

## 2. テイラー展開による多項式近似

### 2.1 テイラー展開による多項式近似

以下のコマンドで数式が多項式近似式を得ることができます。積分式や特殊関数の式を近似して Fortran や C の形式に変換すれば、他のソフトウェアでも数式を簡単に利用することができます。

series : 一般的な級数展開を行います。式の形によりテイラー展開か ローラン展開のどちらかを選択します。

taylor : テイラー展開を行います。

$e^x$  の級数展開

近似する式と、展開する位置、次数を指定します。

```
> restart;  
> series(exp(x)/x, x=0, 8);
```

特殊関数を含む式のテイラー展開

```
> eq:=int(1/GAMMA(t),t=0..x)/x^2;  
> taylor(eq,x=0, 5);
```

定数 gamma が出てきてしまったので、これを数値化します。

```
> evalf(%);
```

剰余項 O() を取り除くため、convert コマンドで式を多項式に変換します。

```
> eq2:=convert(% ,polynom);
```

```
>
```

最後に、元の式と近似式をグラフにしてみましよう。

```
> plot({eq,eq2},x=-1..1);
```

近似した式を C や Fortran の形式に変換し、他のプログラムで使用することも可能です。

```
> CodeGeneration[Fortran](eq2);
```

```
>
```

```
> CodeGeneration[C](eq2);
```

### 2.2 テイラー展開による多項式近似 (応用編)

下の式で定義される半球面を多項式近似し、グラフィックソフトに多項式面としてデータを渡すことにします。ここでは多変数のテイラー展開を行うコマンド mtaylor を使用します。

$$\begin{aligned}x &:= \cos(\text{Pi } u) \sin(\text{Pi } v) \\y &:= \sin(\text{Pi } u) \sin(\text{Pi } v) \\z &:= \cos(\text{Pi } v)\end{aligned}$$

```
> restart:
```

使用する配列の定義と、数値計算の精度の指定 (有効桁数30桁)

```
> t:=array(1..3): taylor_t:=array(1..3): Digits:=30:
```

テイラー展開の次数 order\_t の定義

```
> order_t:=3:
```

半球面の定義

```
> t[1]:=cos(Pi*u)*sin(Pi*v):
```

```
> t[2]:=sin(Pi*u)*sin(Pi*v):
```

```
> t[3]:=cos(Pi*v):
```

多変数の展開を行うコマンド mtaylor の呼び出しと、テイラー展開

( $u=0.5, v=0.5$  の近傍で展開)

```
> taylor_t[1]:=convert(mtaylor(t[1],[u=0.5,v=0.5],order_t),  
  polynom);
```

```
taylor_t[2]:=convert(mtaylor(t[2],[u=0.5,v=0.5],order_t),
polynom);
taylor_t[3]:=convert(mtaylor(t[3],[u=0.5,v=0.5],order_t),
polynom);
```

元の式と近似式をプロット

(近似が十分で無い場合は, order\_t の値を変更し, すべてのコマンドを実行し直して下さい。)

```
> exact:=plot3d(convert(t,list),u=0..1,v=0..1,color=blue):
approx:=plot3d(convert(taylor_t,list),u=0..1,v=0..1,color=red):
plots[display]({exact,approx},view=[-1..1, -1..1, -1..1], style=
wireframe, scaling=constrained);
```

## 係数データの取り出し

多項式 t[1], t[2], t[3] の係数を取り出します。

```
> printf("vvvvvvvvvvvvvv CUT HERE vvvvvvvvvvvvvv/n ORDER 0..%d of
Variable u and v /n U V    COEFFICIENTS", order_t);
for iiii from 0 to order_t do
for jjjj from 0 to order_t do
printf(" %o %o %13.12E/n", iiii, jjjj, evalf(coeff(collect(coeff
(collect(taylor_t[1], u), u, iiii), v), v, jjjj)));
od:
od:
for iiii from 0 to order_t do
for jjjj from 0 to order_t do
printf(" %o %o %13.12E/n", iiii, jjjj, evalf(coeff(collect(coeff
(collect(taylor_t[2], u), u, iiii), v), v, jjjj)));
od:
od:
for iiii from 0 to order_t do
for jjjj from 0 to order_t do
printf(" %o %o %13.12E/n", iiii, jjjj, evalf(coeff(collect(coeff
(collect(taylor_t[3], u), u, iiii), v), v, jjjj)));
od:
od:
print(`^^^^^^^^^^^^^^^^ CUT HERE ^^^^^^^^^^^^^^^^^/n`);
```

この出力を Cut & Paste すれば, 他のソフトでのデータとして利用できます。

また, 下のようにファイルに出力することもできます。

```
> f1:=fopen("C:/temp/mapledata.txt",WRITE);
fprintf(f1,"vvvvvvvvvvvvvv CUT HERE vvvvvvvvvvvvvv/n1 1 %d %d/n",
order_t+1, order_t+1);
for iiii from 0 to order_t do
for jjjj from 0 to order_t do
fprintf(f1," %o %o %13.12E/n", iiii, jjjj, evalf(coeff(collect
(coeff(collect(taylor_t[1], u), u, iiii), v), v, jjjj)));
od:
od:
for iiii from 0 to order_t do
for jjjj from 0 to order_t do
fprintf(f1," %o %o %13.12E/n", iiii, jjjj, evalf(coeff(collect
(coeff(collect(taylor_t[2], u), u, iiii), v), v, jjjj)));
od:
od:
for iiii from 0 to order_t do
for jjjj from 0 to order_t do
fprintf(f1," %o %o %13.12E/n", iiii, jjjj, evalf(coeff(collect
(coeff(collect(taylor_t[3], u), u, iiii), v), v, jjjj)));
od:
od:
fprintf(f1,"^^^^^^^^^^^^^^^^ CUT HERE ^^^^^^^^^^^^^^^^^/n");
```

| L `fclose(f1):`  
[

Copyright (c) 2009, Cybernet Systems, Co.,Ltd. All rights reserved.