

一般函数域上的 QSAlgorithmFF
Maple 风格帮助文档（递归结构版）

release 3.6.0 (Horrocks–Suslin, quotient-safe)
主程序、OneStep、局部循环及其数据流

2026 年 7 月

摘要

本文按照 release 3.6.0 的重构源码说明主算法。公开入口先验证基理想是否定义函数域，再计算么模性与右逆证书；只有两项都通过后才进入递归的 CompleteRow。每个递归层依次处理精确基情形、一个独立启发式子程序、一个由正规化-精确降次-局部循环-Patch 组成的 OneStepReduction，以及少一个变量的递归调用。文档保持 Maple 帮助页的 Calling Sequence、Parameters、Description、Program Hierarchy、Return Value、Examples、Compatibility 和 See Also 结构。

Examples 的使用方式

每一个 Examples 代码框均给出完整变量、 P 和输入行，并附有注释。例子中不人为引入右逆符号；可直接复制到新的 Maple worksheet 中运行。安装目录不同，只需修改 `currentdir(...)` 一行。

$v(0)$ 的含义

`OneStepReduction` 中的 $v(0)$ 表示把有序变量列表 `xvars` 的最后一个多项式变量赋值为 0，即 `subs(xvars[-1] = 0, v)`；不是把向量的最后一个分量直接改成零。

目录

第一章 安装、输入与主算法结构	3
1.1 安装与载入	3
1.2 数学输入与共同参数	3
1.3 四阶段主算法	3
1.4 统一调用树	4
第二章 顶层入口、递归根及主要子程序	5
2.1 QSFULLFF:-CompleteRow	5
2.2 QSFULLFF:-CompleteRowData	6
2.2.1 QSFULLFF:-ValidateFunctionField	7
2.2.2 QSFULLFF:-UnimodularRowData	8
2.3 递归 CompleteRow 核心	9
2.3.1 QSFULLFF:-CompleteRowKx	9
2.3.2 QSFULLFF:-HeuristicCompletionData	10
2.3.3 OneStepReduction: 递归步骤 (2)	11
2.3.3.1 正规化: QSFULLFF:-NormalisationStep	12
2.3.3.2 精确降次: QSReduceFF:-ReduceDegreeData	12
2.3.3.3 QSFULLFF:-LocalLoopData	12
2.3.3.3.1 QSFULLFF:-MaximalIdealData	14
2.3.3.3.2 QSFULLFF:-OneLocalSolData	14
2.3.3.4 QSFULLFF:-PatchData	15
2.3.4 递归步骤 (3)	16
2.3.5 QSFULLFF:-VerifyCompletionFullData	16
2.4 QSFULLFF:-CompleteRowResult	17
第三章 运行控制、缓存与审查要点	18
3.1 日志与总限时	18
3.2 唯一启发式开关	18
3.3 缓存控制	18
3.4 代码审查时应检查的数据不变量	19
附录 A 源文件、模块职责与帮助文档对应关系	20
A.1 为什么帮助文档中的命令名与源文件名不完全相同	20
A.2 载入文件与依赖顺序	20
A.3 核心 Maple 文件职责总表	20
A.4 帮助文档命令到实际实现文件的速查	22
A.5 按主算法阶段阅读源码	22
快速检查清单	23

1 安装、输入与主算法结构

1.1 安装与载入

将发行包解压到普通目录，例如：

C:/QSAAlgorithmFF_release_v3_6_0/

在 Maple 中运行：

```
restart;
# 清空当前 Maple 会话中的变量、模块和缓存状态。

currentdir("C:/QSAAlgorithmFF_release_v3_6_0"):
# 切换到发行包根目录；Windows 路径建议使用正斜杠。

read "load_all.mpl":
# 按依赖顺序载入全部核心模块和 QSFULLFF 公共门面。

QSFULLFF:-Version();
# 应返回 release 3.6.0 的版本字符串。
```

1.2 数学输入与共同参数

程序处理

$$K = \text{Frac}(\mathbb{Q}[t_1, \dots, t_m]/P), \quad R = K[x_1, \dots, x_n],$$

其中 P 必须是素理想。输入幺模行 $f \in R^{1 \times p}$ ，输出 $U \in \text{GL}_p(R)$ ，满足 $fU = e_1$ 。

Parameter	Type / Allowed values	Meaning
row0	非空 list	输入行；条目必须属于 $K[xvars]$ ，分母不能含多项式变量。
xvars0	有序变量 list	多项式变量；顺序决定递归变量次序。
tvars0	参数 list	系数域参数，必须与 $xvars0$ 不交。
Pgens0	多项式 list/set	$P \subseteq \mathbb{Q}[tvars]$ 的生成元；非零时必须 proper 且 prime。
timeLimit	正整数或 infinity	完整公开调用的总限时，默认无界。
printLog	Boolean	打印阶段、递归深度和实时时间。
useHeuristics	Boolean	唯一的可选快路径开关。
localMethod	"horrocks", "residue", "search", "auto"	单个局部问题的求解方法。
printBranches	Boolean	是否逐分支打印 d_i 与 U_i ；只用于 LocalLoop/OneStep。

1.3 四阶段主算法

公开数据入口首先执行两项守卫，然后才进入递归核心：

1. **ValidateFunctionField**：检查变量、基关系、properness 和 primality。
2. **UnimodularRowData**：确认输入行幺模，并计算一个精确右逆；该证书只在根节点计算一次。
3. **递归 CompleteRow**：
 - (1) 精确基情形：长度一、长度二、零变量，以及一变量 CompleteRowKx；
 - (2) HeuristicCompletionData：一个集中、可关闭的启发式子程序；
 - (3) 启发式失败后调用 OneStepReduction 核心：正规化、精确降次、局部循环、Patch；
 - (4) 对 OneStep 的目标行递归调用 CompleteRow。

4. **Final verification:** 根节点再次独立检查行等式、行列式单位性和矩阵条目合法性。

保留的优化

重构只改变控制结构，不删除数学优化：长度二专门算法、按次数和项数选择较好分量、变量顺序优化、Weierstrass fallback、右逆传播、精确 `ReduceDegreeData`、Gröbner/正规形缓存、Horrocks–Suslin、residue/resultant、自动 patch 分母和 quotient-safety 屏障均保留。

1.4 统一调用树

完整调用树

```
QSFULLFF:-CompleteRow
`-- QSFULLFF:-CompleteRowData                                [公开数据根]
  |-- 1. QSFULLFF:-ValidateFunctionField
  |-- 2. QSFULLFF:-UnimodularRowData
  |-- 3. CompleteRowRecursiveData                            [内部递归核心]
    |-- (0) QSFULLFF:-CompleteRowKx
    |-- (1) QSFULLFF:-HeuristicCompletionData
    |-- (2) OneStepReductionCoreData
      |-- QSFULLFF:-NormalisationStep
      |-- QSRREDUCEFF:-ReduceDegreeData
      |-- QSFULLFF:-LocalLoopData
        |-- QSFULLFF:-MaximalIdealData                    [公开诊断接口]
          |-- internal MaximalIdealContainingOverK
          |-- QSFULLFF:-OneLocalSolData
          |-- horrocks / residue / search / auto
          |-- QSFULLFF:-PatchData
        |-- (3) CompleteRowRecursiveData(target,n-1 vars)
    |-- 4. QSFULLFF:-VerifyCompletionFullData/FastData
```

极大理想命令的归属

目录中使用 `QSFULLFF:-MaximalIdealData` 表示用户可调用的数据类型接口；它返回相对生成元、环境生成元、相对维数和理想对象。LocalLoop 内部为了效率只需生成元，因此字面调用的是内部 `QSLocalLoopFF:-MaximalIdealContainingOverK`。两者共享同一底层构造家族，但返回形式不同。

2 顶层入口、递归根及主要子程序

2.1 QSFULLFF:-CompleteRow

CompleteRow

构造并返回经过验证的 Quillen–Suslin 完成矩阵。

[Calling Sequence](#) [Parameters](#) [Description](#) [Program Hierarchy](#) [Return Value](#) [Examples](#) [Compatibility](#) [See Also](#)

Calling Sequence

```
QSFULLFF:-CompleteRow(  
    row0,xvars0,tvars0,Pgens0,  
    'timeLimit'=infinity,  
    'printLog'=false,  
    'useHeuristics'=true,  
    'localMethod'="horrocks"  
)
```

Description

最外层矩阵包装器；内部调用 CompleteRowData，只返回 Data["U"]。未知 keyword 会被拒绝；超时或验证失败时不返回候选矩阵。

Program Hierarchy

Program Hierarchy

公开入口：QSFULLFF:-CompleteRow。直接下级：QSFULLFF:-CompleteRowData。真正的递归核心是 QSAAlgorithmFF 内部的 CompleteRowRecursiveData。

Return Value

$p \times p$ Maple Matrix U ，满足 $fU = e_1$ 。

Examples

```
restart:  
# 清空当前 Maple 会话。  
currentdir("C:/QSAAlgorithmFF_release_v3_6_0"):  
# 切换到程序目录。  
read "load_all.mpl":  
# 载入所有核心模块。  
  
unassign('x','y','a','b','xvars','tvars','Pgens','row','U','V'):  
# 清除本例使用的符号。  
xvars := [x,y]:  
# 多项式变量；y 是当前最后变量。  
tvars := [a,b]:  
# 系数域参数。  
Pgens := [a^2+b^2]:  
# K=Frac(Q[a,b]/(a^2+b^2)).  
row := [b*y+1,x+a,x*y+b]:  
# 待完成的 K[x,y] 中长度三行。  
  
U := QSFULLFF:-CompleteRow(row,xvars,tvars,Pgens,  
    'timeLimit'=600,  
    # 总限时 600 秒。  
    'printLog'=true,  
    # 打印阶段及耗时。  
    'useHeuristics'=false,  
    # 跳过集中启发式阶段，但保留全部精确算法。
```

```
'localMethod'="horrocks"
# 局部问题使用 Horrocks--Suslin。
):

V := QSFul1FF:-VerifyCompletionFullData(row,U,xvars,tvars,Pgens):
# 独立重算最终验证。
V["success"];
# 应返回 true。
V["row_times_U_mod_P"];
# 应返回 [1,0,0]。
```

Compatibility

本文档对应 QSAgorithmFF release 3.6.0，目标运行环境为 Maple 2025。签名、默认值、返回键和调用关系按该发布版源码说明。

See Also

CompleteRowData, HeuristicCompletionData, OneStepReduction。

2.2 QSFul1FF:-CompleteRowData

CompleteRowData

执行四阶段主算法并返回完整递归数据树。

Calling Sequence Parameters Description Program Hierarchy Return Value Examples Compatibility See Also

Calling Sequence

与 CompleteRow 相同。

Description

先验证 P ，再验证幺模性；两项通过后调用递归核心。右逆在根节点计算一次，之后通过列变换和变量专门化传播。递归完成后进行根级最终验证。

Program Hierarchy

Program Hierarchy

这是公开数据根。直接阶段依次为 ValidateFunctionField、UnimodularRowData、内部 CompleteRowRecursiveData 和最终验证。递归核心不重新计算基素性和全局 Gröbner 右逆。

Return Value

Key / Value	Meaning
Data["U"]	最终完成矩阵。
Data["input_validation"]	根级函数域和幺模性守卫数据。
Data["unimodularity_data"]	完整右逆/Bézout 证书。
Data["heuristic_data"]	当前递归层的启发式结果；失败也保留原因。
Data["one_step"]	正规化、降次、local loop 和 patch 的组合数据。
Data["normalisation"], ["degree_reduction"]	兼容旧脚本的阶段字段。
Data["local_loop"], ["patch"]	局部覆盖与 patch 数据。
Data["target_row"], ["recursive_data"]	低一维目标和递归子树。
Data["construction_verification"]	递归核心返回时的验证。
Data["final_verification"]	根级最终独立验证。
Data["verification"]	当前公开结果采用的最终验证表。
Data["runtime_options"], ["elapsed_real_seconds"]	运行参数和实际总时间，由 QSFul1FF 包装层加入。

Examples


```

restart:
# 清空会话。
currentdir("C:/QSAAlgorithmFF_release_v3_6_0"):
# 切换到程序目录。
read "load_all.mpl":
# 载入程序。

unassign('x','y','a','b','xvars','tvars','Pgens','row','Data'):
xvars := [x,y]:
tvars := [a,b]:
Pgens := [a^2+b^2]:
row := [b*y+1,x+a,x*y+b]:
# 定义一般函数域、变量和输入行。

Data := QSFULLFF:-CompleteRowData(row,xvars,tvars,Pgens,
    'timeLimit'=600,
    # 整次调用最多 600 秒。
    'printLog'=true,
    # 打印每个主阶段和递归层耗时。
    'useHeuristics'=false,
    # 禁用可选提前完成和搜索快路径。
    'localMethod'="horrocks"
):

Data["field_is_valid"];
# 应为 true。
Data["unimodularity_data"]["is_unimodular"];
# 应为 true。
Data["verification"]["success"];
# 最终验证应为 true。
Data["elapsed_real_seconds"];
# 实际总运行时间（秒）。
Data["runtime_options"];
# 本次调用真正采用的公开参数。

```

Compatibility

本文档对应 QSAAlgorithmFF release 3.6.0，目标运行环境为 Maple 2025。签名、默认值、返回键和调用关系按该发布版源码说明。

See Also

CompleteRow, HeuristicCompletionData, OneStepReductionData。

2.2.1 QSFULLFF:-ValidateFunctionField

ValidateFunctionField

第一阶段：验证 P 是否定义一般函数域。

[Calling Sequence](#) [Parameters](#) [Description](#) [Program Hierarchy](#) [Return Value](#) [Examples](#) [Compatibility](#) [See Also](#)

Calling Sequence

```
QSFULLFF:-ValidateFunctionField(tvars0,Pgens0)
```

Description

检查参数名、基关系变量、properness 和 primality。成功结果会缓存。

Program Hierarchy

Program Hierarchy

主算法第 1 阶段；只有成功后才进入幺模性判定。实际实现为 `QSAgorithmFF:-ValidateFunctionField`。

Return Value

成功返回 `true`；非法输入抛出异常。

Examples

```
restart:
currentdir("C:/QSAgorithmFF_release_v3_6_0"):
read "load_all.mpl":
# 载入程序。

unassign('a','b','tvars','Pgens'):
tvars := [a,b]:
# 系数参数。
Pgens := [a^2+b^2]:
# 基理想 P。

QSFullFF:-ValidateFunctionField(tvars,Pgens);
# P proper 且 prime 时返回 true。
```

Compatibility

本文档对应 QSAgorithmFF release 3.6.0，目标运行环境为 Maple 2025。签名、默认值、返回键和调用关系按该发布版源码说明。

See Also

`UnimodularRowData`, `CompleteRowData`。

2.2.2 QSFullFF:-UnimodularRowData

UnimodularRowData

第二阶段：验证幺模性并返回右逆/Bézout 证书。

[Calling Sequence](#) [Parameters](#) [Description](#) [Program Hierarchy](#) [Return Value](#) [Examples](#) [Compatibility](#) [See Also](#)

Calling Sequence

```
QSFullFF:-UnimodularRowData(
    row0,xvars0,tvars0,Pgens0,
    checkPrime := true
)
```

Program Hierarchy

Program Hierarchy

主算法第 2 阶段。公开门面转发到 `QSInputFF:-UnimodularRowData`。主入口已经完成第 1 阶段，因此内部传入 `checkPrime=false`，避免重复素性计算。

Return Value

主要键为 `is_unimodular`、`right_inverse`、`bezout_coefficients`、`certificate_verified` 和 `method`。

Examples

```
restart:
currentdir("C:/QSAgorithmFF_release_v3_6_0"):
read "load_all.mpl":

unassign('x','y','a','b','xvars','tvars','Pgens','row','UniData'):
xvars := [x,y]:
```

```

tvars := [a,b]:
Pgens := [a^2+b^2]:
row := [b*y+1,x+a,x*y+b]:
# 定义需要判定的行。

UniData := QSFULLFF:-UnimodularRowData(row,xvars,tvars,Pgens):
# 计算幺模性与一个精确右逆。
UniData["is_unimodular"];
# 幺模时返回 true。
UniData["right_inverse"];
# 返回程序找到的右逆列。
UniData["certificate_verified"];
# 证书经独立检查后应为 true。

```

Compatibility

本文档对应 QSAAlgorithmFF release 3.6.0，目标运行环境为 Maple 2025。签名、默认值、返回键和调用关系按该发布版源码说明。

See Also

[ValidateFunctionField](#), [HeuristicCompletionData](#)。

2.3 递归 CompleteRow 核心

这一节描述内部递归树。它接收已经验证的行和右逆，不再重复计算素性或全局 Gröbner 幺模证书。

递归核心始终先处理：长度一单位行、长度二 Bézout 块、零变量系数域和一变量 CompleteRowKx。这些步骤不受 useHeuristics 控制。

2.3.1 QSFULLFF:-CompleteRowKx

CompleteRowKx

一变量 $K[x]$ 基情形。

[Calling Sequence](#) [Parameters](#) [Description](#) [Program Hierarchy](#) [Return Value](#) [Examples](#) [Compatibility](#) [See Also](#)

Calling Sequence

```
QSFULLFF:-CompleteRowKx(row0,xvar0,tvars0,Pgens0)
```

Description

使用商域感知的 Euclidean/Bézout 块完成一元幺模行。

Examples

```

restart:
currentdir("C:/QSAAlgorithmFF_release_v3_6_0"):
read "load_all.mpl":

unassign('x','a','b','row1','U1','V1'):
tvars := [a,b]:
Pgens := [a^2+b^2]:
row1 := [1+(x-a)*(x+a)*(x^2+b),x+a,(x+a)*(x^2+b)]:
# 一元非平凡幺模行；程序不需要用户提供 Bézout 证书。

U1 := QSFULLFF:-CompleteRowKx(row1,x,tvars,Pgens):
V1 := QSFULLFF:-VerifyCompletionFullData(row1,U1,[x],tvars,Pgens):
V1["success"];
# 应为 true。

```

Compatibility

本文档对应 QSAAlgorithmFF release 3.6.0，目标运行环境为 Maple 2025。签名、默认值、返回键和调用关系按该发布版源码说明。

See Also

CompleteRowData。

2.3.2 QSFULLFF:-HeuristicCompletionData

HeuristicCompletionData

递归步骤 (1)：集中执行可选提前完成。

[Calling Sequence](#) [Parameters](#) [Description](#) [Program Hierarchy](#) [Return Value](#) [Examples](#) [Compatibility](#) [See Also](#)

Calling Sequence

```
QSFULLFF:-HeuristicCompletionData(
    row0,xvars0,tvars0,Pgens0,
    'useHeuristics'=true
)
```

Description

这是重构后唯一调用 `QSEasyFF` 的主算法子程序。它先取得合法右逆，再集中尝试单位分量、右逆零/单位分量、二元 Gröbner 证书、真子行证书和右逆二元证书。未成功只返回 `success=false`，不抛出“算法失败”。

Program Hierarchy

Program Hierarchy

直接上级：内部 `CompleteRowRecursiveData`。直接下级：`QSEasyFF:-TryEasyCompletion`。`useHeuristics=false` 时整个阶段跳过；精确基情形和 `OneStep` 仍继续。

Return Value

Key / Value	Meaning
<code>H["success"]</code>	是否找到提前完成矩阵。
<code>H["enabled"]</code>	本次是否允许启发式。
<code>H["reason"]</code>	未成功时的原因。
<code>H["method"]</code> , <code>H["U"]</code>	成功时的具体方法和完成矩阵。
<code>H["unimodularity_data"]</code>	为启发式阶段准备的右逆证书数据。

Examples

```
restart:
currentdir("C:/QSAAlgorithmFF_release_v3_6_0"):
read "load_all.mpl":

unassign('x','y','a','b','xvars','tvars','Pgens','row','H'):
xvars := [x,y]:
tvars := [a,b]:
Pgens := [a^2+b^2]:
row := [b*y+1,x+a,x*y+b]:

H := QSFULLFF:-HeuristicCompletionData(row,xvars,tvars,Pgens,
    'useHeuristics'=true
):
# 只运行集中启发式阶段，不执行 local loop 或 patch。
H["success"];
# true 表示找到提前完成；false 表示主程序应继续 OneStep。
if H["success"] then H["U"] else H["reason"] end if;
# 查看完成矩阵或未成功原因。
```

Compatibility

本文档对应 QSAAlgorithmFF release 3.6.0，目标运行环境为 Maple 2025。签名、默认值、返回键和调用关系按该发布版源码说明。

See Also

CompleteRowData, DisableHeuristics。

2.3.3 OneStepReduction: 递归步骤 (2)

OneStepReduction

共享的一步核心固定执行：正规化 → 精确降次 → Local Loop → Patch。

Calling Sequence

公开精确命令：

```
QSFULLFF:-OneStepReduction(
    row0,xvars0,tvars0,Pgens0,
    'timeLimit'=infinity,
    'printLog'=false,
    'useHeuristics'=true,
    'localMethod'="horrocks",
    'printBranches'=false
)

QSFULLFF:-OneStepReductionData(...same options...)
```

Description

递归主算法调用内部 `OneStepReductionCoreData`，该核心只做正规化、降次、local loop 与 patch，然后把目标行交回递归调用。公开 `OneStepReduction` 为了满足历史契约 $vU = v(0)$ ，在同一核心之后完成其目标行并作精确 transport；它没有第二套局部算法。

Return Value

矩阵型入口返回 U ，满足 $vU = v(0)$ 。数据型入口还返回 `one_step_core`、`normalisation`、`degree_reduction`、`local_loop`、`patch`、`recursive_data` 和独立验证。

Examples

```
restart:
currentdir("C:/QSAAlgorithmFF_release_v3_6_0"):
read "load_all.mpl":

unassign('x','y','a','b','xvars','tvars','Pgens','row','Step'):
xvars := [x,y]:
tvars := [a,b]:
Pgens := [a^2+b^2]:
row := [b*y+1,x+a,x*y+b]:

Step := QSFULLFF:-OneStepReductionData(row,xvars,tvars,Pgens,
    'timeLimit'=600,
    # 限制整次一步命令。
    'printLog'=true,
    # 打印正规化、降次、local loop、patch 和 transport 时间。
    'useHeuristics'=false,
    # local loop 中不使用整数 search 或小整数点快路径。
    'localMethod'="horrocks",
    'printBranches'=false
    # 改为 true 可逐个打印 d_i 和 U_i。
):

Step["target_row"];
# 原始行把最后变量 y 置零后的目标，应为 [1,x+a,b]。
Step["verification"]["row_times_U_mod_P"];
# 应与 target_row 相同。
```

```
Step["verification"]["success"];
# 应为 true。
```

2.3.3.1 正规化: QSFULLFF:-NormalisationStep

Description

先寻找首项系数属于 K^* 的较优分量/变量；若仅改变变量顺序不能成功，则选择次数和项数较小的非零分量做 Weierstrass/Noether 代换，并记录正反代换和列缩放矩阵。

Examples

```
restart;
currentdir("C:/QSAlgorithmFF_release_v3_6_0");
read "load_all.mpl":

unassign('x','y','a','b','xvars','tvars','Pgens','rowN','N'):
xvars := [x,y]:
tvars := [a,b]:
Pgens := [a^2+b^2]:
rowN := [
    x*y+a*x+b*y+1,
    x^2*y+b*x+a,
    x*y^2+a*y+b
]:
# 每个分量关于 x 和 y 的首项系数都依赖另一个多项式变量,
# 因而没有现成的  $K^*$ -首一分量; auto 将进入 Weierstrass fallback.

N := QSFULLFF:-NormalisationStep(rowN,xvars,tvars,Pgens,"auto"):
N["normalisation_mode"];
# 预期为 "weierstrass".
N["normalized_index"];
# 查看按次数/项数选中的分量。
N["forward_substitution"];
N["backward_substitution"];
# 查看可逆坐标代换。
```

2.3.3.2 精确降次: QSReduceFF:-ReduceDegreeData

正规化产生关于工作变量的首一分量后，程序用它对其他分量做带余除法。该步骤始终开启，不属于启发式；返回初等列矩阵、商、降次前后次数和行列式证书 1。

2.3.3.3 QSFULLFF:-LocalLoopData

LocalLoopData

构造有限局部覆盖，记录每个 M_i, d_i, U_i 。

Calling Sequence Parameters Description Program Hierarchy Return Value Examples Compatibility See Also

Calling Sequence

```
QSFULLFF:-LocalLoopData(
    row0,xvars0,tvars0,Pgens0,
    'localMethod'="horrocks",
    'useHeuristics'=true,
    'printBranches'=false
)
```

Description

从空分母理想开始；每轮找一个包含当前分母理想的极大理想 M_i ，在局部环中构造并验证 U_i ，提取 $d_i \notin M_i$ ，直到 $(d_1, \dots, d_N) = E$ 。

Program Hierarchy

Program Hierarchy

直接上级: OneStepReduction 核心。内部每轮依次执行极大理想分派、OneLocalSolData、VerifyLocalSolution 和 CommonDenominatorData。
公开 MaximalIdealData 是同构造家族的数据型诊断接口。

Return Value

Key / Value	Meaning
L["M_list"]	极大理想生成元列表。
L["U_list"]	对应的局部完成矩阵。
L["d_list"]	entry denominators。
L["GL_denominator_list"]	更保守的 GL denominators。
L["local_data_list"]	每轮局部求解器数据。
L["verification_list"]	每个局部矩阵的独立验证。
L["denominators_generate_E"]	覆盖是否完成。

Examples

```
restart:
currentdir("C:/QSAAlgorithmFF_release_v3_6_0"):
read "load_all.mpl":

unassign('x','y','a','b','xvars','tvars','Pgens','rowN','L'):
xvars := [x,y]:
tvars := [a,b]:
Pgens := [a^2+b^2]:
rowN := [y+1/b,x+a,x*y+b]:
# 第一分量关于 y 首一，适合直接进入 LocalLoopData。

L := QSFULLFF:-LocalLoopData(rowN,xvars,tvars,Pgens,
    'localMethod'="horrocks",
    'useHeuristics'=false,
    'printBranches'=true
):
# 每完成一轮即打印 d_i 和 U_i。
L["number_of_local_solutions"];
L["M_list"];
L["d_list"];
L["U_list"];
L["denominators_generate_E"];
```

Compatibility

本文档对应 QSAAlgorithmFF release 3.6.0，目标运行环境为 Maple 2025。签名、默认值、返回键和调用关系按该发布版源码说明。

See Also

MaximalIdealData, OneLocalSolData, PatchData。

2.3.3.3.1 QSFULLFF:-MaximalIdealData

MaximalIdealData

返回一般函数域极大理想的相对与环境表示。

[Calling Sequence](#) [Parameters](#) [Description](#) [Program Hierarchy](#) [Return Value](#) [Examples](#) [Compatibility](#) [See Also](#)

Calling Sequence

```
QSFULLFF:-MaximalIdealData(F0,xvars0,tvars0,Pgens0)
```

Program Hierarchy

Program Hierarchy

公开诊断接口实际调用 QSMAXIDEALFF:-MaximalIdealData。LocalLoop 内部只需要生成元，故调用轻量分派器 QSLocalLoopFF:-MaximalIdealContainingOverK；一般函数域分支最终进入 QSMAXIDEALFF:-MaximalIdealContaining。两者共享清分母、水平素分支和相对维数下降算法。

Return Value

主要键: relative_generators、ambient_generators、relative_dimension、ambient_ideal_object。

Examples

```
restart:
currentdir("C:/QSAlgorithmFF_release_v3_6_0"):
read "load_all.mpl":

unassign('x','y','a','b','xvars','tvars','Pgens','F','MData'):
xvars := [x,y]:
tvars := [a,b]:
Pgens := [a^2+b^2]:
F := [y-b,x*y-a]:
# K[x,y] 中待包含的 proper 理想生成元。

MData := QSFULLFF:-MaximalIdealData(F,xvars,tvars,Pgens):
MData["relative_generators"];
# K[x,y] 中的极大理想生成元。
MData["ambient_generators"];
# Q[a,b,x,y] 中的水平素理想表示。
MData["relative_dimension"];
# 成功时应为 0。
```

Compatibility

本文档对应 QSAlgorithmFF release 3.6.0，目标运行环境为 Maple 2025。签名、默认值、返回键和调用关系按该发布版源码说明。

See Also

[LocalLoopData](#), [MaximalIdealContaining](#)。

2.3.3.3.2 QSFULLFF:-OneLocalSolData

OneLocalSolData

在一个极大理想处选择并执行局部方法。

[Calling Sequence](#) [Parameters](#) [Description](#) [Program Hierarchy](#) [Return Value](#) [Examples](#) [Compatibility](#) [See Also](#)

Calling Sequence

```
QSFULLFF:-OneLocalSolData(
    row0,xvars0,tvars0,Pgens0,Maxgens0,
    'localMethod'="horrocks",
    'useHeuristics'=true
)
```


Description

- "horrocks": 精确的局部单位-最小首一次数-ReduceDeg-Suslin Lemma 递归;
- "residue": 精确地降到剩余域, 使用 Euclidean reduction、乘法矩阵、resultant 与 Bézout 块;
- "search": 启发式整数线性组合 norm-search; `useHeuristics=false` 时不执行;
- "auto": 依次尝试 Horrocks、residue, 并在允许时最后尝试 search。

Examples

```
restart:
currentdir("C:/QSAAlgorithmFF_release_v3_6_0"):
read "load_all.mpl":

unassign('x','y','a','b','xvars','tvars','Pgens','rowLocal','Maxgens','LD'):
xvars := [x,y]:
tvars := [a,b]:
Pgens := [a^2+b^2]:
rowLocal := [y+1/b,x+a,x*y+b]:
Maxgens := [x+a]:
# 在 E=K[x] 的极大理想 (x+a) 处求局部完成。

LD := QSFulFF:-OneLocalSolData(
    rowLocal,xvars,tvars,Pgens,Maxgens,
    'localMethod'="horrocks",
    'useHeuristics'=false
):
LD["local_method_effective"];
# 应为 "horrocks"。
LD["row_after_U"];
# 局部环中应为 [1,0,0]。
```

Compatibility

本文档对应 QSAAlgorithmFF release 3.6.0, 目标运行环境为 Maple 2025。签名、默认值、返回键和调用关系按该发布版源码说明。

See Also

LocalLoopData, HorrocksData。

2.3.3.4 QSFulFF:-PatchData

PatchData

把局部覆盖粘合成全局 one-step 矩阵。

Calling Sequence Parameters Description Program Hierarchy Return Value Examples Compatibility See Also

Calling Sequence

```
QSFulFF:-PatchData(LocalLoopData0,aval := 0)
```

Description

自动选择 GL 或 entry denominator, 计算 powered-denominator Bézout 系数, 并乘合 $\Delta_i(Y,Z)$ 矩阵, 使约化后的行从 y 移到 $y = \text{aval}$ 。

Examples

```
restart:
currentdir("C:/QSAAlgorithmFF_release_v3_6_0"):
read "load_all.mpl":

unassign('x','y','a','b','xvars','tvars','Pgens','rowN','L','PData'):
xvars := [x,y]:
tvars := [a,b]:
Pgens := [a^2+b^2]:
```

```

rowN := [y+1/b,x+a,x*y+b]:

L := QSFulFF:-LocalLoopData(rowN,xvars,tvars,Pgens,
    'localMethod'="horrocks",
    'useHeuristics'=false,
    'printBranches'=false
):
# 先得到局部矩阵和覆盖分母。
PData := QSFulFF:-PatchData(L,0):
# 把最后变量 y 专门化到 0。
PData["target_row"];
PData["patch_denominator_source"];
# 查看自动选择的是 GL 还是 entry 分母。
PData["bezout_check"];
# 成功时应为 1。

```

Compatibility

本文档对应 QSAlgorithmFF release 3.6.0，目标运行环境为 Maple 2025。签名、默认值、返回键和调用关系按该发布版源码说明。

See Also

LocalLoopData, OneStepReductionData。

2.3.4 递归步骤 (3)

OneStep 核心返回 `target_row`、`target_xvars` 和专门化后的右逆。递归核心直接调用自身，不再执行根级素性或 Gröbner 么模性计算。递归返回的低维矩阵与 OneStep 矩阵相乘，再执行正规化的逆代换。

2.3.5 QSFulFF:-VerifyCompletionFullData

VerifyCompletionFullData

最终独立检查候选矩阵。

Calling Sequence Parameters Description Program Hierarchy Return Value Examples Compatibility See Also

Calling Sequence

```

QSFulFF:-VerifyCompletionFullData(
    row0,U0,xvars0,tvars0,Pgens0,
    checkPrime := true
)

```

Description

重新计算 row0 $U0$ 和 $\det(U0)$ ，检查模 P 后的行等式、行列式属于 K^* ，并确认矩阵分母中没有多项式变量。

Examples

```

restart:
currentdir("C:/QSAlgorithmFF_release_v3_6_0"):
read "load_all.mpl":

unassign('x','y','a','b','xvars','tvars','Pgens','row','U','V'):
xvars := [x,y]:
tvars := [a,b]:
Pgens := [a^2+b^2]:
row := [b*y+1,x+a,x*y+b]:
U := QSFulFF:-CompleteRow(row,xvars,tvars,Pgens,
    'timeLimit'=600,'printLog'=false,
    'useHeuristics'=false,'localMethod'="horrocks"
):
# 先由主程序得到候选矩阵。

V := QSFulFF:-VerifyCompletionFullData(row,U,xvars,tvars,Pgens):

```

```
V["success"];
V["row_times_U_mod_P"];
V["determinant_mod_P"];
V["entries_are_polynomials_over_K"];
```

Compatibility

本文档对应 QSAAlgorithmFF release 3.6.0，目标运行环境为 Maple 2025。签名、默认值、返回键和调用关系按该发布版源码说明。

See Also

CompleteRowData, PrintVerification。

2.4 QSFULLFF:-CompleteRowResult

CompleteRowResult

把完整数据整理为固定的四键结果表。

Calling Sequence Parameters Description Program Hierarchy Return Value Examples Compatibility See Also

Calling Sequence

与 CompleteRow 相同。

Return Value

Result["U"]、Result["data"]、Result["verification"]、Result["success"]。

Examples

```
restart:
currentdir("C:/QSAAlgorithmFF_release_v3_6_0"):
read "load_all.mpl":

unassign('x','y','a','b','xvars','tvars','Pgens','row','Result'):
xvars := [x,y]:
tvars := [a,b]:
Pgens := [a^2+b^2]:
row := [b*y+1,x+a,x*y+b]:

Result := QSFULLFF:-CompleteRowResult(row,xvars,tvars,Pgens,
    'timeLimit'=600,
    'printLog'=false,
    'useHeuristics'=false,
    'localMethod'="horrocks"
):
Result["success"];
Result["U"];
Result["data"]["algorithm_framework"];
```

Compatibility

本文档对应 QSAAlgorithmFF release 3.6.0，目标运行环境为 Maple 2025。签名、默认值、返回键和调用关系按该发布版源码说明。

See Also

CompleteRow, CompleteRowData。

3 运行控制、缓存与审查要点

3.1 日志与总限时

printLog=true 时，日志依次显示：

1. ValidateFunctionField
2. UnimodularRowData
3. CompleteRow recursive core
 - 3.0 base case 或
 - 3.1 HeuristicCompletionData
 - 3.2.1 NormalisationStep
 - 3.2.2 ReduceDegreeData
 - 3.2.3 LocalLoopData
 - 3.2.4 PatchData
 - 3.3 recursive CompleteRow
4. Final VerifyCompletionFast

最后一个出现 START 而没有 END 的阶段，就是中断或超时时正在运行的位置。timeLimit 限制整次公开调用；它不改变算法搜索空间。

3.2 唯一启发式开关

useHeuristics=false 会：

- 跳过整个 HeuristicCompletionData/QSEasyFF 提前完成层；
- 禁用整数线性组合 norm-search；
- 禁用小整数点极大理想快速路径。

它不会关闭精确基情形、正规化、候选选择、精确降次、Horrocks、residue、patch 或递归。

Examples

```
restart:
currentdir("C:/QSAAlgorithmFF_release_v3_6_0"):
read "load_all.mpl":

QSFULLFF:-DisableHeuristics():
# 永久关闭当前会话的启发式快路径。
QSFULLFF:-HeuristicsEnabled();
# 返回 false。
QSFULLFF:-EnableHeuristics():
# 恢复默认状态。
```

3.3 缓存控制

```
QSFULLFF:-ConfigureCaches(
    enabled0 := true,
    maxIdeals0 := 500,
    maxGB0 := 200,
    maxNF0 := 6000,
    maxMembership0 := 4000
)
QSFULLFF:-ResetCaches()
QSFULLFF:-CacheStats()
QSFULLFF:-PrintCacheStats()
```

缓存只记忆精确结果，不改变算法结构。达到某一类上限时清空该类缓存，避免长期会话无限增长。

Examples

```
restart:
currentdir("C:/QSAAlgorithmFF_release_v3_6_0"):
read "load_all.mpl":
```

```

QSFULLFF:-ConfigureCaches(true,500,200,6000,4000):
# 开启并设置共享缓存容量。
Stats := QSFULLFF:-CacheStats():
# 取得当前命中、未命中和条目数。
QSFULLFF:-PrintCacheStats():
# 在 worksheet 中打印统计。
QSFULLFF:-ResetCaches():
# 清空全部共享和证书缓存。

```

3.4 代码审查时应检查的数据不变量

1. 根级 `input_validation["field_is_valid"]` 与 `unimodularity_data["certificate_verified"]`;
2. 每个递归节点的 `right_inverse` 是否满足当前行乘右逆为 1;
3. `heuristic_data` 未成功时是否继续进入 `one_step`;
4. `OneStep` 的 `target_right_inverse` 是否随最后变量专门化正确传播;
5. `LocalLoop` 的 $d_i \notin M_i$ 、局部矩阵验证和分母单位理想覆盖;
6. `Patch` 的 `powered-denominator Bézout` 恒等式与多项式性;
7. 根级 `final_verification["success"]`。

A 源文件、模块职责与帮助文档对应关系

A.1 为什么帮助文档中的命令名与源文件名不完全相同

本项目采用“公开门面—算法模块—基础设施”三层组织方式，因此帮助文档中的命令名不会与磁盘上的文件名逐项一一对应。

三层命名规则

1. 公开门面层：QSFullFF.mpl 定义用户日常调用的 QSFullFF:-... 命令。帮助文档主要按照这一层编排。

2. 算法实现层：真正的数学算法分布在 QSAlgorithmFF.mpl、QSInputFF.mpl、QSLocalSolFF.mpl、QSLocalLoopFF.mpl、QSPatchFF.mpl 等文件中。

3. 基础设施层：QSGroebnerCacheFF.mpl、QSQuotientSafeFF.mpl、QSRuntimeFF.mpl 和 QSVerifyFF.mpl 为多个算法模块共同提供缓存、商环正规化、日志和独立验证。

例如，帮助文档中的

QSFullFF:-MaximalIdealData

公开包装器定义在 QSFullFF.mpl，而一般函数域上的实际构造定义在 QSMaxIdealFF.mpl。同理，QSFullFF:-LocalLoopData 的公开名称位于 QSFullFF.mpl，循环本体则位于 QSLocalLoopFF.mpl。这种组织使用户只需记住一个稳定门面，同时允许维护者分别审查不同数学阶段。

A.2 载入文件与依赖顺序

发行包根目录有两个载入文件。

文件	职责
load_all.mpl	按依赖顺序读取全部数学模块；这是推荐入口。其顺序首先载入共享基础设施，随后载入极大理想、局部算法、Patch、递归核心，最后载入公开门面。
load.mpl	兼容入口；当前版本仅转发到 load_all.mpl，避免旧 worksheet 失效。

load_all.mpl 的读取顺序如下。这个顺序表示**模块依赖**，不等同于一次 QS 计算的算法执行顺序。

模块载入顺序

QSGroebnerCacheFF
-> QSQuotientSafeFF
-> QSRuntimeFF
-> QSMaxIdeal
-> QSMaxIdealFF
-> QSVerifyFF
-> QSInputFF
-> QSLocalResidueFF
-> QSLocalSolFF
-> QSHorrocksFF
-> QSLocalLoopFF
-> QSPatchFF
-> QSReduceFF
-> QSEasyFF
-> QSAlgorithmFF
-> QSFullFF

[稳定公开门面，最后载入]

A.3 核心 Maple 文件职责总表

下表对应发行包 src/core/ 目录中的全部文件。第三列给出它们在帮助文档中的主要公开入口；没有独立帮助页的文件通常属于内部实现或跨模块基础设施。

源文件	主要实现内容	帮助文档/公开入口对应关系
QSFULLFF.mpl	稳定公开门面。统一导出用户可调用的 <code>QSFULLFF:-...</code> 命令, 检查和转发 <code>keyword</code> 参数, 并把底层模块返回值整理为稳定接口。它本身尽量不重复实现代数算法。	帮助文档中绝大多数标题均属于此文件的公开门面, 例如 <code>CompleteRow</code> 、 <code>CompleteRowData</code> 、 <code>OneStepReduction</code> 、 <code>LocalLoopData</code> 。
QSAAlgorithmFF.mpl	主递归核心。实现根级输入准备、精确基情形、独立启发式阶段、 <code>OneStepReduction</code> 核心、少一个变量的递归、矩阵组合、右逆传播, 以及一元 $K[x]$ 的 Euclidean 完成。	<code>CompleteRow</code> 、 <code>CompleteRowData</code> 、 <code>HeuristicCompletionData</code> 、 <code>OneStepReduction(Data)</code> 、 <code>CompleteRowKx</code> 的实际核心。
QSInputFF.mpl	输入行的精确检查与幺模性判定。清除合法的系数域分母, 通过局部化和 Gröbner 计算获得 Bézout/right-inverse 证书, 并独立验证该证书。	<code>QSFULLFF:-IsUnimodularRow</code> 、 <code>QSFULLFF:-UnimodularRowData</code> 、 <code>BezoutCertificate</code> 。
QSEasyFF.mpl	独立启发式完成层。实现 <code>easyf</code> 、 <code>easysr</code> 、 <code>easyg</code> : 检查单位分量、二元 Bézout、proper subrow、右逆中的零分量或单位分量等快捷证书; 所有候选在返回前模 P 验证。	由 <code>QSFULLFF:-HeuristicCompletionData</code> 调用; 受 <code>useHeuristics</code> 控制。
QSLocalSolFF.mpl	两类职责: 全局 <code>NormalisationStep</code> ; 以及给定极大理想后的局部求解分派 <code>OneLocalSolData</code> 。分派到 Horrocks、residue 或 <code>search</code> , 并实现局部单位、成员和最终局部矩阵验证。	<code>NormalisationStep</code> 、 <code>OneLocalSolData</code> 、 <code>VerifyLocalSolution</code> 。
QSHorrocksFF.mpl	精确 Horrocks–Suslin 局部完成。执行局部单位分支、最小局部首次数选择、局部 <code>ReduceDeg</code> 、Suslin Lemma 降次和递归。	<code>OneLocalSolData(localMethod="horrocks")</code> 的实际求解器; 另有 <code>HorrocksData</code> 、 <code>SuslinLemmaData</code> 等研究接口。
QSLocalResidueFF.mpl	精确剩余域/resultant 局部求解器。把尾行降到 $\kappa(\mathfrak{m})[y]$, 执行 Euclidean reduction, 构造乘法矩阵、 <code>norm/resultant</code> 和 Bézout 块。	<code>OneLocalSolData(localMethod="residue")</code> 的实际求解器。
QSLocalLoopFF.mpl	局部覆盖循环。反复判断当前分母是否生成单位理想, 选择包含当前分母理想的极大理想, 调用单个局部解, 验证 U_i , 提取 d_i , 并保存每个分支的 M_i, d_i, U_i 。	<code>QSFULLFF:-LocalLoopData</code> ; 返回 <code>M_list</code> 、 <code>d_list</code> 、 <code>U_list</code> 。
QSMaxIdeal.mpl	处理 $P = 0$ 或纯有理函数域 $\mathbb{Q}(t)[x]$ 情形的确定性极大理想构造。使用素分解、维数、独立变量和专门化。	主要是 <code>QSLocalLoopFF:-MaximalIdealContainingOverK</code> 的特殊情形下级; 普通用户通常不直接调用。
QSMaxIdealFF.mpl	一般函数域 $K = \text{Frac}(\mathbb{Q}[t]/P)$ 上的极大理想构造。通过环境环中的水平素理想、收缩到 P 、相对维数下降和相对生成元恢复得到 $K[x]$ 的极大理想。	<code>QSFULLFF:-MaximalIdealContaining</code> 、 <code>QSFULLFF:-MaximalIdealData</code> 的实际实现。
QSPatchFF.mpl	Patch 阶段。选择可覆盖的分母列表, 求 powered-denominator Bézout 系数, 构造并相乘 $\Delta_i(Y, Z)$, 得到把最后变量移动到指定值的全局矩阵。	<code>QSFULLFF:-PatchData</code> ; 也是 <code>OneStepReduction</code> 的最后一个主要子步骤。
QSReduceFF.mpl	正规化后的全局精确降次。选择首项系数属于 K^* 的较好分量, 用其对其他分量做多项式带余除法, 并记录初等列矩阵和行列式证书。	<code>OneStepReduction</code> 内部的 <code>ReduceDegreeData</code> ; 通常不需要用户直接调用。
QSGroebnerCacheFF.mpl	共享有界缓存与模 P 正规形层。缓存 <code>PolynomialIdeal</code> 、Gröbner 基、normal form 和理想成员测试, 并提供统一的 <code>ReduceModuloP</code> 、 <code>IsZeroModuloP</code> 。	<code>ConfigureCaches</code> 、 <code>ResetCaches</code> 、 <code>CacheStats</code> ; 被几乎所有代数模块横向使用。
QSQuotientSafeFF.mpl	商环安全屏障。对行列式、矩阵逆和其他重新组合表达式的分子分母先模 P 约化, 再判断零、单位、变量依赖或提取分母, 防止环境代表元中的伪项造成误判。	无独立日常帮助页; 由 <code>LocalLoop</code> 、 <code>Patch</code> 、 <code>Verify</code> 和 <code>OneStep</code> 等模块内部调用。
QSRuntimeFF.mpl	轻量运行日志和实际时间统计。负责 <code>START/END</code> 、递归缩进、步骤耗时和总耗时; 不改变任何代数分支。	主程序的 <code>printLog</code> 与 <code>elapsed_real_seconds</code> 。

源文件	主要实现内容	帮助文档/公开入口对应关系
QSVerifyFF.mpl	独立验证器。重新检查 fU 、目标行、 $\det(U) \in K^*$ 、矩阵条目的多项式性以及模 P 正规形，不信任构造阶段的 success 标志。	VerifyCompletionFastData、VerifyCompletionFullData、PrintVerification。

A.4 帮助文档命令到实际实现文件的速查

帮助页以公开门面名称为标题；下表显示门面包装器和真正执行算法的文件。

帮助文档中的公开命令	核心实现文件	继续调用的主要模块
QSFulFF:-CompleteRow、CompleteRowData	QSAAlgorithmFF.mpl	QSInputFF、QSEasyFF、OneStep 子树、递归和 QSVerifyFF。
QSFulFF:-ValidateFunctionField	QSAAlgorithmFF.mpl	PolynomialIdeal 与 Gröbner/缓存基础设施。
QSFulFF:-UnimodularRowData	QSInputFF.mpl	QSGroebnerCacheFF 和证书验证。
QSFulFF:-HeuristicCompletionData	QSAAlgorithmFF.mpl	QSEasyFF.mpl。
QSFulFF:-OneStepReduction(Data)	QSAAlgorithmFF.mpl	QSLocalSolFF、QSReduceFF、QSLocalLoopFF、QSPatchFF。
QSFulFF:-NormalisationStep	QSLocalSolFF.mpl	候选选择、变量顺序和 Weierstrass 代换。
QSFulFF:-LocalLoopData	QSLocalLoopFF.mpl	极大理想家族、OneLocalSolData、局部验证和分母提取。
QSFulFF:-MaximalIdealData	QSMaIdealFF.mpl	环境理想、水平性、相对维数； $P = 0$ 时可使用 QSMaIdeal.mpl 的轻量分支。
QSFulFF:-OneLocalSolData	QSLocalSolFF.mpl	QSHorrocksFF、QSLocalResidueFF 或 search 分支。
QSFulFF:-PatchData	QSPatchFF.mpl	powered Bézout、DeltaMatrix 和多项式性检查。
QSFulFF:-CompleteRowKx	QSAAlgorithmFF.mpl	商域感知的 Euclidean/gcdex 基情形。
QSFulFF:-VerifyCompletionFullData	QSVerifyFF.mpl	quotient-safe 行列式和矩阵条目检查。
QSFulFF:-ConfigureCaches 等	QSGroebnerCacheFF.mpl	汇总 easy、unimodular、residue、unit-ideal 和 maximal-ideal 缓存。

A.5 按主算法阶段阅读源码

若目标是审查主算法而不是逐文件浏览，建议按下列顺序阅读。

1. 公开参数和返回结构：QSFulFF.mpl。
2. 主递归骨架：QSAAlgorithmFF.mpl，重点查看 PrepareInputData、CompleteRowRecursiveData、HeuristicCompletionCoreData、OneStepReductionCoreData 和矩阵组合过程。
3. 根级数学前提：QSInputFF.mpl；启发式阶段再看 QSEasyFF.mpl。
4. OneStepReduction：依次阅读 QSLocalSolFF.mpl、QSReduceFF.mpl、QSLocalLoopFF.mpl 和 QSPatchFF.mpl。
5. 局部求解细节：Horrocks 路径阅读 QSHorrocksFF.mpl；residue 路径阅读 QSLocalResidueFF.mpl。
6. 极大理想：一般函数域阅读 QSMaIdealFF.mpl； $P = 0$ 的特殊分支阅读 QSMaIdeal.mpl。
7. 共享正确性基础：QSGroebnerCacheFF.mpl、QSQuotientSafeFF.mpl、QSVerifyFF.mpl；运行过程观察则阅读 QSRuntimeFF.mpl。

最重要的对应关系

帮助文档按 QSFulFF:-... 公共命令组织；审查源码时先在 QSFulFF.mpl 找到包装器，再沿其转发目标进入实际模块。不要仅根据帮助页标题猜测文件名。例如 QSFulFF:-MaximalIdealData 的实际算法在 QSMaIdealFF.mpl，而不是名为 MaximalIdealData.mpl 的独立文件。

快速检查清单

1. 使用 XeLaTeX 编译本文档；不需要安装 Noto CJK 字体。
2. Maple 中先 `restart`，再从发行包根目录读取 `load_all.mpl`。
3. 使用有序 `xvars`；其顺序决定递归变量顺序。
4. 日常计算使用 `CompleteRowData`，只取矩阵时使用 `CompleteRow`。
5. 关闭启发式后，日志仍应出现正规化、降次、LocalLoop、Patch 和递归。
6. 检查 `Data["verification"]["success"]`，并在最终验收时调用 `VerifyCompletionFullData`。

运行认证说明

当前构建环境没有 Maple 内核。本文档中的签名、调用关系和返回键依据 release 3.6.0 源码静态核对；最终动态认证应在 Maple 2025 中运行具体案例，并以独立验证器结果为准。