情報系の物理学 演習 8

G99P102-8 永田貴彦

出題日　:2001 年　1 月 10 日
提出期限:2001 年　1 月 31 日
提出日　:2001 年　1 月 31 日

1　問題

$$u(t,x) = -2\frac{\partial^2}{\partial x^2}\log\left(1+e^{kx-k^3t+c}\right)$$

を計算して、u をハイパボリックセカンドで表し、アニメーションでその動きをシミュレートせよ。

2　式の計算

$$\begin{aligned}
u(t,x) &= -2\frac{\partial^2}{\partial x^2}\log\left(1+e^{kx-k^3t+c}\right) \\
&\quad 1+e^{kx-k^3t+c} = T \text{ とおく} \\
&= -2\frac{\partial}{\partial x}\frac{\partial T}{\partial x}\frac{\partial}{\partial T}\log(T) \\
&= -2\frac{\partial}{\partial x}\frac{\partial T}{\partial x}\frac{1}{T} \\
&\quad \frac{\partial T}{\partial x} = ke^{kx-k^3t+c} = k(T-1) \text{ より} \\
&= -2\frac{\partial}{\partial x}\frac{k(T-1)}{T} \\
&= -2k\frac{\partial T}{\partial x}\frac{\partial}{\partial T}\left(1-\frac{1}{T}\right) \\
&= -2k^2(T-1)\frac{1}{T^2} \\
&= -2k^2\frac{e^{kx-k^3t+c}}{\left(1+e^{kx-k^3t+c}\right)^2}
\end{aligned}$$

$$
\begin{aligned}
&= -2k^2 \frac{1}{\left(e^{\frac{1}{2}(kx-k^3t+c)} + e^{\frac{1}{2}(kx-k^3t+c)}\right)^2} \\
&\quad \cosh(z) = \frac{e^z + e^{-z}}{2} \quad \mathrm{sech}(z) = \frac{1}{\cosh(z)} \\
&= -\frac{k^2}{2}\mathrm{sech}^2\left(\frac{1}{2}(kx - k^3t + c)\right)
\end{aligned}
$$

3 プログラムの概略

今回は、動きをシミュレートしなければいけないため、Java でアプレットを作ることにした。

3.1 u(t,x)

$$u(t,x) = -\frac{k^2}{2}\mathrm{sech}^2\left(\frac{1}{2}(kx - k^3t + c)\right)$$

この式の通りにプログラムを書く。

```
private double u(double x,double t) {
  return (-k*k*Math.pow(sech((k * x - k*k*k*t + c)/2),2) / 2);
}
```

また、**sech** は基本 API にないため、これも追加する。

```
private double sech(double z) {
  return (2 / (Math.pow(Math.E,z) + Math.pow(Math.E,-z)));
}
```

3.2 波の描画

なるべく滑らかな波を描きたいため、x 軸方向に 1 ピクセルごと動かしながら、その点における y 座標を求めることにする。

したがって、描画処理を

```
for (int i=0;i<getSize().width;i++) {
（描画処理);
}
```

という**for** 文でくくってやる。

次に、y 座標を求めるわけであるが、時間**t** は**run()** 中で、**Thread** によって、処理されている。

x 軸方向は、1 ピクセル = 1 として考え、更に、何倍かに拡大して描画し、中心を 0 としている。

```
x = (i - cw) / wTime;
```

ここで、**cw** は中心の座標、**wTime** は x 軸方向の倍率である。これで、座標変換が終わり、y 軸を求めればいい。

今回は、波を幾つか重ねるため、変数**k** の値をずらしながら、y 軸方向に重ねていく。ここで、y 軸は下向きが正方向であるため、実際のグラフとは上下が逆になる。ただ、この KdV 方程式は $y <= 0$ となるため見た目は正方向にでてくるからそのままにしておく。

```
for (k = 1;k<=maxK;k++) {
  d += u(x,t);
}
```

ここで、**maxK** は波の数で、maxK = 5 なら、波が 5 つ表示される。今回は**KdV** 方程式の変数**k** を整数で、1 から指定しているが、実際には負でも少数でもいい。ただ、整数のほうがプログラムがしやすいからである。

また、d は、y 軸方向の値である。

これを、実際に描画する。この際、前回の値をとっておき、線を引くことで、滑らかに見えるようにする。

```
OSG.drawLine(i-1,(int)(hTime*dt)+ch,i,(int)(hTime*d)+ch);
```

ch は y 軸方向の中心である。また、**hTime** は y 軸方向の倍率であり、**dt** は前回の値である。

全体としては以下のようになる。

```
for (int i=0;i<getSize().width;i++) {
  d = 0;
  x = (i - cw) / wTime;
  for (k = 1;k<=maxK;k++) {
    d += u(x,t);
  }
  OSG.drawLine(i-1,(int)(hTime*dt)+ch,i,(int)(hTime*d)+ch);
  dt = d;
}
```

プログラム全体としては、**Soliton.java** を参照してください。

4　動作結果

Soliton.html を、**appletviewer** または**Browser** で開いて下さい。

動作確認した環境

- Windows2000+JDK1.3+appletviewer
- Windows2000+JDK1.3+Internet Explorer 5.01