Finite fields with GP

A. Page

IMB
Inria/Université de Bordeaux

16/01/2019

INTMOD's and POLMOD's

You can perform operations in quotients with `Mod`.

```
? p = randomprime(2^100)
% = 792438309994299602682608069491
? a = Mod(2,p);
? type(a)
% = "t_INTMOD"
? a^(p-1)
% = Mod(1, 792438309994299602682608069491)
? a.mod == p
% = 1
? lift(a) \\lift to Z
% = 2
```

INTMOD's and POLMOD's

You can perform operations in quotients with `Mod`.

```
? T = x^2+1;
? b = Mod(x+a, T);
? type(b)
% = "t_POLMOD"
? b^(p+1)
% = Mod(Mod(5,79243...69491),x^2+1)
? b.pol
% = Mod(1, 79243...69491)*x + Mod(2,79243...69491)
? b.mod == T
% = 1
```

Creation of finite fields and FFELT's

There is no finite field structure, finite fields are represented only by elements.

```
? c = ffgen(3^8,'c) \\generator of F_3^8 as a field
% = c
? type(c)
% = "t_FFELT"
? c.p
% = 3
? c.mod \\defining polynomial, lifted to Z
% = c^8 + c^7 + 2*c^6 + c^3 + 2*c^2 + 2*c + 1
? polisirreducible(c.mod*Mod(1,3))
% = 1
? c.f \\degree over F_3
% = 8
```

Creation of finite fields and FFELT's

```
? d = c^9+1
% = 2*c^7 + 2*c^6 + 2*c^4 + 2*c^3 + c + 2
? d.pol
% = 2*c^7 + 2*c^6 + 2*c^4 + 2*c^3 + c + 2
? type(d.pol)
% = "t_POL"
```

You can directly get an irreducible polynomial with `ffinit`.

```
? ffinit(3,5)
% = Mod(1,3)*x^5+Mod(1,3)*x^4+Mod(2,3)*x^3+Mod(1,3)
```

You can also supply your own defining polynomial. We do not check for irreducibility.

```
? ffgen(x^2+Mod(1,3))
% = x
```

Creation from number fields

You can create finite fields as residue fields of prime ideals.

```
? nf = nfinit(y^8-2*y^7+9*y^6-2*y^5+38*y^4-34*y^3\
?             +31*y^2-6*y+1);
? pr = idealprimedec(nf,2)[1]; [pr.e,pr.f]
% = [2, 2]
? g = nfmodpr(nf,y,pr)
% = y + 1
```

You can also initialise a structure with `modprinit` to avoid recomputing information.

```
? modpr = nfmodprinit(nf,pr);
? nfmodpr(nf,y^2+1,modpr)
% = y + 1
? nfmodprlift(nf,g+1,modpr) \\find a preimage
% = [0, 1, 0, 0, 0, 0, 0, 0]~
```

Operations with elements

You can use many generic functions with finite field elements.

```
? [c,c+1;2*c,1]^-1
% = [...]
? d = random(c) \\random element in the field
% = c^5 + 2*c^4 + c^3 + 2*c^2 + c
? issquare(d)
% = 1
? trace(d) \\over F_3
% = Mod(2, 3)
? norm(d)
% = Mod(1, 3)
? minpoly(d^82)
% = Mod(1,3)*x^4+Mod(1,3)*x^2+Mod(1,3)*x+Mod(1,3)
```

Operations with elements

You can use many generic functions with finite field elements.

```
? factor(x^5+x^3+c) \\variants factormodSQF
                          and factormodDDF
% = [x + (2*c^5 + c^4 + 2*c) 1]
    [x^2 + (c^7 + 2*c^6 + ... + c^2 + 2) 1]
    [x^2 + (2*c^7 + c^6 + ... + 2*c^2 + 1) 1]
? polrootsmod(x^7+x+c)
% = [c^7 + 2*c^6 + c^5 + c^3 + 2*c + 2,
     2*c^7 + c^6 + c^2 + 1]~
```

Operations related to the multiplicative structure

Warning : the field generator is not necessarily a primitive root (group generator) !

```
? fforder(c)
% = 1640
? z = ffprimroot(c)
% = c^6 + c^5 + c^3 + c^2 + 2*c + 1
? fforder(z)
% = 6560
? n = fflog(c,z)
% = 6332
? c == z^n
% = 1
```

Reminder : there are corresponding functions on rings $\mathbb{Z}/N\mathbb{Z}$: `znorder`, `znprimroot`, `znlog`.

Curves

Elliptic curves operations and hyperelliptic curves point counting are available over finite fields.

```
? E = ellinit([c,1]);
? E.cyc \\structure of the group of points
% = [80, 80]
? ellissupersingular(E)
% = 1
? hyperellcharpoly(x^7+c*x+2) \\y^2 = x^7+c*+2
% = x^6 - 81*x^5 + ... + 282429536481
```

See `?12` for more !

Maps between finite fields

There is a structure for maps between finite fields.

```
? d = ffgen([3,24],'d)
% = d
? Mcd = ffembed(c,d); \\compute some embedding
? ffembed(d,c)
  ***   at top-level: ffembed(d,c)
  ***                 ^------------
  *** ffembed: domain error in ffembed: d is not a
? c2 = ffmap(Mcd,c^5+c+1) \\apply the map
% = d^20 + 2*d^18 + ... + 2*d^5 + 1
? F = fffrobenius(d,8); \\8-th power of Frobenius
? ffmap(F, d) == d
% = 0
? ffmap(F, c2) == c2
% = 1
```

Extending finite fields

You can construct extensions of finite fields defined by an irreducible polynomial with `ffextend`.

```
? T = ffinit(3,5)
% = Mod(1,3)*x^5+Mod(1,3)*x^4+Mod(2,3)*x^3+Mod(1,3)
? [e,Mde] = ffextend(d, T, 'e);
? e.f
% = 120
? fforder(e)
% = 242
? ffmap(Mde, d)
% = ...
```

Composing maps

You can compute the composition of maps :
`ffcompomap`$(f, g) = f \circ g$.

```
? Mce = ffcompomap(Mde,Mcd);
? ffmap(Mce, c) == ffmap(Mde, ffmap(Mcd, c))
% = 1
? ffcompomap(F,Mcd) == Mcd
% = 1
? ffcompomap(F,F) == fffrobenius(d,16)
% = 1
```

Preimages

You can compute the partial inverse of a map with `ffinvmap`.

```
? Mdc = ffinvmap(Mcd);
? ffmap(Mdc, ffmap(Mcd, c^3+c+1))
% = c^3 + c + 1
? ffmap(Mdc, d)
% = [] \\sentinel value: no preimage
? Mec = ffcompomap(Mdc, ffinvmap(Mde));
? ffmap(Mec, ffmap(Mce, c))
? ffinvmap(fffrobenius(c,3)) == fffrobenius(c,5)
% = 1
```

Relative extensions

Given a map between finite fields, we could compute the relative trace, norm, characteristic polynomial, a relative expression for elements of the larger field, etc.

Which names / interface would you like to see ?