CP315 - Portfolio Part 1

Mason Cooper
Wilfrid Laurier University

February 26, 2017

Contents

1 Horner's Scheme

The following is my code for Horner's scheme:

$horner := proc(f, x)$
$locald, h; h := f;$
$d := degree(h);$
$numapprox[hornerform](select(u- > degree(u, x) <= dendproc, h), x)$
$endproc$

- $x^8 - x^7 + x^4 - x^3 + x + 1$

$$1 + (1 + (-1 + (1 + (-1 + x)x^3)x)8x^2)x$$

- $x^{400} - x^{300} + x^{200} - x^{100} + 1$

$$1 + (-1 + (1 + (x^1 00 - 1)x^{100})x^{100})x^{100}$$

- $x^1 4 + x^1 1 + x^8 + x^5$

$$(1 + (1 + (x^3 + 1)x^3)x^3)x^5$$

2 Taylor's theorem

My Taylor's theorem code,

$mytaylor := proc(f, x0, d) local r, i;$
$r := eval(f, x = x0);$
$for i to d do r := r + (eval(diff(f, x i),$
$x = x0)) * (x - x0)^i / factorial(i)$
$end do$
$end proc$

The following is the output from my code for Taylor's theorem and the output from Maple's built in Taylor command for comparison:

- $f(x) = cos(x)$, up to degree $k = 10$, around $x_0 = 0$

$$1 - \frac{1}{2}x^2 + \frac{1}{24}x^4 - \frac{1}{720}x^6 + \frac{1}{40320}x^8 - \frac{1}{3628800}x^{10}$$

$$1 - \frac{1}{2}x^2 + \frac{1}{24}x^4 - \frac{1}{720}x^6 + \frac{1}{40320}x^8 + O(x^{10})$$

- $f(x) = e^x$, up to degree $k = 10$, around $x_0 = 0$

$$1 + ln(e)x + \frac{1}{2}ln(e)^2x^2 + \frac{1}{6}ln(e)^3x^3 + \frac{1}{24}ln(e)^4x^4 + \frac{1}{120}ln(e)^5x^5$$

$$+\frac{1}{720}ln(e)^6x^6+\frac{1}{5040}ln(e)^7x^7+\frac{1}{40320}ln(e)^8x^8+\frac{1}{3628800}ln(e)^9x^9+\frac{1}{3628800}ln(e)^{10}x^{10}$$

$$1 + ln(e)x + \frac{1}{2}ln(e)^2x^2 + \frac{1}{6}ln(e)^3x^3 + \frac{1}{24}ln(e)^4x^4 + \frac{1}{120}ln(e)^5x^5$$

$$+\frac{1}{720}ln(e)^6x^6+\frac{1}{5040}ln(e)^7x^7+\frac{1}{40320}ln(e)^8x^8+\frac{1}{3628800}ln(e)^9x^9+O(x^{10})$$

3 Polynomials

My code for sign change,

$signchange := proc(f)$
$localmax, min, i, h, result, t, s; h := f;$
$result := [];$
$max := 100;$
$min := -100;$
$t := eval(f, x = min);$
$forifrommintomaxdo$
$s := eval(f, x = i);$
$ifs = 0then$
$result := [op(result), [i]];$
$s := eval(f, x = i + 1);$
$if0 < tand0 < sort < 0ands < 0then$
$result := [op(result), [i]]$
$else$
$t := s$
$endif$
$elif0 < tands < 0then$
$result := [op(result), [i - 1, i]];$
$t := s$
$elift < 0and0 < sthen$
$result := [op(result), [i - 1, i]];$
$t := s$
$endif$
$enddo;$
$returnresult$
$endproc$

My Vieta formulae,

$vieta := proc(f, i)$
$localj, d, a, h, s;$
$h := f;$
$a := [];$
$d := degree(f);$
$forjfrom0toddo$
$a := [op(a), coeff(h, x, j)]$
$enddo;$
$d := d + 1;$
$s := (-1)^i * \frac{a[d-i]}{a[d]};$
$returns$
$endproc$

With my sign change code, the function $x^3 - 44x^2 + 564x - 1728$ had sign changes at,

$$[[4, 5], [18], [21, 22]]$$

Using Vieta's formulae I was able to compute the following,

$$p_1 + p_2 + p_3 = 44$$

$$p_1 p_2 p_3 = 1728$$

$$p_1 p_2 + p_1 p_3 + p_2 p_3 = 564$$

$$p_1^2 + p_2^2 + p_3^2 = 808$$

$$p_1^3 + p_2^3 + p_3^3 = 15920$$

$$\frac{1}{p_1^2} + \frac{1}{p_2^2} + \frac{1}{p_3^2} = 808$$

4 Equation Solving

4.1 Bisection method

My code for bisection method,

$bisection := proc(l, h, f)$
$locala, b, c, m, n;$
$Digits := 4;$
$a := l;$
$b := h;$
$while \frac{1}{10000} < (1/2) * b - (1/2) * ado$
$c := \frac{a+b}{2};$
$m := evalf(eval(f, x = c));$
$n := evalf(eval(f, x = a));$
$ifm = 0then$
$returnr = c$
$endif;$
$ifn * m < 0thenb := c$
$else$
$a := c$
$endif$
$enddo;$
$returnr = evalf(\frac{a+b}{2})$
$endproc$

The following is output of my code for bisection method,

- $f(x) = x^3 + x - 1, x \in [0, 1]$

$$r = .6823$$

- $f(x) = cosx - x, x \in [0, 1]$

$$r = .7391$$

4.2 Fixed-point iteration method

My code for Fixed-point iteration method,

$FPI := proc(f, x0, max)$
$localp0, p1, i;$
$Digits := 9;$
$p0 := evalf(x0);$
$p1 := p0;$
$forifrom0tomaxdo$
$p0 := p1;$
$p1 := eval(f, x = p0);$
$i := i + 1$
$enddo;$
$returnp1$
$endproc$

The following is output of my code for Fixed-point iteration method,

- $g(x) = \frac{(x+\frac{2}{x})}{2}$
$$1.41421356$$
- $x^3 = 2 * x + 2$
$$1.76931316$$

4.3 Newton's method

My code for Newton's method,

$newton := proc(f, x0, max)$
$localp0, p1, i;$
$Digits := 9;$
$h := f;$
$i := 0;$
$p0 := evalf(x0);$
$p1 := p0;$
$whilei < maxdo$
$p0 := p1;$
$p1 := p0 - (\frac{eval(h,x=p0))}{(eval((h',x),x=p0))};$
$i := i + 1$
$enddo;$
$returnp1$
$endproc$

The following is output of my code for Newton's method,

- $f(x) = x^3 + x - 1, x_0 = 0.1$

$$0.682327804$$

- $x^2 + 1 = 0, x_0 = 2]$

$$2.40087675$$

5 Systems of Linear Equations

5.1 Gauss Elimination

My code for Gauss Elimination method,

$GaussElimination := proc(A)$
$localB, X, i, j, k, d, c;$
$B := A;$
$d := Dimension(B);$
$X := [seq(x[i], i = 1..d[1])];$
$forjtod[1]doforitod[1]do$
$ifj < ithen$
$c := \frac{B[i,j]}{B[j,j]};$
$forktod[2]do$
$B[i, k] := B[i, k] - c * B[j, k]$
$enddo$
$endif$
$enddo$
$enddo;$
$X[d[1]] := \frac{B[d[1],d[1]+1]}{B[d[1],d[1]]};$
$forifromd[1] - 1by - 1to1do$
$c := 0;$
$forjfromi + 1tod[1]do$
$c := c + B[i, j] * X[j]$
$enddo;$
$X[i] := \frac{B[i,d[2]]-c}{B[i,i]}$
$enddo;$
$returnX$
$endproc$

The following is output of my code for Gauss Elimination,

$$[1, 2, 4]$$

5.2 Gauss-Jordan Elimination

My code for Gauss-Jordan Elimination method,

$GaussJordan := proc(A, n)$
$locali, j, B, k;$
$B := A;$
$foritondo$
$forjfromi + 1tondo$
$ifabs(B[i, i]) < abs(B[j, i])$
$thenB[i, j] = B[j, i]$
$endif$
$enddo;$
$B[i] := \frac{B[i]}{B[i,i]};$
$forktondo$
$ifk <> i$
$thenB[k] := -B[i] * B[k, i] + B[k]$
$endif$
$enddo$
$enddo;$
$returnB$
$endproc$

The following is output of my code for Gauss-Jordan Elimination,

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 4 \end{bmatrix}$$

5.3 The Hilbert matrix

The following is output of my code for Gauss Elimination and Gauss-Jordan Elimination on the Hilbert matrix,

Gauss Elimination, $H_5 \cdot \vec{x} = \vec{1}$

$$Determinantis, \frac{1}{266716800000}$$

$$[5, -120, 630, -1120, 630]$$

Gauss-Jordan Elimination, $H_5 \cdot \vec{x} = \vec{1}$ and Determinant,

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 5 \\ 0 & 1 & 0 & 0 & 0 & -120 \\ 0 & 0 & 1 & 0 & 0 & 630 \\ 0 & 0 & 0 & 1 & 0 & -1120 \\ 0 & 0 & 0 & 0 & 1 & 630 \end{bmatrix}$$

Gauss-Jordan Elimination, H_5^{-1}

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 25 & -300 & 1050 & -1400 & 630 \\ 0 & 1 & 0 & 0 & 0 & -300 & 4800 & -18900 & 26880 & -12600 \\ 0 & 0 & 1 & 0 & 0 & 1050 & -18900 & 79380 & -117600 & 56700 \\ 0 & 0 & 0 & 1 & 0 & -1400 & 26880 & -117600 & 179200 & -88200 \\ 0 & 0 & 0 & 0 & 1 & 630 & -12600 & 56700 & -88200 & 44100 \end{bmatrix}$$

Condition Number,

1-Form
$Multiply(MatrixNorm(H, 1), MatrixNorm(MatrixInverse(H), 1));$

$$943656$$

Frobenius
$Multiply(MatrixNorm(H, Frobenius), MatrixNorm(MatrixInverse(H), Frobenius))$

$$\frac{1}{504}\sqrt{(}3700542505)\sqrt{(}15871330)$$