# Metacyclics

## A GAP package for constructing and computing invariants of finite metacyclic groups

Version 1.0.0

January 23, 2023

**Àngel García-Blàzquez**
**Ángel del Río Mateos**

**Àngel García-Blàzquez** Email: angel.garcia11@um.es

**Ángel del Río Mateos** Email: adelrio@um.es
Homepage: https://www.um.es/adelrio/

# Copyright



The package can be downloaded from https://www.um.es/adelrio/MetaCyc.php

# Contents

# Chapter 1

# Metacyclics

## 1.1 Metacyclic Groups and Metacyclic Parameters

A group $G$ is said to be *metacyclic* if it has a normal cyclic subgroup $A$ with $G/A$ cyclic. In that case $A$ is called a *kernel* of $G$. In case $A$ has order $m$ and $G/A$ has order $n$ then $G$ has a presentation as follows:

$$G = \left\langle a, b \mid a^m = 1, b^n = a^s, a^b = a^t \right\rangle, \tag{1.1}$$

where $m, n, s$ and $t$ are integers satisfying

$$m > 0, \quad n > 0, \quad m \mid t^n - 1, \quad m \mid s(t-1). \tag{1.2}$$

In that case we say that $[m, n, s, t]$ is a list of *metacyclic parameters*. If moreover, the group $G$ satisfies (1.1) then we say that $[m, n, s, t]$ is a list of metacyclic parameters of $G$, in short, *MCParameters* of $G$. These parameters determine the group, but the group can have different lists of MCParameters.

### 1.1.1 IsMetacyclic

▷ `IsMetacyclic(G)` (operation)

This function takes a finite group $G$ as input and returns `True` if the group is metacyclic and `False` otherwise. If the given input is not a finite group then the function fails.

### 1.1.2 MCParameters

▷ `MCParameters(G)` (operation)

This function takes a group $G$ and if $G$ is finite and metacyclic then it returns some MCParameters of $G$. If $G$ is not a finite metacyclic group then the function fails.

### 1.1.3 AreMCParameters

▷ `AreMCParameters(x)` (operation)

This function takes a 4-tuple of integers and returns `True` if the input is a list MCParameters and `False` otherwise. If the argument is not a list of four integers then the function fails.

### 1.1.4 MetacyclicGroupPC

▷ `MetacyclicGroupPC(x)` (operation)

This function takes a 4-tuple $[m,n,s,t]$ and if it is a list of MCParameters then it returns a Pc group with a presentation as in (1.1). The group is built using a Power-conjugate presentation. If the input is not a list of MCParameters then the function fails.

### 1.1.5 MinimalKernel

▷ `MinimalKernel(x)` (operation)

This function takes a group $G$ and returns a normal cyclic subgroup $A$ of the smallest possible order such that $G/A$ is also cyclic. If there is no such group, the function fails.

```
gap> G:=SmallGroup(380,7);;
gap> IsMetacyclic(G);
 false
gap> G:=SmallGroup(380,2);
<pc group of size 380 with 4 generators>
gap>  IsMetacyclic(G);
true
gap>  Size(MinimalKernel(G));
19
gap> MCParameters(G);
[ 19, 20, 19, 18 ]
gap> H:=MetacyclicGroupPC([19,20,19,18 ]);;
gap> IdSmallGroup(H);
[ 380, 2 ]
gap> AreMCParameters([19,20,19,18]);
true
gap> AreMCParameters([2, 2, 2, 2]);
false
gap> m162:=Filtered([1..55],i->IsMetacyclic(SmallGroup(162,i)));
[ 1, 2, 3, 6, 7, 8, 9, 23, 25, 26, 27 ]
gap> List(m162,i->MCParameters(SmallGroup(162,i)));
[ [ 81, 2, 81, 80 ], [ 1, 162, 1, 1 ], [ 9, 18, 9, 8 ], [ 9, 18, 9, 2 ],
  [ 27, 6, 27, 26 ], [ 3, 54, 3, 2 ], [ 27, 6, 27, 17 ], [ 9, 18, 9, 1 ],
  [ 9, 18, 9, 4 ], [ 3, 54, 3, 1 ], [ 9, 18, 3, 4 ] ]
```

As the MCParameters of a group are not unique we may encountered isomorphic groups with different MCParameters.

```
gap> G:=MetacyclicGroupPC([5,76,5,2]);;
gap> H:=MetacyclicGroupPC([5,76,5,3]);;
gap> MCParameters(G);
[ 5, 76, 5, 2 ]
gap> MCParameters(H);
[ 5, 76, 5, 3 ]
gap> IdSmallGroup(G);
[ 380, 5 ]
gap> IdSmallGroup(H);
[ 380, 5 ]
```

## 1.2 Invariants of Metacyclic Groups

Let $G$ be a metacyclic group. Then the list of *metacyclic invariants* of $G$ is a unique distinguished list of MCParameters of $G$, and $\mathrm{MCINV}(G)$ is a tuple $\mathrm{MCINV}(G) = [m,n,s,\Delta]$ where $m,n$ and $s$ are the first three entries of the list of metacyclic invariants of $G$ and $\Delta$ is a cyclic subgroup of the group of units of a certain divisor $m'$ of $m$. See [GBdR23] for the definitions of the operator MCINV and the list of metacyclic invariants of a finite group. Two metacyclic groups $G$ and $H$ are isomorphic if and only if the have the same list of metacyclic invariants, if and only if $\mathrm{MCINV}(G) = \mathrm{MCINV}(H)$.

### 1.2.1 MetacyclicInvariants

▷ `MetacyclicInvariants(G)` (operation)

This function takes a group $G$ and if it is finite and metacyclic then it returns the list of metacyclic invariants of $G$. The input can be given as the group itself or some MCParameters. If the input is neither or the group is not finite and metacyclic then the function fails.

### 1.2.2 MCINV

▷ `MCINV(G)` (operation)

This function takes a group $G$ and if it is finite and metacyclic then it returns $\mathrm{MCINV}(G)$. The input can be given as the group itself or some MCParameters. In the input is neither or the group is not finite and metacyclic then the function fails.

### 1.2.3 MCINVData

▷ `MCINVData(G)` (operation)

This function takes a group $G$ and if it is finite and metacyclic then it returns a 5-tuple of integers $[m,n,s,m',t]$ such that $\mathrm{MCINV}(G) = [m,n,s,\langle t \rangle_{m'}]$ where $\langle t \rangle_{m'}$ denotes the multiplicative group of $\mathbb{Z}/m'\mathbb{Z}$ of the class containing $t$. The input can be given as the group itself or some MCParameters. In the input is neither or the group is not finite and metacyclic then the function fails.

If $G$ and $H$ are two finite metacyclic groups then they are isomorphic if and only if the first four entries of MCINVData(G) and MCINVData(H) coincide and $\langle t_G \rangle_{m'} = \langle t_H \rangle_{m'}$ where $m'$ is the common fourth entry and $t_G$ and $t_H$ are the last entries.

```
gap> G:=MetacyclicGroupPC([5,76,5,2]);;
gap> H:=MetacyclicGroupPC([5,76,5,3]);;
gap> MetacyclicInvariants(G);
[ 5, 76, 5, 2 ]
gap> MetacyclicInvariants(H);
[ 5, 76, 5, 2 ]
gap> G:=SmallGroup(162,9);;
gap> MCParameters(G);
[ 27, 6, 27, 17 ]
gap> MetacyclicInvariants(G);
[ 27, 6, 27, 8 ]
gap> MetacyclicInvariants([27,6,27,17]);
[ 27, 6, 27, 8 ]
gap> x:=MCINV(G);
[ 27, 6, 27, <group with 1 generator> ]
gap> MCINV([27,6,27,17]);
[ 27, 6, 27, <group with 1 generator> ]
gap> MCINV([27,6,27,17])[4]=x[4];
true
gap> y:=MCINVData(G);
[ 27, 6, 27, 27, 17 ]
gap> x[4]=Group(ZmodnZObj(y[5],y[4]));
true
gap> MCINVData([27,6,27,17]);
[ 27, 6, 27, 27, 8 ]
gap> G:=SmallGroup(384,533);;
gap> MetacyclicInvariants(G);MCINV(G);MCINVData(G);
[ 8, 48, 4, 5 ]
[ 8, 48, 4, <group of size 1 with 1 generator> ]
[ 8, 48, 4, 4, 1 ]
```

### 1.2.4 AreIsomorphicMetacyclicGroups

▷ AreIsomorphicMetacyclicGroups(*G*, *H*) (operation)
▷ AreIsomorphicMetacyclicGroups(*G*, *[m,n,s,r]*) (operation)
▷ AreIsomorphicMetacyclicGroups(*[m,n,s,r]*, *G* ) (operation)
▷ AreIsomorphicMetacyclicGroups(*[m,n,s,r]*, *[m',n',s',r']*) (operation)

This function returns true if the two inputs represent isomorphic finite metacyclic groups and false if one of them represent a finite metacyclic group and the other one is a group non-isomorphic to the first one. The metacyclic groups may be given either by the group itself or by their MCParameters. If any of the inputs do not represent a finite metacyclic group then the function fails.

```
gap> G:=MetacyclicGroupPC([5,76,5,2]);;
gap> H:=MetacyclicGroupPC([5,76,5,3]);;
gap> AreIsomorphicMetacyclicGroups(G,H);
true
gap> G:=SmallGroup(162,9);;
gap> AreIsomorphicMetacyclicGroups(G,[27,6,27,8]);
true
gap> AreIsomorphicMetacyclicGroups([27,6,27,17],[27,6,27,8]);
true
gap> AreIsomorphicMetacyclicGroups([27,6,27,17],[27,6,27,26]);
false
gap> AreIsomorphicMetacyclicGroups([8,2,4,5],[4,4,2,3]);
true
```

## 1.3 The Classification of Finite Metacyclic Groups

### 1.3.1 The Classification Algorithm

▷ `MetacyclicGroupsByOrder(N)` (operation)

This function takes a positive integer $N$ and returns a list formed by the metacyclic invariants of all the metacyclic groups of order $N$. It uses Algorithm 4 of [GBdR23]).

```
gap> MetacyclicGroupsByOrder(21);
[ [ 1, 21, 1, 0 ], [ 7, 3, 7, 2 ] ]
gap> x:=MetacyclicGroupsByOrder(256);
[ [ 1, 256, 1, 0 ], [ 2, 128, 2, 1 ], [ 4, 64, 2, 3 ], [ 4, 64, 4, 1 ],
  [ 4, 64, 4, 3 ], [ 8, 32, 4, 3 ], [ 8, 32, 4, 5 ], [ 8, 32, 8, 1 ],
  [ 8, 32, 8, 3 ], [ 8, 32, 8, 5 ], [ 8, 32, 8, 7 ], [ 16, 16, 4, 5 ],
  [ 16, 16, 8, 3 ], [ 16, 16, 8, 5 ], [ 16, 16, 8, 7 ], [ 16, 16, 8, 9 ],
  [ 16, 16, 16, 1 ], [ 16, 16, 16, 3 ], [ 16, 16, 16, 5 ], [ 16, 16, 16, 7 ],
  [ 16, 16, 16, 9 ], [ 16, 16, 16, 15 ], [ 32, 8, 8, 5 ], [ 32, 8, 8, 9 ],
  [ 32, 8, 16, 3 ], [ 32, 8, 16, 7 ], [ 32, 8, 16, 15 ], [ 32, 8, 32, 7 ],
  [ 32, 8, 32, 15 ], [ 32, 8, 32, 31 ], [ 64, 4, 32, 15 ], [ 64, 4, 32, 31 ],
  [ 64, 4, 64, 31 ], [ 64, 4, 64, 63 ], [ 128, 2, 64, 63 ],
  [ 128, 2, 64, 127 ], [ 128, 2, 128, 127 ] ]
gap> List(x,MCINVData);
[ [ 1, 256, 1, 1, 0 ], [ 2, 128, 2, 2, 1 ], [ 4, 64, 2, 4, 3 ],
  [ 4, 64, 4, 4, 1 ], [ 4, 64, 4, 4, 3 ], [ 8, 32, 4, 4, 3 ],
  [ 8, 32, 4, 4, 1 ], [ 8, 32, 8, 8, 1 ], [ 8, 32, 8, 4, 3 ],
  [ 8, 32, 8, 4, 1 ], [ 8, 32, 8, 8, 7 ], [ 16, 16, 4, 4, 1 ],
  [ 16, 16, 8, 4, 3 ], [ 16, 16, 8, 4, 1 ], [ 16, 16, 8, 8, 7 ],
  [ 16, 16, 8, 8, 1 ], [ 16, 16, 16, 16, 1 ], [ 16, 16, 16, 4, 3 ],
  [ 16, 16, 16, 4, 1 ], [ 16, 16, 16, 8, 7 ], [ 16, 16, 16, 8, 1 ],
  [ 16, 16, 16, 16, 15 ], [ 32, 8, 8, 4, 1 ], [ 32, 8, 8, 8, 1 ],
  [ 32, 8, 16, 4, 3 ], [ 32, 8, 16, 8, 7 ], [ 32, 8, 16, 16, 15 ],
  [ 32, 8, 32, 8, 7 ], [ 32, 8, 32, 16, 15 ], [ 32, 8, 32, 32, 31 ],
  [ 64, 4, 32, 16, 15 ], [ 64, 4, 32, 32, 31 ], [ 64, 4, 64, 32, 31 ],
  [ 64, 4, 64, 64, 63 ], [ 128, 2, 64, 64, 63 ], [ 128, 2, 64, 128, 127 ],
  [ 128, 2, 128, 128, 127 ] ]
```

```
gap> Size(MetacyclicGroupsByOrder(40000));
377
gap> Size(MetacyclicGroupsByOrder(16*9*25*7));
712
```

# References


[GBdR23] À. García-Blázquez and Á. del Río. A classification of metacyclic groups by group invariants. *http://arxiv.org/abs/2301.08683*, 2023. 6, 8

# Index