Extended Euclid's Algorithm via Backward Recurrence Relations

S. P. Glasby
Department of Mathematics and Computing Science
The University of the South Pacific
Suva, Fiji Islands

Introduction Given elements a and b of a Euclidean ring R, Euclid's algorithm is most useful for computing the greatest common divisor $\gcd(a,b)$, or when $\gcd(a,b)$ is invertible, computing the inverse of $a+bR$ in the quotient ring R/bR. The second problem is usually solved by computing $x,y\in R$ satisfying $ax+by=\gcd(a,b)$ via the familiar backward substitution method. It seems less well known that solving $ax+by=\gcd(a,b)$ can be performed more efficiently using a "backward" recurrence relation. Many books, such as [**1**, **2**, **3**], use a "forward" recurrence relation method. I shall argue that the backward recurrence relation method is both pedagogically more natural for students, and more efficient for hand computations.

The ring $F[X]$ of polynomials over a field F and the ring $\mathbb{Z}$ of integers are examples of Euclidean rings. There is a "degree" function $\delta : R \to \mathbb{N}\cup\{-\infty\}$. Given $a,b\in R$ where $b\neq 0$ there exist a "quotient" q, and a "remainder" r in R such that $a=qb+r$ where $\delta(r)<\delta(b)$. The reader unfamiliar with Euclidean rings in general need not worry: in our examples, $R=\mathbb{Z}$ and q and r are the usual quotients and remainders, i.e., $q=\lfloor a/b\rfloor$ (where $\lfloor\ \rfloor$ denotes the greatest integer function), $r=a-qb$, and $\delta(r)=|r|$.

Given $a_0,a_1\in R$ where $a_1\neq 0$, there exist quotients $q_1,\dots,q_r$ and remainders $a_2,\dots,a_{r+1}$ in R such that $a_2,\dots,a_r$ are nonzero and

$$a_0=q_1a_1+a_2,\quad a_1=q_2a_2+a_3,\quad \dots\quad a_{r-1}=q_ra_r+a_{r+1}, \tag{1}$$

where $\delta(a_{r+1})<\cdots<\delta(a_2)<\delta(a_1)$. Since $\mathbb{N}\cup\{-\infty\}$ has no infinite descending sequences, this process can not continue indefinitely, so we shall assume that $a_r\neq a_{r+1}=0$. It follows from $\gcd(a_{i-1},a_i)=\gcd(a_i,a_{i+1})$ that $\gcd(a_0,a_1)=a_r$. Furthermore, by writing equations (1) in the form $a_{i+1}=a_{i-1}-q_ia_i$ and using backward substitution, one can show that there exist $x,y\in R$ such that $a_0x+a_1y=a_r$.

That is, starting with $a_r = a_{r-2} - q_{r-1}a_{r-1}$, one substitutes $a_{r-1} = a_{r-3} - q_{r-2}a_{r-2}$ and then, after collecting terms, $a_{r-2} = a_{r-4} - q_{r-3}a_{r-3}$ is substituted, etc.

When writing the equations (1) one sees the same terms written repeatedly: $a_2, \ldots, a_{r-1}$ each appear three times, and a_1, a_r appear twice. This suggests abbreviating these equations by

$$\begin{array}{cccccc} & q_1 & q_2 & \cdots & q_r & \\ \hline a_0 & a_1 & a_2 & \cdots & a_r & 0 \end{array} \tag{2}$$

A backward recurrence relation One can start with a_0, a_1 and generate higher a_i via the *forward recurrence* $a_{i+1} = -q_ia_i + a_{i-1}$, or less conventionally, start with $a_{r+1} = 0$ and a_r and generate lower a_i via a *backward recurrence*: $a_{i-1} = q_ia_i + a_{i+1}$. The backward substitution method corresponds to solving the equations $a_{i-1}x_i + a_iy_i = a_r$ for x_i and y_i until one finds x_1 and y_1. It turns out to be more convenient to consider the following related equations: $a_{i-1}x_{i-1} + a_i(-1)^{r+i}y_{i-1} = a_r$. Our initial conditions are then $x_r = 1$, $y_r = 0$ and $x_{r-1} = 0$, $y_{r-1} = 1$. A recurrence relation can be determined as follows:

$$\begin{aligned} a_r &= a_ix_i + a_{i+1}(-1)^{r+i+1}y_i \\ &= a_ix_i + (a_{i-1} - q_ia_i)(-1)^{r+i+1}y_i \\ &= a_{i-1}(-1)^{r+i+1}y_i + a_i(-1)^{r+i}(q_iy_i + (-1)^{r+i}x_i). \end{aligned}$$

Take $x_{i-1} = (-1)^{r+i+1}y_i$ and $y_{i-1} = q_iy_i + (-1)^{r+i}x_i$. Eliminating x_i gives the following recurrence relation:

$$y_r = 0, \quad y_{r-1} = 1 \quad \text{and} \quad y_{i-1} = q_iy_i + y_{i+1}.$$

Hence we may add an extra row to table (2) to compute the y's:

$$\begin{array}{cccccccccc} & q_1 & \cdots & & q_i & & \cdots & q_r & \\ \hline a_0 & a_1 & \cdots & q_ia_i + a_{i+1} & a_i & a_{i+1} & \cdots & a_r & 0 \\ y_0 & y_1 & \cdots & q_iy_i + y_{i+1} & y_i & y_{i+1} & \cdots & y_r & \end{array}$$

Recall that

$$\begin{aligned} a_r &= a_{i-1}x_{i-1} + a_i(-1)^{r+i}y_{i-1} \\ &= a_{i-1}(-1)^{r+i+1}y_i + a_i(-1)^{r+i}y_{i-1}. \end{aligned}$$

Putting $i = 1$ gives $a_0 y_1 - a_1 y_0 = (-1)^r a_r$. As an illustration, we compute $\gcd(74, 54)$ and solve $74x + 54y = \gcd(74, 54)$:

	1	2	1	2	3	
74	54	20	14	6	2	0
11	8	3	2	1	0	

Therefore $74 \times 8 - 54 \times 11 = (-1)^5 2$, so $x = -8$ and $y = 11$ is a solution.

The above calculation has advantages over the familiar method:

$$\begin{aligned} 2 &= 1 \times 14 - 2 \times 6 \\ &= 1 \times 14 - 2 \times (20 - 1 \times 14) = -2 \times 20 + 3 \times 14 \\ &= -2 \times 20 + 3 \times (54 - 2 \times 20) = 3 \times 54 - 8 \times 20 \\ &= 3 \times 54 - 8 \times (74 - 1 \times 54) = -8 \times 74 + 11 \times 54. \end{aligned}$$

Forward recurrence relations An alternative approach is to set $a_0 X_i + a_1 Y_i = a_i$ and seek the values of X_r and Y_r. Our initial conditions are $X_0 = 1, Y_0 = 0$ and $X_1 = 0, Y_1 = 1$. Furthermore,

$$\begin{aligned} a_{i+1} &= a_{i-1} - q_i a_i \\ &= (a_0 X_{i-1} + a_1 Y_{i-1}) - q_i (a_0 X_i + a_1 Y_i) \\ &= a_0 (X_{i-1} - q_i X_i) + a_1 (Y_{i-1} - q_i Y_i) \end{aligned}$$

so we may take $X_{i+1} = X_{i-1} - q_i X_i$ and $Y_{i+1} = Y_{i-1} - q_i Y_i$. Unlike the backward method, these recurrence relations are not coupled: one may solve for the X's independently of the Y's, and vice versa. Hence we may add two extra rows to table (2) to compute the X's and the Y's:

	q_1	$\cdots$		q_i		$\cdots$	q_r	
a_0	a_1	$\cdots$	a_{i-1}	a_i	$a_{i-1} - q_i a_i$	$\cdots$	a_r	0
X_0	X_1	$\cdots$	X_{i-1}	X_i	$X_{i-1} - q_i X_i$	$\cdots$	X_r	
Y_0	Y_1	$\cdots$	Y_{i-1}	Y_i	$Y_{i-1} - q_i Y_i$	$\cdots$	Y_r	

For example, if $a_0 = 74$ and $a_1 = 54$, this method gives

	1	2	1	2	3	
74	54	20	14	6	2	0
1	0	1	−2	3	−8	
0	1	−1	3	−4	11	

Therefore $74 \times (-8) + 54 \times 11 = 2$, yielding the same answer as before.

Comparing the methods The backward recurrence method was motivated by backward substitution which is taught to most students, and so is pedagogically preferable to the forward recurrence method. From the point of view of hand calculations, the backward recurrence method requires half the effort, and students are less likely to make a sign error. The X_i, Y_i, and y_i can be expressed in terms of polynomials in $q_1, \ldots, q_{r-1}$, called *continuants* (see [**2**]). Computing the y's instead of the X's and the Y's has half the computational complexity, as the polynomials arising are very similar. It follows from a symmetry property of continuants that $y_0 = (-1)^{r-1} Y_r$ and $y_1 = (-1)^r X_r$.

The forward method is good for computers when long computations are involved as the q's need not be stored — once q_i has been used to compute a_{i+1}, X_{i+1} and Y_{i+1}, it can be discarded. This makes writing a computer program for the forward method slightly easier than the backward method. This is no advantage for hand calculation, as the student will invariably have recorded each q_i on paper.

The point made in the first paragraph needs qualification. If one uses the forward recurrence method to compute $d = \gcd(a, b)$ and X, then Y can be computed, using three divisions, from the equation $Y = (1 - a'X)/b'$ where $a' = a/d$, $b' = b/d$. When computing the inverse of $a + bR$ in the quotient ring R/bR, the value of Y is irrelevant, and so one need only compute d and X.

When $R = \mathbb{Z}$ and a_0, a_1 are positive, it is annoying that the signs of the X's and the Y's alternate. This can be avoided by redefining X_i and Y_i as follows: $a_0 X_i - a_1 Y_i = (-1)^i a_i$. Then the forward recurrence relations become

$$\begin{aligned} X_0 = 1,\ X_1 = 0 \quad &\text{and} \quad X_{i+1} = X_{i-1} + q_i X_i, \\ Y_0 = 0,\ Y_1 = 1 \quad &\text{and} \quad Y_{i+1} = Y_{i-1} + q_i Y_i. \end{aligned}$$

This, however, is undesirable from the view point of teaching as then different rows of our table are computed via different rules (namely $a_{i+1} = a_{i-1} - q_i a_i$, $X_{i+1} = X_{i-1} + q_i X_i$, and $Y_{i+1} = Y_{i-1} + q_i Y_i$).

References

[1] Henri Cohen, *A Course in Computational Algebraic Number Theory*, Graduate Texts in Mathematics 138, Springer-Verlag, New York, NY, 1995

[2] Donald E. Knuth, *The Art of Computer Programming, vol. 2: Semi-Numerical Algorithms*, Addison-Wesley, Reading, MA, 1995

[3] M. Pohst and H. Zassenhaus, Algorithmic algebraic number theory, in *Encyclopedia of Mathematics and its Applications*, Addison-Wesley, Reading, MA, 1989