

KETpic の機能拡張 とくに メタコマンドとその応用

高遠 節夫 (東邦大)

金子 真隆・山下 哲 (木更津高専)

北原 清志 (工学院大)

教材作成と K_ETpic

数学教材作成の要件

- (1) 数式
- (2) 記号
- (3) 図
- (4) 表
- (5) レイアウト

数学教材作成の要件

- (1) 数式
- (2) 記号
- (3) 図
- (4) 表
- (5) レイアウト

手書きはすべての要件を満たす

数学教材作成の要件

- (1) 数式
- (2) 記号
- (3) 図
- (4) 表
- (5) レイアウト

手書きはすべての要件を満たす
でも、正確さ、きれいさ、再現性に難点

数学教材作成の要件

(1) 数式	$\text{T}_{\text{E}}\text{X}$	ワープロ
(2) 記号	$\text{T}_{\text{E}}\text{X}$	ワープロ
(3) 図	$\text{T}_{\text{E}}\text{X}$	ワープロ
(4) 表	$\text{T}_{\text{E}}\text{X}$	ワープロ
(5) レイアウト	$\text{T}_{\text{E}}\text{X}$	ワープロ

数学教材作成の要件

(1) 数式	TeX	ワープロ
(2) 記号	TeX	ワープロ
(3) 図	TeX	ワープロ
(4) 表	TeX	ワープロ
(5) レイアウト	TeX	ワープロ

数学教材作成の要件

(1) 数式	TeX	ワープロ
(2) 記号	TeX	ワープロ
(3) 図	TeX	ワープロ
(4) 表	TeX	ワープロ
(5) レイアウト	TeX	ワープロ

数学教材作成の要件

(1) 数式	TeX	ワープロ
(2) 記号	TeX	ワープロ
(3) 図	TeX	ワープロ
(4) 表	TeX	ワープロ
(5) レイアウト	TeX	ワープロ

数学教材作成の要件

(1) 数式

~~TeX~~

(2) 記号

~~TeX~~

+K_ETpic メタコマンド

(3) 図

~~TeX~~

⇒ K_ETpic 描画コマンド

(4) 表

~~TeX~~

⇒ K_ETpic 作表コマンド

(5) レイアウト

~~TeX~~

⇒ “layer”環境

KE_ETpic とは

(1) T_EX 描画のための CAS マクロパッケージ

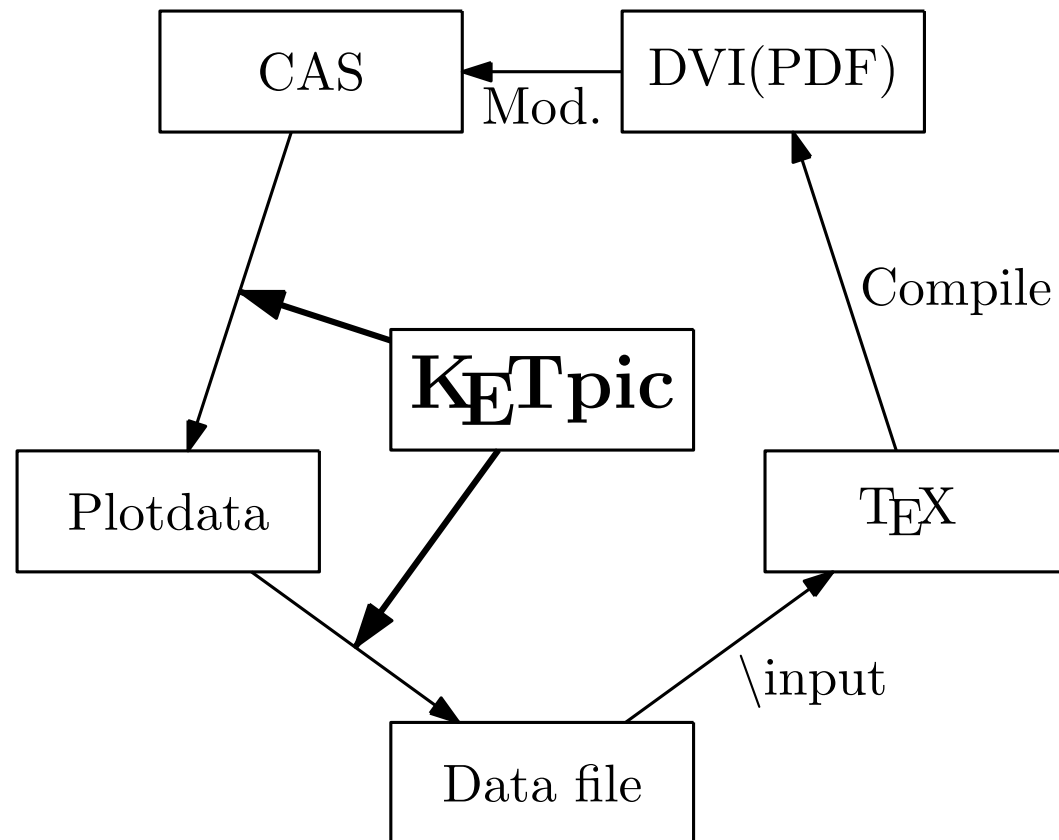
Maple, Mathematica, Maxima, Risa/Asir
Scilab, Matlab, R

(2) CAS の機能で複雑な図の作成が可能 プログラミング, 数式記述能力

(3) 次のサイトからフリーでダウンロードできる.

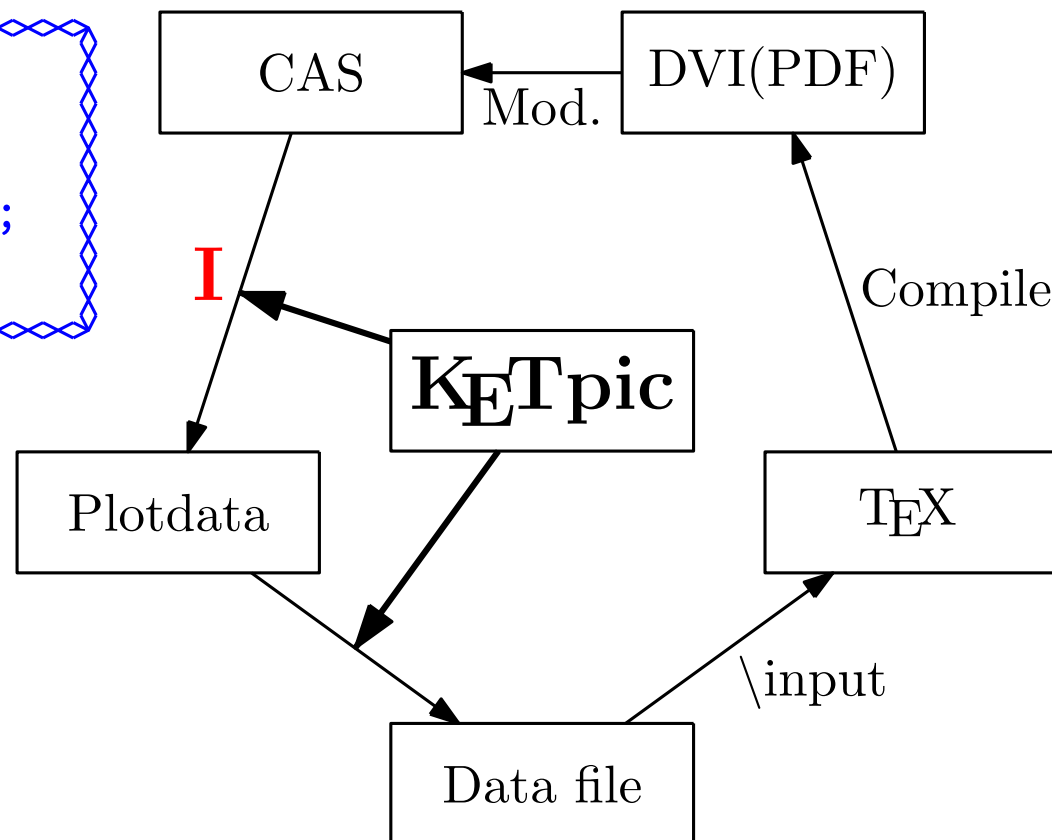
<http://ketpic.com>

KE_ETpicによる描画



K_ETpicによる描画

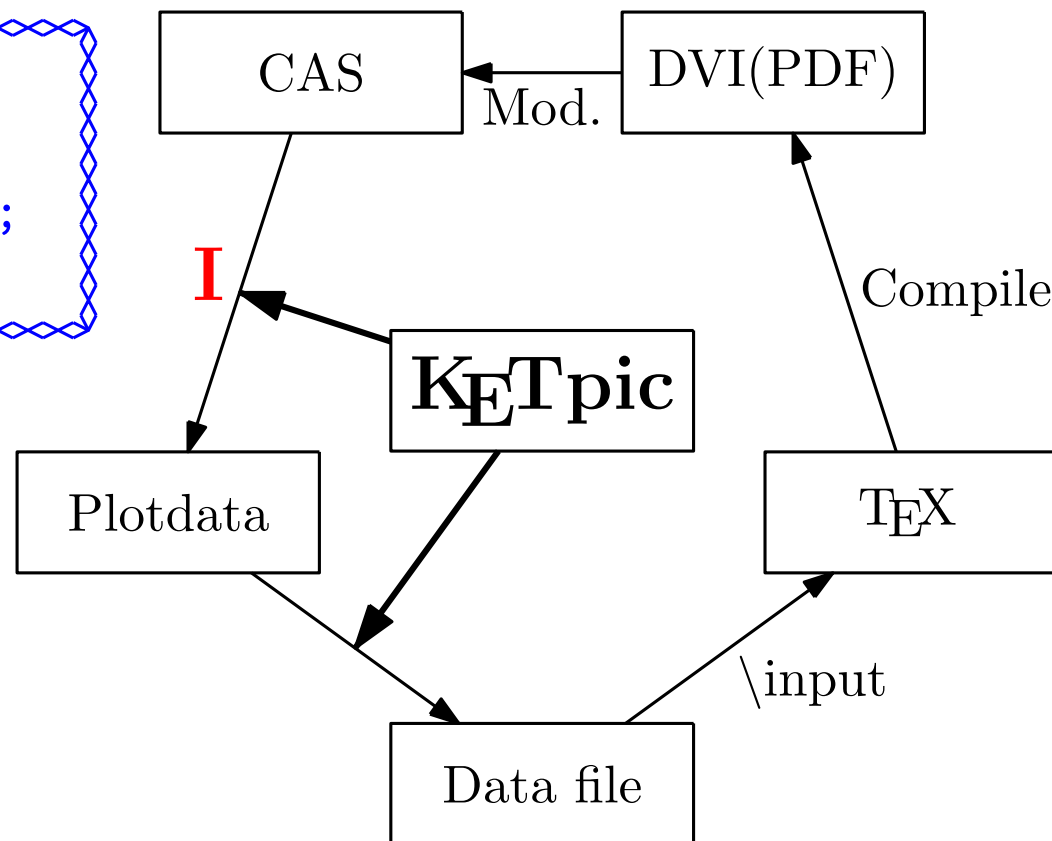
```
G=list();  
for a=[2, 2.5, 3, 3.5]  
    G($+1)=Plotdata('a^x','x');  
end;
```



K_ETpicによる描画

```
G=list();  
for a=[2, 2.5, 3, 3.5]  
    G($+1)=Plotdata('a^x','x');  
end;
```

読みやすさがポイント

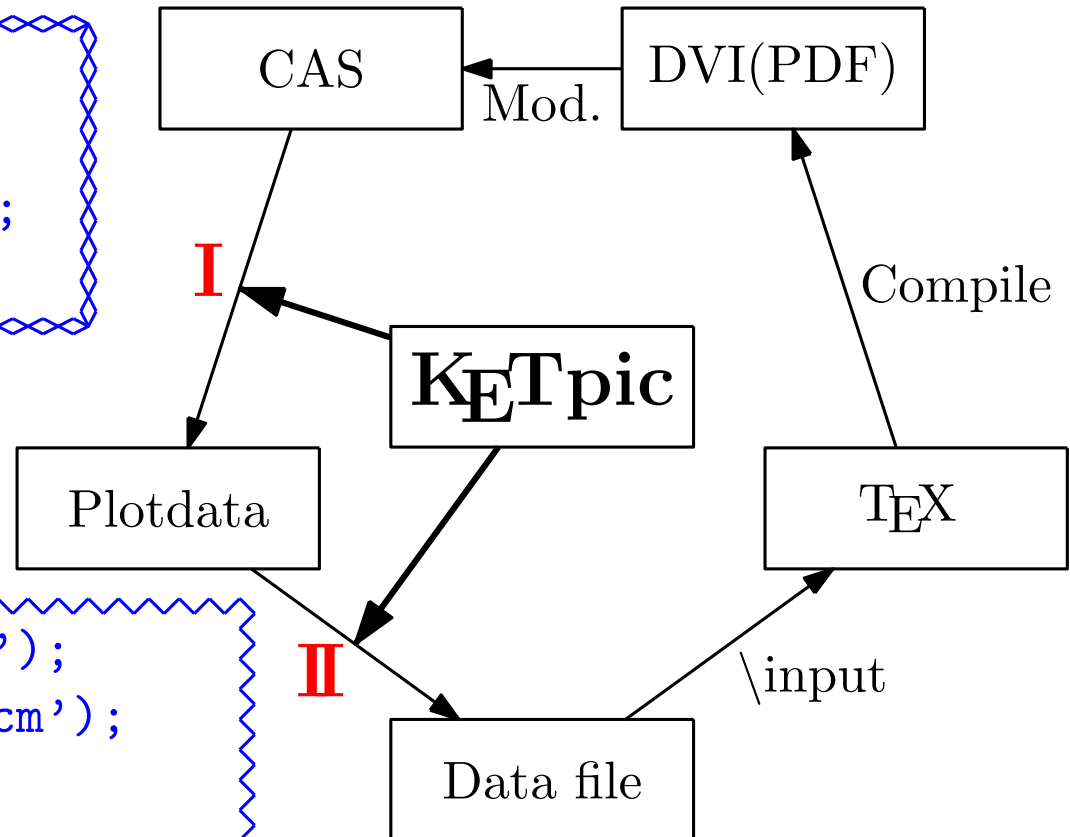


KE_ETpicによる描画

```
G=list();  
for a=[2, 2.5, 3, 3.5]  
  G($+1)=Plotdata('a^x','x');  
end;
```

読みやすさがポイント

```
Openfile('fig.tex');  
  Beginpicture('1cm');  
    Drwline(G);  
  Endpicture(1);  
Closefile()
```

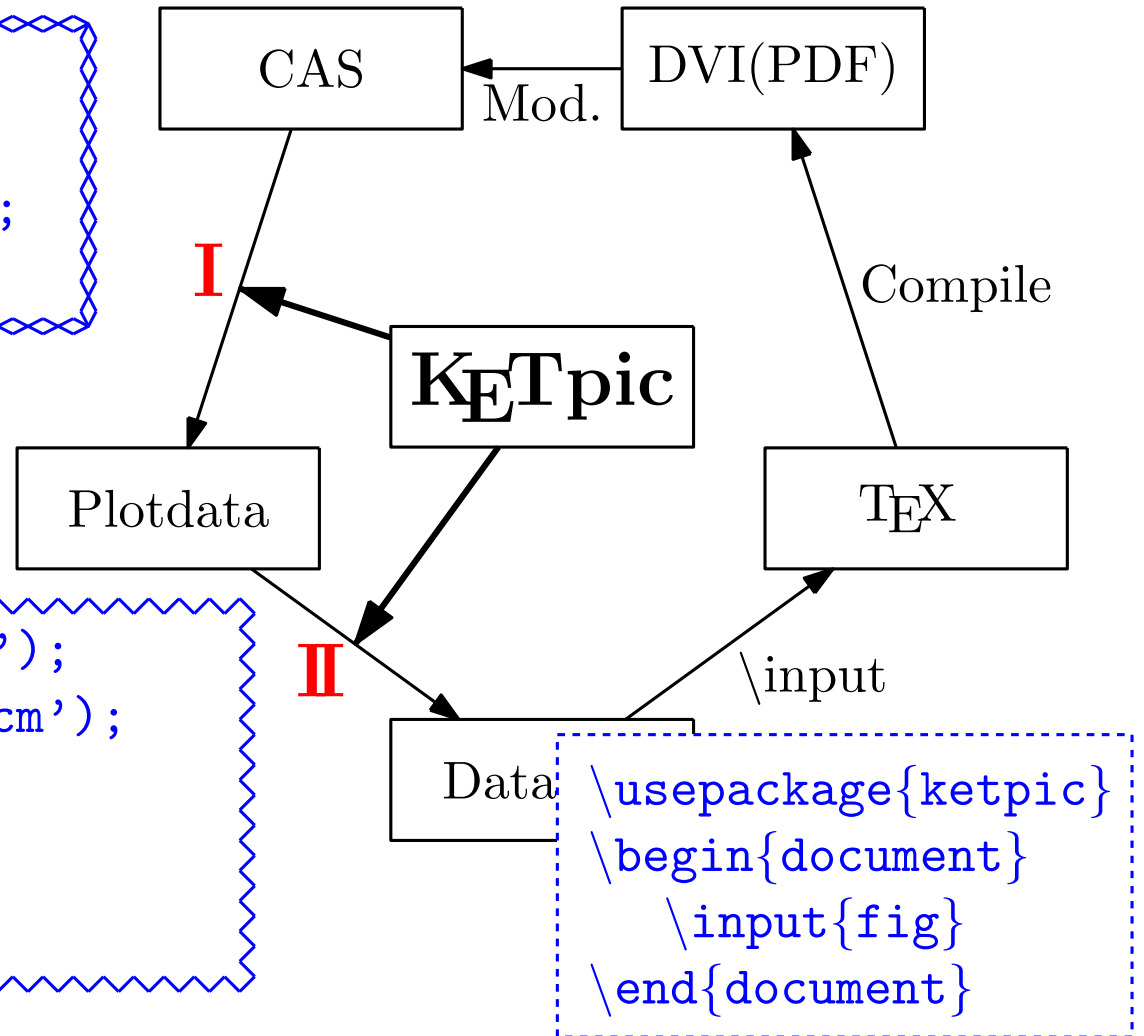


KE_ETpicによる描画

```
G=list();  
for a=[2, 2.5, 3, 3.5]  
  G($+1)=Plotdata('a^x','x');  
end;
```

読みやすさがポイント

```
Openfile('fig.tex');  
  Beginpicture('1cm');  
    Drwline(G);  
  Endpicture(1);  
Closefile()
```



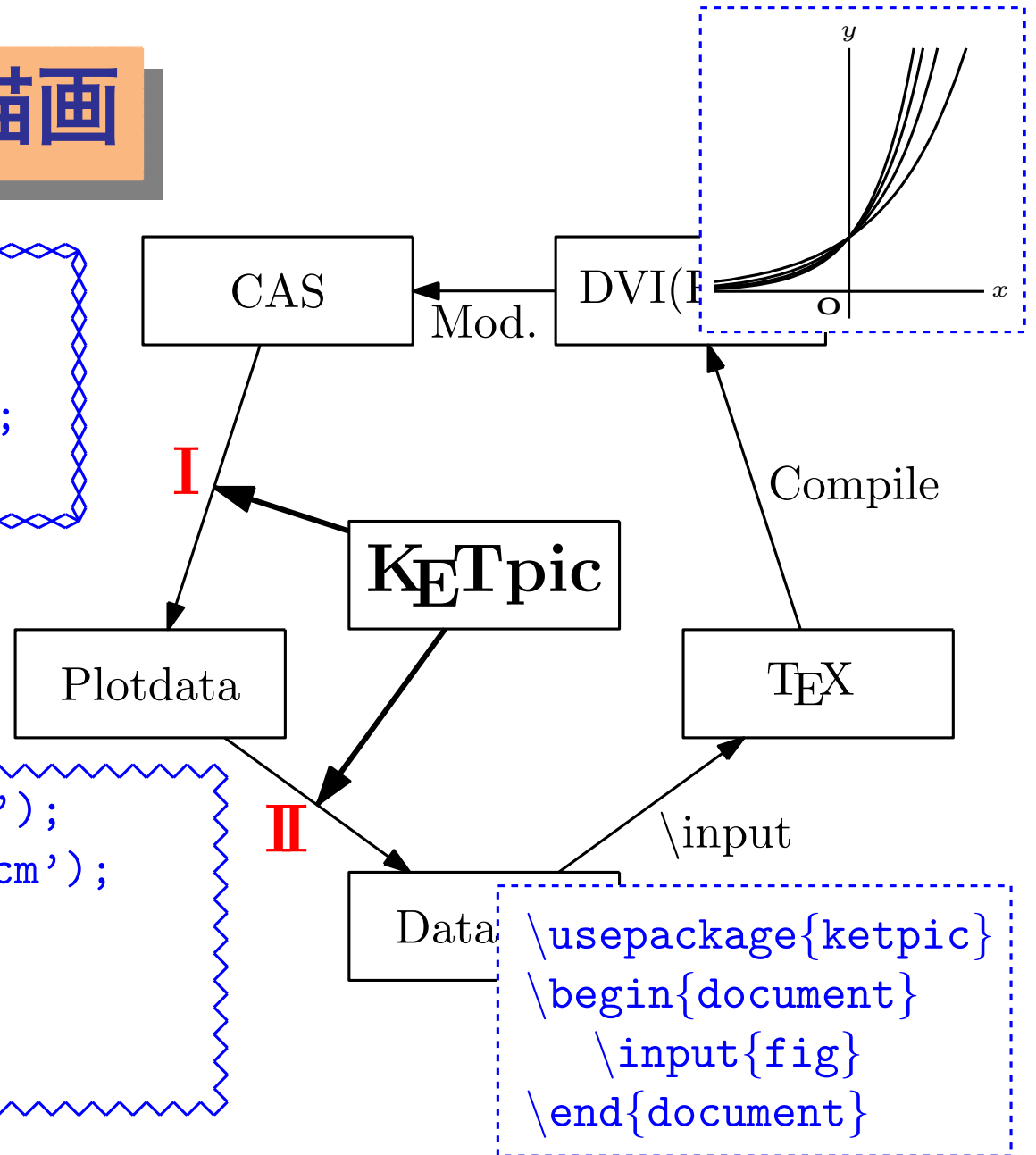
KE_ETpicによる描画

```
G=list();  
for a=[2, 2.5, 3, 3.5]  
  G($+1)=Plotdata('a^x','x');  
end;
```

読みやすさがポイント

```
Openfile('fig.tex');  
  Beginpicture('1cm');  
    Drwline(G);  
  Endpicture(1);  
Closefile()
```

```
Data: \usepackage{ketpic}  
       \begin{document}  
         \input{fig}  
       \end{document}
```



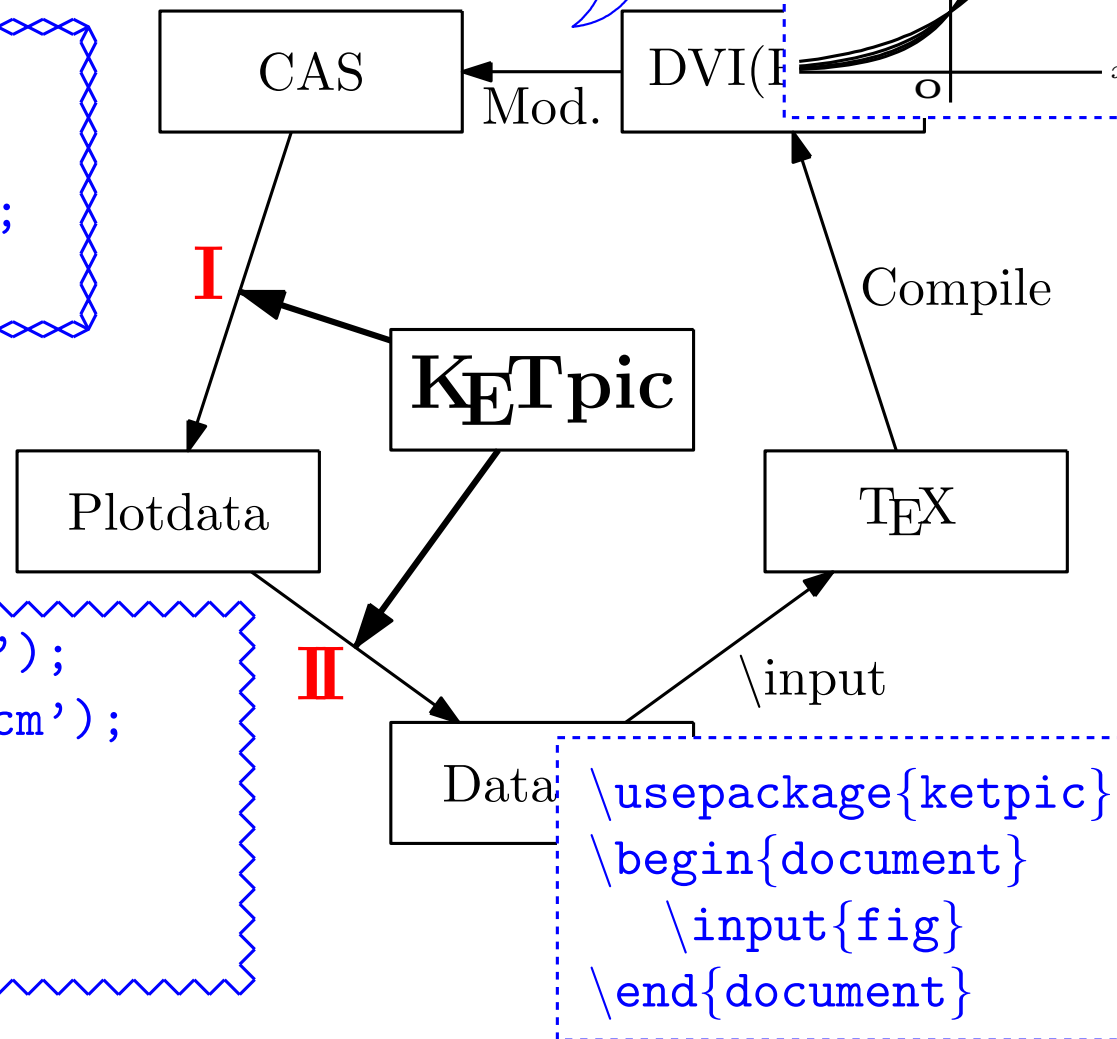
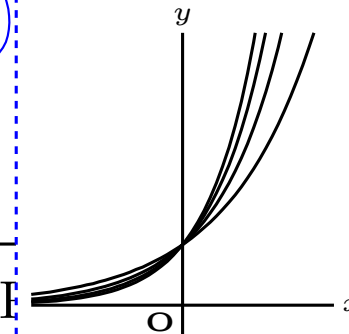
KE_ETpicによる描画

```
G=list();
for a=[2, 2.5, 3, 3.5]
  G($+1)=Plotdata('a^x','x');
end;
```

読みやすさがポイント

```
Openfile('fig.tex');
Beginpicture('1cm');
  Drwline(G);
Endpicture(1);
Closefile()
```

スパイラル
アップ

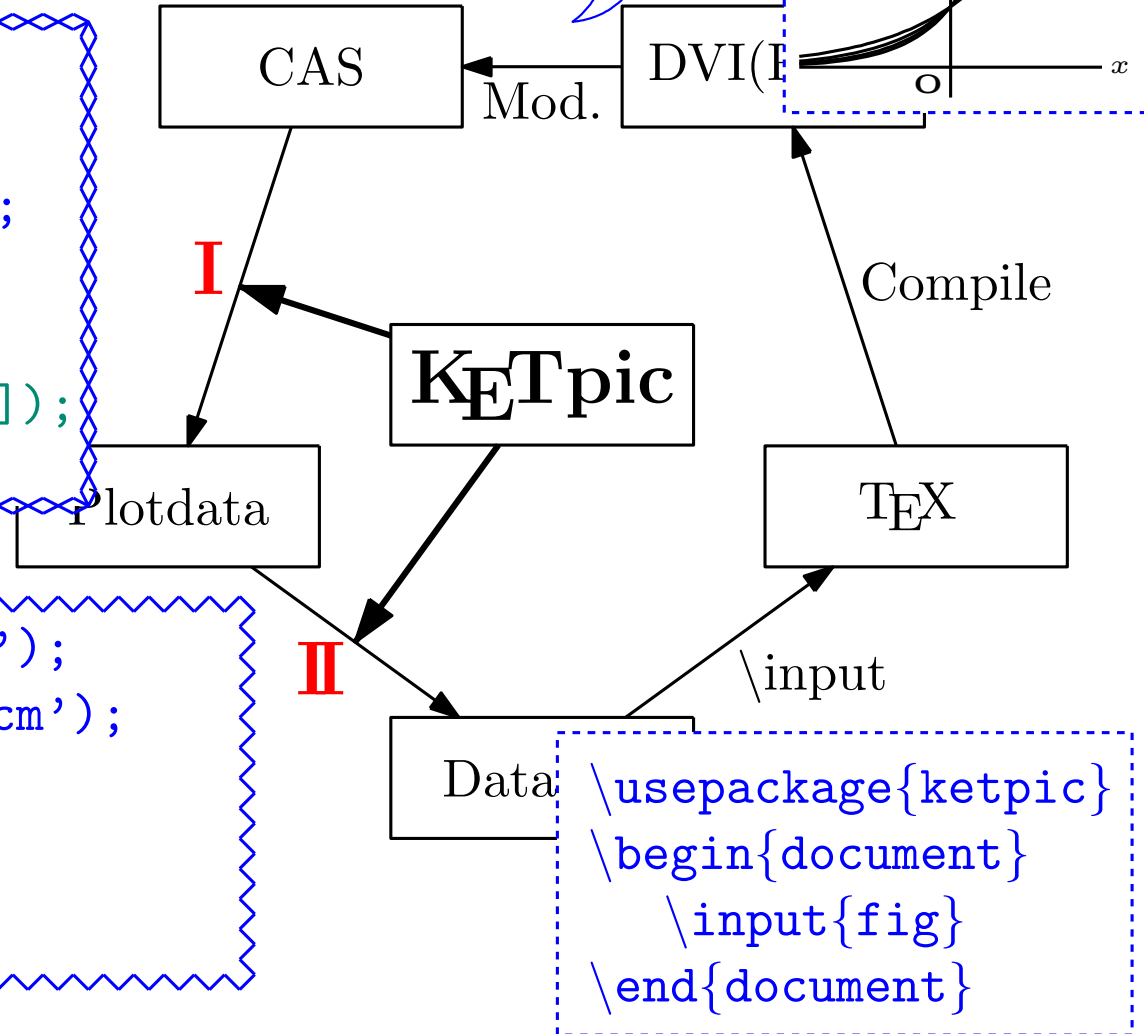
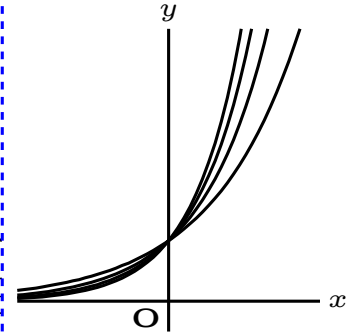


KE_ETpicによる描画

```
G=list();
for a=[2, 2.5, 3, 3.5]
  G($+1)=Plotdata('a^x','x');
end;
H=0.5;
Gb=Framedata([-H,H],[1-H,1+H]);
```

```
Openfile('fig.tex');
Beginpicture('1cm');
  Drwline(G);
Endpicture(1);
Closefile()
```

スパイラル
アップ

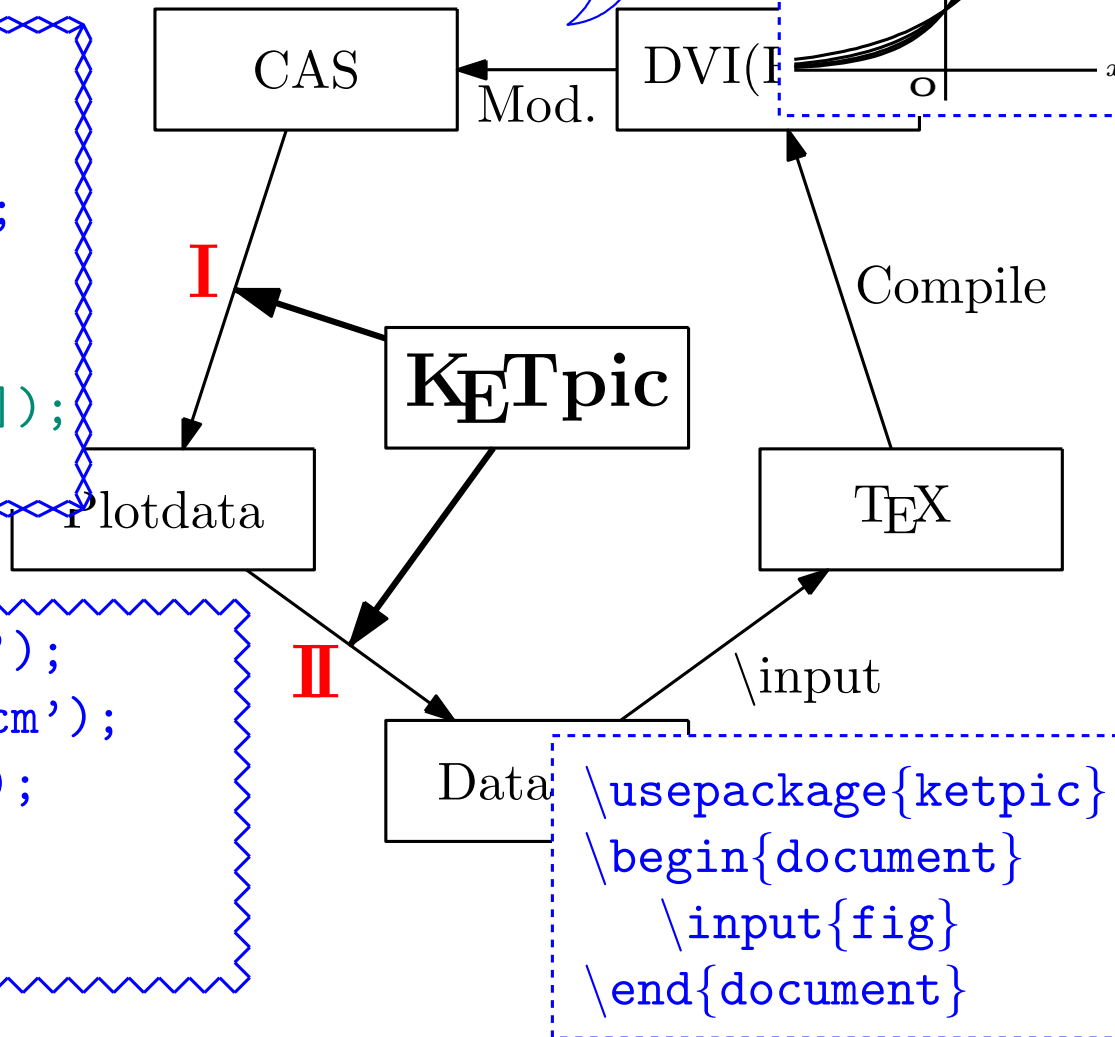
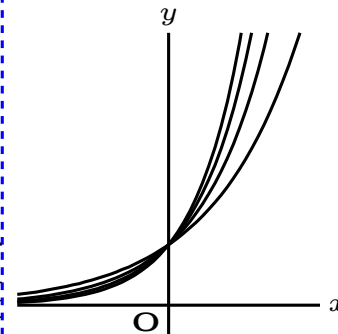


KE_ETpicによる描画

```
G=list();
for a=[2, 2.5, 3, 3.5]
  G($+1)=Plotdata('a^x','x');
end;
H=0.5;
Gb=Framedata([-H,H],[1-H,1+H]);
```

```
Openfile('fig.tex');
Beginpicture('1cm');
  Drwline(G, Gb);
Endpicture(1);
Closefile()
```

スパイラル
アップ

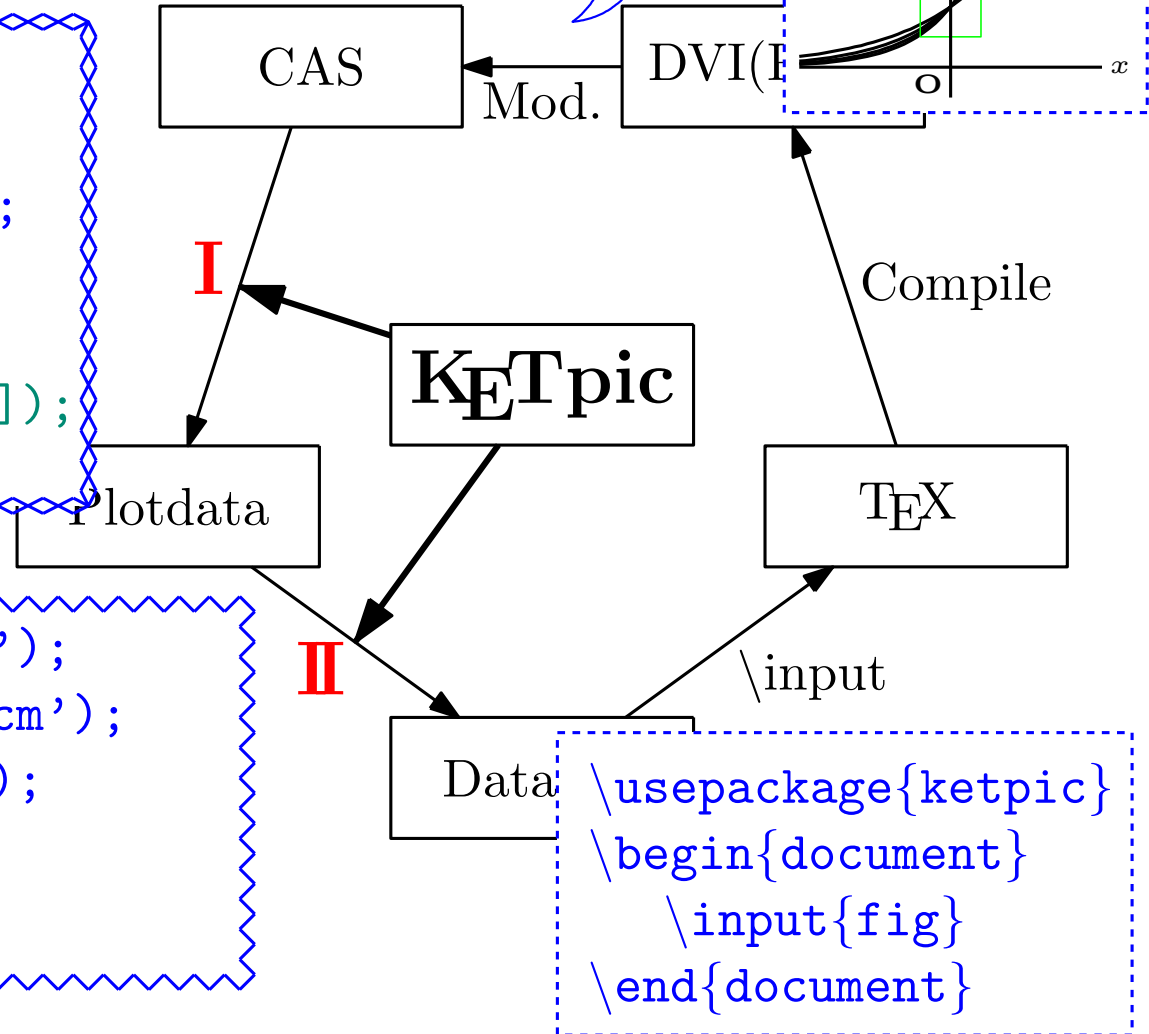
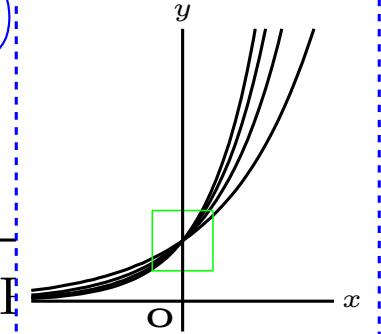


KE_ETpicによる描画

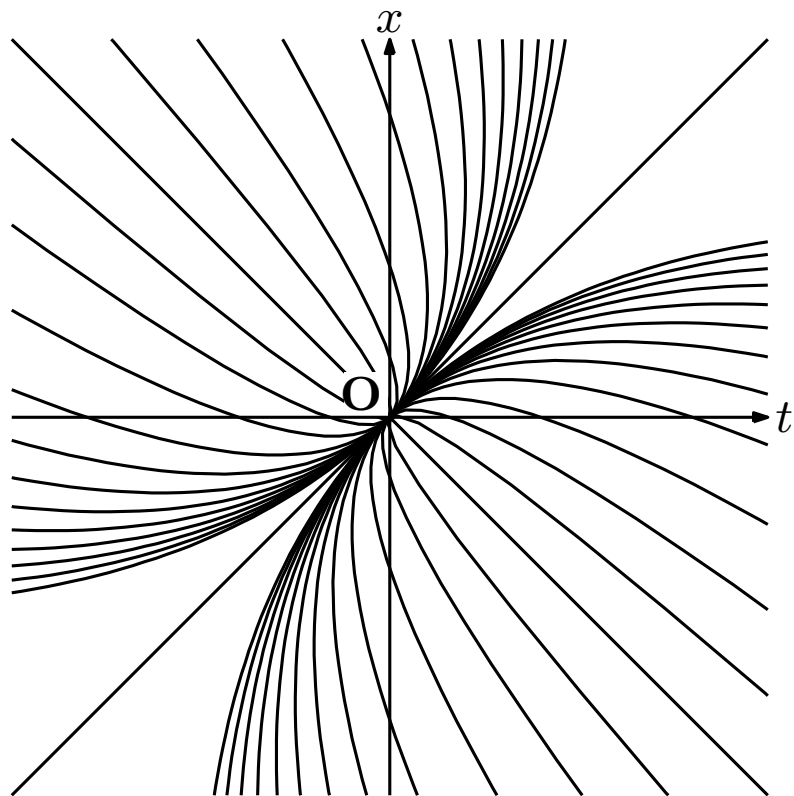
```
G=list();
for a=[2, 2.5, 3, 3.5]
  G($+1)=Plotdata('a^x','x');
end;
H=0.5;
Gb=Framedata([-H,H],[1-H,1+H]);
```

```
Openfile('fig.tex');
Beginpicture('1cm');
  Drwline(G, Gb);
Endpicture(1);
Closefile()
```

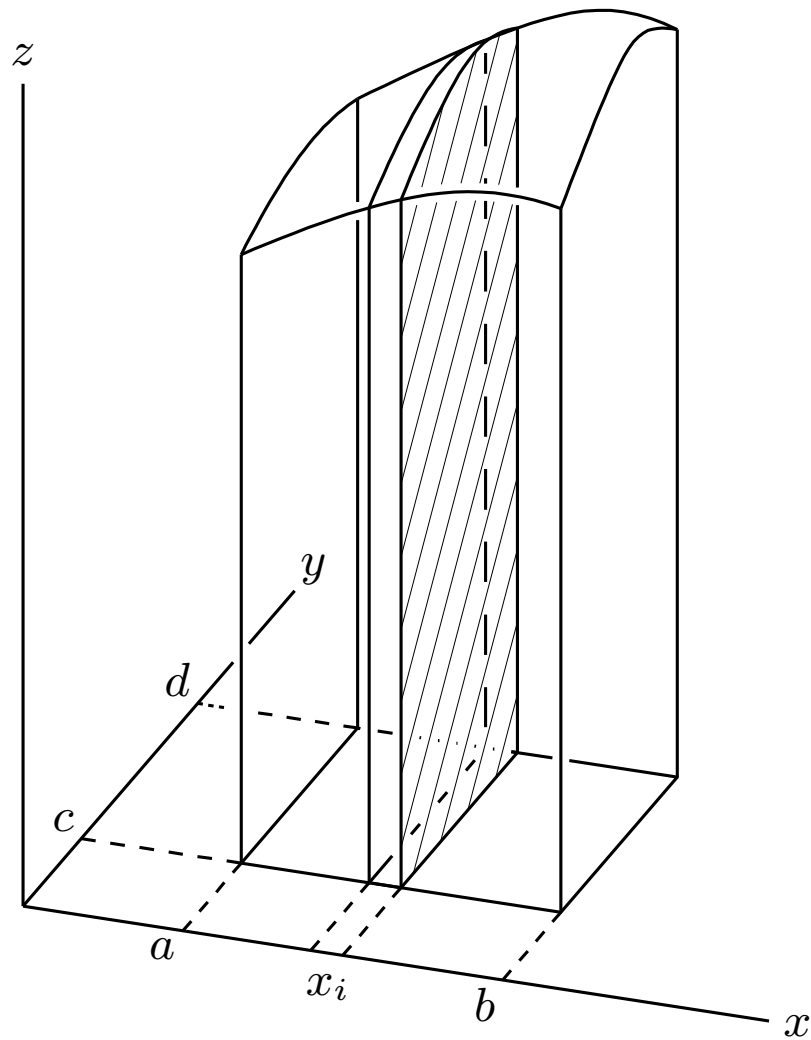
スパイラル
アップ



作图例



$$z = f(x, y)$$



KETrpic のメタコマンド

K_ETpic の進展

K_ETpic は T_EX 文書への挿図ツールとして強力

K_ETpic の進展

K_ETpic は T_EX 文書への挿図ツールとして強力
実は, T_EX マクロ作成にも有用である

K_ETpic の進展

K_ETpic は T_EX 文書への挿図ツールとして強力

実は, T_EX マクロ作成にも有用である

K_ETpic にマクロ作成用のコマンドを追加

K_ETpic の進展

K_ETpic は T_EX 文書への挿図ツールとして強力

実は, T_EX マクロ作成にも有用である

K_ETpic にマクロ作成用のコマンドを追加

メタコマンド

K_ETpic の進展

K_ETpic は T_EX 文書への挿図ツールとして強力

実は, T_EX マクロ作成にも有用である

K_ETpic にマクロ作成用のコマンドを追加

メタコマンド

描画コマンドとのコンビネーション

⇒ 図的要素をもつマクロを作成

簡単なマクロ例

1 2 3 4 5 6 7 8 9 0



簡単なマクロ例

```
Openphr( '\cirmark' );  
  Tmp=Circledata( [0,0] ,1);  
  F1=Scaledata( Tmp, Sx, Sy);  
  Beginpicture( '1mm' );  
    Shade( list( F1) );  
    Drwline( F1);  
  Endpicture( 0);  
Closephr();
```

簡単なマクロ例

1

```
Openphr('\circmark');  
  Tmp=Circledata([0,0],1);  
F1=Sc Openphr('\dashmark#1');  
Begin  Beginpicture('1mm');  
  Sha  Dashline(F1);  
  Drw  Letter([0,0],'c','#1');  
Endpi  Endpicture(0);  
Closeph Closephr();
```

簡単なマクロ例

1

```
Openphr('\circmark');  
  Tmp=Circledata([0,0],1);  
  F1=Sc Openphr('\dashmark#1');  
  Beginpicture('1mm');  
    Sha Dashline(F1);  
    Drw Letter([0,0],'c','#1');  
  Endpicture  
Closephr();
```

より複雑な場合

プログラミングで必要なもの

- (1) ユーザ変数
- (2) 条件分岐
- (3) 繰り返し

より複雑な場合

プログラミングで必要なもの

- (1) ユーザ変数
- (2) 条件分岐
- (3) 繰り返し

しかし、いずれも $\text{T}_\text{E}\text{X}$ プログラミングでは**難関**

KETrpic の追加メタコマンド

(現在は Scilab 版のみ)

(1) 変数 = カウンタ

```
Texsetctr(2, '#1+2*Texctr(1)');
```

(2) 条件分岐

```
Texif(Texthctr(1) + '=0');  
Texelse();  
Texendif();
```

(3) 繰り返し

```
Texfor(1,1, '#1');  
Texendfor(1);
```

少し複雑なマクロ例

“`\diachain{n}`”

少し複雑なマクロ例

“`\diachain{n}`”

`\diachain{6}`



少し複雑なマクロ例

“`\diachain{n}`”

```
Texnewcmd('\diachain',1);
  Tmp=Framedata([0,0],L);
  G=Rotatedata(Tmp,%pi/4);
  Texfor(1,'1','\#1');
    Beginpicture('1mm');
      Drwline(G,2);
      Texsetctr(2,'Texctr(1)/2*2-Texctr(1)');
      Texif(Texthctr(2)+'=0');
        Shade(G);
      Texendif();
    Endpicture(0);
  Texendfor(1);
Texend();
```



少し複雑なマクロ例

“\diachain{n}”

```
Texnewcmd('\diachain',1);
  Tmp=Framedata([0,0],L);
  G=Rotatedata(Tmp,%pi/4);
  Texfor(1,'1','\#1');
    Beginpicture('1mm');
      Drwline(G,2);
      Texsetctr(2,'Texctr(1)/2*2-Texctr(1)');
      Texif(Texthectr(2)+'=0');
        Shade(G);
      Texendif();
    Endpicture(0);
  Texendfor(1);
Texend();
```



応用としての layer 環境

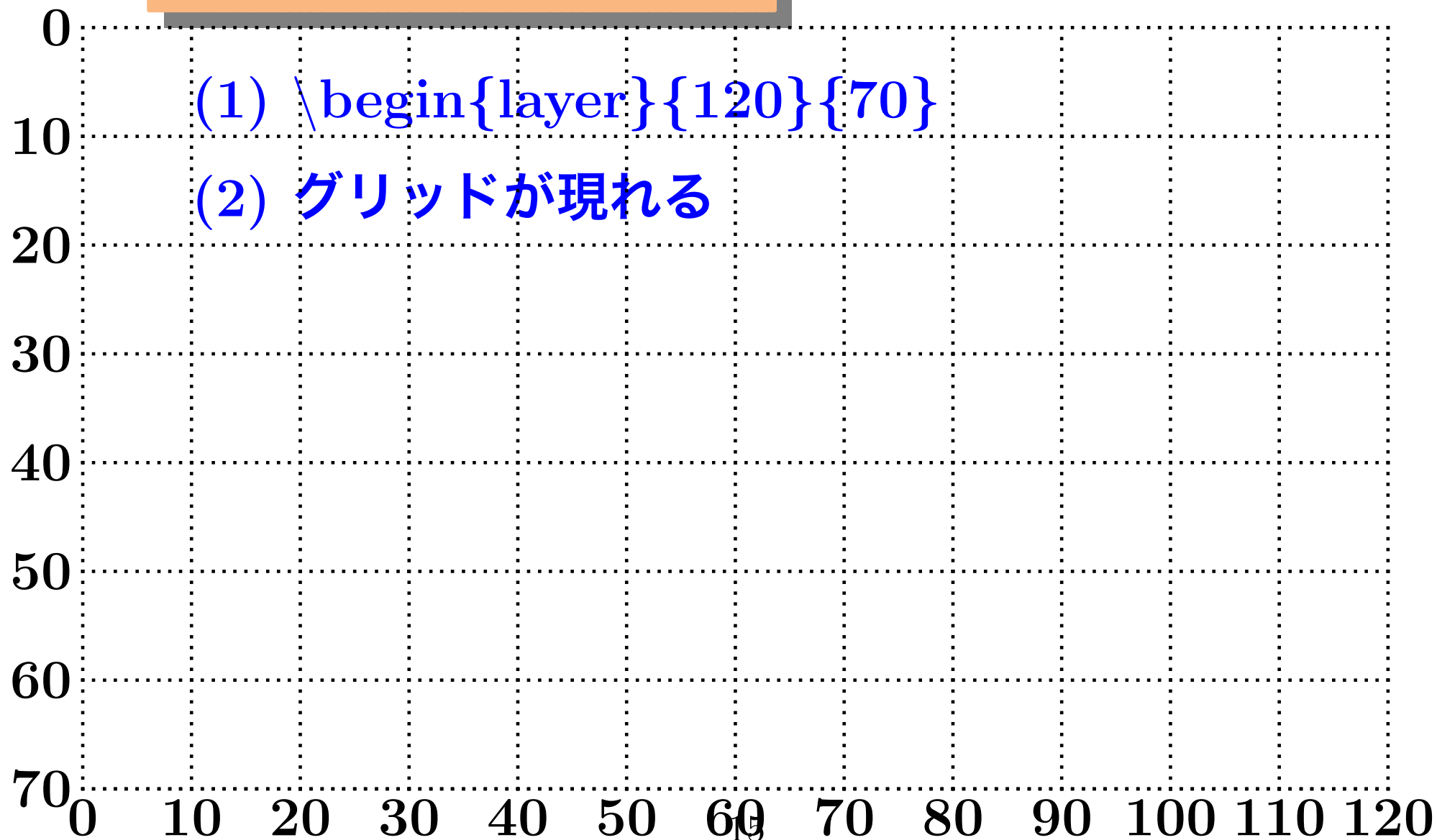
layer 環境

- (1) “overpic.sty” と似ているが、より汎用的
- (2) 文書中に記号などを配置
- (3) 柔軟なレイアウト

layer 環境の使用法

(1) `\begin{layer}{120}{70}`

layer 環境の使用法

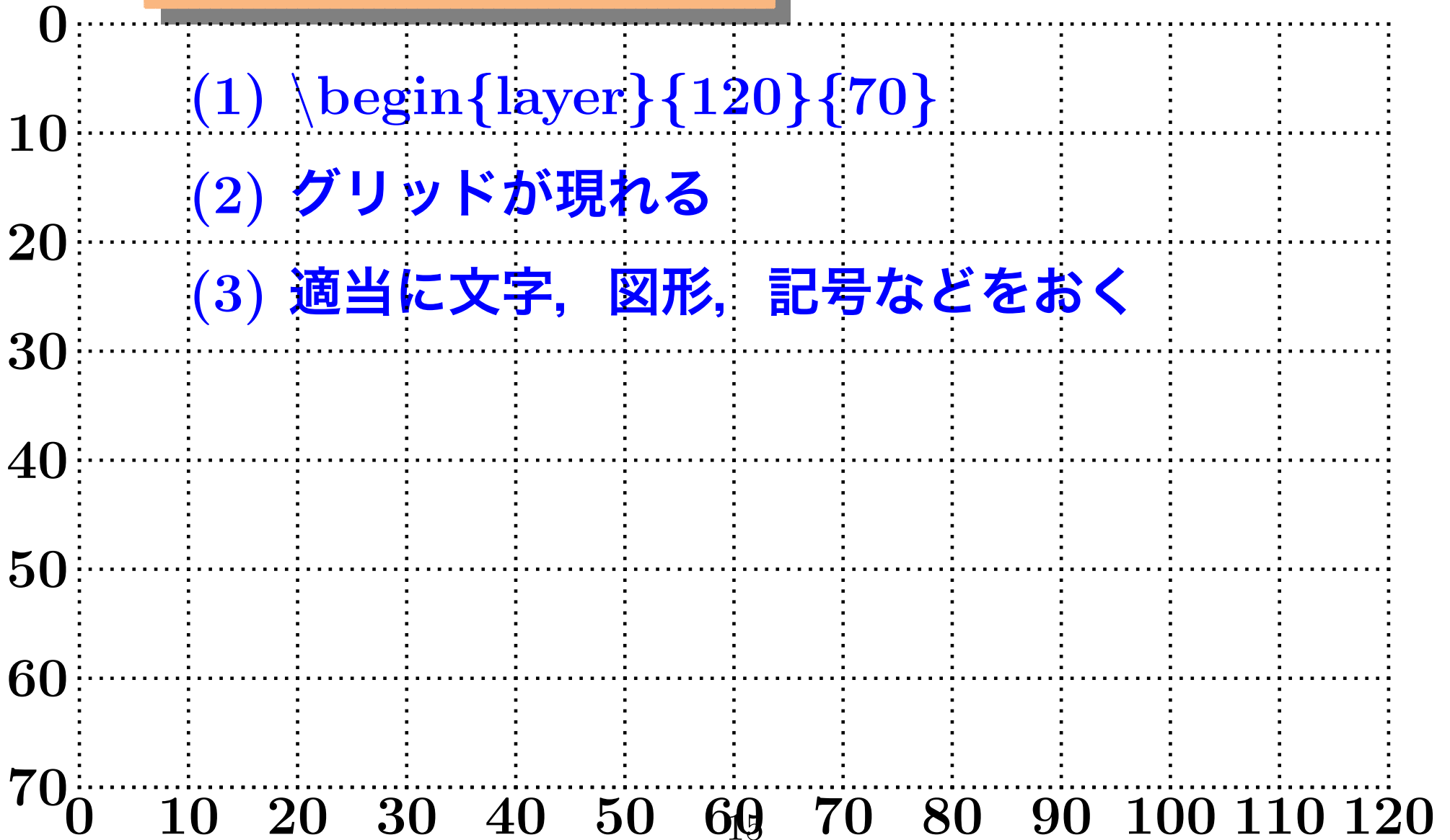


layer 環境の使用法

(1) `\begin{layer}{120}{70}`

(2) グリッドが現れる

(3) 適当に文字, 図形, 記号などをおく



layer 環境の使用法

(1) `\begin{layer}{120}{70}`

(2) グリッドが現れる

(3) 適当に文字, 図形, 記号などをおく

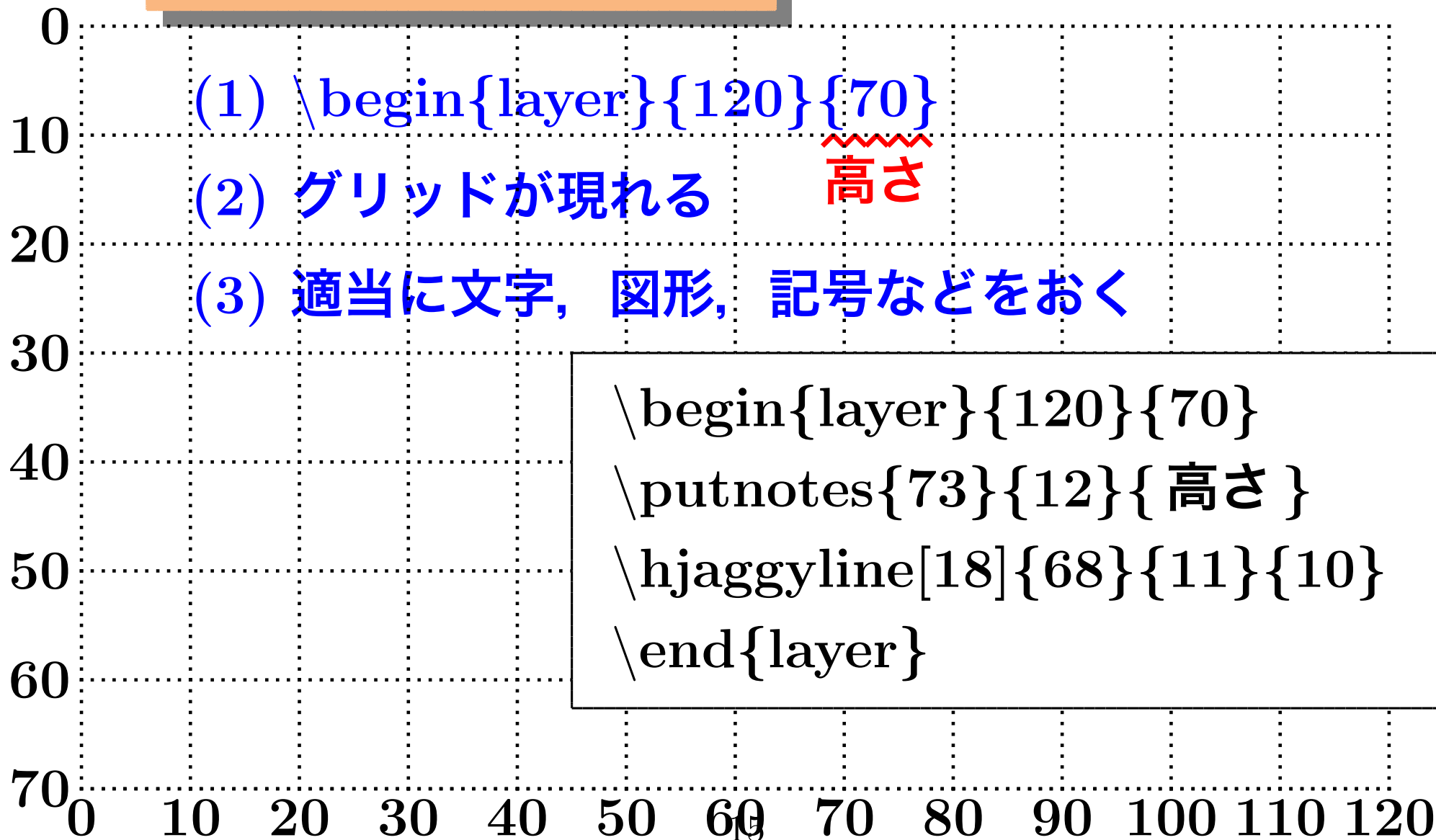
```
\begin{layer}{120}{70}
```

```
\putnotes{73}{12}{高さ}
```

```
\hjaggyline[18]{68}{11}{10}
```

```
\end{layer}
```

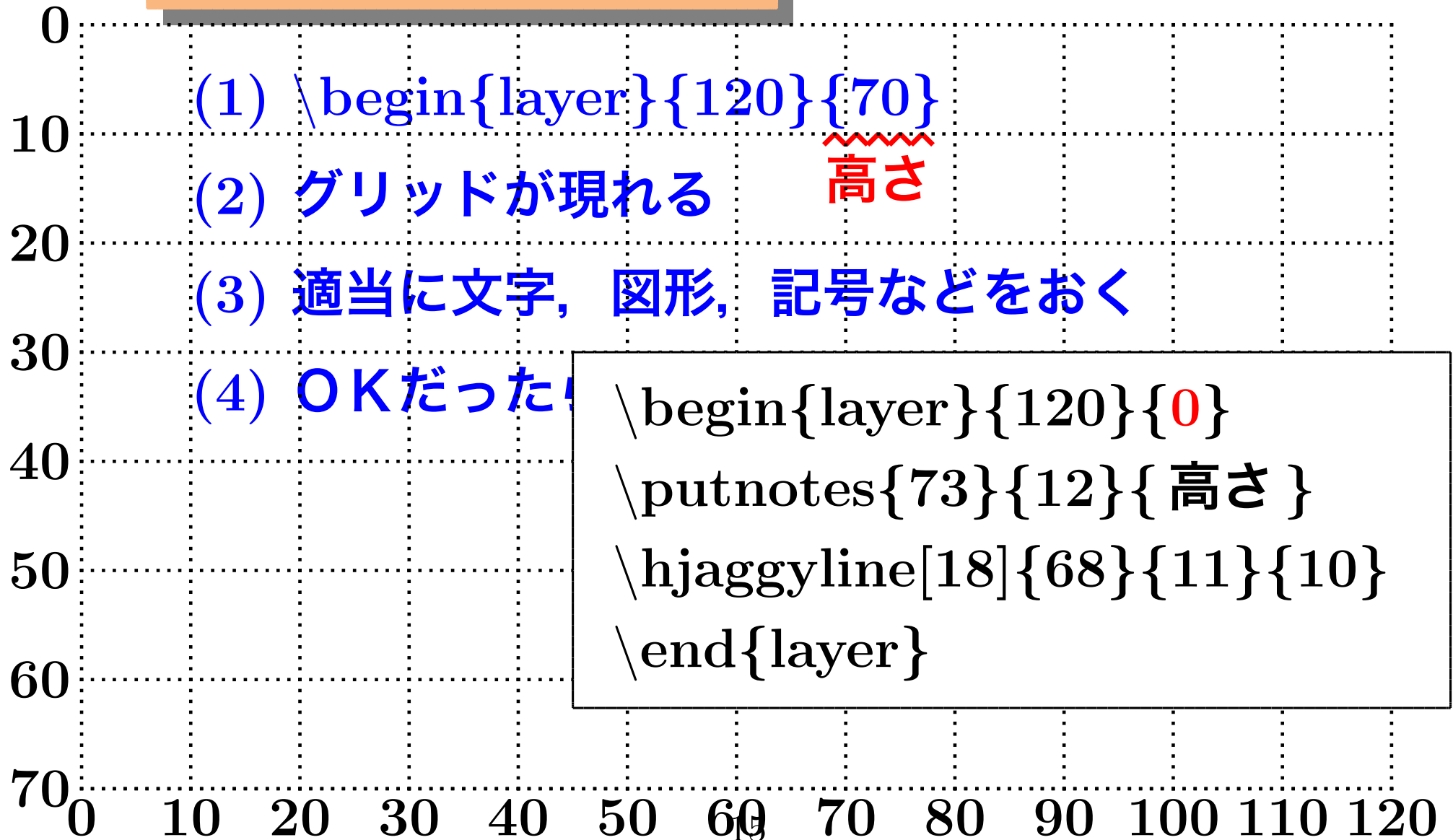
layer 環境の使用法



layer 環境の使用法



layer 環境の使用法



layer 環境の使用法

- (1) `\begin{layer}{120}{70}`
- (2) グリッドが現れる 高さ
- (3) 適当に文字, 図形, 記号などをおく
- (4) OKだったら, 高さを 0 にする
- (5) グリッドが消える

まとめ

- (1) K_ET_pic は T_EX 作図用に開発された.
- (2) 最近, メタコマンドを追加した.
- (3) 作図コマンドとメタコマンドのコンビネーションで, T_EX の「ふつうの」ユーザーでも, いろいろなマクロの作成が可能になった.
- (4) メタコマンドは, まだ機能上の制約も多い.
- (5) しかし, K_ET_pic は単なる T_EX 作図を超えたツールになったと言える.

ご清聴ありがとうございました

Děkuji vám

