SHANGHAI JIAO TONG UNIVERSITY, UM-SJTU JOINT INSTITUTE

Introduction to Numerical Analysis
HW6

Yu Cang
018370210001

July 2, 2018

1 RICHARDSON EXTRAPOLATION

1. *Proof.* When $k = 0$, it's clear that $A_0(t) = a_0 + O(t)$.
   Assume that the statement is still valid when $k = n-1$, which means $A_{n-1} = a_0 + O(t^n)$
   Thus, when $k = n$, apply the induction hypothesis

$$\begin{aligned} A_n(t) &= \frac{r^n A_{n-1}(t) - A_{n-1}(rt)}{r^n - 1} \\ &= \frac{r^n(a_0 + O(t^n)) - (a_0 + r^n O(t^n))}{r^n - 1} \\ &= a_0 + \frac{r^n}{r^n - 1}(O(t^n) - O(t^n)) \\ &= a_0 + O(t^{n+1}) \end{aligned} \tag{1.1}$$

   Hence, the statement is valid. □

2. a) *Proof.* When $t_m = \frac{t_0}{r_0^m}$, and $r_0 > 1$

$$\lim_{m\to\infty} A_n(t_m) = a_0 + O((\frac{t_0}{r_0^m})^{n+1}) = a_0 + \frac{O(t_0^{n+1})}{r_0^{m(n+1)}} = 0 \tag{1.2}$$

   The equation above is valid as $O(t_0^{n+1})$ is finite. □

b) *Proof.* As $O(t_0^{n+1})$ is finite, it's clear that

$$A_n(t_m) = a_0 + O((\frac{t_0}{r_0^m})^{n+1}) = a_0 + O(r_0^{-m(n+1)}) \tag{1.3}$$

$\square$

3. Algorithm used to do extrapolation is given as below.

Algorithm 1 Calculation of the extrapolation matrix and the improved quadrature

Input: $A(t)$, n, m, a, b, r_0
Output: M, a_0

$h \leftarrow b - a$
$i \leftarrow 0$
while $i \neq m$ **do**
  $M[i][0] \leftarrow A(\frac{h}{2^i})$
  $i \leftarrow i + 1$
end while
$j \leftarrow 1$
while $j \neq n$ **do**
  $i \leftarrow j$
  while $i \neq m$ **do**
    $M[i][j] \leftarrow \frac{r_0^i}{r_0^i - 1} M[i][j-1] - \frac{1}{r_0^i - 1} M[i-1][j-1]$
    $i \leftarrow i + 1$
  end while
  $j \leftarrow j + 1$
end while
$a_0 \leftarrow M[m-1][n-1]$
return M, a_0

4. a) With proper combination, lower-order terms inside original quadratures can be eliminated when generating the new quadrature.

   b) Consider the Runge's Function

$$f(x) = \frac{1}{1 + 25x^2} \tag{1.4}$$

(to be added...)

2 INTEGRATION

3 GAUSS'S METHOD

1. a) *Proof.* It's clear that $w(x)$ is positive.
   And

$$\int_{-1}^{1} w(x)dx = \int_{-1}^{1} \sqrt{1 - x^2}dx = \int_{-\pi/2}^{\pi/2} cos^2\theta d\theta = \frac{\pi}{2} \tag{3.1}$$

Thus, $w(x)$ is a weight function. $\square$

b) *Proof.* Since

$$\int_{-1}^{1} w(x)q_m(x)q_n(x)dx = \int_{\pi}^{0} \sqrt{1-cos^2\theta}\frac{sin(m+1)\theta}{sin\theta}\frac{sin(n+1)\theta}{sin\theta}(-sin\theta)d\theta$$
$$= \int_{0}^{\pi} sin(m+1)\theta sin(n+1)\theta d\theta \tag{3.2}$$

It's clear that when $n = m$, the value of the equation above is $\frac{\pi}{2}$, and is 0 when $n \neq m$. Thus, q_k ($k \in \mathbb{N}$) defines a sequence of orthogonal polynomials for weight function $w(x)$. $\square$

c)

$$p_k(x) = \sqrt{\frac{2}{\pi}}q_k(x) \tag{3.3}$$

2. a) x_k are roots of orthogonal polynomial over $[a, b]$, whose order is $n+1$ and weight is $w(x)$.

b) With the Lagrange interpolation function $f(x) = \sum_{i=0}^{n} l_i(x)f(x_i)$, A_k can be determined as

$$A_k = \int_{-1}^{1} l_k(x)w(x)dx \tag{3.4}$$

As $x_k = cos(\frac{k\pi}{n+2})$, it can be further resolved that $A_k = \frac{\pi}{n+2}sin^2\frac{(k+1)\pi}{n+2}$.

c) *Proof.* Apply Hermitan interpolation polynomials $H_{2n+1}(x)$ onto $f(x_k)$ s.t. for $k = 0, 1, ..., n$

$$H_{2n+1}(x_k) = f(x_k) \tag{3.5}$$

and

$$H'_{2n+1}(x_k) = f'(x_k) \tag{3.6}$$

Thus

$$f(x) = H_{2n+1}(x) + \frac{f^{(2n+2)}(\xi)}{(2n+2)!}\omega^2(x) \tag{3.7}$$

Then integrate over $[-1, 1]$ with weight $w(x)$

$$I = \int_{-1}^{1} f(x)w(x)dx = \int_{-1}^{1} H_{2n+1}(x)w(x)dx + R_n[f] \tag{3.8}$$

Thus

$$R_n[f] = I - \sum_{k=0}^{n} A_k f(x_k) = \int_{-1}^{1} \frac{f^{(2n+2)}(\xi)}{(2n+2)!}\omega^2(x)w(x)dx \tag{3.9}$$

Since $\omega^2(x)w(x) \geq 0$, it can be concluded from the first intermediate value theorem that

$$R_n[f] = \frac{f^{(2n+2)}(\eta)}{(2n+2)!}\int_{-1}^{1} \omega^2(x)w(x)dx \triangleq c\frac{f^{(2n+2)}(\eta)}{(2n+2)!} \tag{3.10}$$

$\square$