In the name of God

Lin-Bairstow Method

Compiled by Naser Bagheri
Student ID : 9016393

Module

for

The Lin-Bairstow Method

Quadratic Synthetic Division

Let the polynomial $P(x)$ of degree n have coefficients $\{a_i\}_{i=0}^{n}$. Then $P(x)$ has the familiar form

(1) $$P(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_k x^k + \ldots + a_3 x^3 + a_2 x^2 + a_1 x + a_0.$$

Let $T(x) = x^2 - r x - s$ be a fixed quadratic term. Then $P(x)$ can be expressed as

(2) $$P(x) = (x^2 - r x - s) Q(x) + u(x - r) + v,$$

where $R(x) = u(x - r) + v$ is the remainder when $P(x)$ is divided by $T(x) = (x^2 - r x - s)$. Here $Q(x)$ is a polynomial of degree $n - 2$ and can be represented by

(3) $$Q(x) = b_n x^{n-2} + b_{n-1} x^{n-3} + \ldots + b_4 x^2 + b_3 x + b_2.$$

If we set $b_1 = u$ and $b_0 = v$, then

(4) $$P(x) = (x^2 - r x - s) Q(x) + R(x),$$

where

(5) $$R(x) = b_1 (x - r) + b_0$$

and equation (4) can be written

(6) $$P(x) = (x^2 - r x - s)(b_n x^{n-2} + b_{n-1} x^{n-3} + \ldots + b_4 x^2 + b_3 x + b_2) + b_1 (x - r) + b_0.$$

The terms in (6) can be expanded so that $P(x)$ is represented in powers of x.

(7) $$P(x) = b_n x^n + (b_{n-1} - r b_n) x^{n-1} + (b_{n-2} - r b_{n-1} - s b_n) x^{n-2} + \ldots + (b_k - r b_{k+1} - s b_{k+2}) x^k + \ldots + (b_3 - r b_4 - s b_5) x^3 + (b_2 - r b_3 - s b_4) x^2 + (b_1 - r b_2 - s b_3) x + (b_0 - r b_1 - s b_2)$$

The numbers b_k are found by comparing the coefficients of x^k in equations (1) and (7). The coefficients $\{b_k\}_{k=0}^{n}$ of $Q(x)$ and $R(x)$ and are computed recursively.

(8) Set $b_n = a_n$, and

$b_{n-1} = a_{n-1} + r b_n$, and then

$b_k = a_k + r b_{k+1} + s b_{k+2}$ for $k = n-2, n-3, \ldots, 2, 1, 0$.

Proof **Lin-Bairstow Method** **Lin-Bairstow Method**

Example 1. Use quadratic synthetic division to divide $P(x) = x^5 + 6x^4 - 20x^2 + 22x + 8$ by $T(x) = x^2 + 2x - 3$.

Solution 1.

Heuristics

In the days when "hand computations" were necessary, the quadratic synthetic division tableau (or table) was used. The coefficients $\{a_k\}_{k=0}^{n}$ of the polynomial are entered on the first row in descending order, the second and third rows are reserved for the intermediate computation steps ($+ r b_{k+1}$ and $+ s b_{k+2}$) and the bottom row contains the coefficients b_n, $b_{n-1} = a_{n-1} + r b_n$ and $\{b_k = a_k + r b_{k+1} + s b_{k+2}\}_{k=0}^{n-2}$.

Input	a_n	a_{n-1}	a_{n-2}	a_{n-3}	$\cdots$	a_k	$\cdots$	a_2	a_1		a_0
r	$\downarrow$	$+ r b_n$	$+ r b_{n-1}$	$+ r b_{n-2}$	$\cdots$	$+ r b_{k+1}$	$\cdots$	$+ r b_3$	$+ r b_2$		$+ r b_1$
s	$\downarrow$		$+ s b_n$	$+ s b_{n-1}$	$\cdots$	$+ s b_{k+2}$	$\cdots$	$+ s b_4$	$+ s b_3$		$+ s b_2$
	$\nearrow$	$\nearrow$	$\nearrow$	$\nearrow$		$\nearrow$		$\nearrow$	$\nearrow$		
	b_n	b_{n-1}	b_{n-2}	b_{n-3}	$\cdots$	b_k	$\cdots$	b_2	b_1	|	b_0
									Output		Output

Example 2. Use the "quadratic synthetic division tableau" to divide $P(x) = x^5 + 6x^4 - 20x^2 + 22x + 8$ by $T(x) = x^2 + 2x - 3$.

Solution 2.

Using vector coefficients

As mentioned above, it is efficient to store the coefficients $\{a_{[[k]]}\}_{k=1}^{n+1}$ of a polynomial $P(x)$ of degree n in the vector $a = \{a_{[[1]]}, a_{[[2]]}, \ldots, a_{[[n]]}, a_{[[n+1]]}\}$. Notice that this is a shift of the index for $a_{[[k]]}$ and the polynomial $P(x)$ is written in the form

$$P(x) = \sum_{k=0}^{n} a_{[[k+1]]} x^k .$$

Given the quadratic $T(x) = x^2 - r x - s$, the quotient $Q(x)$ and remainder $R(x)$ are

$$Q(x) = \sum_{k=0}^{n-2} b_{[[k+3]]} x^k$$

and

$$R(x) = b_{[[2]]} (x - r) + b_{[[1]]} .$$

The recursive formulas for computing the coefficients $\{b_{[[k]]}\}_{k=0}^{n+1}$ of $Q(x)$ and $R(x)$ are

$b_{[[n+1]]} = a_{[[n+1]]}$, and

$b_{[[n]]} = a_{[[n]]} + r\, b_{[[n+1]]}$, and then

$b_{[[k]]} = a_{[[k]]} + r\, b_{[[k+1]]} + s\, b_{[[k+2]]}$ for $k = n-1, \ldots, 3, 2, 1$.

Example 3. Use the vector form of quadratic synthetic division to divide
$P(x) = x^5 + 6x^4 - 20x^2 + 22x + 8$ by $T(x) = x^2 + 2x - 3$.

Solution 3.

The Lin-Bairstow Method

We now build on the previous idea and develop the Lin-Bairstow's method for finding a quadratic factor $(x^2 - rx - s)$ of $P(x)$. Suppose that we start with the initial guess

(9) $\quad x^2 - r_0 x - s_0$

and that $P(x)$ can be expressed as

$$P(x) = (x^2 - r_0 x - s_0)\,Q(x) + u(x - r_0) + v.$$

When u and v are small, the quadratic (9) is close to a factor of $P(x)$. We want to find new values r_1 and s_1 so that

(10) $\quad x^2 - r_1 x - s_1$

is closer to a factor of $P(x)$ than the quadratic (9).

Observe that u and v are functions of r and s, that is

$u = u(r, s)$, and

$v = v(r, s)$.

The new values r_1 and s_1 satisfy the relations

$r_1 = r_0 + \Delta r$, and

$s_1 = s_0 + \Delta s$.

The differentials of the functions u and v are used to produce the approximations

$v(r_1, s_1) \approx v(r_0, s_0) + v_r(r_0, s_0)\Delta r + v_s(r_0, s_0)\Delta s$

and

$u(r_1, s_1) \approx u(r_0, s_0) + u_r(r_0, s_0)\Delta r + u_s(r_0, s_0)\Delta s$

The new values r_1 and s_1 are to satisfy

$v(r_1, s_1) = 0$, and

$u(r_1, s_1) = 0$.

When the quantities Δr and Δs are small, we replace the above approximations with equations and obtain the linear system:

$0 = v(r_0, s_0) + v_r(r_0, s_0)\Delta r + v_s(r_0, s_0)\Delta s$

(11)

$0 = u(r_0, s_0) + u_r(r_0, s_0)\Delta r + u_s(r_0, s_0)\Delta s$

All we need to do is find the values of the partial derivatives $v_r(r_0, s_0)$, $v_s(r_0, s_0)$, $u_r(r_0, s_0)$ and $u_s(r_0, s_0)$ and then use Cramer's rule to compute Δr and Δs. Let us announce that the values of the partial derivatives are

$v_r(r_0, s_0) = c_1$
$v_s(r_0, s_0) = c_2$
$u_r(r_0, s_0) = c_2$
$u_s(r_0, s_0) = c_3$

where the coefficients $\{c_k\}$ are built upon the coefficients $\{b_k\}$ given in (8) and are calculated recursively using the formulas

(12) Set $c_n = b_n$, and

$c_{n-1} = b_{n-1} + r\,c_n$, and then

$$c_k = b_k + r\,c_{k+1} + s\,c_{k+2} \quad \text{for} \quad k = n-2, n-3, \ldots, 2, 1.$$

The formulas in (12) use the coefficients $\{b_k\}$ in (8). Since

$b_0 = v(r_0, s_0)$, and

$b_1 = u(r_0, s_0)$, and

the linear system in (11) can be written as

$$c_1 \Delta r + c_2 \Delta s = -b_0$$
$$c_2 \Delta r + c_3 \Delta s = -b_1$$

Cramer's rule can be used to solve this linear system. The required determinants are

$$d_0 = \begin{vmatrix} c_1 & c_2 \\ c_2 & c_3 \end{vmatrix}, \quad d_1 = \begin{vmatrix} -b_0 & c_2 \\ -b_1 & c_3 \end{vmatrix}, \text{ and } \quad d_2 = \begin{vmatrix} c_1 & -b_0 \\ c_2 & -b_1 \end{vmatrix}.$$

and the new values r_1 and s_1 are computed using the formulas

$$r_1 = r_0 + \frac{d_1}{d_0},$$

and

$$s_1 = s_0 + \frac{d_2}{d_0}.$$

Proof **Lin-Bairstow Method** **Lin-Bairstow Method**

The iterative process is continued until good approximations to r and s have been found. If the initial guesses r_0 and s_0 are chosen small, the iteration does not tend to wander for a long time before converging. When $x \approx 0$, the larger powers of x can be neglected in equation (1) and we have the approximation

$$0 \approx P(x) = a_2 x^2 + a_1 x + a_0.$$

Hence the initial guesses for r_0 and s_0 could be $r_0 = -\frac{a_1}{a_2}$ and $s_0 = -\frac{a_0}{a_2}$, provided that $a_2 \neq 0$.

If hand calculations are done, then the quadratic synthetic division tableau can be extended to form an easy way to calculate the coefficients $\{c_k\}$.

Input	a_n	a_{n-1}	a_{n-2}	a_{n-3}	$\cdots$	a_k	$\cdots$	a_2	a_1	a_0
r	$\downarrow$	$+ r\, b_n$	$+ r\, b_{n-1}$	$+ r\, b_{n-2}$	$\cdots$	$+ r\, b_{k+1}$	$\cdots$	$+ r\, b_3$	$+ r\, b_2$	$+ r\, b_1$
s	$\downarrow$		$+ s\, b_n$	$+ s\, b_{n-1}$	$\cdots$	$+ s\, b_{k+2}$	$\cdots$	$+ s\, b_4$	$+ s\, b_3$	$+ s\, b_2$
	$\nearrow$	$\nearrow$	$\nearrow$	$\nearrow$		$\nearrow$		$\nearrow$	$\nearrow$	
	b_n	b_{n-1}	b_{n-2}	b_{n-3}	$\cdots$	b_k	$\cdots$	b_2	b_1	| b_0
	$\downarrow$	$+ r\, c_n$	$+ r\, c_{n-1}$	$+ r\, c_{n-2}$	$\cdots$	$+ r\, c_{k+1}$	$\cdots$	$+ r\, c_3$	$+ r\, c_2$	
	$\downarrow$		$+ s\, c_n$	$+ s\, c_{n-1}$	$\cdots$	$+ s\, c_{k+2}$	$\cdots$	$+ s\, c_4$	$+ s\, c_3$	
	c_n	c_{n-1}	c_{n-2}	c_{n-3}	$\cdots$	c_k	$\cdots$	c_2	c_1	

Bairstow's method is a special case of Newton's method in two dimensions.

Algorithm (Lin-Bairstow Iteration). To find a quadratic factor of $P(x)$ given an initial approximation $x^2 - r_0 x - s_0$.

Computer Programs Lin-Bairstow Method Lin-Bairstow Method

Mathematica Subroutine (Lin-Bairstow Iteration).

```
Bairstow[r00_, s00_, max_] :=
 Module[{j = 0, k, n = Length[a] - 1, r0 = N[r00], s0 = N[s00]},
  c = b = Table[0, {i, n + 1}];
  Print["x^2 - ", "r"_j, " x - ", "s"_j, " = ", PaddedForm[x^"2" - r0 x - s0, {16, 15}] ];
  While[ j < max,
   b[[n+1]] = a[[n+1]];
   b[[n]] = a[[n]] + r0 b[[n+1]];
   For[ k = n - 1, 1 ≤ k, k--,
    b[[k]] = a[[k]] + r0 b[[k+1]] + s0 b[[k+2]]; ];
   c[[n+1]] = b[[n+1]];
   c[[n]] = b[[n]] + r0 c[[n+1]];
   For[ k = n - 1, 2 ≤ k, k--,
    c[[k]] = b[[k]] + r0 c[[k+1]] + s0 c[[k+2]]; ];
   d0 = Det[( c[[2]]  c[[3]]
              c[[3]]  c[[4]] )];
   d1 = Det[( -b[[1]]  c[[3]]
              -b[[2]]  c[[4]] )];
   d2 = Det[( c[[2]]  -b[[1]]
              c[[3]]  -b[[2]] )];
   r1 = r0 + d1/d0;
   s1 = s0 + d2/d0;
   j = j + 1;
   Print["x^2 - ", "r"_j, " x - ", "s"_j, " = ", PaddedForm[x^"2" - r1 x - s1, {16, 15}] ];
   r0 = r1;
   s0 = s1;  ];
  Return[x^2 - r1 x - s1]; ]
```

Example 4. Given $P(x) = x^4 + x^3 + 3x^2 + 4x + 6$. Start with $r_0 = -2.1$ and $s_0 = -1.9$ and use the Lin-Bairstow method to find a quadratic factor of $P(x)$.

Solution 4.

Research Experience for Undergraduates

Lin-Bairstow Method **Lin-Bairstow Method** Internet hyperlinks to web sites and a bibliography of articles.

Download this Mathematica Notebook Lin-Bairstow Method

Example 1. Use quadratic synthetic division to divide $P(x) = x^5 + 6x^4 - 20x^2 + 22x + 8$ by $T(x) = x^2 + 2x - 3$.

Solution 1.

First, construct the polynomial $P(x)$.

```
a5 = 1; a4 = 6; a3 = 0; a2 = -20; a1 = 22; a0 = 8;
n = 5;
P[x_] = Sum[a_k x^k, {k, 0, n}];
Print["P[x] = ", P[x]];
```

$P[x] = 8 + 22x - 20x^2 + 6x^4 + x^5$

Second, given $T(x) = x^2 - rx - s$ construct the polynomials $Q(x)$ and $R(x)$.

```
r = -2;
s = 3;
T[x_] = x^2 - r x - s;
b_n = a_n;
b_(n-1) = a_(n-1) + r b_n;
For[ k = n - 2, 0 ≤ k, k--,
 b_k = a_k + r b_(k+1) + s b_(k+2); ];
Q[x_] = Sum[b_(k+2) x^k, {k, 0, n-2}];
R[x_] = b_1 (x - r) + b_0;
Print["P[x] = ", P[x] ];
Print["T[x] = ", T[x] ];
Print["    r = ", r];
Print["    s = ", s];
Print[""];
Print["a"_n, " = ", a_n, "  and  ", "b"_n, " = ", "a"_n, " = ", b_n];
Print["a"_(n-1), " = ", a_(n-1), "  and  ", "b"_(n-1), " = ", "a"_(n-1), " + r ", "b"_(n-1), " = ", a_(n-1), " + (", r, ")(", b_n, ") = ", b_(n-1)];
For[ k = n - 2, 0 ≤ k, k--,
 Print["a"_k, " = ", a_k, "  and  ", "b"_k, " = ", "a"_k, " + r ", "b"_(k+1), " + s ", "b"_(k+2), " = ", a_k, " + (", r, ")(", b_(k+1),
  ") + (", s, ")(", b_(k+2), ") = ", b_k]; ];
Print[""];
Print["Q[x] = ", Sum["b"_(k+2) x^k, {k, 0, n-2}] ];
Print["R[x] = b_1 (x-r) + b_0" ];
Print[""];
Print["Q[x] = ", Q[x] ];
Print["R[x] = ", R[x] ];
```

$P[x] = 8 + 22x - 20x^2 + 6x^4 + x^5$
$T[x] = -3 + 2x + x^2$
$r = -2$
$s = 3$

$a_5 = 1$ and $b_5 = a_5 = 1$
$a_4 = 6$ and $b_4 = a_4 + r\, b_4 = 6 + (-2)(1) = 4$
$a_3 = 0$ and $b_3 = a_3 + r\, b_4 + s\, b_5 = 0 + (-2)(4) + (3)(1) = -5$
$a_2 = -20$ and $b_2 = a_2 + r\, b_3 + s\, b_4 = -20 + (-2)(-5) + (3)(4) = 2$
$a_1 = 22$ and $b_1 = a_1 + r\, b_2 + s\, b_3 = 22 + (-2)(2) + (3)(-5) = 3$
$a_0 = 8$ and $b_0 = a_0 + r\, b_1 + s\, b_2 = 8 + (-2)(3) + (3)(2) = 8$

$Q[x] = b_2 + x\, b_3 + x^2 b_4 + x^3 b_5$
$R[x] = b_1 (x - r) + b_0$

$Q[x] = 2 - 5x + 4x^2 + x^3$
$R[x] = 8 + 3(2 + x)$

Third, verify that $P(x) = (x^2 - r x - s) Q(x) + b_1 (x - r) + b_0$.

```
Print["P[x] = ", P[x] ];
Print["r = ", r];
Print["s = ", s];
Print["T(x) = (x^2 - r x - s)"];
Print["T[x] = ", T[x] ];
Print[""];
Print["Q[x] = ", Q[x] ];
Print["b1 = ", b1];
Print["b0 = ", b0];
Print[""];
Print["P(x) = T(x)Q(x) + R(x)"];
Print["P(x) = T(x)Q(x) + b1(x - r) + b0"];
Print[""];
Print["P(x) = (x^2 - r x - s)Q(x) + b1(x - r) + b0"];
Print["P(x) = (", T[x], ")(", Q[x], ") + (", b1, ")(", x - r, ") + (", b0, ")"];
Print["P(x) = (", T[x], ")(", Q[x], ") + (", Expand[b1 (x - r)], ") + (", b0, ")"];
Print["P(x) = ", b1 (x - r) + b0, " + ", T[x] Q[x]];
Print["P(x) = ", Expand[ b1 (x - r) + b0], " + (", Expand[T[x] Q[x]], ")"];
Print["P(x) = ", Expand[T[x] Q[x] + b1 (x - r) + b0]];
Print["Is this the original polynomial ?"];
Print[ExpandAll[ P[x] == T[x] Q[x] + b1 (x - r) + b0] ];
```

```
P[x] = 8 + 22 x - 20 x^2 + 6 x^4 + x^5
r = -2
s = 3
T(x) = (x^2 - r x - s)
T[x] = -3 + 2 x + x^2

Q[x] = 2 - 5 x + 4 x^2 + x^3
b1 = 3
b0 = 8

 P (x) = T (x) Q (x) + R (x)
 P (x) = T (x) Q (x) + b1 (x - r) + b0

 P (x) = (x^2 - r x - s) Q (x) + b1 (x - r) + b0
P(x) = (-3 + 2 x + x^2)(2 - 5 x + 4 x^2 + x^3) + (3)(2 + x) + (8)
P(x) = (-3 + 2 x + x^2)(2 - 5 x + 4 x^2 + x^3) + (6 + 3 x) + (8)
P(x) = 8 + 3 (2 + x) + (-3 + 2 x + x^2) (2 - 5 x + 4 x^2 + x^3)
P(x) = 14 + 3 x + (-6 + 19 x - 20 x^2 + 6 x^4 + x^5)
P(x) = 8 + 22 x - 20 x^2 + 6 x^4 + x^5
 Is this the original polynomial ?
True
```

We are done.

Aside. We can have *Mathematica* compute the quotient and remainder using the built in procedures PolynomialQuotient and `PolynomialRemainder`. This is just for fun!

```
Q[x_] = PolynomialQuotient[P[x], T[x], x];
R[x_] = PolynomialRemainder[P[x], T[x], x];
Print["P[x] = ", P[x] ];
Print["T[x] = ", T[x] ];
Print[""];
Print["Q[x] = ", Q[x] ];
Print["R[x] = ", R[x] ];
Print[""];
Print["P[x] = T[x]Q[x] + R[x]" ];
Print["P[x] = ", T[x] Q[x], " + (", R[x], ")" ];
Print["P[x] = (", Expand[T[x] Q[x]], ") + (", R[x], ")" ];
Print["P[x] = ", Expand[T[x] Q[x] + R[x]] ];
```

```
P[x] = 8 + 22 x - 20 x^2 + 6 x^4 + x^5
T[x] = -3 + 2 x + x^2

Q[x] = 2 - 5 x + 4 x^2 + x^3
R[x] = 14 + 3 x

 P[x] = T[x] Q[x] + R[x]
P[x] = (-3 + 2 x + x^2) (2 - 5 x + 4 x^2 + x^3) + (14 + 3 x)
P[x] = (-6 + 19 x - 20 x^2 + 6 x^4 + x^5) + (14 + 3 x)
P[x] = 8 + 22 x - 20 x^2 + 6 x^4 + x^5
```

Example 2. Use the "quadratic synthetic division tableau" to divide

$P(x) = x^5 + 6x^4 - 20x^2 + 22x + 8$ by $T(x) = x^2 + 2x - 3$.

Solution 2.

Use the "quadratic synthetic division tableau"

Input							
	a_5	a_4	a_3	a_2	a_1		a_0
r	↓	$+ r b_5$	$+ r b_4$	$+ r b_3$	$+ r b_2$		$+ r b_1$
s	↓		$+ s b_5$	$+ s b_4$	$+ s b_3$		$+ s b_2$
	↗	↗	↗	↗	↗		
	b_5	b_4	b_3	b_2	b_1	|	b_0
					Output		Output

Since $T(x) = x^2 - r x - s$ we use $r = -2$ and $s = 3$.

Input							
	1	6	0	- 20	22		8
r = -2	↓	+ (-2) 1	+ (-2) 4	+ (-2) (-5)	+ (-2) 2		+ (-2) 3
s = 3	↓		+ (3) 1	+ (3) (4)	+ (3) (-5)		+ (3) 2
	1	4	- 5	2	3		8
	b_5	b_4	b_3	b_2	b_1	|	b_0
					Output		Output

Then simplify and get.

Input							
	1	6	0	- 20	22		8
r = -2	↓	- 2	- 8	+ 10	- 4		- 6
s = 3	↓		+ 3	+ 12	- 15		+ 6
	1	4	- 5	2	3		8
	b_5	b_4	b_3	b_2	b_1	|	b_0
					Output		Output

Thus we have

$$Q(x) = b_5 x^3 + b_4 x^2 + b_3 x + b_2$$

$$Q(x) = x^3 + 4x^2 - 5x + 2$$

and

$$R(x) = b_1 (x - r) + b_0$$

$$R(x) = 3(x + 2) + 8$$

This agrees with our previous computation in Example 1.

We are done.

Aside. We can let *Mathematica* verify the result.

```
P[x_] = x^5 + 6 x^4 - 20 x^2 + 22 x + 8;
T[x_] = x^2 + 2 x - 3;
Q[x_] = x^3 + 4 x^2 - 5 x + 2;
R[x_] = 3 (2 + x) + 8;
Print["P[x] = ", P[x] ];
Print["T[x] = ", T[x] ];
Print[""];
Print["Q[x] = ", Q[x] ];
Print["R[x] = ", R[x] ];
Print[""];
Print["P[x] = T[x]Q[x] + R[x]" ];
Print["P[x] = ", T[x] Q[x], " + (", R[x], ")" ];
Print["P[x] = (", Expand[T[x] Q[x]], ") + (", R[x], ")" ];
Print["P[x] = ", Expand[T[x] Q[x] + R[x]] ];
```

```
P[x] = 8 + 22 x - 20 x^2 + 6 x^4 + x^5
T[x] = -3 + 2 x + x^2

Q[x] = 2 - 5 x + 4 x^2 + x^3
R[x] = 8 + 3 (2 + x)

 P[x] = T[x] Q[x] + R[x]
P[x] = (-3 + 2 x + x^2) (2 - 5 x + 4 x^2 + x^3) + (8 + 3 (2 + x))
P[x] = (-6 + 19 x - 20 x^2 + 6 x^4 + x^5) + (8 + 3 (2 + x))
P[x] = 8 + 22 x - 20 x^2 + 6 x^4 + x^5
```

Example 3. Use the vector form of quadratic synthetic division to divide

$P(x) = x^5 + 6x^4 - 20x^2 + 22x + 8$ by $T(x) = x^2 + 2x - 3$.

Solution 3.

First, construct the polynomial $P(x)$.

```
a = {8, 22, -20, 0, 6, 1};
n = Length[a] - 1;
P[x_] = Sum[a[[k+1]] x^k, {k, 0, n}];
Print["P[x] = ", P[x]];
```

$P[x] = 8 + 22x - 20x^2 + 6x^4 + x^5$

Second, given $T(x) = x^2 - rx - s$ construct the polynomials $Q(x)$ and $R(x)$.

```
r = -2;
s = 3;
T[x_] = x^2 - r x - s;
b = Table[0, {i, n + 1}];
b[[n+1]] = a[[n+1]];
b[[n]] = a[[n]] + r b[[n+1]];
For[k = n - 1, 1 ≤ k, k--,
 b[[k]] = a[[k]] + r b[[k+1]] + s b[[k+2]]; ];
Q[x_] = Sum[b[[k+3]] x^k, {k, 0, n-2}];
R[x_] = b[[2]] (x - r) + b[[1]];
Print["P[x] = ", P[x]];
Print["T[x] = ", T[x]];
Print[""];
Print["Q[x] = ", Q[x]];
Print["R[x] = ", R[x]];
```

$P[x] = 8 + 22x - 20x^2 + 6x^4 + x^5$

$T[x] = -3 + 2x + x^2$

$Q[x] = 2 - 5x + 4x^2 + x^3$

$R[x] = 8 + 3(2 + x)$

Third, verify that $P(x) = (x^2 - r x - s)\,Q(x) + b_{[[2]]}\,(x - r) + b_{[[1]]}$.

```
Print["P[x] = ", P[x] ];
Print["r = ", r];
Print["s = ", s];
Print["T(x) = (x^2 - r x - s)"];
Print["T[x] = ", T[x] ];
Print[""];
Print["Q[x] = ", Q[x] ];
Print["b[[2]] = ", b[[2]]];
Print["b[[1]] = ", b[[1]]];
Print[""];
Print["P(x) = T(x)Q(x) + R(x)"];
Print["P(x) = T(x)Q(x) + b[[2]](x-r) + b[[1]]"];
Print[""];
Print["P(x) = (x^2 - r x - s)Q(x) + b[[2]](x-r) + b[[1]]"];
Print["P(x) = (", T[x], ")(", Q[x], ") + (", b[[2]], ")(", x - r, ") + (", b[[1]], ")"];
Print["P(x) = (", T[x], ")(", Q[x], ") + (", Expand[b[[2]] (x - r)], ") + (", b[[1]], ")"];
Print["P(x) = ", b[[2]] (x - r) + b[[1]], " + ", T[x] Q[x]];
Print["P(x) = ", Expand[ b[[2]] (x - r) + b[[1]]], " + (", Expand[T[x] Q[x]], ")"];
Print["P(x) = ", Expand[T[x] Q[x] + b[[2]] (x - r) + b[[1]]]];
Print["Is this the original polynomial ?"];
Print[ExpandAll[ P[x] == T[x] Q[x] + b[[2]] (x - r) + b[[1]]] ];
```

P[x] = $8 + 22 x - 20 x^2 + 6 x^4 + x^5$

r = -2

s = 3

T(x) = $(x^2 - r x - s)$

T[x] = $-3 + 2 x + x^2$

Q[x] = $2 - 5 x + 4 x^2 + x^3$

$b_{[[2]]} = 3$

$b_{[[1]]} = 8$

P(x) = T(x) Q(x) + R(x)

P(x) = T(x) Q(x) + $b_{[[2]]}$ (x − r) + $b_{[[1]]}$

$P(x) = (x^2 - r x - s)\,Q(x) + b_{[[2]]}\,(x - r) + b_{[[1]]}$

$P(x) = (-3 + 2 x + x^2)(2 - 5 x + 4 x^2 + x^3) + (3)(2 + x) + (8)$

$P(x) = (-3 + 2 x + x^2)(2 - 5 x + 4 x^2 + x^3) + (6 + 3 x) + (8)$

$P(x) = 8 + 3(2 + x) + (-3 + 2 x + x^2)(2 - 5 x + 4 x^2 + x^3)$

$P(x) = 14 + 3 x + (-6 + 19 x - 20 x^2 + 6 x^4 + x^5)$

$P(x) = 8 + 22 x - 20 x^2 + 6 x^4 + x^5$

Is this the original polynomial ?

True

Example 4. Given $P(x) = x^4 + x^3 + 3x^2 + 4x + 6$. Start with $r_0 = -2.1$ and $s_0 = -1.9$ and use the Lin-Bairstow method to find a quadratic factor of $P(x)$.

Solution 4.

Enter the coefficients of the polynomial.

```
a = {6, 4, 3, 1, 1};
n = Length[a] - 1;
P[x_] = Sum[a[[k+1]] x^k, {k, 0, n}];
Print["P[x] = ", P[x] ];
```

$P[x] = 6 + 4x + 3x^2 + x^3 + x^4$

Enter the starting values $r_0 = -2.1$ and $s_0 = -1.9$ and call the subroutine `Bairstow`.

```
r0 = -2.1;
s0 = -1.9;
Bairstow[r0, s0, 5];
```

$x^2 - r_0\ x - s_0 = 1.900000000000000 + 2.100000000000000\,x + x^2$

$x^2 - r_1\ x - s_1 = 1.949988192338374 + 1.989302820836306\,x + x^2$

$x^2 - r_2\ x - s_2 = 2.000150979221003 + 1.999992769904759\,x + x^2$

$x^2 - r_3\ x - s_3 = 1.999999991124103 + 1.999999996545100\,x + x^2$

$x^2 - r_4\ x - s_4 = 2.000000000000000 + 2.000000000000000\,x + x^2$

$x^2 - r_5\ x - s_5 = 2.000000000000000 + 2.000000000000000\,x + x^2$

Verify that a quadratic factor has been found.

```
T[x_] = x^2 - r1 x - s1;
Q[x_] = Sum[b[[k+3]] x^k, {k, 0, n-2}];
R[x_] = b[[2]] (x - r1) + b[[1]];
Print["P[x] = ", P[x] ];
Print["T[x] = ", T[x] ];
Print[""];
Print["Q[x] = ", Q[x] ];
Print["R[x] = ", R[x], " = ", Expand[R[x]] ];
Print[""];
Print["P[x] ≈ Q[x] T[x]" ];
Print["P[x] ≈ (", Q[x], ")(", T[x], ")" ];
```

$P[x] = 6 + 4x + 3x^2 + x^3 + x^4$
$T[x] = 2. + 2.x + x^2$

$Q[x] = 3. - 1.x + x^2$
$R[x] = 0. + 0.(2. + x) = 0. + 0.x$

$P[x] \approx Q[x]T[x]$
$P[x] \approx (3. - 1.x + x^2)(2. + 2.x + x^2)$

We are done.

Aside. We can let *Mathematica* find the factors too. This is just for fun.

```
Print["P[x] = ", P[x] ];
Print["P[x] = ", Factor[P[x]] ];
```

$P[x] = 6 + 4x + 3x^2 + x^3 + x^4$
$P[x] = (3 - x + x^2)(2 + 2x + x^2)$

Aside. We can let *Mathematica* find the roots too. This is just for fun.

```
Print[P[x] == 0 ];
Solve[P[x] == 0, x]
```

$6 + 4x + 3x^2 + x^3 + x^4 == 0$
$\{\{x \to -1 - i\}, \{x \to -1 + i\}, \{x \to \frac{1}{2}(1 - i\sqrt{11})\}, \{x \to \frac{1}{2}(1 + i\sqrt{11})\}\}$

```
Print[P[x] == 0 ];
NSolve[P[x] == 0, x]
```

$6 + 4x + 3x^2 + x^3 + x^4 == 0$
$\{\{x \to -1. - 1.i\}, \{x \to -1. + 1.i\}, \{x \to 0.5 - 1.65831 i\}, \{x \to 0.5 + 1.65831 i\}\}$

```
//*********************************************
// Programmer :  Naser . Bagheri  (9016393)
//*********************************************
#include <iostream>
#include <cmath>
using namespace std;
#define ESP 0.001
#define F(x) (x)*(x)*(x) + (x) + 10
#define a3 1
#define a2 0
#define a1 1
#define a0 10
//#define c3  0void main()
int main()
{
  double u,v,u1,v1,u2,v2,b3,b2,p,b1,b0,c2,c1,c0,U,V;
  int i=1;
  float c3=0;
  cout<<"\nEnter the value of u: ";
  scanf("%lf",&u);
  cout<<"\nEnter the value of v: ";
  scanf("%lf",&v);
  b3=a3;
  b2=a2+u*b3;
  b1=a1+u*b2+v*b3;
  b0=a0+u*b1+v*b2;
  c2=b3;
  c1=b2+u*c2+v*c3;
  c0=b1+u*c1+v*c2;
  p=c1*c1-c0*c2;
  U=((-(b1*c1-b0*c2))/(p));
  V=((-(b0*c1-c0*b1))/(p));
  u1=u+U;
  v1=v+V;
  cout<<"\n\n b0 = %lf"<<b0;
  cout<<"\n\n b1 = %lf"<<b1;
  cout<<"\n\n b2 = %lf"<<b2;
  cout<<"\n\n b3 = %lf"<<b3;
  cout<<"\n\n c0 = %lf"<<c0;
  cout<<"\n\n c1 = %lf"<<c1;
  cout<<"\n\n c2 = %lf"<<c2;
  cout<<"\n\n c3 = %lf"<<c3;
  cout<<"\n\n * * *  u = %lf * * *"<<u1;
  cout<<"\n\n * * *  v = %lf * * *"<<v1;

  do
  {
   u=u1;
   v=v1;
   b3=a3;
   b2=a2+u*b3;
   b1=a1+u*b2+v*b3;
   b0=a0+u*b1+v*b2;
   c2=b3;
   c1=b2+u*c2+v*c3;
   c0=b1+u*c1+v*c2;
   p=c1*c1-c0*c2;
   U=((-(b1*c1-b0*c2))/(p));
```

```
    V=((-(b0*c1-c0*b1))/(p));
    u2=u+U;
    v2=v+V;
    cout<<"\n\n b0 = %lf"<<b0;
   cout<<"\n\n b1 = %lf"<<b1;
   cout<<"\n\n b2 = %lf"<<b2;
   cout<<"\n\n b3 = %lf"<<b3;
   cout<<"\n\n c0 = %lf"<<c0;
   cout<<"\n\n c1 = %lf"<<c1;
   cout<<"\n\n c2 = %lf"<<c2;
   cout<<"\n\n c3 = %lf"<<c3;
   cout<<"\n\n * * * u = %lf * * *"<<u2;
   cout<<"\n\n * * * v = %lf * * *"<<v2;


    if(fabs(u1 - u2) < ESP && fabs(v1-v2) < ESP)
    {
     cout<<"\n\nREAL ROOT = %.3lf"<<u2;
     cout<<"\n\nREAL ROOT = %.3lf"<<v2;
     i=0;
    }
    else
    {
      u1 = u2;
      v1 = v2;
    }
   }while(i!=0);
}

/*
----------------------------------
     OUT PUT
-----------------------------------
Enter the value of u: 1.8
Enter the value of v: -4
-----------------------------------------------
 b0 = 3.232000
 b1 = 0.240000
 b2 = 1.800000
 b3 = 1.000000

 c0 = 2.720000
 c1 = 3.600000
 c2 = 1.000000
 c3 = 0.000000

  u = 2.031250
  v = -5.072500

 b0 = -0.194891
 b1 = 0.053477
 b2 = 2.031250
 b3 = 1.000000

 c0 = 3.656953
 c1 = 4.271250
 c2 = 1.000000
 c3 = 0.000000
```

```

   u = 2.002230
   v = -5.002025

  b0 = -0.001389
  b1 = 0.006900
  b2 = 2.002230
  b3 = 1.000000

  c0 = 6.121717
  c1 = 5.552230
  c2 = 1.000000
  c3 = 0.000000

   u = 2.000623
   v = -5.000003

  b0 = 0.001859
  b1 = 0.002490
  b2 = 2.000623
  b3 = 1.000000

  c0 = 11.224619
  c1 = 8.108540
  c2 = 1.000000
  c3 = 0.000000

   u = 2.000287
   v = -4.999767

REAL ROOT = 2.000
REAL ROOT = -5.000
*/

```

Applied Mathematics > Numerical Methods > Root-Finding >

Bairstow's Method

A procedure for finding the quadratic factors for the complex conjugate roots of a polynomial $P(x)$ with real coefficients.

$$[x-(a+i\,b)]\,[x-(a-i\,b)]=x^2+2\,a\,x+(a^2+b^2)\equiv x^2+B\,x+C. \quad (1)$$

Now write the original polynomial as

$$P(x)=(x^2+B\,x+C)\,Q(x)+R\,x+S \quad (2)$$

$$R(B+\delta B,C+\delta C)\approx R(B,C)+\frac{\partial R}{\partial B}\,d\,B+\frac{\partial R}{\partial C}\,d\,C \quad (3)$$

$$S(B+\delta B,C+\delta C)\approx S(B,C)+\frac{\partial S}{\partial B}\,d\,B+\frac{\partial S}{\partial C}\,d\,C \quad (4)$$

$$\frac{\partial P}{\partial C}=0=(x^2+B\,x+C)\,\frac{\partial Q}{\partial C}+Q(x)+x\,\frac{\partial R}{\partial C}+\frac{\partial S}{\partial C} \quad (5)$$

$$-Q(x)=(x^2+B\,x+C)\,\frac{\partial Q}{\partial C}+x\,\frac{\partial R}{\partial C}+\frac{\partial S}{\partial C} \quad (6)$$

$$\frac{\partial P}{\partial B}=0=(x^2+B\,x+C)\,\frac{\partial Q}{\partial B}+x\,Q(x)+x\,\frac{\partial R}{\partial B}+\frac{\partial S}{\partial B} \quad (7)$$

$$-x\,Q(x)=(x^2+B\,x+C)\,\frac{\partial Q}{\partial B}+x\,\frac{\partial R}{\partial B}+\frac{\partial S}{\partial B}. \quad (8)$$

Now use the two-dimensional Newton's method to find the simultaneous solutions.

REFERENCES:

Press, W. H.; Flannery, B. P.; Teukolsky, S. A.; and Vetterling, W. T. *Numerical Recipes in FORTRAN: The Art of Scientific Computing, 2nd ed.* Cambridge, England: Cambridge University Press, pp. 277 and 283-284, 1989.

Referenced on Wolfram|Alpha: Bairstow's Method

CITE THIS AS:

Weisstein, Eric W. "Bairstow's Method." From *MathWorld*--A Wolfram Web Resource. http://mathworld.wolfram.com/BairstowsMethod.html