Computing Gauss–Manin Connections for Families of Projective Hypersurfaces

Sebastian Pancratz
Hertford College
University of Oxford

A thesis submitted for the
Transfer of Status

Michaelmas 2009

Contents

Preface

In this thesis, we present an approach to computing in the algebraic de Rham cohomology spaces associated to a family of projective hypersurfaces over a field of characteristic zero. We take a special interest in such families containing a diagonal fibre. As a particular application, we consider the computation of the action of the Gauss–Manin connection. While the general algorithm is well-known in the literature, we demonstrate that a change in the implementation of the key step, replacing an approach based on Gröbner basis computations by a direct method with sparse matrices, improves the typical run-time of computations by several orders of magnitude.

The main body of this thesis is structured as follows. In Chapter 1, we introduce the main objects of our study. We first consider algebraic de Rham cohomology spaces and the action of the Gauss–Manin connection on these in some generality, but eventually restrict to the case of families of hypersurfaces. In Chapters 2 and 3, we present detailed descriptions of the algorithms that form the main focus of this work. In the former, we consider computations in de Rham cohomology, whereas in the latter we use this to formulate an algorithm for computing the matrix of the Gauss–Manin connection. We explicitly demonstrate the success of the new approach in Chapter 5, by comparing our implementation of it with a pre-existing set of routines by Lauder. Finally, in Chapter 6 we provide a list of suggestions which might further improve our current implementation.

Chapter 1

Introduction

In this chapter we introduce the main objects of our study. We are interested in the computation of the Gauss–Manin connection matrix of the de Rham complex associated to a smooth hypersurface in projective space. In the following sections, while we provide some background and references, our aim is to provide a clean introduction and to arrive at a computationally feasible setup.

1.1 Algebraic de Rham cohomology

We begin by recalling some definitions and results from the algebraic theory of Kähler differentials. We first consider affine sheaves via their underlying rings, following Hartshorne [11], and then extend this to schemes.

Consider a ring A (commutative and with multiplicative identity), and let B be an A-algebra and M a B-module.

Definition 1.1. An *A-derivation* of B into M is a map $d\colon B \to M$ which satisfies the following three properties:

(i) d is additive;

(ii) the Leibniz rule holds: $d(bb') = bdb' + b'db$ for all $b, b' \in B$;

(iii) $da = 0$ for all $a \in A$.

Definition 1.2. A B-module $\Omega^1_{B/A}$ together with an A-derivation $d\colon B \to \Omega^1_{B/A}$ is called a *module of relative differential forms* if the following universal property is satisfied: for any B-module M with an A-derivation $d'\colon B \to M$ there is a unique B-module homomorphism $f\colon \Omega^1_{B/A} \to M$ such that $d' = f \circ d$.

The existence of the (essentially unique) module of relative differential forms of B over A can be shown by construction. Namely, we can consider the quotient of the free B-module generated by all symbols db, for $b \in B$, by the submodule generated by expressions of the three forms in Definition 1.1 together with the derivation sending an element b to the image of db.

This construction can be generalised from affine sheaves to schemes via a glueing argument, yielding the following result:

Proposition 1.3. Let $\pi\colon X \to S$ be a morphism of schemes. Then there exists a unique quasi-coherent sheaf $\Omega^1_{X/S}$ on X such that for any affine open subset $V \subset S$, any affine open subset $U \subset \pi^{-1}(V)$ and any $x \in U$ we have

$$\Gamma(U, \Omega^1_{X/S}) \cong \Omega^1_{\mathcal{O}_X(U)/\mathcal{O}_S(V)}, \quad (\Omega^1_{X/S})_x \cong \Omega^1_{\mathcal{O}_{X,x}/\mathcal{O}_{S,\pi(x)}}.$$

Proof. See Liu [16, Proposition 6.1.17]. □

Definition 1.4. Let $\pi\colon X \to S$ be a morphism of schemes. The sheaf $\Omega^1_{X/S}$ is called the *sheaf of relative* 1*-forms* of X over S. For $i \in \mathbf{N}$, we also define the *sheaf of relative* i*-forms* of X over S by

$$\Omega^i_{X/S} = \bigwedge_{\mathcal{O}_X}^{i} \Omega^1_{X/S}$$

with the additional convention that $\Omega^0_{X/S} = \mathcal{O}_X$.

In the above situation, one can check that the derivation $d\colon \mathcal{O}_X \to \Omega^1_{X/S}$ induces a family of maps, turning the above sequence of sheaves into a complex:

Proposition 1.5. Let $\pi\colon X \to S$ be a morphism of schemes. Then there exists a unique family of maps $d\colon \Omega^i_{X/S} \to \Omega^{i+1}_{X/S}$ such that the following conditions are satisfied:

(i) d is $\pi^{-1}(\mathcal{O}_S)$-linear and $d(ab) = da \wedge b + (-1)^i a \wedge db$ for a homogeneous of degree i;

(ii) $d \circ d = 0$;

(iii) $da = d_{X/S}(a)$ for a of degree zero.

Proof. See Illusie [13, §1]. □

Definition 1.6. This complex is called the *(algebraic) de Rham complex*. We refer to an i-form as *closed* if it is in the kernel of d, and *exact* if it is in the image of d. Moreover, we define the *relative de Rham cohomology sheaf* $\mathcal{H}^i_{dR}(X/S)$ to be

$$\mathcal{H}^i_{dR}(X/S) = \mathcal{R}^i\pi_*(\Omega^\bullet_{X/S})$$

where $\mathcal{R}^i$ is the ith hyperderived functor of π_*. We refer to its global sections as the *ith relative de Rham cohomology of X over S* and denote this by $H^i_{dR}(X/S)$.

1.2 Gauss–Manin connection

In this section, we give a brief introduction to the Gauss–Manin connection in the algebraic sense, following Katz and Oda [14]. We consider two smooth schemes X and S over a field K together with a smooth K-morphism $\pi\colon X \to S$.

Definition 1.7. A *connection* on a quasi-coherent $\mathcal{O}_S$-module $\mathcal{E}$ is defined to be a homomorphism ρ of abelian sheaves

$$\rho\colon \mathcal{E} \to \Omega^1_{S/K} \otimes_{\mathcal{O}_S} \mathcal{E}$$

such that

$$\rho(se) = s\rho(e) + ds \otimes e$$

for sections s and e of $\mathcal{O}_S$ and $\mathcal{E}$, respectively, over an open subset of S.

Remark 1.8. We observe that this can be extended to the whole de Rham complex as follows. For every $i \in \mathbf{N}$, given $\omega \in \Omega^i_{S/K}$ and $e \in \mathcal{E}$ let $\omega \wedge \rho(e)$ denote the image of $\omega \otimes \rho(e)$ under the map

$$\Omega^i_{S/K} \otimes_{\mathcal{O}_S} \left(\Omega^1_{S/K} \otimes_{\mathcal{O}_S} \mathcal{E}\right) \to \Omega^{i+1}_{S/K} \otimes_{\mathcal{O}_S} \mathcal{E}, \quad \omega \otimes (\tau \otimes e) \mapsto (\omega \wedge \tau) \otimes e.$$

We then obtain a homomorphism of abelian sheaves ρ_i given by

$$\rho_i\colon \Omega^i_{S/K} \otimes_{\mathcal{O}_S} \mathcal{E} \to \Omega^{i+1}_{S/K} \otimes_{\mathcal{O}_S} \mathcal{E}, \quad \omega \otimes e \mapsto d\omega \otimes e + (-1)^i \omega \wedge \rho(e).$$

Definition 1.9. The complex $\Omega^\bullet_{X/K}$ admits a decreasing filtration given by

$$F^j = \operatorname{Im}\Big(\Omega^{\bullet-j}_{X/K} \otimes_{\mathcal{O}_X} \pi^*\big(\Omega^j_{S/K}\big) \to \Omega^\bullet_{X/K}\Big).$$

Forming a spectral sequence $\{E_r, d_r\}_{r\geq 0}$ as in Griffiths and Harris [9, §3.5, p. 440], we find that

$$E_1^{i,j} = \Omega^i_{S/K} \otimes_{\mathcal{O}_S} \mathcal{H}^j_{dR}(X/S).$$

The *(algebraic) Gauss–Manin connection* ∇_{GM} is now defined as the differential $d_1^{0,j}$, i.e.,

$$\nabla_{GM} = d_1^{0,j}\colon \mathcal{H}^j_{dR} \to \Omega^1_{S/K} \otimes_{\mathcal{O}_S} \mathcal{H}^j_{dR}(X/S).$$

Remark 1.10. From the description by Katz and Oda [14] and following the practical description by Kedlaya [15], we can describe the action of the Gauss–Manin connection further in the case when S is affine, say $S = \operatorname{Spec}(A)$. In this case, we may apply the global section functor $\Gamma(S, -)$ throughout and consider

$$\nabla_{GM}\colon H^j_{dR}(X/S) \to \Omega^1_{A/K} \otimes_A H^j_{dR}(X/S).$$

The action of ∇_{GM} can be computed as follows. Given $\omega \in H^j_{dR}(X/S)$, arbitrarily lift this to $\tilde{\omega} \in \Omega^j_{X/K}$. Computing the exterior derivative, which decomposes as $d_{X/K} = d_S + d_{X/S}$, the image of ω under the Gauss–Manin connection is then given as the projection of $d_{X/K}(\tilde{\omega})$ onto $\Omega^1_{A/K} \otimes_A H^j_{dR}(X/S)$.

1.3 Hypersurfaces in projective space

In this section we make the results from the previous sections concrete, specialising to the case of hypersurfaces in projective space. In the development of a computationally feasible description of the Gauss–Manin connection in this situation, we follow Abbott, Kedlaya and Roe [1, §3.2] when computing in de Rham cohomology and Kedlaya [15, §3.2] when expressing the action of the Gauss–Manin connection. We begin by setting the scene with some notation, which we shall use throughout this thesis:

Notation 1.11. We consider a non-singular hypersurface X in $\mathbf{P}^n(K) \times \mathbf{A}^1(K)$, where K is a field of characteristic zero and $n \geq 2$, defined by a homogeneous polynomial $P \in K[t][x_0, \ldots, x_n]$ of degree d. We let U denote the open complement $\mathbf{P}^n(K) \times \mathbf{A}^1(K) - X$.

Moreover, we assume that we may also view this as two smooth K-schemes X and S together with a smooth proper K-morphism $\pi\colon X \to S$, where we consider an affine subspace $S = \operatorname{Spec}(A)$ of the t-line. We denote the fibre of X above t in S by X_t.

Let us consider a smooth fibre X_t. The embedding $X_t \hookrightarrow \mathbf{P}^n(K)$ induces maps $H^i_{dR}(\mathbf{P}^n(K)/K) \to H^i_{dR}(X_t/K)$, which by the Lefschetz hyperplane theorem [9, §1.2, p. 156] are bijective for $0 \leq i \leq n-2$ and injective for $i = n-1$. A direct computation [1, Corollary 3.1.4] shows that $H^i_{dR}(\mathbf{P}^n(K))$ has, as a K-vector space, dimension 1 if $i = 0, 2, \ldots, 2n$ and 0 otherwise. Using Poincaré duality, it then follows that, for all $0 \leq i \leq 2n-2$ with $i \neq n-1$,

$$\dim_K H^i_{dR}(X_t/K) = \begin{cases} 1 & \text{if } i \text{ is even,} \\ 0 & \text{otherwise.} \end{cases} \tag{1.1}$$

In particular, $H^{n-1}_{dR}(X_t/K)$ is the only cohomology group that needs to be computed. Defining $U_t = \mathbf{P}^n(K) - X_t$, from [8, (10.16)], we have one of the following two exact sequences

$$0 \to H^n_{dR}(U_t/K) \to H^{n-1}_{dR}(X_t/K) \to 0, \tag{1.2}$$

$$0 \to H^n_{dR}(U_t/K) \to H^{n-1}_{dR}(X_t/K) \to H^{n+1}_{dR}(\mathbf{P}^n(K)/K) \to 0, \tag{1.3}$$

as n is even or odd, respectively.

Now define the n-form

$$\Omega = \sum_{i=0}^{n} (-1)^i x_i dx_0 \wedge \cdots \wedge \widehat{dx_i} \wedge \cdots \wedge dx_n. \tag{1.4}$$

A calculation as in [8, §4] then shows that the above cohomology group $H^n_{dR}(U_t/K)$ is isomorphic as a K-vector space to the quotient of the group of n-forms $Q\Omega/P^k$ with $k \in \mathbf{N}$ and $Q \in K[x_0, x_1, \dots, x_n]$ homogeneous of degree $kd - (n+1)$ by the subgroup generated by

$$\frac{(\partial_i Q)\Omega}{P^k} - k\frac{Q(\partial_i P)\Omega}{P^{k+1}} \tag{1.5}$$

for all $0 \leq i \leq n$, where here and in the following ∂_i denotes the partial derivative operator with respect to x_i.

Since the fibres of the relative de Rham cohomology $H^i_{dR}(X/S)$ can be identified with the de Rham cohomology $H^i_{dR}(X_t/K)$ of the fibres, it follows that we can calculate the action of the Gauss–Manin connection $\nabla_{GM} \colon H^{n-1}_{dR}(X/S) \to \Omega^1_{A/K} \otimes_A H^{n-1}_{dR}(X/S)$ via the induced map $H^n_{dR}(U/S) \to \Omega^1_{A/K} \otimes_A H^n_{dR}(U/S)$, which by abuse of notation we shall refer to as ∇_{GM}, too. Let $\omega \in H^n_{dR}(U/S)$, where we may assume it is of the form $Q\Omega/P^k$ as described above. Since S is an affine curve, we have that $\Omega^1_{A/K}$ is free of rank one, generated by the symbol dt. Then the action of ∇_{GM} is given by

$$\nabla_{GM} \colon \omega \mapsto d_U(\omega) = d_S(\omega) + d_{U/S}(\omega) = \frac{\partial}{\partial t}\omega \wedge dt \tag{1.6}$$

where the term $d_{U/S}(\omega)$ vanishes by definition of Ω. We will describe how a unique representative can be obtained for the right-hand side in Chapters 2 and 3.

Chapter 2

Computing in de Rham Cohomology

This chapter is devoted to an in-depth description of the computation in de Rham cohomology, exploiting the vector space isomorphism given in Section 1.3 of the introduction. In the first section, we begin by setting up the notation and then describe the so-called *reduction of poles* in some generality. A particular problem, that of finding the coordinates of an element of a multivariate polynomial ideal, is treated as a black box, but we turn to it in the second section where we specialise to the case when the family of projective hypersurfaces contains a diagonal fibre. Finally, we provide further details on the matrix computations involved in the third section.

2.1 Reduction of poles

This section is based on [1, Remark 3.2.5], describing a *reduction of poles* procedure also referred to as the Griffiths–Dwork method. We continue with the same Notation 1.11, but since for large parts of the discussion it will not matter that $K(t)$ is a function field, we also define $L = K(t)$.

From the description of $H^n_{dR}(U/S)$ in the introduction, with its elements being represented by n-forms $Q\Omega/P^i$ for $i \in \mathbf{N}$, it is clear that it can be equipped with a filtration whose ith part consists of all elements which can be represented by n-forms as above with $\deg Q = kd - (n+1)$ for $1 \leq k \leq i+1$.

We can obtain a basis for $H^n_{dR}(U/S)$ respecting this filtration as follows. For every $k \in \mathbf{N}$, we find an independent set B_k of polynomials of degree $kd-(n+1)$ generating the quotient of the space of all such polynomials by the Jacobian ideal $(\partial_0 P, \dots, \partial_n P)$. This yields a generating set $\bigcup_{k\in\mathbf{N}} \mathcal{B}_k$ for $H^n_{dR}(U/S)$ where $\mathcal{B}_k = \{Q\Omega/P^k : Q \in B_k\}$. However, it follows from a theorem of Macaulay [8, §4, (4.11)] that in fact the set $\mathcal{B}_1 \cup \cdots \cup \mathcal{B}_n$ already forms a generating set. In Section 2.2, we shall exhibit an explicit basis of monomials in the case where the family of projective hypersurfaces contains a diagonal fibre.

Now, to obtain a unique representative for the class of $Q\Omega/P^k$ in terms of the basis elements in $\mathcal{B}_1 \cup \cdots \cup \mathcal{B}_n$, we express Q in terms of $\partial_0 P, \ldots, \partial_n P$ as well as elements of B_k, and then iteratively reduce the pole order of the first part according to the relations given by the expressions from Equation (1.5). Assume that we have at our disposal a method DECOMPOSE which, given a polynomial Q of degree $kd-(n+1)$ lying in the ideal generated by the Jacobian ideal $(\partial_0 P, \ldots, \partial_n P)$ together with the set B_k, returns an expression of the form $Q = Q_0 \partial_0 P + \cdots + Q_n \partial_n P + \gamma_k$ with $Q_0, \ldots, Q_n$ homogeneous polynomials in $L[x_0, \ldots, x_n]$ and γ_k in the L-span of B_k. The reduction of poles procedure can then be formalised as in Algorithm 1 below, which we will refer to as REDUCE. In this generality, the correctness of the algorithm depends on a theorem of Macaulay [8, §4, (4.11)].

Algorithm 1 Reduce $Q\Omega/P^k$ in cohomology

Input:
- P is a homogeneous polynomial in $L[x_0, \ldots, x_n]$ of degree d, defining a non-singular hypersurface, where L is a field of characteristic zero.
- For $1 \leq k \leq n$, B_k is a basis for all homogeneous polynomials of degree $kd-(n+1)$ modulo the Jacobian ideal $(\partial_0 P, \ldots, \partial_n P)$.
- Q is a homogeneous polynomial of degree $kd-(n+1)$.

Output: Polynomials γ_i in the L-span of B_i, for $1 \leq i \leq n$, such that $Q\Omega/P^k \equiv \gamma_1 \Omega/P^1 + \cdots + \gamma_n \Omega/P^n$.

procedure REDUCE($P, B_1, \ldots, B_n, Q$)
    while $k \geq n+1$ **do**
        $Q_0, \ldots, Q_n \leftarrow$ DECOMPOSE($Q, \partial_0 P, \ldots, \partial_n P, B_k$)
        $k \leftarrow k-1$
        $Q \leftarrow \frac{1}{k} \sum_{i=0}^n \partial_i Q_i$
    end while
    $\gamma_{k+1}, \ldots, \gamma_n \leftarrow 0$
    while $Q \notin B_k$ **do**
        $Q_0, \ldots, Q_n \leftarrow$ DECOMPOSE($Q, \partial_0 P, \ldots, \partial_n P, B_k$)
        $\gamma_k \leftarrow Q - \sum_{i=0}^n Q_i \partial_i P$
        $k \leftarrow k-1$
        $Q \leftarrow \frac{1}{k} \sum_{i=0}^n \partial_i Q_i$
    end while
    if $Q \neq 0$ **then**
        $\gamma_k \leftarrow Q$
        $k \leftarrow k-1$
    end if
    $\gamma_1, \ldots, \gamma_k \leftarrow 0$
    return $\gamma_1, \ldots, \gamma_n$
end procedure

2.2 Reduction of poles using linear algebra

In this section, we specialise to the case of a smooth family of projective hypersurfaces containing a diagonal fibre. In this case, we exhibit a basis of monomials $B_1 \cup \cdots \cup B_n$ such that the corresponding set $\mathcal{B}_1 \cup \cdots \cup \mathcal{B}_n$ forms a basis for $H^n_{dR}(U/S)$ and re-express the problem of decomposing a homogeneous polynomial from the previous section in the language of linear algebra. From this description, we furnish an explicit reduction procedure in terms of matrices. The approach is based on a generalisation of Sylvester matrices from two polynomials to $n+1$ polynomials, following Macaulay [17].

The decomposition problem from the previous section can be formulated as follows:

Problem 2.1. Given a homogeneous polynomial $Q \in L[x_0, \dots, x_n]$ of degree $kd-(n+1)$, for some $k \in \mathbf{N}$, we try to find homogeneous polynomials $Q_0, \dots, Q_n$ in $L[x_0, \dots, x_n]$ such that

$$Q \equiv Q_0 \partial_0 P + \cdots + Q_n \partial_n P \tag{2.1}$$

modulo the L-span of B_k.

Remark 2.2. Immediately, we see that, for each $0 \le i \le n$, either Q_i is identically zero or has degree $(k-1)d-n$ since $\partial_i P$ is homogeneous of degree $d-1$ and also the elements of B_k have degree $kd-(n+1)$.

For the remaining part of this section, and in fact this thesis, we consider the following basis sets B_k, for $k \in \mathbf{N}$, which as in the previous section induce a generating set $\mathcal{B}_1 \cup \cdots \cup \mathcal{B}_n$ of $H^n_{dR}(U/S)$ as an L-vector space respecting the aforementioned filtration:

Notation 2.3. Given a multi-index $i = (i_0, \dots, i_n) \in \mathbf{N}_0^{n+1}$, we let x^i denote the monomial $x_0^{i_0} \cdots x_n^{i_n}$ and set $|i| = i_0 + \cdots + i_n$.

Definition 2.4. For $k \in \mathbf{N}$, we define the following sets of monomials,

$$F_k = \{x^i : |i| = kd-(n+1)\}, \tag{2.2}$$

$$B_k = \{x^i : |i| = kd-(n+1) \text{ and } i_j < d-1 \text{ for } 0 \le j \le n\}. \tag{2.3}$$

Moreover, we let $\mathcal{F}_k$ denote the L-vector space spanned by the n-forms $x^i\Omega/P^k$ for $x^i \in F_k$ and $\mathcal{B}_k$ denote the set of n-forms $x^i\Omega/P^k$ with $x^i \in B_k$.

We shall defer the proof of the statement that the corresponding set $\mathcal{B}_1 \cup \cdots \cup \mathcal{B}_n$ indeed forms a basis of $H^n_{dR}(U/S)$, at least in the case when the family of hypersurfaces given by P contains a diagonal fibre, until the end of this section.

Assumption 2.5. In order to ensure that our computations are not vacuous, we assume that $H^n_{dR}(U/S) \neq 0$. After setting

$$\ell = \left\lceil \frac{n+1}{d} \right\rceil, \quad u = \left\lfloor \frac{(n+1)(d-1)}{d} \right\rfloor = n+1-\ell, \tag{2.4}$$

we note that B_k is non-empty if and only if $\ell \leq k \leq u$. The assumption is thus equivalent to $\ell \leq u$. Finally, it can easily be verified that this is equivalent to the statement that $d \geq 2$ whenever n is odd and $d \geq 3$ whenever n is even.

When considering Problem 2.1, it turns out that, in the case when the family of hypersurfaces given by P contains a diagonal fibre, we can place further restrictions on the polynomials $Q_0, \dots, Q_n$. We thus consider the following problem instead:

Problem 2.6. Given a homogeneous polynomial $Q \in L[x_0, \dots, x_n]$ of degree $kd-(n+1)$, for some $k \in \mathbf{N}$, we try to find homogeneous polynomials $Q_0, \dots, Q_n$ in $L[x_0, \dots, x_n]$ such that

$$Q \equiv Q_0 \partial_0 P + \cdots + Q_n \partial_n P \tag{2.5}$$

modulo the L-span of B_k. Moreover, for each $1 \leq j \leq n$, the polynomial Q_j may only contain non-zero coefficients for monomials of degree $(k-1)d-n$ that are not divisible by any of the monomials $x_0^{d-1}, \dots, x_{j-1}^{d-1}$.

Definition 2.7. For $k \in \mathbf{N}$, we define the following sets of monomials in $L[x_0, \dots, x_n]$. Let $\mathcal{R}_k = F_k - B_k$, containing the monomials of total degree $kd-(n+1)$ and partial degree at least $d-1$ with respect to some of the $n+1$ variables. Let $\mathcal{C}_k^{(0)}$ be the set of monomials of total degree $(k-1)d-n$, and then inductively, for $j = 1, \dots, n$, define $\mathcal{C}_k^{(j)}$ to be the set of monomials in $\mathcal{C}_k^{(j-1)}$ except for those divisible by x_{j-1}^{d-1}. Moreover, we define the multi-set $\mathcal{C}_k$ as the disjoint union of $\mathcal{C}_k^{(0)}, \dots, \mathcal{C}_k^{(n)}$. We shall write an element of this multi-set as (j, g), referring to a monomial g in $\mathcal{C}_k^{(j)}$.

With this set-up, the following theorem provides a solution to Problem 2.6 in the cases that we are interested in:

Theorem 2.8. Suppose that the family of projective hypersurfaces given by the polynomial P in $K[t][x_0, \dots, x_n]$ contains a diagonal fibre. Let $k \in \mathbf{N}$ and suppose that $\mathcal{R}_k$ and $\mathcal{C}_k$ are non-empty. For $0 \leq j \leq n$, let $V_k^{(j)}$ be the L-vector space of polynomials with basis $\mathcal{C}_k^{(j)}$, and let V_k denote their cartesian product $V_k = V_k^{(0)} \times \cdots \times V_k^{(n)}$. Let W_k be the L-vector space of polynomials with basis $\mathcal{R}_k$. Then the L-linear map

$$\phi_k \colon V_k \to W_k, (Q_0, \dots, Q_n) \mapsto Q_0 \partial_0 P + \cdots + Q_n \partial_n P \tag{2.6}$$

is an isomorphism of L-vector spaces.

Proof. We first show that, for all $k \in \mathbf{N}$, the multi-sets $\mathcal{R}_k$ and $\mathcal{C}_k$ have the same cardinality:

We construct the following bijection $\mathcal{R}_k \to \mathcal{C}_k$, representing the monomials by their exponent tuple. Let $i = (i_0, \ldots, i_n)$ be in $\mathcal{R}_k$. If $i_0 \geq d-1$, we define the image as $(i_0 - d - 1, i_1, \ldots, i_n) \in \mathcal{C}_k^{(0)}$. More generally, if $i_0 < d-1, \ldots, i_{j-1} < d-1$ and $i_j \geq d-1$, the image is $(i_0, \ldots, i_{j-1}, i_j - d - 1, i_{j+1}, \ldots, i_n) \in \mathcal{C}_k^{(j)}$. It is easy to verify that this map is indeed a bijection.

In order to establish that the map $\phi_k \colon V_k \to W_k$ is an isomorphism of L-vector spaces, we now exhibit its matrix with respect to the given basis:

Let $k \in \mathbf{N}$ and suppose that $\mathcal{R}_k$ and $\mathcal{C}_k$ are non-empty, that is to say, $k \geq n/d + 1$. We define the auxiliary matrix Δ_k with row and column index sets $\mathcal{R}_k$ and $\mathcal{C}_k$, respectively, as follows. Given $f \in \mathcal{R}_k$ and $(j, g) \in \mathcal{C}_k$, we set the corresponding entry in Δ_k to be the monomial coefficient of f/g in $\partial_j P$ if g divides f and 0 otherwise. It is immediate that Δ_k is the matrix representing ϕ_k with respect to the bases $\mathcal{C}_k$ and $\mathcal{R}_k$ of V_k and W_k, respectively.

The assumption that the family X of projective hypersurfaces given by P contains a diagonal hypersurface means that for some t_0 the fibre X_{t_0} is given by an equation of the form

$$P_{t_0}(x_0, \ldots, x_n) = \alpha_0 x_0^d + \cdots \alpha_n x_n^d = 0 \tag{2.7}$$

with $\alpha_0, \ldots, \alpha_n \in K^\times$.

We aim to show that the determinant of Δ_k is non-zero in L. Since the specialisation to the diagonal fibre viz. evaluation of the matrix at $t = t_0$ commutes with computing the determinant, it suffices to show that the determinant of $(\Delta_k)\big|_{t=t_0}$ is non-zero in K.

Since, for $0 \leq j \leq n$, $\partial_j P_{t_0}(x_0, \ldots, x_n) = d\alpha_j x_j^{d-1}$, there is precisely one non-zero entry in each column of Δ_k. Namely, in column $(j, g) \in \mathcal{C}_k$ and row $g x_j^{d-1} \in \mathcal{R}_k$ there is the non-zero entry $d\alpha_j$. Thus, the determinant of $(\Delta_k)\big|_{t=t_0}$ is given by the product of all its non-zero entries, which implies that it is non-zero, concluding the proof. □

In principle, by including any of the numerous methods available for solving linear equations, we are in a position to furnish a routine Decompose, which we formalise in Algorithm 2.

We conclude this section by establishing that the set $\mathcal{B}_1 \cup \cdots \cup \mathcal{B}_n$ is indeed a basis for $H^n_{dR}(U/S)$, as claimed earlier, using the reduction of poles procedure.

Theorem 2.9. Suppose that the family of projective hypersurfaces given by P contains a diagonal fibre. Then the set $\mathcal{B}_1 \cup \cdots \cup \mathcal{B}_n$ defined in Definition 2.4 is a basis for the L-vector space $H^n_{dR}(U/S)$.

Proof. We know that $H^n_{dR}(U/S)$ is spanned by the classes of the n-forms $Q\Omega/P^k$ for all homogeneous polynomials Q of degree $kd - (n+1)$ and $k \in \mathbf{N}$. By a theorem of Macaulay [8, §4, (4.11)], we may assume that $1 \leq k \leq n$, that is to say, any class in $H^n_{dR}(U/S)$ can be represented by an n-form with a pole of order at most n.

Algorithm 2 Obtain co-ordinates for Q in the Jacobian ideal modulo basis elements

Input:
- Q is a homogeneous polynomial of degree $kd-(n+1)$.
- $\partial_0 P, \dots, \partial_n P$ are the partial derivatives of a homogeneous polynomial P in $K[t][x_0, \dots, x_n]$ of degree d, which defines a smooth family of projective hypersurfaces containing a diagonal fibre, where K is a field of characteristic zero.
- B_k is the set of all monomials of total degree $kd-(n+1)$ and partial degree less than $d-1$.

Output: Homogeneous polynomials $Q_0, \dots, Q_n$ such that $Q \equiv Q_0 \partial P_0 + \dots + Q_n \partial_n P$ modulo the $K(t)$-span of B_k.

The multi-sets $\mathcal{R}_k$ and $\mathcal{C}_k$ are as in Definition 2.7, the matrix Δ_k is as in the proof of Theorem 2.8.

procedure DECOMPOSE($Q, \partial_0 P, \dots, \partial_n P, B_k$)

Step I. Let b be the vector of length $|\mathcal{R}_k|$ such that the entry corresponding to the monomial $x^i \in \mathcal{R}_k$ is the coefficient of x^i in Q.

Step II. Let v be the unique vector of length $|\mathcal{C}_k|$ satisfying $\Delta_k v = b$. From the description of $\mathcal{C}_k$ as a disjoint union, we can write v accordingly as $\left(v^{(0)}, \dots, v^{(n)}\right)$ where, for $0 \le j \le n$, $v^{(j)}$ is a vector of length $|\mathcal{C}_k^{(j)}|$.

Step III. For $j = 0, \dots, n$, set $Q_j = \sum_{g \in \mathcal{C}_k^{(j)}} v_g^{(j)} g$, where $v_g^{(j)}$ is the entry in $v^{(j)}$ corresponding to the monomial $g \in \mathcal{C}_k^{(j)}$.

Step IV. **return** $Q_0, \dots, Q_n$

end procedure

Without loss of generality, we may thus start the reduction of poles procedure with a homogeneous polynomial Q of degree $(n+1)d-(n+1)$. Then, since $B_{n+1} = \emptyset$ and $\mathcal{R}_{n+1} = F_{n+1}$, Theorem 2.8 shows that there exist homogeneous polynomials $Q_0, \dots, Q_n$ either zero or homogeneous of degree $nd-(n+1)$ such that $Q = Q_0 \partial_0 P + \dots + Q_n \partial_n P$. Continuing with the reduction of poles procedure as described in Algorithm 1, we obtain an expression for $Q\Omega/P^{n+1}$ as an L-linear combination of elements in $\mathcal{B}_1 \cup \dots \cup \mathcal{B}_n$. This shows that this set spans the vector space $H^n_{dR}(U/S)$.

To see that this set is linearly independent, note that it contains only monomials whose partial degrees are strictly less than $d-1$. However, since P is a homogeneous polynomial of degree d and the family of hypersurfaces contains a diagonal fibre, it follows that, for each $0 \le i \le n$, the partial derivative $\partial_i P$ is a homogeneous polynomial of degree $d-1$ and contains precisely one monomial term with partial degree equal to $d-1$, namely that of the monomial x_i^{d-1}. It follows that the elements of $B_1 \cup \dots \cup B_n$ cannot be reduced further modulo the Jacobian ideal and hence that the set $\mathcal{B}_1 \cup \dots \cup \mathcal{B}_n$ is linearly independent. □

2.3 Sparse matrix techniques

In this section we present a technique for repeatedly solving a sparse system of linear equations as in *Step II.* of Algorithm 2. The methods we present in this section are discussed in great detail in a survey by Reid [18].

Before continuing, let us briefly justify why spare matrix methods are appropriate in this case:

Remark 2.10. Let σ and $\sigma_0, \ldots, \sigma_n$ denote the number of non-zero terms of the polynomials P and $\partial_0 P, \ldots, \partial_n P$, respectively. We observe that the auxiliary matrix Δ_k described in the proof of Theorem 2.8 contains at most σ_j non-zero entries in column (j, g), since the non-zero entries in this column are coefficients of the polynomial $\partial_j P$ corresponding to monomials in $\mathcal{R}_k$ via multiplication by g. In particular, the number of non-zero entries in the matrix is bounded above by $\sigma\,|\mathcal{R}_k|$.

Notation 2.11. In order to simplify the notation to what is relevant in this section, we shall consider the system of linear equations given by

$$Ax = b \tag{2.8}$$

where A is a non-singular $n \times n$ matrix over a field of characteristic zero. Moreover, for later reference we let τ denote the number of non-zero entries in A.

Our first observation is that the reduction of poles procedure from the previous section requires a linear system as in Equation (2.8) to be solved multiple times for the same matrix A but with different column vectors b, which suggests using an approach involving some pre-processing on the matrix A. A popular such approach is the LUP decomposition.

The LUP decomposition of matrix A is a way of performing the classical Gaussian elimination with bookkeeping. It consists of a permutation matrix P, a unit lower-triangular matrix L and an upper triangular matrix U such that $PA = LU$. Once these are known, the original system can be solved in a two step process as

$$Ly = Pb, \quad Ux = y \tag{2.9}$$

which consists of two triangular systems. We briefly note that, using a standard dense approach [2], the LUP decomposition can be computed using a number cubic in n of operations in the base field, and that solving the two triangular systems only requires a number quadratic in n of operations in the base field.

However, assuming that the matrix A is sparse, with further pre-processing a much better performance can be realised. As a first step, not in its own right but as prerequisite to the next step, we permute the rows of the matrix to ensure that the diagonal contains only non-zero entries. We can achieve this here because we assume that the matrix is non-singular. Effectively, this problem is the same as that of computing a perfect matching in the $n \times n$ bipartite graph with vertex sets $\{v_i\}_{i=1}^n$ and $\{w_i\}_{i=1}^n$, where an edge $v_i w_j$ is present if and only the element A_{ij} is non-zero. If we find a perfect matching $\{v_{\pi i} w_i\}_{i=1}^n$ in the form of a permutation $\pi \in S_n$, then, after defining the permutation matrix P_0 via

$$(P_0)_{ij} = \begin{cases} 1 & \text{if } j = \pi i, \\ 0 & \text{otherwise,} \end{cases} \tag{2.10}$$

we conclude that $P_0 A$ has a zero-free diagonal. A detailed discussion of this problem together with an implementation can be found in the work of Duff [4, 3]. We end our discussion here by quoting the worst-case and experimental average-case complexity results as $\mathcal{O}(\tau n)$ and $\mathcal{O}(\tau + n)$, respectively.

The second step is to compute the block-triangularization of the matrix $P_0 A$. We compute a permutation matrix Q_0 such that the matrix $Q_0 P_0 A Q_0^t$ is block-triangular, that is, is of the form

$$Q_0 P_0 A Q_0^t = \begin{pmatrix} A^{(11)} & & & \\ A^{(21)} & A^{(22)} & & \\ \vdots & & \ddots & \\ A^{(N1)} & A^{(N2)} & \dots & A^{(NN)} \end{pmatrix} \tag{2.11}$$

where each diagonal block $A^{(kk)}$ is square and can itself not be symmetrically permuted to block-triangular form. Again an asymptotically optimal algorithm for this problem stems from the realm of graphs: it is Tarjan's linear time algorithm for computing the strongly connected components in a directed graph. In this setting, its asymptotic complexity is given by $\mathcal{O}(n + \tau)$. For further information, we refer to the joint work of Duff and Reid [6, 5].

Finally, this enables us to solve N potentially much smaller systems of linear equations instead of the one we started with. We can rewrite our original system of equations $Ax = b$ as $A'y = c$ where $A' = Q_0 P_0 A Q_0^t$, $y = (Q_0^t)^{-1} x$ and $c = Q_0 P_0 b$ and now use the fact that this system is block-triangular. This allows us to instead solve the sequence of systems of linear equations

$$A^{(kk)} y_k = c_k - \sum_{j=1}^{k-1} A^{(kj)} y_j \tag{2.12}$$

for $k = 1, \dots, N$, where we implicitly think of the column vectors y and c to be divided into N corresponding blocks $y_1, \dots, y_N$ and $c_1, \dots, c_N$. An important consequence of this approach is that after the above two steps of pre-processing, the off-diagonal blocks are not changed further; in particular, the phenomenon of fill-in during Gaussian elimination, which refers to the introduction of new non-zero entries, is limited to the diagonal blocks.

Chapter 3

Computing the Gauss–Manin Connection Matrix

We now describe the action of the Gauss–Manin connection ∇_{GM} on basis elements of $H^n_{dR}(U/S)$. Suppose that we are given a basis element $x^i\Omega/P^k \in \mathcal{B}_k$, where for $1 \leq k \leq n$ the set $\mathcal{B}_k$ is defined as in Definition 2.4. Following the description in Section 1.3, the action of the connection is given by exterior differentiation, that is, differentiation with respect to t. We first compute

$$\frac{d}{dt}\left(\frac{x^i\Omega}{P^k}\right) = \frac{-kx^iP_t\Omega}{P^{k+1}}, \tag{3.1}$$

where $P_t = dP/dt$. The second step is to repeatedly apply the reduction of poles procedure in de Rham cohomology in order to express the n-form above as

$$\frac{d}{dt}\left(\frac{X^i\Omega}{P^k}\right) \equiv \frac{\gamma_{k+1}\Omega}{P^{k+1}} + \cdots + \frac{\gamma_1\Omega}{P} \tag{3.2}$$

where, for $1 \leq i \leq k+1$, γ_i is an element in the $K(t)$-span of B_i.

We formalise this for later reference in Algorithm 3.

Algorithm 3 Computing the Gauss–Manin connection matrix

Input: Homogeneous polynomial P in $K[t][x_0, \ldots, x_n]$ of degree d, which defines a smooth family of projective hypersurfaces containing a diagonal fibre, where K is a field of characteristic zero.

Output:
- Basis $B_1 \cup \cdots \cup B_n$ for $H^n_{dR}(U/S)$.
- Matrix M for the Gauss–Manin connection on $H^n_{dR}(U/S)$ with respect to this basis.

procedure GMConnection(P)

Step I. Compute the partial derivatives $\partial_0 P, \ldots, \partial_n P$ and the exterior derivative dP/dt.

Step II. Compute the basis sets $B_1, \ldots, B_n$. In the following, we use the convenient way to index their union by (i, f), referring to a polynomial $f \in B_i$.

Step III. Compute the auxiliary matrices Δ_k, for $k = \lfloor n/d \rfloor + 1, \ldots, n + 1$, and perform pre-processing on each auxiliary matrix as described in Sections 2.2 and 2.3.

Step IV. For all $(j, g) \in \bigcup_{k=1}^{n} B_k$, let $Q = -jgP_t$ and set $\gamma_1, \ldots, \gamma_n$ to the output of Reduce($P, B_1, \ldots, B_n, Q$). Then, for each $(i, f) \in \bigcup_{k=1}^{n} B_k$, let $M_{(i,f),(j,g)}$ be the coefficient of f in γ_i.

Step V. **return** $B_1 \cup \cdots \cup B_n$, M

end procedure

Chapter 4

Complexity Analysis

In this section we develop a complexity analysis for the computation of the Gauss–Manin connection following our algorithm outlined before. We shall make one major simplifying assumption, namely that field artihmetic in $K(t)$ and integer arithmetic can be realised in constant space and time. As some examples in Chapter 5 show, this assumption can be far from true; however, it is hoped that this analysis nonetheless provides a useful starting point.

4.1 Main analysis

We begin by gathering a few estimates of quantities we shall use later. First, we compute the size of the Gauss–Manin connection matrix itself. From the Hodge diamond, we find that

$$\begin{aligned} \dim_{K(t)} H^n_{dR}(U/S) &= \sum_{k=1}^{n} |B_k| \\ &= (-1)^n \left(n - d \sum_{j=0}^{n-1} \binom{n+1}{j} (-d)^{n-1-j} \right) \\ &= \frac{d-1}{d} \big((d-1)^n - (-1)^n \big). \end{aligned} \tag{4.1}$$

The size of the row index sets $\mathcal{R}_k$, for $k = 2, \dots, n+1$, as well as the graded parts B_k of the basis, for $k = 1, \dots, n$ can be bounded above by $\binom{kd-1}{n}$, since this is the number of monomials in $n+1$ variables of total degree $kd - (n+1)$.

A useful inequality for binomial coefficients in this context is $(\alpha/\beta)^\beta \leq \binom{\alpha}{\beta} \leq (\alpha e/\beta)^\beta$ for integers $0 < \beta \leq \alpha$, where $e = \sum_{m=0}^{\infty} 1/m!$. We thus find, for $k = 1, \dots, n+1$, that $\binom{kd-1}{n}$ is $\mathcal{O}((de)^n)$ and that $\log \binom{kd-1}{n}$ is $\mathcal{O}(n \log d)$.

In the following, we consider the various steps involved in the computation of the connection matrix as outlined in Algorithm 3.

Step I. The computation of the Jacobian ideal $(\partial_0 P, \dots, \partial_n P)$ and the derivative $\partial P/\partial t$ depends, of course, on the exact representation used for multivariate polynomials. But a valid estimate is given by $\mathcal{O}(n\sigma \log \sigma)$ for time and $\mathcal{O}(n\sigma)$ for space, where here and in the following σ denotes the number of terms in P.

Step II. The computation of the basis sets $B_1, \dots, B_n$ can be accomplished in linear time and space, that is, an asymptotic estimate for each is given by $\mathcal{O}(d^n)$.

Step III. The next step is the computation and pre-processing of the auxiliary matrices. We consider each matrix Δ_k separately.

As the number of non-zero entries in each column $(j, g) \in \mathcal{C}_k$ is at most σ, the total number of non-zero entries is bounded by $\sigma\binom{kd-1}{n}$. We thus obtain a space estimate of $\mathcal{O}(\sigma(de)^n)$. In fact, to quickly deal with the overall space complexity, note that all subsequent work with the matrix Δ_k involves at most a constant number of temporary rows, thus giving an overall estimate of $\mathcal{O}(\sigma(de)^n)$.

If we compute the entries in the matrix by firstly ranging over all columns, secondly ranging over all monomial terms in appropriate partial derivative of P and then performing a binary search to find the correct row, a time estimate of $\mathcal{O}\big((de)^n \sigma n \log d\big)$ can be achieved.

The next two stages of pre-processing, ensuring that the matrix has a zero-free diagonal and the block-triangularisation, take time bounded by $\mathcal{O}\big(\sigma(de)^{2n}\big)$. Suppose now that this step results in a partition of $|\mathcal{R}_k|$ as $n_k^{(1)} + \cdots + n_k^{(N_k)}$. Then the subsequent LUP decomposition takes time $\mathcal{O}\big(\big(n_k^{(1)}\big)^3 + \cdots + \big(n_k^{(N_K)}\big)^3\big)$.

Step IV. For each column in the Gauss–Manin connection matrix, we create a polynomial corresponding to the basis element to be reduced. Here, the process of creating the polynomial is irrelevant, and we only need to the consider the reduction process and the extraction of the specific coefficients afterwards.

We first develop a bound for Algorithm 1. For the moment, we ignore the subroutine Decompose but take into account the time it takes to transform a polynomial into a vector and vice versa. For a polynomial of degree $kd - (n+1)$ this takes time $\mathcal{O}\big(|\mathcal{R}_k| \log \binom{kd-1}{n}\big)$, which is $\mathcal{O}((de)^n n \log d)$, and space $\mathcal{O}\big(\binom{kd-1}{n}\big)$, which is $\mathcal{O}((de)^n)$. Checking whether a polynomial of degree $kd-(n+1)$ lies in B_k and if necessary reducing the pole order of the associated element in $H_{dR}^n(U/S)$ by one takes time

$$\mathcal{O}\bigg(n\binom{kd-1}{n}\bigg) + \mathcal{O}\bigg(n\binom{kd-1}{n}\sigma \log\bigg(\binom{kd-1}{n}\sigma\bigg)\sigma\bigg) + \mathcal{O}\bigg(n\binom{kd-1}{n} \log\binom{kd-1}{n}\bigg),$$

which is $\mathcal{O}\big(n^2(de)^n \sigma \log d\big)$. It takes space $\mathcal{O}\big(\sigma\binom{kd-1}{n}\big)$, which is $\mathcal{O}(\sigma(de)^n)$.

Therefore, by using the upper bounds for the case $k = n$ when k ranges through $1, \dots, n$, we obtain the following weak upper bound as an estimate for the total time required by Algorithm 1 throughout the computation of the connection matrix,

$$\mathcal{O}\Big(\sum_{k=1}^{n} |B_k|\, n^3 (de)^n \sigma \log d\Big),$$

which is $\mathcal{O}\big(d^{2n} e^n n^3 \sigma \log d\big)$. Since the space requirements do not hold simultaneously, an estimate for the total space required is just $\mathcal{O}(\sigma (de)^n)$.

We now attend to the subroutine DECOMPOSE. A call to Algorithm 1 with a polynomial of degree $kd - (n+1)$ triggers calls to DECOMPOSE possibly for all values $k, k-1, \ldots, 2$. In particular, the number of calls to DECOMPOSE with parameter k, for $k = 2, \ldots, n+1$, is bounded above by $|B_{k-1}| + |B_k| + \cdots |B_n|$, which we relax to $\mathcal{O}(d^n)$.

Solving an instance of the system of linear equations given by Δ_k takes time $\mathcal{O}\big(\big(n_k^{(1)}\big)^2 + \cdots + \big(n_k^{(N_k)}\big)^2\big)$, and it takes no additional space.

Summarising all steps, the time estimate we arrive at is given by

$$\mathcal{O}\Big(n\sigma (de)^{2n} + d^{2n} e^n n^3 \sigma \log d + d^n \sum_{k=2}^{n+1} \big((n_k^{(1)})^2 + \cdots + (n_k^{(N_k)})^2\big) + n \sum_{k=2}^{n+1} \big((n_k^{(1)})^3 + \cdots + (n_k^{(N_k)})^3\big)\Big)$$

and the total amount of space required can be bounded by $\mathcal{O}(n (de)^n \sigma)$.

In order to simplify this further, we consider two special cases. In the worst case the block-triangularisation of Δ_k, for $k = 2, \ldots, n$, results in one block. In this case, the above estimate simplifies to

$$\mathcal{O}(n^2 (de)^{3n} + n^3 d^{2n} e^n \sigma \log d).$$

In the sparse case, which is much more typical, the block-triangularisation results in a linear number of blocks of constant size. In that case, the estimate we obtain is

$$\mathcal{O}(n (de)^{2n} \sigma + n^3 d^{2n} e^n \sigma \log d).$$

4.2 Multivariate polynomials

For reference, we provide some complexity results for the basic operations in the multivariate polynomial ring $K(t)[x_0, \ldots, x_n]$. The performance, of course, depends on the data structures and algorithms used: we base the following results on a basic representation of multivariate polynomials backed by balanced binary trees with monomials packed into single integers. While more specialised representations exist, for the moment we make the above choice since here the multivariate polynomial arithmetic does not seem to be a principal limiting factor and because this choice is easy to implement with small overhead.

We present the estimates in Table 4.1, where the letters f, g, h denote polynomials, m a monomial and c an element in the base field $K(t)$. Moreover, σ_f, σ_g and σ_h denote the number of non-zero terms in f, g and h, respectively. Finally, the operation $c \leftarrow \textsc{Coeff}(f, m)$ refers to extracting the coefficient c of the monomial m from the polynomial f. Note that the space complexity only considers the amount of space that needs to be allocated for the computation of the operation, that is, it does not include space already allocated holding the input.

Table 4.1: Complexity bounds for operations in $K(t)[x_0, \ldots, x_n]$

Operation	Time	Space
$f \leftarrow g$	$\mathcal{O}(\sigma_f + \sigma_g)$	$\mathcal{O}(\sigma_g)$
$f \leftarrow f \pm g$	$\mathcal{O}(\sigma_g \log(\sigma_f + \sigma_g))$	$\mathcal{O}(\log \sigma_f + \sigma_g)$
$f \leftarrow g \pm h$	$\mathcal{O}(\sigma_f + \sigma_g + \sigma_h \log(\sigma_g + \sigma_h))$	$\mathcal{O}(\sigma_g + \sigma_h)$
$f \leftarrow f \pm cm$	$\mathcal{O}(\log \sigma_f)$	$\mathcal{O}(1)$
$c \leftarrow \textsc{Coeff}(f, m)$	$\mathcal{O}(\sigma_f)$	$\mathcal{O}(1)$
$f \leftarrow cf$	$\mathcal{O}(\sigma_f)$	$\mathcal{O}(\log \sigma_f)$
$f \leftarrow cg$	$\mathcal{O}(\sigma_f + \sigma_g)$	$\mathcal{O}(\sigma_g)$
$f \leftarrow gh$	$\mathcal{O}(\sigma_g \sigma_h)$	$\mathcal{O}(\sigma_g \sigma_h)$
$f \leftarrow f \pm gh$	$\mathcal{O}(\sigma_g \sigma_h \log(\sigma_f + \sigma_g \sigma_h))$	$\mathcal{O}(\sigma_g \sigma_h)$
$f \leftarrow \partial_i g$	$\mathcal{O}(\sigma_g \log \sigma_g)$	$\mathcal{O}(\sigma_g)$
$f \leftarrow \partial g / \partial t$	$\mathcal{O}(\sigma_g \log \sigma_g)$	$\mathcal{O}(\sigma_g)$

Chapter 5

Examples

In this chapter, we present a few numerical examples. For each example, we include a comparison of the run-time and memory usage of a previously existing MAGMA routine[1] by Lauder and our new implementation in C. The MAGMA code was executed on the machine `wolverine` at the Mathematical Institute, comprising eight Dual Core AMD Opteron processors running at 2.2GHz as well as 32GB of memory and running MAGMA version 2.15-12. The new implementation was executed on a personal laptop with two Intel Core 2 Duo processors at 2.26GHz and 2GB of memory. Finally, we note that, for the smallest examples, the comparison of the memory requirements is not fair in the sense that, even without executing user specific code, the MAGMA set-up available to the author required about 7.19MB of memory and therefore only the excess of this should be attributed to Lauder's routines.

To begin with, the following Table 5.1 illustrates the numerical size of the problems we need to consider even for relatively small dimensions and degrees.

Table 5.1: Dimensions of the graded parts of $H^n_{dR}(U/S)$

d		n											
		2			3				4				
3	k	1	2	3	1	2	3	4	1	2	3	4	5
	$\lvert\mathcal{B}_k\rvert$	1	1	0	0	6	0	0	0	5	5	0	0
	rank $\mathcal{F}_k$	1	10	28	0	10	56	165	0	5	70	330	1001
4	k	1	2	3	1	2	3	4	1	2	3	4	5
	$\lvert\mathcal{B}_k\rvert$	3	3	0	1	19	1	0	0	30	30	0	0
	rank $\mathcal{F}_k$	3	21	55	1	35	165	455	0	35	330	1365	3876
5	k	1	2	3	1	2	3	4	1	2	3	4	5
	$\lvert\mathcal{B}_k\rvert$	6	6	0	4	44	4	0	1	101	101	1	0
	rank $\mathcal{F}_k$	6	36	91	4	84	364	969	1	126	1001	3876	10626

[1] Specifically, we used the routines `GriffithsRed-v1.4.m` and `ConnMatrix-v1.4.m`.

5.1 A toy example

We begin by giving many details of the computation in the case of the toy example given by $P(X, Y, Z) = X^3 + Y^3 + Z^3 + tXYZ$, containing the diagonal and only one varying cross-term, which is even symmetric in all variables.

After computing the derivatives, we find that $\ell = 1$ and $u = 2$, and hence compute the basis sets $B_1 = \{1\}$ and $B_2 = \{XYZ\}$.

The auxiliary matrix Δ_2 is computed as

$$\begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & t & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & t & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & t & 0 & 0 & 0 & 0 & 0 \\ 0 & t & 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & t & 3 & 0 & 0 \\ 0 & 0 & t & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \end{pmatrix} \tag{5.1}$$

but in this initial form it still has three zero entries on the diagonal. The permutation P_0 such that $P_0 A$ has a zero-free diagonal is given by the cycle (3 5 4). The subsequent permutation to block-triangular form is given by (2 6 4)(3 7)(5 8). Together they transform Δ_2 to $Q_0 P_0 \Delta_2 Q_0^t$ which is

$$\begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & t & 0 & 0 & 0 & 0 & 0 \\ 0 & t & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & t & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & t & 0 & 0 \\ 0 & 0 & 0 & 0 & t & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & t & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \end{pmatrix}. \tag{5.2}$$

The above matrix is block-triangular with block sizes $1, 3, 3, 1, 1$. The following LUP decomposition performed on the two 3×3 submatrices yields the matrix

$$\begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & t & 0 & 0 & 0 & 0 & 0 \\ 0 & t/3 & 3 & -t^2/3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & t/3 & (t^3+27)/9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & t & 0 & 0 \\ 0 & 0 & 0 & 0 & t/3 & 3 & -t^2/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & t/3 & (t^3+27)/9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \end{pmatrix}. \tag{5.3}$$

We do not include full details for the computation of Δ_3, which is a 28×28 matrix. However, in order to illustrate the usefulness of applying sparse rather then dense matrix techniques, we point out that the corresponding block sizes partition 28 as $19 \times 1 + 3 \times 3$.

In order to compute the Gauss–Manin connection matrix with respect to our choice of basis, we need to reduce the two elements $-XYZ\Omega/P^2$ and $-2X^2Y^2Z^2\Omega/P^3$ corresponding to the basis elements Ω/P and $XYZ\Omega/P^2$, respectively. This gives

$$\nabla_{GM}\Big(\frac{\Omega}{P}\Big) = \frac{-XYZ\Omega}{P}, \tag{5.4}$$

$$\nabla_{GM}\Big(\frac{XYZ\Omega}{P^2}\Big) = \frac{-2X^2Y^2Z^2\Omega}{P^3} \equiv \frac{t/(t^3+27)\Omega}{P} + \frac{-3t^2/(t^3+27)\Omega}{P^2}. \tag{5.5}$$

Finally, the run-time and memory requirements of the Magma routines and the new C implementation are given by 0.0035s and 0.00032s as well as 7.32MB and 136KB, respectively.

5.2 Quartic surfaces

We now consider a sequence of quartic surfaces with an increasing number of cross-terms. This highlights the practical limitations of the previous Magma code and demonstrates the usefulness of the approach investigated in this thesis.

From Table 5.1 we see that $|B_1| = 1$, $|B_2| = 19$, $|B_3| = 1$ and $|\mathcal{R}_2| = 16$, $|\mathcal{R}_3| = 164$, $|\mathcal{R}_4| = 455$. We are particularly interested in how these last three cardinalities are partitioned by the block-triangularisation since this provides a measure of the usefulness of the sparse matrix techniques employed here. For a partition of the matrix Δ_k into N_k diagonal blocks of size $n_1, \dots, n_{N_k}$, we include the two quantities $\alpha_k = |\mathcal{R}_k|^2 / (n_1^2 + \cdots + n_{N_k}^2)$ and $\beta_k = |\mathcal{R}_k|^3 / (n_1^3 + \cdots + n_{N_k}^3)$, rounded to the nearest integer. These two quantities give an indication of the improvement in run-time gained by employing sparse matrix techniques instead of standard dense matrix techniques. The former value pertains to the quadratic time routines, which in particular includes the solving of linear systems as part of the reduction process, whereas the latter is relevant for the LUP decomposition, which in the current implementation requires cubic time.

5.2.1 A quartic surface with one cross-term

We consider the family of surfaces given by

$$W^4 + X^4 + Y^4 + Z^4 + tWXYZ. \tag{5.6}$$

The partitions we obtain are $16 = 16 \times 1$, $164 = 104 \times 1 + 15 \times 4$ and $455 = 391 \times 1 + 16 \times 4$, which yield values $\alpha = (\alpha_k)_{k=1}^3 = (16, 78, 319)$ and $\beta = (\beta_k)_{k=1}^3 = (256, 4145, 66570)$.

As expected from the small number of terms and high level of symmetry, both Lauder's Magma routine and the new implementation compute this quickly in 0.045s and 0.0138s, requiring 7.71MB and 1.35MB of memory, respectively.

We also note that Gerkmann [7, p. 61] mentions a run-time of about 7s in this case.

5.2.2 A quartic surface with three cross-terms

We consider the family of surfaces given by

$$W^4 + X^4 + Y^4 + Z^4 + t(W^3X + WYZ^2 + XZ^3). \tag{5.7}$$

The block sizes give rise to partitions $16 = 16 \times 1$, $164 = 70 \times 1 + 8 \times 4 + 1 \times 62$ and $455 = 253 \times 1 + 33 \times 4 + 1 \times 70$, from which we find $\alpha = (16, 6, 36)$ and $\beta = (256, 18, 273)$.

The time and space requirements for the two routines are 27.88s, 2.013s and 18.84MB, 5.29MB. In particular, the new routine is about 14 times faster.

5.2.3 A quartic surface with four cross-terms

We consider the family of surfaces given by

$$W^4 + X^4 + Y^4 + Z^4 + t(W^3X + WYZ^2 + XZ^3 + WXYZ). \tag{5.8}$$

The partitions we obtain are $16 = 16\times1$, $164 = 53\times1+4\times4+1\times95$ and $455 = 231\times1+29\times4+1\times108$. This gives $\alpha = (16, 3, 17)$ and $\beta = (256, 5, 75)$.

The time and space requirements for the two routines are 4190.610s, 24.97s and 238.38MB, 16.62MB. At this point, the new implementation is already more than 165 times faster.

5.2.4 A quartic surface with six cross-terms

We consider the family of surfaces given by

$$W^4 + X^4 + Y^4 + Z^4 + t(2W^3X + 7W^2XY - 11WX^2Y + 13X^2YZ + 17X^2Z^2 - WXYZ). \tag{5.9}$$

The block-triangularisations result in the following partitions of the ranks of the auxiliary matrices: $16 = 14\times1+1\times2$, $164 = 38\times1+4\times2+4\times4+1\times102$ and $455 = 181\times1+18\times2+22\times4+7\times5+1\times115$. We hence find that $\alpha = (14, 3, 15)$ and $\beta = (186, 4, 62)$.

For this example, Lauder's Magma routine takes 4.52 days and requires 3.57GB of memory. The new implementation completes the computation in 277.02s and uses 66.7MB. That is, the new implementation is more than 1400 times faster.

5.2.5 A quartic surface with seven cross-terms

We consider the family of surfaces given by

$$\begin{aligned}W^4 + X^4 + Y^4 + Z^4 + t(&-3W^3X + 5W^3Y + 7W^2XY \\ &- 23WX^2Y - 29X^2YZ + 31Y^2Z^2 - 37WXYZ) \quad (5.10)\end{aligned}$$

In this case, we obtain the partitions $16 = 14 \times 1 + 1 \times 2$, $164 = 32 \times 1 + 1 \times 4 + 1 \times 128$ and $455 = 152 \times 1 + 15 \times 4 + 3 \times 28 + 1 \times 159$ and values $\alpha = (14, 2, 7)$ and $\beta = (186, 2, 23)$.

The previous Magma routine solved this example in 34.04 days using 12.5GB of memory. In contrast to this, the run-time of the new C implementation has only doubled when compared to the previous example and is 527.28s, with a memory requirement of 126.5MB.

5.3 A quintic surface

We finally give an example which begins to show limitations of the approach discussed in this work:

$$(1-t)(W^5 + X^5 + Y^5 + Z^5) + t\big((WXZ + Y^3)(W^2 + XY + Z^2)\big). \quad (5.11)$$

For this example, the new implementation requires about 190 minutes and 979.5MB. The author did not obtain the corresponding data from Lauder's Magma routine since it was terminated after running for over 34 days and using nearly 5GB of memory.

Chapter 6

Further Improvements

In this chapter, we briefly discuss possible further improvements for the computation of the Gauss–Manin connection matrix based on this work. We present these in an increasing level of abstraction.

Basic operations in the base field. The author has written a fast implementation of the rational function field $\mathbf{Q}(t)$ based on the C library FLINT [10]. While Henrici's algorithms [12] for addition and multiplication in quotient fields lead to a fast implementation of these basic operations, for our application it would be more appropriate to consider the ternary operation $x \leftarrow x + yz$ as the basic operation.

Multivariate polynomial rings. The current implementation of multivariate polynomials is based on very fast monomials packed into single words. On top of these, only the classical sparse algorithms for polynomial addition and multiplication are implemented. The degree and the number of terms of the polynomials we consider are likely to be large enough for the implementation of more advanced algorithms to be beneficial.

Auxiliary matrices. The auxiliary matrices are implemented using a hybrid data structure between sparse and dense matrices. More specifically, while the entries are stored in a dense matrix, we also maintain a sparse data structure for the structural information. Currently, the LUP decomposition and the solving of linear systems is carried out using dense techniques. Since when applying this implementation to curves or surfaces the reduction process typically takes up between 60% and 70% of the run-time, almost all of which is contributed by the solving of linear systems, a purely sparse approach for all but the smallest (sparse) blocks would lead to an improved performance.

Multi-modular techniques. The sizes of the integer coefficients of the rational functions that we consider are large enough to make a multi-modular approach appear interesting. Moreover, since FLINT also contains a fast implementation for the polynomial ring $\mathbf{F}_p[t]$, where p is a prime number, it would be straightforward to modify the existing implementation. However, it remains to obtain a computationally feasible bound on the number and sizes of the primes that we need to incorporate in the reconstruction process via the Chinese remainder theorem.

Parallelisation. We note that the computation and the pre-processing of the auxiliary matrices Δ_k are independent for varying k. Moreover, the reduction of poles of the images $\nabla_{GM}(\omega)$ are independent for varying basis elements $\omega \in \mathcal{B}_1 \cup \cdots \cup \mathcal{B}_n$. When one is interested in computing the connection for a single hypersurface, these above tasks should be handled in parallel. However, when one is interested in computing the connection for many different hypersurfaces as part of a numerical experiment, the parallelisation should be realised with coarser granularity.

References

[1] T.G. Abott, K.S. Kedlaya, and D. Roe, *Bounding Picard numbers of surfaces using p-adic cohomology*, Arithmetic, Geometry, and Coding Theory (AGCT-10), 2006.

[2] T. H. Cormen, C. H. Leiserson, and R. L. Rivest, *Introduction to algorithms*, The MIT Press, Cambridge, Massachusetts, 1990.

[3] I.S. Duff, *Algorithm 575: Permutations for a zero-free diagonal*, ACM Trans. Math. Softw. **7** (1981), no. 3, 387–390.

[4] ______, *On algorithms for obtaining a maximum transversal*, ACM Trans. Math. Softw. **7** (1981), no. 3, 315–330.

[5] I.S. Duff and J.K. Reid, *Algorithm 529: Permutations to block triangular form [F1]*, ACM Trans. Math. Softw. **4** (1978), no. 2, 189–192.

[6] ______, *An implementation of Tarjan's algorithm for the block triangularization of a matrix*, ACM Trans. Math. Softw. **4** (1978), no. 2, 137–147.

[7] R. Gerkmann, *Relative rigid cohomology and deformation of hypersurfaces*, International Mathematics Research Papers **2007** (2007), no. 1.

[8] P. Griffiths, *On the periods of certain rational integrals: I, (resp. II)*, The Annals of Mathematics, Second Series **90** (1969), no. 3, 460–495 (resp. 496–541).

[9] P. Griffiths and J. Harris, *Principles of algebraic geometry*, John Wiley & Sons, 1978.

[10] W.B. Hart, D. Harvey, et al., *Fast library for number theory.*

[11] R. Hartshorne, *Algebraic geometry*, Graduate Texts in Mathematics, vol. 52, Springer, 1977.

[12] P. Henrici, *A subroutine for computations with rational numbers*, J. ACM **3** (1956), no. 1, 6–9.

[13] L. Illusie, *Frobenius and Hodge degeneration*, Introduction to Hodge Theory, vol. 8, SMF/AMS Texts and Monographs, no. 3, American Mathematical Society, Société Mathématique de France, 1996, pp. 96–145.

[14] N.M. Katz and T. Oda, *On the differentiation of de Rham cohomology classes with respect to parameters*, J. Math. Kyoto Univ. **8** (1968), no. 2, 199–213.

[15] K.S. Kedlaya, *p-adic cohomology: from theory to practice*, p-adic Geometry: Lectures from the 2007 Arizona Winter School (D. Savitt and D.S. Thakur, eds.), University Lecture Series, vol. 45, American Mathematical Society, 2008, pp. 175–201.

[16] Q. Liu, *Algebraic geometry and arithmetic curves*, Oxford Graduate Texts in Mathematics, vol. 6, Oxford University Press, Oxford, 2002.

[17] F.S. Macaulay, *The algebraic theory of modular systems*, Cambridge Mathematical Library, Cambridge University Press, 1994.

[18] J.K. Reid, *Solution of linear systems of equations: Direct methods*, Sparse Matrix Techniques (Copenhagen) (V.A. Barker, ed.), vol. 572, Springer-Verlag, 1976.