Programmierpraktikum

Claudius Gros, SS2012

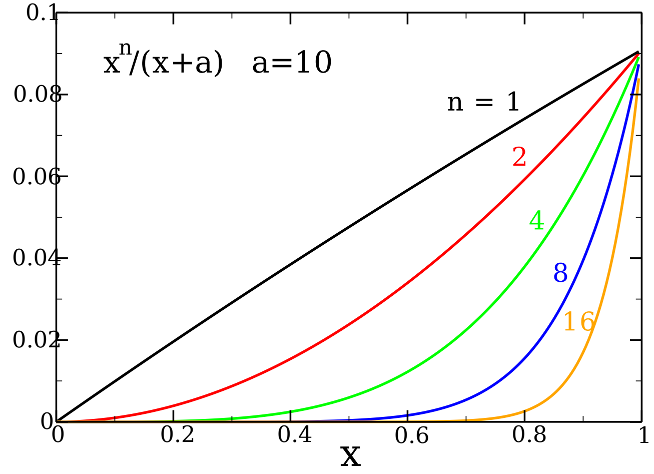
Institut für theoretische Physik
Goethe-University Frankfurt a.M.

Summation, Recursion and Stablilty

problem setting

- evaluation of

$$y_n = \int_0^1 \frac{x^n}{x+a}\, dx$$



- by decomposition and summation
- by recursion and inverse recursion
- for various n and $a \gg 1$

decomposition into a series

$$y_n \;=\; \int_0^1 \frac{x^n}{x+a}\,dx$$

- term decomposition

$$\frac{x^n}{x+a} \;=\; \frac{x^n-(-a)^n+(-a)^n}{x+a} \;=\; \frac{x^n-(-a)^n}{x-(-a)} \;+\; \frac{(-a)^n}{x+a}$$

- polynomial devision

$$\frac{x^n-(-a)^n}{x-(-a)} \;=\; x^{n-1} \;+\; (-a)x^{n-2} \;+\; (-a)^2x^{n-3} \;+\; \ldots \;+\; (-a)^{n-2}x \;+\; (-a)^{n-1}$$

- integration of partial terms

$$y_n \;=\; \frac{1}{n} \;+\; \frac{(-a)}{n-1} \;+\; \frac{(-a)^2}{n-2} \;+\; \ldots \;+\; \frac{(-a)^{n-2}}{2} \;+\; \frac{(-a)^{n-1}}{1} \;+\; (-a)^n\,\log\Bigg(\frac{1+a}{a}$$

careful when subtracting large terms

- summing up and down yields different results for $n > 2$
- substraction of very large terms leads to large relative errors
- $a = 10$

```
n, y_n (up/down):     0     0.0953101798043249     0.0953101798043249
n, y_n (up/down):     1     0.0468982019567507     0.0468982019567507
n, y_n (up/down):     2     0.0310179804324928     0.0310179804324928
n, y_n (up/down):     3     0.0231535290083968     0.0231535290084016
n, y_n (up/down):     4     0.0184647099159747     0.0184647099160125
n, y_n (up/down):     5     0.0153529008402984     0.0153529008389668
n, y_n (up/down):     6     0.0131376582721714     0.0131376582624891
n, y_n (up/down):     7     0.0114805602934212     0.0114805602325932
n, y_n (up/down):     8     0.0101943966001272     0.0101943983716508
n, y_n (up/down):     9     0.0091671049594879     0.0091671236256787
n, y_n (up/down):    10     0.0083286762237549     0.0083287335918189
n, y_n (up/down):    11     0.0076236724853516     0.0076222290336939
n, y_n (up/down):    12     0.0071258544921875     0.0071148651228605
n, y_n (up/down):    13     0.0057373046875000     0.0057442655768129
n, y_n (up/down):    14     0.0156250000000000     0.0142329159691716
n, y_n (up/down):    15    -0.0937500000000000    -0.0717638696291832
n, y_n (up/down):    16     0.8750000000000000     0.7797726530809349
n, y_n (up/down):    17    -6.0000000000000000    -7.9918313757219790
n, y_n (up/down):    18    80.0000000000000000    67.9816784898641800
n, y_n (up/down):    19  -512.0000000000000000  -807.3893324901815000
n, y_n (up/down):    20  4096.0000000000000000  6540.9419220573470000
```

relative errors

- numbers/measurements $a_\alpha,\ b_\beta$
  with small relative errors $\alpha,\ \beta$

$$a_\alpha = a\,(1 \pm \alpha) \qquad b_\beta = b\,(1 \pm \beta) \qquad a, b, \alpha, \beta > 0 \qquad \alpha, \beta \ll 0$$

- substracting two large numbers $a, b \gg 0$

$$a_\alpha - b_\beta = (a - b) \pm (a\,\alpha + b\,\beta) = (a - b)\left(1 \pm \frac{a\,\alpha + b\,\beta}{|a - b|}\right)$$

- may lead to very large relative errors

$$\boxed{\frac{a\,\alpha + b\,\beta}{|a - b|}}$$

- for $a \approx b$

straightforward recursion

$$y_n = \int_0^1 \frac{x^n}{x+a}\,dx = \int_0^1 \frac{x^{n-1}(x+a-a)}{x+a}\,dx = \int_0^1 x^{n-1}\,dx - a\int_0^1 \frac{x^{n-1}}{x+a}\,dx$$

- recursion relation

$$y_n = \frac{1}{n} - a\,y_{n-1}, \qquad y_0 = \log\left(\frac{a+1}{a}\right)$$

- recursion is unstable

```
 n      forward recursion
 0     0.0953101798043249
 1     0.0468982019567507
 2     0.0310179804324935
 3     0.0231535290083985
 4     0.0184647099160155
 5     0.0153529008398455
 6     0.0131376582682119
 7     0.0114805601750236
 8     0.0101943982497636
 9     0.0091671286134746
10     0.0083287138652537
11     0.0076219522565537
12     0.0071138107677960
13     0.0057849692451174
14     0.0135788789773972
15    -0.0691221231073049
16     0.7537212310730486
```



```
17    -7.4783887813187210
18    74.8394433687427600
19  -748.3418021084802000
20  7483.4680210848010000
```

stability analyis of recursion formula

$$y_n \;=\; -\,a\,y_{n-1} \,+\, \frac{1}{n}$$

small perturbation

consider time evolution of two closed-by trajetories: $y_n,\ \tilde{y}_n$ $\qquad y_n - \tilde{y}_n = \epsilon_n \ll 1$

$$\epsilon_n \;=\; -\,a\,\epsilon_{n-1}$$

- unstable (exponentially) for $|-a| > 1$
- Lyapunov exponent $\lambda > 0$

$$e^{\lambda} \;\equiv\; |-a|$$

- error growth exponentially with number n of iterations (butterfly effect)

$$\epsilon_n = e^{n\lambda}\epsilon_0$$

dynamical system theory
dynamical systems with positive (negative) Lyapunov exponents are denoted chaotic (regular)

forward vs backward recursion

$$y_n \;=\; -a\,y_{n-1} \;+\; \frac{1}{n}\,, \qquad\qquad y_{n-1} \;=\; \frac{1}{a}\left(\frac{1}{n}-y_n\right)$$

- error $\epsilon_n \ll 1$

$$y_n \;\to\; y_n \pm \epsilon_n\,, \qquad\qquad y_{n-1} \pm \epsilon_{n-1} = \frac{1}{a}\left(\frac{1}{n}-y_n \pm \epsilon_n\right)$$

- stable for $a > 1$

$$\epsilon_{n-1} = \frac{\epsilon_n}{a}$$

- starting with $y_{50} = 0$

n	forward recursion	backward recursion
0	0.0953101798043249	0.0953101798043249
1	0.0468982019567507	0.0468982019567514
2	0.0310179804324935	0.0310179804324860
3	0.0231535290083985	0.0231535290084733
4	0.0184647099160155	0.0184647099152671
5	0.0153529008398455	0.0153529008473289
6	0.0131376582682119	0.0131376581933773
7	0.0114805601750236	0.0114805609233700

```
  8     0.0101943982497636     0.0101943907661000
  9     0.0091671286134746     0.0091672034481114
 10     0.0083287138652537     0.0083279655188863
 11     0.0076219522565537     0.0076294357202278
 12     0.0071138107677960     0.0070389761310555
 13     0.0057849692451174     0.0065333156125223
 14     0.0135788789773972     0.0060954153033482
 15    -0.0691221231073049     0.0057125136331850
 16     0.7537212310730486     0.0053748636681501
 17    -7.4783887813187210     0.0050748927302638
 18    74.8394433687427600     0.0048066282529171
 19  -748.3418021084802000     0.0045652964181972
```

summation/recursion code

- all four methods for evaluating $y_n = \int_0^1 \frac{x^n}{x+a}\,dx$
- examine the dependence of the results on the value of a
- what happens for $a \in]0,1[$?

```
import java.lang.Math;   // standard import, always useful
/** Summing up terms may lead to cancellation issues.
  * y_n = \int_0^1 x^n/(x+a)dx
  */
public class InteMultiMethods {

public static void main(String[] args) {
  int maxN = 21;
  double a = 10.0;
  System.out.printf("\na = %6.1f\n\n",a);
// *** ***************** ***
// *** summation of terms ***
// *** ***************** ***
  for (int n=0; n<maxN; n+=2)
    {
    double resultSumUp   = performSumUp(n,a);
```



```java
    double resultSumDown = performSumDown(n,a);
    System.out.printf("n, y_n (up/down):   %3d %22.16f %22.16f\n",
                      n,resultSumUp,resultSumDown);
    }
  System.out.println(" ");
// *** ****************** ***
// *** recursion relation ***
// *** ****************** ***
  System.out.printf("%3s %22s %22s\n","n",
                    "forward recursion","backward recursion");
  for (int n=0; n<maxN; n+=2)
    {
    double resultForward  = forwardRecursion(n,a);
    double resultBackward = backwardRecursion(n,a);
    System.out.printf("%3d %22.16f %22.16f\n",
                      n,resultForward,resultBackward);
    }
}  // end of InteMultiMethods.main

/** Straight summing up of integrated terms.
  */
public static double performSumUp(int N, double a) {
  double result = Math.pow(-a,1.0*N)*Math.log((1+a)/a);
  for (int i=0; i<N; i++)
    result +=  Math.pow(-a,1.0*i)/(N-i);
  return result;
```

```
  }    //  end of summationUP

/** Straight summing down of integrated terms.
  */
public static double performSumDown(int N, double a) {
  double result = Math.pow(-a,1.0*N)*Math.log((1+a)/a);
  for (int i=N-1; i>=0; i--)
    result +=  Math.pow(-a,1.0*i)/(N-i);
  return result;
  }    //  end of summationDown

/** Solving the forward recursion.
  */
public static double forwardRecursion(int N, double a) {
  double result = Math.log((1+a)/a);
  for (int i=1; i<=N; i++)
    result =  -a*result + 1.0/i;
  return result;
  }    //  end of forwardRecursion

/** Solving the backward recursion.
  */
public static double backwardRecursion(int N, double a) {
  int startN = 50;
  double result = 0.0;
  for (int i=startN; i>N; i--)
```

```
    result = (1.0/i-result)/a;
  return result;
  }    //  end of forwardRecursion

}    //  end of class InteMultiMethods
```