SOLVING CUBIC CONGRUENCES MODULO A PRIME ON THE TI-89

Joseph Fadyn
Southern Polytechnic State University
1100 South Marietta Parkway
Marietta, Georgia 30060
jfadyn@spsu.edu

INTRODUCTION

We consider the problem of solving the cubic congruence:

$$Ax^3 + Bx^2 + Cx + D \equiv 0 \pmod{p} \qquad (1)$$

where p is a prime greater than 3 using the TI-89. We assume that p does not divide A, for otherwise the congruence reduces to $Bx^2 + Cx + D \equiv 0 \pmod{p}$, which is quadratic. Quadratic congrences are discussed in the paper "Solving Quadratic Congruences Modulo a Prime On The TI-89" [2] which appears in the Proceedings of the ICTCM 2010. To solve the cubic congruence, we will follow the method of Zhi-Hong Sun as discussed in his paper "Cubic and Quartic Congruences Modulo a Prime" [4] (Journal of Number Theory, 2003). First, we observe that the cubic congruence has 0, 1 or 3 solutions (unless roots are repeated). Determining the number of solutions will be our first goal. To begin, by multiplying (1) by the inverse of A mod p, we may assume that the cubic is monic, of the form: $f(x) = x^3 + a_1 x^2 + a_2 x + a_3 \equiv 0 \pmod{p}$. Following Sun, we denote the number of solutions as $N_p(f(x))$. Let (a/p) be the Legendre symbol, and let $D = a_1^2 a_2^2 - 4 a_2^3 - 4 a_1^3 a_3 - 27 a_3^3 + 18 a_1 a_2 a_3$ be the discriminant of f(x). It is well known that: $N_p(f(x)) = 1$ if $(D/p) = -1$ and $N_p(f(x)) = 0$ or 3 if $(D/p) = 1$. To decide between the cases of 0 or 3 solutions, Sun uses the third-order recurring sequence $\{s_n\}$ defined by:

$$s_0 = 3,\ s_1 = -a_1,\ s_2 = a_1^2 - 2a_2,\ s_{n+3} + a_1 s_{n+2} + a_2 s_{n+1} + a_3 s_n = 0,\ (n \geq 0)$$

Then the result (Theorem 4.1 of Sun) states that if $p \nmid \left(a_1^2 - 3a_2\right)$ then $N_p(f(x)) = 0$ if and only if $s_{p+1}(a_1, a_2, a_3) \equiv a_2 \pmod{p}$, and $N_p(f(x)) = 3$ if and only if $s_{p+1}(a_1, a_2, a_3) \equiv a_1^2 - 2a_2 \pmod{p}$. Thus we are immediately faced with the problem of determining an efficient way of finding $s_{p+1}(a_1, a_2, a_3)$. Although Sun does not use matrix methods in his paper, this technique is suggested in a review of his paper in Zentralblatt Math [5]. Accordingly, let us define:

$$A = \begin{bmatrix} -a_1 & -a_2 & -a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \qquad v = \begin{bmatrix} a_1^2 - 2a_2 \\ -a_1 \\ 3 \end{bmatrix}$$

We compute $A^k \mathbf{v}$. Then the [1,1] element of $A^k \mathbf{v}$ will be s_k for $k > 3$. Programs to compute s_{p+1} and $s_{(p+1)/2}$ (which is needed later) employ the "fast matrix algorithm" of successive squaring and reducing by the modulus at each step. These programs follow:

```
sponemat(a1,a2,a3,n,e)
Prgm
DelVar b
RandMat(3,3) -> b : -a1 -> b[1,1] : -a2 -> b[1,2] : -a3 -> b[1,3] : 1 -> b[2,1] : 0 -> b[2,2]
0-> b[2,3] : 0 -> b[3,1] : 1 -> b[3,2] : 0 -> b[3,3]
matsp (b, e – 2, n) : pone * [a1^2 – 2*a2 ; -a1 ; 3] -> temp
tmp * [a1^2 – 2*a2 ; -a1 ; 3] -> te
EndPrgm

matsp(b, e, n)
Prgm
DelVar tmp, pone, tmp, te, mm:identity (romDim(b)) -> tmp: mod(b,n) -> m: m -> mm
While e ≠ 0 : diva(e,2) -> d: mod(tmp*m^d[1,2],n) -> tmp: d[1,1] -> e : mod(m^2,n) ->
m: EndWhile: mod(mm^2 * tmp^2 , n) -> pone
EndPrgm
```

The TI-89 function diva() is described in [2]. Accordingly, we write down the beginnings of our main program cunsolm() to solve cubic congruences modulo p:

```
cunsolm()
Prgm
ClrIo : modinv(a0,p) -> inv: mod(inv*a1,p) ->a1: modinv(a2,p) ->a2: modinv(a3,p)->a3
mod(((a1^2 – 3*a2)^3, p) -> a : 0->norots
If mod(a,p) = 0 Then : Goto stpa : EndIf
sponemat(a1,a2,a3,p,(p+1)/2): mod(temp[1,1],p) -> sponem : mod(te[1,1],p) -> sphalfm
If sponem = mod(a1^2 – 2*a2,p) Then
   Disp "THREE SOLUTIONS" : Pause: Goto stpe
ElseIf  sponem = mod(a2,p) Then : Disp "NO SOLUTION" : 0->norots: Goto stpf
Else
 Disp "ONE SOLUTION. IT IS:"
    mod(mod(-a1*sponem+2*a1*a2-9*a3,p)*modinv(mod(3*sponem-2*a1^2
+3*a2,p),p),p)->rt[1]: Display rt[1]: Pause : 1->norots: Goto stpf
EndIf
Lbl stpa
```

In this block of code, the TI-89 program modinv(a,m) which finds the inverse of a modulo m is displayed in [2]. The code above determines the number of solutions (in the case that $p \nmid \left(a_1^2 - 3\,a_2\right)$. The code also gives the solution in the case that there is a unique solution. This solution is provided by Sun on page 3 of his paper. The case that p|a is handled in the transfer to stpa. Although this information is not provided in Sun's paper, I have shown (using results from Sun) that if p|a then $x \equiv \left(\frac{1}{3}\right)\left(-a_1 + b^{1/3}\right) \pmod{p}$. Here $b = -2\,a_1^3 + 9\,a_1\,a_2 - 27\,a_3$

Using this, we may proceed with our coding of cunsolm():

```
Lbl stpa
mod(-2*a1^3+9*a1*a2-27*a3,p) ->b: modinv(3,p) -> tin
If mod(b,p)=0 Then: Disp "There is a Triple Root": Disp "It Is:"
mod(-a1*tin,p)->rt[1]: 1->norots: Pause: Goto stpf: EndIf
If mod(p,3)=2 Then: Disp "One Solution. It Is:": curtmdp(b,p)
mod((-a1+z)*tin,p)->rt[1]: Disp rt[1]: Pause: 1->norots: Goto stpf: EndIf
mdexp(b,(p-1)/3,p): mod(z,p) -> z: If z ≠0 Then: Disp "There is NO SOLUTION"
0->norots: Pause: Goto stpf: EndIf
curtmdp(b,p): Disp "There are THREE SOLUTIONS": Disp "They Are:"
mod((-a1*crot)*tin,p) ->rt[1]: Disp rt[1] : Pause: mod(crot+rt[1],p)->crot2
mod((-a1*crot2)*tin,p)->rt[2]: Disp rt[2] : Pause: mod(crot+rt[3],p)->crot3
mod((-a1*crot3)*tin,p)->rt[3]: Disp rt[3] : Pause: 3->norots: Goto stpf
Lbl stpe
```

This program segment calls the cube root function curtmdp(b,p) which follows:

```
curtmdp(cc,p)
Prgm
DelVar ppa, f: p->ppa
If mod(p,3)=2 Then
  exeualg(p-1,3): mod(v+p-1,p)->v : mdexp(cc,v,p): Disp "One Cube Root:"
   Disp  mod(z,p): Goto stp10
EndIf
shanks3(cc,p): Display "Other Cube Roots Are:" DelVar x
x^2 + mod(3*crot,ppa)*x+mod(3*crot^2,ppa)->f(x): qcon(ppa):
Disp mod(crot+rt[1],p): Pause: disp "And": Disp mod(crot+rt[2],p)
Lbl stp10
EndPrgm
```

Program calls here include calls to exeualg(), shanks3() and qcon(). These represent the extended Euclidean algorithm, Shank's Algorithm (for cube roots) and a quadratic congruence solver for prime moduli p. The coding for these follow:

```
exeualg(a,b)
Prgm
ClrIO: DelVar s0,v0,s1,v1: 1->s0: 0->v0: 0->s1: 1->v1
If a = gcd(a,b) Then: 1->s:0->v: End If
Ib b = gcd(a,b) Then: 0->s: 1->v: EndIf
While b≠0: diva(a,b)->d: s0-d[1,1]*s1->s: v0-d[1,1]*v1->v: s1->s0: v1->v0
s->s1: v->v1: b->a: d[1,2]->b: Disp [[s,v]]: EndWhile: Disp [[s,v]]
EndPrgm

shanks3(dd,p)
Prgm
```

```
mdexp(dd,(p-3)/3,p): If mod(z,p) ≠1 Then: Disp "No Cube Root Exists": Stop:EndIf
1->n: p-1->h: While fPart(h/3)=0: h/3->h: n+1->n: EndWhile: n-1->n
(p-1)/3^n->k: 3->q: mdexp(q,(p-1)/3,p): While z=1: q+1->q: mdexp(q,(p-1)/3,p)
EndWhile: mdexp(dd,k,p): z->r: If mod(k,3)=1 Then mdexp(dd,(k-1)/3,p)
Else: mdexp(dd,(k+1)/3,p): EndIf: z->t: Lbl stp1: 0->i: mdexp(r,3^I,p): While z≠1
1+i->i: mdexp(r,3^I,p): EndWhile: If i=0 Then: If mod(k,3)=2 Then
Disp "One (of 3) Solutions Is:": mod(t,p)->crot: Else Disp "One (of 3) Solutions Is:"
modinv(mod(t,p),p)->crot: Disp crot: EndIf: Go to stp8: Else: k*3^(n-i-1)->v
mdexp(q,v,p): z->u: mod(t*u,p)->t: mod(r*u^3,p)->r: Goto stp1: EndIf: Lbl stp8
EndPrgm
```

```
qucon(p)
Prgm
DelVar r: ClrIO: 0->norots: f(0)->c: d(f(x),x)|x=0 ->b: d(f(x),x,2)/2 ->a
If mod(a,p)=0 Then: If mod(b,p) ≠0 Then: Disp "Linear, One Solution:": Pause:
mod(modinv(b,p)*-c,p)->rt[1]: Disp rt[1]: Pause: 1->norots: Goto stp6: Else:
0->norots: Goto stp6: EndIf: EndIf: mod(b^2-4*a*c,p)->di: mdexp(di,(p-1)/2,p)
0->norots: If p=2 Then: 0->j: For i,0,1,1 If mod(f(i),p)=0 Then: i->rt[j+1]: j+1->j
norots+1->norots: EndIf: EndFor: Goto stp12: EndIf: If mod(di,p)=0 Then
mod(modinv(2*a,p)*-b,p)->rt[1]: 1->norots: Goto stp6: ElseIf z=p-1 Then: Goto stp20
Else: End If: If fPart(√(di))=0 Then: mod(modinv(2*a,p)*(-b+(√(di)),p)->rt[1]
mod(modinv(2*a,p)*(-b-(√(di)),p)->rt[2]: 2->norots: Goto stp6: EndIf
If mod(p,4)=3 Then: mdexp(di,(p+1)/4,p): mod(modinv(2*a,p)*(-b+z),p)->rt[1]
mod(modinv(2*a,p)*(-b-z),p)->rt[2]: 2->norots: Goto stp6: EndIf
If mod(p,8)=5 Then mdexp(di,(p+3)/8,p): z->w: If mod(z^2,p)=di Then
mod(modinv(2*a,p)*(-b+z),p)->rt[1]: mod(modinv(2*a,p)*(-b-z),p)->rt[2]
2->norots: Goto stp6: EndIf: EndIf: If mod(p,8) = 5 Then: mdexp(2,(p-1)/4,p)
mod(w*z,p)->z: If mod(z^2,p)=di Then: mod(modinv(2*a,p)*(-b+z),p)->rt[1]
mod(modinv(2*a,p)*(-b-z),p)->rt[2]: 2->norots: Goto stp6: EndIf: EndIf
1->n: p-1->h: While fPart(h/2)=0 h/2->h: n+1->n: (p-1)/2^n->k: 2->q
mdexp(q,(p-1)/2,p): While z=1: q+1->q: mdexp(q,(p-1)/2,p): Endwhile
mdexp(di,k,p): z->r: mdexp(di,(k+1)/2,p): z->t: Lbl stp1: 0->i: mdexp(r,2^i,p)
While z ≠ 1: 1+i->i: mdexp(r,2^i,p): EndWhile: If i=0 Then
mod(modinv(2*a,p)*(-b+z),p)->rt[1]: mod(modinv(2*a,p)*(-b-z),p)->rt[2]
2->norots: Goto stp6: Else: k*2^(n-i-1)->v: mdexp(q,v,p): z->u: mod(t*u,p)->t
mod(r*u^2,p)->r: Goto stp1: EndIf: Lbl stp6: Lbl stp12: norots->j: Lbl stp20:
EndPrgm
```

In the last program segment of cunsolm() the transfer of control to stpe deals with the three solution case. This case is more difficult. On page 3 of Sun, we have that if $N_p(f(x)) = 3$, $p \nmid D$, and $x_0 = (1/2)((-a_3/p)\, s_{(p+1)/2} - a_1) \equiv -a_1 \pmod p$, then: $x \equiv x_0$, $-\frac{1}{2}a_1 - \frac{1}{2}x_0 + \frac{1}{2}\frac{d}{3x_0^2 + 2a_1x_0 + a_2}$

$$-\frac{1}{2}a_1 - \frac{1}{2}x_0 - \frac{1}{2}\frac{d}{3x_0^2 + 2a_1x_0 + a_2}$$

are the three solutions mod p, where d is an integer such that $d^2 \equiv D \pmod{p}$. We continue our coding to cunsolm() as follows:

```
Lbl stpe
mod(cdisc(a1,a2,a3),p)->dis: mod(-a3,p)->a3t: mdexp(a3t,(p-1)/2),p): If z=p-1 Then
-1->z: EndIf: modinv(2,p)->toin: mod((z*sphalfm-a1)*toin,p)->so
Of dis≠0 and so≠mod(-a1,p) Then: Disp "Three Solutions": Disp "One Solution Is:"
so->rt[1]: Disp rt[1]: Pause: sqrtmdp(dis,p):
mod(rot[1]*modinv(mod(3*so^2+2*a1*so+a2,p),p),p)
->qua: Disp "Other Solutions Are:" : mod((-a1-so+qua)*toin,p)->rt[2]: Disp rt[2]:
Pause: mod((-a1-so-qua)*toin,p)->rt[3]: Disp rt[3]: 3->norots: Goto stpf: EndIf
```

This program segment calls the function cdisc (the cubic discriminant) and the program sqrtmdp (which extracts the square root modulo a prime p when one exists). Listings of these follow:

```
cdisc(a1,a2,a3): Func: a1^2*a2^2-4*a2^3-4*a1^3*a3-27*a3^2+18*a1*a2*a3: EndFunc
```

```
sqrtmdp(dd,p): Prgm: DelVar r,rot: ClrIO: mod(dd,p)->dd: mdexp(dd,(p-1)/2,p)
If mod(dd,p)=0 Then: Disp "Root Is:" 1->numrots: 0->rot[1]: Goto stp6: ElseIf Z=p-1
Then: Disp "No Root Exists": 0->numrots: Goto stp6: Else: Disp "Two Roots Exist"
EndIf: If mod(p,4)=3 Then: mdexp(dd,(p+1)/4,p): Disp "Roots Are:" : mod(z,p)->rot[1]
Disp rot[1]: Disp "And:" : mod(-z,p)->rot[2]: Disp rot[2]: 2->numrots: Goto stp6: EndIf
If mod(p,8)=5 Then: mdexp(dd,(p+3)/8,p): z->w: If mod(z^2,p)=dd Then: Disp "Roots
Are:" mod(z,p)->rot[1]: Disp rot[1]: Disp "And:" mod(-z,p)->rot[2]: Disp rot[2]
2->numrots: Goto stp6: EndIf: End If: If mod(p,8)=5 Then:mdexp(2,(p-1).4,p):
mod(w*z,p)->z: If mod(z^2,p)=dd Then: Disp "Roots Are:" : Disp "Roots Are:"
mod(z,p)->rot[1}: Disp rot[1]: Disp "And:" mod(-z,p)->rot[2]: Disp rot[2]: 2->numrots
Goto stp6: EndIf: EndIf: 1->n: p-1->h: While fPart(h/2)=0: h/2->h: n+1->n: EndWhile
n-1->n: (p-1)/2^n->h: 2->q: mdexp(q,(p-1)/2,p): While z=1: q+1->q:mdexp(q,(p-1)/2,p)
EndWhile: mdexp(dd,k,p): z->r: mdexp(dd,(k+1)/2,p): z->t: Lbl stp1: 0->i
mdexp(r,2^i,p): While z≠1: 1+i->i: mdexp(r,2^i,p): EndWhile: If i=1 Then: Disp "Roots
Are:": mod(t,p)->rot[1]: Disp rot[1]: Disp "And": mod(-t,p)->rot[2]: Disp rot[2]
2->numrots: Goto stp6: Else: k*2^(n-i-1)->v: mdexp(q,v,p): z->u: mod(t*u,p)->t
mod(r*u^2,p)->r: Goto stp1: EndIf: Lbl stp6: Pause: Lbl stp12: EndPrgm
```

Next we consider the case where p|D. Lemma 4.1 in Sun provides the three solutions in the case that $p \nmid \left(a_1^2 - 3a_2\right)$.

And the two fold root: $-a_1 + \dfrac{a_1 a_2 - 9a_3}{a_1^2 - 3a_2}$

$$-\frac{a_1 a_2 - 9 a_3}{2 a_1^2 - 6 a_2}$$

are the three solutions mod p in this case. We have already considered the case where $p|(a_1^2 - 3a_2)$ earlier. We continue with the next program segment of cunsolm():

If dis=0 Then: DelVar qu, qu3
mod((a1*a2-9*a3)*modinv(mod(a1^2-3*a2,p),p,),p)->qu: mod(qu*toin,p)->qu2
Disp "Three Roots": Disp "A Two-Fold Root Is:" : mod(-qu2,p)->rt[1]: Disp rt[1]: Pause Disp "The Other Root Is:": mod(-a1*qu,p)->rt[2]: Disp rt[2]: 2->norots: Goto stpf: EndIf

Next we consider the case where $N_p(f(x))=3$, $p \nmid a \cdot (b^2 - 4 \cdot a)$ and $p \equiv 1 \pmod 3$. This is handled by Theorem 4.5 in Sun. We comment that if $p \mid a\ (b^2 - 4a)$, then either $p \mid a$ or $p \mid D$. This is obvious since Sun shows that $b^2 - 4a \equiv -27\ D \pmod p$. The cases $p \mid a$ and $p \mid D$ have already been dealt with. Theorem 4.5 assures us in this case that the cubic congruence $z^3 \equiv b^2 - 4a \pmod p$ has three solutions z_1, z_2 and z_3 and that for i = 1,2,3 , the three solutions to the congruence modulo p are given by:

$$x \equiv \left(\frac{1}{6} \frac{(z_i - a_1)^2 + 3 a_1^2 - 12 a_2}{z_i} \right)$$

We continue with the next program segment on cunosolm():

If mod(p,3)=1 Then: mod(-2*a1^3+9*a1*a2-27*a3,p)->b: mod((a1^2-3*a2)^3,p)->a
sqrtmdp(mod(b^2-4*a),p),p): rot[1]->y: curtmdp(mod(4*(b-y),p),p): crot->ze1
mod(crot+rt[1],p)->ze2: mod(crot+rt[2],p)->ze3: Disp "The Three Roots Are:"
modinv(6*ze1,p)->in1: mod(((ze1-a1)^2+3*(a1^2-4*a2))*in1,p)->rt[1]: Disp rt[1]:
Pause : modinv(6*ze2,p)->in2: mod(((ze2-a1)^2+3*(a1^2-4*a2))*in2,p)->rt[2]: Disp
rt[2]: Pause: modinv(6*ze3,p)->in3: mod(((ze3-a1)^2+3*(a1^2-4*a2))*in3,p)->rt[3]:
Disp rt[3]: 3->norots: Goto stpf: EndIf

We are left with the case that $N_p(f(x))=3$, $x_0 = (1/2)((-a_3/p)\ s_{(p+1)/2} - a_1) \equiv -a_1 \pmod p$, and $p \equiv 2 \pmod 3$. I could find no resolution of this case in Sun. For this case, I will use a more classical method described in Morain ("Solving Equations of Small Degree Using Large Primes" [3]). Morain's approach solves the (depressed) cubic:

$$X^3 + uX + v \equiv 0 \pmod p \qquad (2)$$

where $x = X - a_1 / 3$. If α and β are solutions of the quadratic equation: $y^2 + (3vy)/u - u/3 = 0$, then Lemma 5.1 of Morain states that if X_0 is a root of (2) and $z = (X_0 - \alpha) / (X_0 - \beta)$, then $z^3 = A = \alpha / \beta$. The difficulty in using this method in the case we are working on is that the discriminant of the quadratic is generally not a quadratic residue mod p so that z is an element of the field **GF**(p^2), which is a finite field of order p^2. Therefore z^3 is also in **GF**(p^2), and so finding z requires finding cube roots in that same field. For this task we will employ an algorithm listed in the book "Algorithmic Number Theory" [1] by Bach and Shallit on pages 160-161 for finding

roots in finite fields. This algorithm finds only one root, but then the other two roots of z^3 which we will need are fairly easy to obtain. We continue with the coding of the program cubts2() which will deal with this case as well as other supporting programs:

```
cubts2(a1,a2,a3,p)
Prgm: alphandb(a1,a2,a3,p): mdexp(ddq,(p-1)/2,p): z->z3: If bbq=0 or ze=1 Then:
Goto stpb: EndIf: mod(-bbq,p)->bbqp: mddiv(aaq,bbq,aaq,bbqp,p):
rrotinfg(did[1,1],ddq,did[1,2],3,p): modinv(3,p)->tin:
mdpd(aaq,ddq,bbqp,cubrt[1,1],cubrt[1,2],p)
mddiv(uu[1,1]-aaq,uu[1,2]-bbq,cubrt[1,1]-1,cubrt[1,2],ddq,p): Disp "A Solution Is:"
mod(did[1,1]-tin*a1,p)->rut1: Disp rut1: rut1->rt[1]: Pause
mdpd(aaq,ddq,bbqp,cubrt2[1,1],cubrt2[1,2],p)
mddiv(uu[1,1]-aaq,uu[1,2]-bbq,cubrt2[1,1]-1,cubrt2[1,2],ddq,p): Disp "A Solution Is:"
mod(did[1,1]-tin*a1,p)->rut2: Disp rut2: rut2->rt[2]: Pause
mdpd(aaq,ddq,bbqp,cubrt3[1,1],cubrt3[1,2],p)
mddiv(uu[1,1]-aaq,uu[1,2]-bbq,cubrt3[1,1]-1,cubrt3[1,2],ddq,p): Disp "A Solution Is:"
mod(did[1,1]-tin*a1,p)->rut3: Disp rut3: rut3->rt[3]: Lbl stpb : EndPrgm

alphandb(a1,a2,a3,p)
Prgm: modinv(3,p)->tin: If mod(a1,p)=0 Then: mod(a2,p)->uu: mod(a3,p)->vv:
Goto stp8: EndIf: mod(a2-a1^2*tin,p)->uu
mod(a3*s*a1^3+modinv(27,p)-a1*a2*tin,p)->vv: Lbl stp8
mod(3*vv*modinv(uu,p),p)->uuu: mod(-uu*tin,p)->vvv: x^2+uuu*x+vvv->f(x)
mod(modinv(2,p),p)->bbq: mod(-uuu*bbq,p)->aaq: mod(uuu^2-4*vvv,p)->ddq
EndPrgm

mddiv(a,b,c,d,e,p)
Prgm: randMat(1,2)->did: modinv(mod(c^2-d^2*e,p),p)->mi:
mod(mi*(a*c-b*d*e),p)->did[1,1]: mod(mi*(c*b-a*d),p)->did[1,2]: EndPrgm

rrotinfg(c,e,d,r,p)
Prgm: p^2-1->pq: ClrIO: mdexp(c,e,d,pq/r,p):If pp[1,1]≠1 or pp[1,2] ≠0 Then Disp "No
Root Exists": Stop:EndIf: For i,2,p: For j,3,p: mdexpi(i,e,j,pq/r,p): If pp[1,1]≠1 or
pp[1,2]≠0:Then: i->h1: j->h2: Goto stp2: EndIf: EndFor: EndFor: Lbl stp2: 1->s: While
fPart(pq/3)=0 : pq/3->pq: s+1->s: EndWhile: s-1->s: (p^2-1)/3^s->t: mdexpi(c,e,d,t,p)
pp->ar: mdexpi(c,e,d,r^s,p): pp->at: mdexp(h1,e,h2,t,p): pp_>gg: 0->ee: For i,1,s-1
For j,0,r-1:mdexpi(gg[1,1],e,gg[1,2],ee+j*r^i,p)
mdpd(pp[1,1],e,pp[1,2],ar[1,1],ar[1,2],p): mdexpi(uu[1,1],e,uu[1,2],r^(s-i-1),p)
If pp[1,1]=1 and pp[1,2]=0 Then :Goto stp3: EndIf: EndFor: Lbl stp3: ee+j*r^i->ee
EndFor: modinv(r,t)->rp: If ee=0 Then: [1,0]->br: Else: qinvmdp(gg,e,p)
mdexpi(qqin[1,1],e,qqin[1,2],ee/r,p): pp->br: EndIf: mdexpi(at[1,1],e,at[1,2],rp,p)
pp->bt: s->ss: exeualg(t,r^ss): s->alp: v->bet: If alp<0 Then: qinvmdp(br,e,p)
mdexpi(qqin[1,1],e,qqin[1,2],abs(alp),p): pp->bra: Else: mdexpi(br[1,1],e,br[1,2],alp,p)
pp->bra:EndIf:If bet<0 Then: qinvmdp(bt,e,p): mdexpi(qqin[1,1],e,qqin[1,2],abs(bet),p)
Else: mdexpi(bt[1,1],e,bt[1,2],bet,p): pp->btb: EndIf:
```

mdpd(bra[1,1],e,bra[1,2],btb[1,1],btb[1,2],p): uu->cubrt: Disp "One Cube Root Is:"
Disp cubrt: Pause: mdexpi(gg[1,1],e,gg[1,2],(p^2-1)/r,p): pp->cru: Disp "The Other Cube Roots Are:" : mdpd(cru[1,1],e,cru[1,2],cubrt[1,1],cubrt[1,2],p): uu->cubrt2: Disp cubrt2: Disp "And:" mdpd(cru[1,1],e,cru[1,2],cubrt2[1,1],cubrt2[1,2],p): uu->cubrt3
Disp cubrt3: EndPrgm

mdpd(a,b,k,c,e,n)
Prgm: DelVar uu: [mod(a*c+k*e*b,n),mod(c*k+a*e,n)]->uu: EndPrgm

mdexpi(a,b,c,e,n)
Prgm: DelVar pp,mm: [1,0]->pp: mod(a,n)->aa: mod(c,n)->cc: [11,cc]->mm: While e≠0: diva(e,2)-> d: If d[1,2]=0 Then: Goto stp10: EndIf:
mdpd(pp[1,1],b,pp[1,2],mm[1,1],mm[1,2],n): uu->pp: Lbl stp10: d[1,1]->e
mdpd(mm[1,1],b,mm[1,2],mm[1,1],mm[1,2],n): uu->mm:EndWhile: Disp pp: EndPrgm

qinvmdp(qq,e,p)
Prgm: DelVar qqin: mod(qq[1,1]^2-qq[1,2]^2*e,p)->kq: modinv(kq,p)->kp:
RandMat(1,2)->qqin: mod(kp*qq[1,1],p)->qqin[1,1]: mod(kp*-qq[1,2]),p)->qqin[1,2]
EndPrgm

We are now able to complete the coding of our main program cunsolm() as follows:

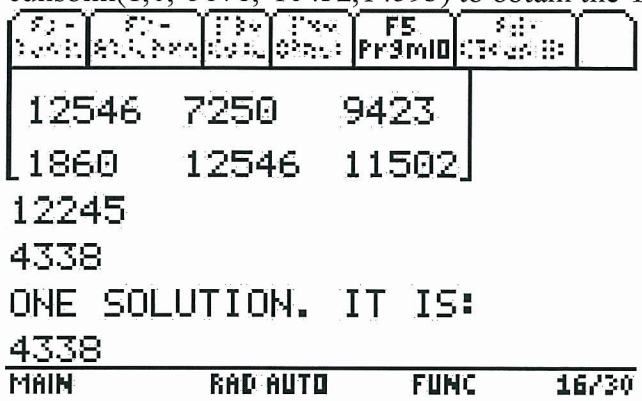
cubts2(a1,a2,a3,p)
If ze=1 Then: Goto stpc: EndIf: 3->norots: Goto stpf: Lbl stpc: sqrtmdp(ddd,p)
mod(aaq*bbq*rot[1],p)->tp: mod(aaq-bbq*rot[1],p)->bot: mdiv(bot,p)->boti
mod(tp*boti,p)->zcube: curtmdp(zcube,p): mod(z,p)->croot: modinv(3,p)->tin
modinv(croot-1,p)->crootin: mod(bot*croot-tp+crootin,p)->xzero
mod(xzero-tin*a1,p)->xone: Disp "A Solution Is:" : Disp xone: xone->rt[1]:Pause
modinv(2,p)->toin: mod(-3*xone^2-2*a1*xone+a1^2-4*a2,p)->undr
sqrtmdp(undr,p): Disp "Other Solutions Are:" mod(toin*(-a1-xone+rot[1]),p)->rt[2]:
Disp rt[2]: Pause: Disp "And:": mod(toin*(-a1-xone-rot[1]),p)->rt[3]: Disp rt[3]:
3->norots: Lbl stpf: EndPrgm

This completes the coding of our main program cunsolm(). Let's consider some examples:

Example 1: $x^3 - 5678x - 10432 \equiv 0 \pmod{14593}$.

We have $a_1 = 0$, $a_2 = -5678$, $a_3 = -10432$, $p = 14593$. If we run sponemat as follows: sponemat(0,-5678,-10432,14593,14594), we obtain $s_{p+1} = s_{14594} = 12245$. It is easily shown that $s_{14594} \not\equiv a_1^2 - 2a_2 \equiv 5156 \pmod{14593}$ and $s_{14594} \not\equiv a_2 \equiv 8915 \pmod{14593}$ and that $a_1^2 - 3a_2 \equiv 2441 \pmod{14593}$, so that $p \nmid \left(a_1^2 - 3a_2\right)$. Therefore there is one solution as indicated by Sun on page 3 of his paper. We employ:

cunsolm(1,0,-5678,-10432,14593) to obtain the TI-89 screen output:

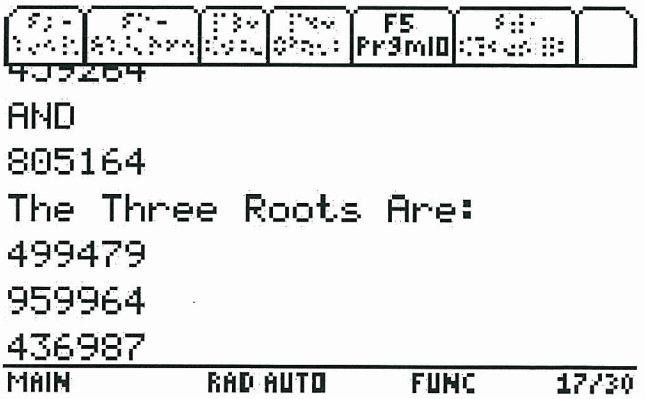


Example 2: $x^3 + 78964x^2 - 898432x - 345876 \equiv \pmod{987697}$.

We have $a_1 = 78964$, $a_2 = -898432$, $a_3 = -345876$, $p = 987697$. Note that $p \equiv 1 \pmod 3$. This is a nontrivial example of the case $N_p(f(x))=3$, $x_0 = (1/2)((-a_3/p)\, s_{(p+1)/2} - a_1) \equiv -a_1 \pmod p$, and $D \not\equiv 0 \pmod p$, so that we are unable to use the solution x_0 displayed on page 3 of Sun. We see this via the computations:

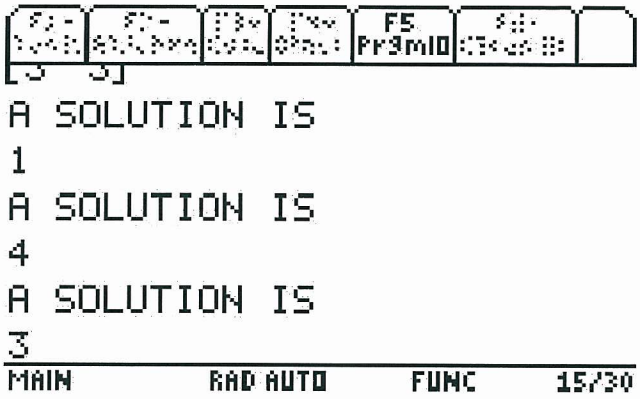
sponemat(78964,-898432,-345876,987697,(987697+1)/2) = 908733 = $s_{(p+1)/2}$, modinv(2,987697) = 493849 (the inverse of 2 modulo p), mdexp(-a3,(p-1)/2,p) yields: mdexp(345876,493849,987697), which gives $z = 1 = (-a_3/p)$. Next we compute: $x_0 \equiv$ mod(493849*(1-908733-78964),987697) to obtain 908733, and indeed,

$-a_1 \equiv -78964 \equiv 908733 \pmod{987697}$. It is also easily verified that $D \not\equiv 0 \pmod p$, using: mod(cdisc(78964,-898432,-345876),987697) $\equiv 478403 \pmod{987697}$. In this case our solution is obtained from Theorem 4.5 on page 36 of Sun which we coded in program cunsolm() on page 6 of this paper. The solutions modulo 987697 are: $x \equiv 499479$, $x \equiv 959964$, and $x \equiv 436987$ as indicated by running cunsolm() as follows: cunsolm(1, 78964,-898432,-345876,987697) to obtain the TI-89 screen output:



Example 3: $x^3 + 3x^2 + 8x + 10 \equiv 0 \pmod{11}$.

We have $a_1 = 3$, $a_2 = 8$, $a_3 = 10$, $p = 11$. Note that $p \equiv 2 \pmod 3$. It is easily verified that $s_{(p+1)/2} = s_6 = 8$, the inverse of 2 mod 11 is 6 and that $(-a_3/p) = (-10/11) = 1$. Now we compute $x_0 \equiv 6(1*8 - 3) \bmod 11 \equiv 8 \pmod{11} \equiv -3 \pmod{11} \equiv -a_1 \pmod p$. We are in the case where $N_p(f(x))=3$, $x_0 = (1/2)((-a_3/p)\, s_{(p+1)/2} - a_1) \equiv -a_1 \pmod p$, and $D \not\equiv 0 \pmod p$, so this example will test our implementation of Morain's technique which we described on page 7 of this paper. Accordingly we run: cunsolm(1,3,8,10,11) to obtain the TI-89 screen output:



References

1. E. Bach and J. Shallit, *Algebraic Number Theory, Volume 1*, The MIT Press, 1996.
2. J.N. Fadyn, *Solving Quadratic Congruences Modulo a Prime on The TI-89,* Proceedings of the ICTCM 2010.
3. Morain, Francois, *Solving Equations of Small Degree Modulo Large Primes*, 1989.
4. Z.H. Sun, *Cubic and Quartic Congruences Modulo A Prime*, Journal of Number Theory, 102, No. 1, 41-89 (2003).
5. Zentralblatt Math 1868-2008, Zbl 1033.11003 , 2008.