The Primitive Euclidean Algorithm and Intermediate Expression Swell.

Michael Monagan, Spring 2021.

```
> restart;
> a := 3*x^3+x^2+x+5;
```

$$a := 3x^3 + x^2 + x + 5 \quad (1)$$

```
> b := 5*x^2-3*x+1;
```

$$b := 5x^2 - 3x + 1 \quad (2)$$

Pseudo division in ℤ[x]: prem(a, b, x) = rem(lc(b)^(deg(a) - deg(b) + 1), a, b, x) = rem(5^2 a, b, x)

```
> prem(a,b,x);
```

$$52x + 111 \quad (3)$$

```
> rem(5^2*a,b,x);
```

$$52x + 111 \quad (4)$$

```
> pr := prem(a,b,x,'m','pq');
```

$$pr := 52x + 111 \quad (5)$$

```
> m, pq;
```

$$25, 15x + 14 \quad (6)$$

```
> expand( m*a=b*pq+pr );
```

$$75x^3 + 25x^2 + 25x + 125 = 75x^3 + 25x^2 + 25x + 125 \quad (7)$$

An example of pseudo-division in ℤ[y][x]

```
> a := 5*x^3+y*x^2+3;
```

$$a := 5x^3 + yx^2 + 3 \quad (8)$$

```
> b := 3*y^2*x^2 + 5*x + 7*y;
```

$$b := 3y^2x^2 + 5x + 7y \quad (9)$$

In this example the multiplier $m = lc(a)^2 = 3^2 \cdot y^4$

```
> pr := prem(a,b,x,'m');
```

$$pr := -120xy^3 + 6y^4 + 125x + 175y \quad (10)$$

```
> m;
```

$$9y^4 \quad (11)$$

```
> rem(m*a,b,x);
```

$$(-120y^3 + 125)x + 6y^4 + 175y \quad (12)$$

The Primitive Euclidean Algorithm

```
> g := 2*x-3;
```

$$g := 2x - 3 \quad (13)$$

```
> a := expand( g*randpoly(x,degree=4,dense) );
```

$$a := -14x^5 + 65x^4 - 176x^3 - 23x^2 + 456x - 261 \quad (14)$$

```
> b := expand( g*randpoly(x,degree=4,dense) );
```

$$b := -112x^5 + 168x^4 - 124x^3 + 380x^2 - 437x + 219 \quad (15)$$

```
> content(a,x);
```

$$1 \quad (16)$$

```
> content(b,x);
```

$$1 \quad (17)$$

```
> r[0] := primpart(a,x);
```

```
r[1] := primpart(b,x);
k := 1;
while r[k] <> 0 do
    pr := prem(r[k-1],r[k],x);
    r[k+1] := primpart(pr,x);
    k := k+1;
od;
g := r[k-1];
g := gcd(content(a,x),content(b,x))*g; # attach gcd of contents
```

$$r_0 := -14\,x^5 + 65\,x^4 - 176\,x^3 - 23\,x^2 + 456\,x - 261$$

$$r_1 := -112\,x^5 + 168\,x^4 - 124\,x^3 + 380\,x^2 - 437\,x + 219$$

$$k := 1$$

$$pr := -4928\,x^4 + 17976\,x^3 + 7896\,x^2 - 57190\,x + 32298$$

$$r_2 := -352\,x^4 + 1284\,x^3 + 564\,x^2 - 4085\,x + 2307$$

$$k := 2$$

$$pr := -146318080\,x^3 + 160375552\,x^2 + 200787904\,x - 168203328$$

$$r_3 := -2286220\,x^3 + 2505868\,x^2 + 3137311\,x - 2628177$$

$$k := 3$$

$$pr := 5568818946161152\,x^2 - 12794178883681536\,x + 6661425696659712$$

$$r_4 := 179778504202\,x^2 - 413035217061\,x + 215051191137$$

$$k := 4$$

$$pr := -14164158620635468468339428800\,x + 21246237930953202702509143200$$

$$r_5 := -2\,x + 3$$

$$k := 5$$

$$pr := 0$$

$$r_6 := 0$$

$$k := 6$$

$$g := -2\,x + 3$$

$$g := -2\,x + 3 \qquad \textbf{(18)}$$

We need to multiply g by -1 to make it unit normal in $\mathbb{Z}[x]$.
Notice that the degree of the pseudo-remainder pr is decreasing until we get pr = gcd BUT the size of the integer coefficients are growing. Let us spy on the size of the integer coefficients for a larger example.

```
> a := expand( g*randpoly(x,degree=20,dense) );
  b := expand( g*randpoly(x,degree=20,dense) );
```

$$a := 8\,x^{21} + 154\,x^{20} - 229\,x^{19} - 154\,x^{18} + 350\,x^{17} - 406\,x^{16} + 328\,x^{15} - 274\,x^{14} + 247\,x^{13} + 99\,x^{12} - 205\,x^{11} - 16\,x^{10} + 59\,x^9 - 204\,x^8 + 226\,x^7 - 196\,x^6 - 81\,x^5 + 409\,x^4 - 288\,x^3 - 130\,x^2 + 78\,x + 216$$

$$b := -74\,x^{21} + 157\,x^{20} - 243\,x^{19} + 173\,x^{18} + 74\,x^{17} - 109\,x^{16} + 340\,x^{15} - 89\,x^{14} \qquad \textbf{(19)}$$

$$+ 152 x^{13} - 167 x^{12} + 34 x^{11} - 277 x^{10} + 263 x^9 + 131 x^8 - 53 x^7 - 221 x^6 + 282 x^5 - 425 x^4 + 137 x^3 + 224 x^2 - 92 x + 93$$

```
> r[0] := primpart(a,x):
  r[1] := primpart(b,x):
  k := 1:
  while r[k] <> 0 do
      printf("k=%2d  deg=%2d  size=%d\n",
             k,degree(r[k],x),length(maxnorm(r[k])) );
      pr := prem(r[k-1],r[k],x);
      r[k+1] := primpart(pr,x);
      k := k+1;
  od:
  g := r[k-1];
  g := gcd(content(a,x),content(b,x))*g; # attach gcd of contents
k= 1  deg=21  size=3
k= 2  deg=20  size=5
k= 3  deg=19  size=9
k= 4  deg=18  size=13
k= 5  deg=17  size=17
k= 6  deg=16  size=21
k= 7  deg=15  size=25
k= 8  deg=14  size=29
k= 9  deg=13  size=33
k=10  deg=12  size=39
k=11  deg=11  size=43
k=12  deg=10  size=48
k=13  deg= 9  size=53
k=14  deg= 8  size=57
k=15  deg= 7  size=61
k=16  deg= 6  size=65
k=17  deg= 5  size=71
k=18  deg= 4  size=75
k=19  deg= 3  size=79
k=20  deg= 2  size=84
k=21  deg= 1  size=1
```

$$g := 2x - 3$$

$$g := 2x - 3 \qquad \textbf{(20)}$$

Notice that the size of the coefficients increases by just over 4 digits each iteration. This is a linear growth in the size of the coefficients. The degree is dropping but coefficients grow until we get to the gcd.

I'm going to redo the example with two variables instead of one so for $\mathbb{Z}[x,y]$ and we are going to watch what happens to the degree of the pseudo-remainders in y.

```
> g := 2*x-3*y-5;
```

$$g := 2x - 3y - 5 \qquad \textbf{(21)}$$

```
> a := expand( g*randpoly([x,y],degree=10,dense) );
```

$$a := -102 x^{11} + 307 x^{10} y - 229 x^9 y^2 - 59 x^8 y^3 + 54 x^7 y^4 + 139 x^6 y^5 - 237 x^5 y^6 + 286 x^4 y^7 - 251 x^3 y^8 + 253 x^2 y^9 - 388 x y^{10} + 141 y^{11} + 445 x^{10} - 668 x^9 y + 24 x^8 y^2 - 26 x^7 y^3 + 72 x^6 y^4 + 141 x^5 y^5 + 113 x^4 y^6 - 279 x^3 y^7 - 219 x^2 y^8 - 376 x y^9 + 352 y^{10} - 365 x^9 - 110 x^8 y - 362 x^7 y^2 + 669 x^6 y^3 + 265 x^5 y^4 - 15 x^4 y^5 - 255 x^3 y^6 - 293 x^2 y^7 + 22 x y^8 + 354 y^9 - 329 x^8 + 75 x^7 y + 448 x^6 y^2 - 481 x^5 y^3 \qquad \textbf{(22)}$$

$$+ 143 x^4 y^4 - 432 x^3 y^5 - 380 x^2 y^6 - 194 x y^7 + 481 y^8 - 39 x^7 - 281 x^6 y - 161 x^5 y^2 - 184 x^4 y^3 + 327 x^3 y^4 - 433 x^2 y^5 + 436 x y^6 + 651 y^7 + 259 x^6 + 699 x^5 y - 201 x^4 y^2 + 515 x^3 y^3 - 497 x^2 y^4 + 468 x y^5 + 386 y^6 + 276 x^5 + 492 x^4 y + 171 x^3 y^2 - 16 x^2 y^3 + 229 x y^4 - 195 y^5 + 314 x^4 - 164 x^3 y - 310 x^2 y^2 - 53 x y^3 - 71 y^4 + 240 x^3 - 459 x^2 y - 392 x y^2 + 232 y^3 + 142 x^2 - 73 x y + 250 y^2 - 331 x - 361 y - 60$$

```
> b := expand( g*randpoly([x,y],degree=9,dense) );
```

$$b := -50 x^{10} - 117 x^9 y + 168 x^8 y^2 + 2 x^7 y^3 + 147 x^6 y^4 + 2 x^5 y^5 + 201 x^4 y^6 + 67 x^3 y^7 + 202 x^2 y^8 - 159 x y^9 - 108 y^{10} + 225 x^9 + 246 x^8 y + 286 x^7 y^2 + 687 x^6 y^3 + 98 x^5 y^4 + 718 x^4 y^5 + 220 x^3 y^6 - 95 x^2 y^7 - 206 x y^8 - 453 y^9 - 236 x^8 + 257 x^7 y + 352 x^6 y^2 - 98 x^5 y^3 + 108 x^4 y^4 - 138 x^3 y^5 - 108 x^2 y^6 - 19 x y^7 - 389 y^8 - 171 x^7 - 6 x^6 y - 40 x^5 y^2 - 391 x^4 y^3 - 208 x^3 y^4 + 353 x^2 y^5 + 347 x y^6 - 43 y^7 + 332 x^6 + 168 x^5 y - 424 x^4 y^2 - 641 x^3 y^3 + 159 x^2 y^4 + 526 x y^5 - 174 y^6 - 64 x^5 - 142 x^4 y + 216 x^3 y^2 + 12 x^2 y^3 + 363 x y^4 - 15 y^5 + 388 x^4 - 15 x^3 y - 333 x^2 y^2 - 150 x y^3 - 430 y^4 - 365 x^3 + 86 x^2 y - 236 x y^2 - 27 y^3 - 96 x^2 - 340 x y + 596 y^2 - 454 x + 526 y + 485 \quad \textbf{(23)}$$

```
> r[0] := primpart(a,x):
  r[1] := primpart(b,x):
  k := 1:
  while r[k] <> 0 do
      printf("k=%2d  degx=%2d  degy=%3d  size=%d\n",
              k,degree(r[k],x),degree(r[k],y),length(maxnorm(r[k])) )
  ;
      pr := prem(r[k-1],r[k],x);
      r[k+1] := primpart(pr,x);
      k := k+1;
  od:
  g := r[k-1];
  g := gcd(content(a,x),content(b,x))*g; # attach gcd of con
k= 1  degx=10  degy= 10  size=3
k= 2  degx= 9  degy= 11  size=7
k= 3  degx= 8  degy= 14  size=12
k= 4  degx= 7  degy= 19  size=16
k= 5  degx= 6  degy= 26  size=21
k= 6  degx= 5  degy= 35  size=26
k= 7  degx= 4  degy= 46  size=31
k= 8  degx= 3  degy= 59  size=36
k= 9  degx= 2  degy= 74  size=41
k=10  degx= 1  degy=  1  size=1
```

$$g := 2 x - 3 y - 5$$

$$g := 2 x - 3 y - 5 \quad \textbf{(24)}$$

There is now a two dimensional linear growth. The degree in y is growing and the size of the integer coefficients is also growing. This means the intermediate pseudo-remainders, before we get to the gcd, can be much larger than the input polynomials. Let us make the example larger and time Maple executing the primitive Euclidean algorithm and then time Maple doing the gcd using whatever algorithm it uses.

```
> a := expand( g*randpoly([x,y],degree=40,dense) ):
```

```
> b := expand( g*randpoly([x,y],degree=39,dense) ):
> st := time():
  r[0] := primpart(a,x):
  r[1] := primpart(b,x):
  k := 1:
  while r[k] <> 0 do
      printf("k=%2d  degx=%2d  degy=%3d  size=%d\n",
              k,degree(r[k],x),degree(r[k],y),length(maxnorm(r[k])) )
  ;
      pr := prem(r[k-1],r[k],x);
      r[k+1] := primpart(pr,x);
      k := k+1;
  od:
  g := r[k-1];
  g := gcd(content(a,x),content(b,x))*g; # attach gcd of contents
  time=time()-st;
```

```
k= 1  degx=40  degy= 40  size=3
k= 2  degx=39  degy= 41  size=7
k= 3  degx=38  degy= 44  size=12
k= 4  degx=37  degy= 49  size=16
k= 5  degx=36  degy= 56  size=21
k= 6  degx=35  degy= 65  size=26
k= 7  degx=34  degy= 76  size=32
k= 8  degx=33  degy= 89  size=37
k= 9  degx=32  degy=104  size=42
k=10  degx=31  degy=121  size=48
k=11  degx=30  degy=140  size=54
k=12  degx=29  degy=161  size=59
k=13  degx=28  degy=184  size=65
k=14  degx=27  degy=209  size=71
k=15  degx=26  degy=236  size=76
k=16  degx=25  degy=265  size=82
k=17  degx=24  degy=296  size=88
k=18  degx=23  degy=329  size=94
k=19  degx=22  degy=364  size=101
k=20  degx=21  degy=401  size=107
k=21  degx=20  degy=440  size=113
k=22  degx=19  degy=481  size=119
k=23  degx=18  degy=524  size=125
k=24  degx=17  degy=569  size=132
k=25  degx=16  degy=616  size=137
k=26  degx=15  degy=665  size=144
k=27  degx=14  degy=716  size=150
k=28  degx=13  degy=769  size=156
k=29  degx=12  degy=824  size=163
k=30  degx=11  degy=881  size=169
k=31  degx=10  degy=940  size=176
k=32  degx= 9  degy=1001  size=182
k=33  degx= 8  degy=1064  size=188
k=34  degx= 7  degy=1129  size=195
k=35  degx= 6  degy=1196  size=202
k=36  degx= 5  degy=1265  size=208
k=37  degx= 4  degy=1336  size=214
k=38  degx= 3  degy=1409  size=220
k=39  degx= 2  degy=1484  size=226
k=40  degx= 1  degy=  1  size=1
```

$$g := -2\,x + 3\,y + 5$$

$$g := -2\,x + 3\,y + 5$$

$$time = 346.809 \qquad \textbf{(25)}$$

```
> st := time(): gcd(a,b); time()-st;
```

$$2\,x - 3\,y - 5$$

$$0.024 \tag{26}$$

Maple is NOT using the primitive Euclidean algorithm! If we added a third variable z to this example, Maple might never finish. Basically, the complexity of the primitive Euclidean algorithm is exponential in the number of variables.

In the 1970s polynomial gcd computation was a major area of research because it was impossible to compute the gcd of two polynomials in many variables for even small degrees. This phenomenon where the intermediate expressions are much larger than the inputs and output(s) has been termed *intermediate expression swell.* It also occurs in Gaussian elimination when applied to a matrix of integers or polynomials.