation-in-r.html

Quellen[2]

[mathe-werkstatt] Computeralgebrasysteme
http://www.mathe-werkstatt.de/themen/cas.htm

[CRAN] Pakete: https://cran.r-project.org/

[math.ethz.ch] Integrate https://stat.ethz.ch/R-manual/R-devel/library/stats/html/integrate.html