Algebra Symboliczna

Wykład V

Andrzej Odrzywolek

Instytut Fizyki, Zakład Teorii Względności i Astrofizyki

11.07.2007, środa, 13:15

Dane kontaktowe

dr Andrzej Odrzywołek

pokój 447, IV piętro

E-mail: `odrzywolek@th.if.uj.edu.pl`

Wykład: środy 13.15-15.00 s. 128

Ćwiczenia: piątki 10.30-12.00

Konsultacje: środy ∼11-13, czwartki 10-12

WWW: http://ribes.if.uj.edu.pl/alsymb/

Obróbka danych w MATHEMATICE

MATHEMATICA ma możliwośći:

- importu danych w licznych formatach np. txt, xls, tsv
- graficznej prezentacji danych
- interpolacji danych
- dopasowywania modeli np. „fitowanie” wielomianów
- operowania na relatywnie dużych zbiorach

Uwaga!

Możliwości w zakresie importu/eksportu/wizualizacji danych zostały znacznie poszerzone w wersji 6.0

Uwagi ogólne

Ważne porady

- dane na ogół nie muszą być specjalnie formatowane — MATHEMATICA doskonale nadaje się do ich późniejszej obróbki . . .
- . . . ale wcześniejsze ich przygotowanie znacznie uprości(przyspieszy) pracę
- operowanie bardzo dużymi zbiorami danych niesie ryzyko „zawieszenia" interfejsu użytkownika w przypadku pomyłki: dane zostaną wyświetlone na ekranie jako część komunikatu o błędzie!
- zawsze należy przetestować kod na okrojonej wersji danych; jeżeli wszystko działa poprawnie wczytujemy pełne dane kończąc linie wyswietlające „duże" wyniki średnikami aby uniknać zaśmiecenia lub nawet zawieszenia interfejsu użytkownika

Uwagi ogólne (2)

Ważne informacje na temat wersji 6.0

- Z punktu widzenia obróbki danych MATHEMATICA 6.0 stanowi istotny krok do przodu
- problem z wyświetlaniem dużych danych rozwiązano poprzez automatyczne ograniczenie ilości danych na ekranie
- instrukcja **Short** „skraca” dane
- nowe wbudowane instrukcje graficzne np. **LogListPlot, ListPlot3D** itd
- rysowanie danych nie posiadających uporzadkowanej struktury (typu siatki prostokątnej)
- pre-definowane dane np: **IsotopeData**, **CountryData** itd (automatycznie ściągane z internetowej bazy danych)

Uwagi ogólne (3)

Pod pojeciem dane należy rozumieć nie tylko zbiór liczb uzyskanych w wyniku fizycznego eksperymentu ale również wyniki długotrwałych obliczeń

Przykłady

- Rowiązanie 100 równań z jedną niewiadomą, gdzie uzyskanie jednego trwa np. godzinę
- Wynik działania **DSolve** – funkcji rozwiązującej równania różniczkowe

Odczyt i zapis danych

Dane pochodzące z doświadczenia lub eksperymentu numerycznego najczęściej mają postać plików tekstowych różnej postaci

Odczyt danych

- Dane z pliku `"plik.ext"` ładujemy instrukcją **Import["plik.ext"]**
- Dla danych o „uporządkowanej" i znanej strukturze można użyć **ReadList**

Format danych jest ustalany automatycznie na postawie rozszerzenia. Obsługiwane typy to m. in.:

- .txt (plik tekstowy)
- .csv (dane oddzielone przecinkami)
- .tsv (dane oddzielone tabulatorami)
- .xls (dane w formaci excela)

Odczyt i zapis danych (2)

Odczyt danych: przykłady

- **Import["dane.tsv"]**
- **Import["http://ribes.if.uj.edu.pl/alsymb/przyklady/fitowanie_nieliniowe/dane/Si_data.txt","TSV"]**
- **dane = ReadList["/home/andrzej/dane.txt",{Real,Real,Real}]**
- ręczne wpisanie danych (np. z I pracowni):

$$\textbf{\textit{dane}} = \{\{0.1, 0.000098273, 4.69687 \times 10^{-6}\}, \{0.2, 0.00945858, 0.000413368\}, \{0.3, 0.0507545, 0.00199369\}, \{0.4, 0.123371, 0.00571356\}, \{0.5, 0.214963, 0.00880628\}, \{0.6, 0.322484, 0.00741434\}, \{0.7, 0.424869, 0.0169502\}\}$$

Odczyt i zapis danych (3)

Zapis danych

- Dane zawarte w zmiennej **dane** do pliku `"plik.ext"` zapisujemy instrukcją **Export["plik.ext", dane]**
- Format danych jest ustalany automatycznie na postawie rozszerzenia
- Jeżeli format jest niezgodny z rozszerzeniem wymuszamy go poprzez trzeci argument:
  Export["plik.ext", dane, "format danych"]

Przykłady

- **Export["dane.tsv", dane]**
- **Export["dane.dat", dane, "TSV"]**

Manipulacja danymi

Niech przykladowy plik tekstowy "dane.txt" w katalogu /home/andrzej/ pochodzi z programu w C:

```
for(i = 0; i < N_points; i++ ) {

  if( PRINT_SPECTRUM )
    printf("\%lf\t\%e\t\%e\n", E , Spectrum[i] / Norm, Error[i]/ Norm);

};
```

Efekt uruchomienia programu:

```
0.100000 9.827301e-05 4.696871e-06
0.200000 9.458575e-03 4.133682e-04
0.300000 5.075447e-02 1.993688e-03
0.400000 1.233705e-01 5.713560e-03
0.500000 2.149632e-01 8.806281e-03
0.600000 3.224839e-01 7.414336e-03
```

Manipulacja danymi (2)

Dane odczytujemy w MATHEMATICE:
In:= dane = ReadList["/home/andrzej/dane.txt",{Real,Real,Real}]

$$\texttt{Out} = \{\{0.1, 0.000098273, 4.69687 \times 10^{-6}\}, \{0.2, 0.00945858, 0.000413368\},$$
$$\{0.3, 0.0507545, 0.00199369\}, \{0.4, 0.123371, 0.00571356\}, \{0.5, 0.214963, 0.00880628\}$$

Aby odwołać się do 4-tej pozycji listy stosujemy następujacą notację: **In:=dane[[4]]**
Out = {0.4, 0.123371, 0.00571356}

Uwaga

- Dane są reprezentowane w postaci listy
- Lista jest indeksowana zaczynając od 1
- Ujemne indeksy pozwalają na odczytanie listy od tyłu
- Przykład: **dane[[-1]]** wskazuje na ostatni element listy.

Manipulacja danymi (3)

Wyrażenie: **dane[[5]]** jest 3-elementową listą więc możemy się odwołać np. do drugiego elementu:

In:= dane[[5]][[2]]

```
Out = 0.123371
```

lub równoważnie:

In:= dane[[5,2]]

```
Out = 0.123371
```

UWAGA na bardzo częste błędy !

- Do elementów list odwołujemy się *podwójnymi* nawiasami kwadratowymi !
- Listy są numerowane od 1 !

Prosta wizualizacja danych

ListPlot

Dane zawierające 1 lub dwie kolumny można rysować bezpośrednio

- Dla jednej kolumny na osi X jest numer elementu, na Y dane
- Dla dwóch kolumn na osi X jest pierwsza kolumna, na Y druga

Przykłady

ListPlot[dane]
ListPlot[dane, PlotJoined → True]

Uwaga

Aby przykłady wyżej zadziałały **dane** muszą zawierać maksymalnie 2 kolumny.
Dane zawierające więcej kolumn należy przekształcić (patrz dalej)

Przekształcanie danych

Najprostszą metodą przekształcania danych jest przepisanie za pomocą **Table**

Przykład

- Tablica **w** za pomocą której można wykreślić lub interpolować **dane** zawarte w kolumnie 1 i 2:

w = Table[{ dane[[ii,1]],dane[[ii,2]] }, {ii,1,Dimensions[dane][[1]]}]

Uwaga

Dimensions[dane] zwraca ona listę wymiarów tablicy **dane**, np:

In:=wymiar = Dimensions[dane]

Out={100,3}

In:=wymiar[[1]]

Out=100

wymiar[[2]]

Out=3

Wbudowane operacje na listach (subiektywny wybór)

Wybór operacji na listach

1. **Flatten** – „spłaszczenie zagnieżdżonych list” np:
   In:= Flatten[{a,{b,{c,d}},{e,f,g} }]
   `Out= {a,b,c,d,e,f,g}`
2. **Transpose** -- przestawienie kolumn z wierszami
3. **Drop** -- porzucenie pierwszego elementu listy
4. **Prepend** -- dodaj na początku
5. **Append** -- dodaj na końcu
6. **Select[lista,kryterium]** -- wybierz elementy spełniające dane kryterium np:
   Select[{-1,0,10,100,3}, #<10 &]

Uwaga!

Instrukcje powyżej są zoptymalizowane pod kątem wydajności

Interpolacja danych

Interpolation

Dane w postaci tablicy liczb można w łatwy sposób przekształcić do postaci funkcji:

In:= h = dane // Interpolation

```
Out = InterpolatingFunction[{{0, 10}}, <>]
```

In:= Plot[h[t],{t,0,10}]

Znaczenie f. interpolujących

1. Interpolowanie danych jest niezwykle efektywnym narzędziem, dzięki któremu możemy tworzyć *nowe* funkcje w większości zastosowań równoważne f. wbudowanym jak np. **Sin,Log**
2. Obiekty **InterpolatingFunction[<>]** są bardzo szybkie.
3. Idealne dla operowania wynikami długotrwałych obliczeń

Fitowanie liniowe

Fit

Linię prostą dopasowujemy w następujacy sposób:

In:= fit = Fit[wykres, {1, x}, x]

```
Out= 5.53503 + 5.44446 x
```

Rysowanie danych i f. dopasowanej na jednym wykresie:

rys1 = ListPlot[wykres]

rys2 = Plot[fit,{x,0,1}]

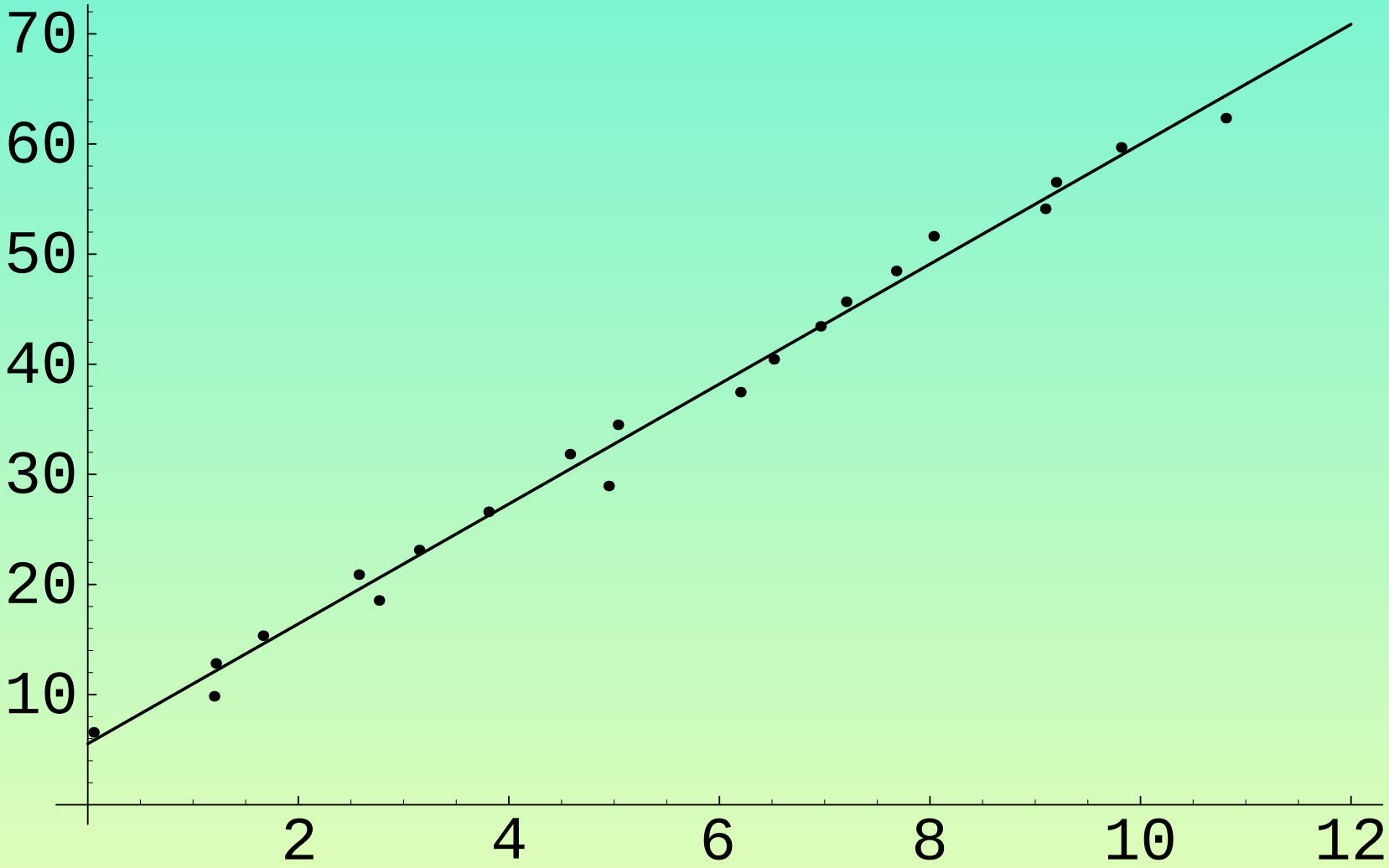
Show[rys1,rys2]

Uwaga!

Fitowanie (dopasowywanie) musi być liniowe w *nieznanych parametrach*. Same funkcje bazowe mogą być nieliniowe np:

Fit[wykres, {Sin[x], Exp[x] }, x]

Przykładowy wynik fitowania



Regress

Jeżeli potrzebujemy bardziej szczegółowej analizy danych, musimy dokonać regresji liniowej. Można to zrobić za pomocą komendy **Regress** z pakietu **Statistics`LinearRegression`**

In:=Regress[wykres, {1, x}, x, Weights → wagi]

$Out = \{$ParameterTable $\rightarrow$

	Estimate	SE	TStat	PValue
1	5.6467	0.637727	8.85443	3.59665×10^{-8}
x	5.46892	0.102845	53.1765	0.

,

RSquared $\rightarrow$ 0.993326, AdjustedRSquared $\rightarrow$ 0.992974,

EstimatedVariance $\rightarrow$ 0.836394, ANOVATable $\rightarrow$

	DF	SumOfSq	MeanSq	FRatio	PValue
Model	1	2365.1	2365.1	2827.74	0.
Error	19	15.8915	0.836394		
Total	20	2380.99			

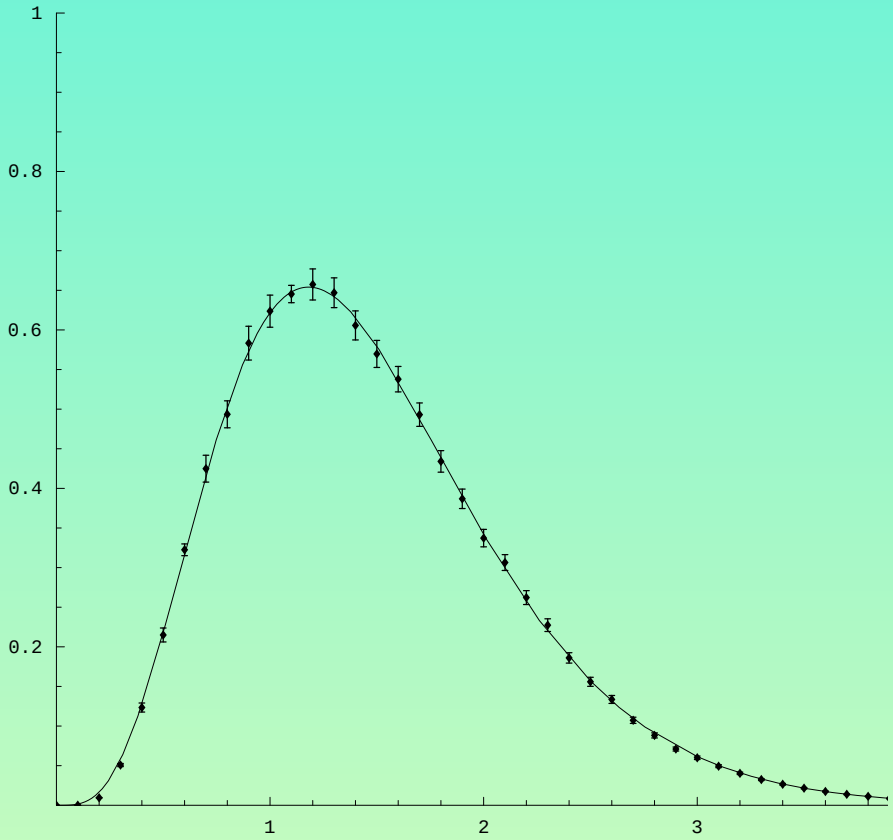
$\}$

Regresja i fitowanie *nieliniowe*

Wprowadzenie

1. Dopasowywanie modelu nieliniowego w parametrach jest zadaniem znacznie trudniejszym od fitowanie liniowego które może być wykonane np. z użyciem dobrego kalkulatora
2. Fitowanie nieliniowe wymaga zwykle użycia specjalistycznego oprogramowania np. Origin, Root itp.
3. MATHEMATICA jest w tej dziedzinie bardzo mocnym narzędziem
4. Fitowanie nieliniowe nie zawsze musi się udać
5. modelem może być cokolwiek, nie tylko funkcje zadane wzorem, np. równanie różniczkowe (dzięki zastosowaniu np. f. interpolujących)

Regresja i fitowanie *nieliniowe*



Dopasowywanie funkcji z nieliniową zależnością od parametrów wykonujemy komendami **NonlinearFit, NonlinearRegress** z pakietu **Statistics**

Wartości startowe parametrów *muszą* być podane. Ich przybliżone wartości trzeba znaleźć eksperymentalnie (graficznie).

Przykład

$$\mathbf{NonlinearRegress}[\mathit{wykres}, ax^q e^{-bx}, x, \{\{a, 50.1, 200\}, \{b, 1, 3\}, \{q, 2, 4\}\}]$$