Dø1

Department
of
PURE MATHEMATICS

19-12-15-81 ISSN 0332-5407

A numerical study of algebraic space curves.

by

Audun Holme

UNIVERSITY OF BERGEN
Bergen, Norway

19-12-15-81 ISSN 0332-5407

A numerical study of algebraic space curves.

by

Audun Holme

Department of Mathematics
University of Bergen
Norway

Table of contents.

§0. Introduction.

The basic and important Classification Problem in projective algebraic geometry is the following:

Let k be a field, say $k = \overline{k}$, and consider the set $V_{N,n}$ of all smooth, projective subvarieties of $\mathbb{P}^N_k = \mathbb{P}^N$ of dimension n. PGL(N+1,k) acts on $V_{N,n}$ in the obvious way, two varieties are said to be projectively equivalent if they belong to the same orbit. The problem then is to classify all varieties in $V_{N,n}$ up to projective equivalence.

Already for $n = 1$ this problem is unsolved. In fact, the lack of information even in this case is demonstrated by the fact that a simple and suggestive question concerning the classification of curves in $\mathbb{P}^3$ has stood open for almost 100 years.

In 1882 two great works on the classification of smooth curves in $\mathbb{P}^3$ appeared, written by G. Halphen [Hal] and M. Noether [No1], [No2]. They had been submitted to the Akademie der Wisseschaften zu Berlin in competition for the Steinersche Preis for 1882, which had been announced for the best treatise on the classification of space curves.

The two authors actually shared the price, having written two extensive and fundamental articles, which were to have a deep and far reaching influence in the years to come.

But the results obtained were in no way complete or definitive. Moreover, Halphen's article is very hard to read, and contains passages which need further justification, see L. Gruson and C. Peskine [G-P].

In particular, Halphen claims a theorem which for a given integer $d \geq 1$ would yield a list of those integers g such that there exists a smooth curve in $\mathbb{P}^3$ of degree d and genus g. Unfortunately, the proof of this assestion contains a gap, and Halphen's assestion thus remains an open conjecture. See R. Hartshorne [Har2] for further comments on this.

It is quite probable that this question will be settled along the lines set out by Halphen, however. In fact, research by Gruson and Peskine, based on a critical reading of Halphen, seems to suggest this.[1)]

1) See note added in proof at the end of this paper.

Also, it should be noted that the classical works cited above yield verifications of the genera lists for all $d \leq 20$.

The basic problem behind the present article is the following: To find a collection of curve generating algorithms which up to projective equivalence yield all smooth, projective curves in $\mathbb{P}^3$ of a given degree d. A solution to this problem would certainly settle the Classification Problem for curves in a complete and very satisfactory way.

Of course a large number of more or less elementary curve generating algorithms are easily constructed. Thus for instance chapters IV, V of [Har1] yield a list of such algorithms. It would be rather surprising if already this list should be sufficient, and as a first step one might want to convince oneself that additional curve generating algorithms are really required. Since we strongly believe in Halphen's conjecture, an obvious approach is to test the conjecture on the output of our elementary curve generating algorithms.

So we select three specific algorithms, which were suggested by some initial experimentation.

Surprisingly, it turns out that these are capable of filling Halphen's conjectural genera lists for all $d \leq 100$. This is done on the UNIVAC 1100 Computer at the University of Bergen, the programs being written in SIMULA-67.

For practical reasons we proceed by first running a main program, which implements two curve generating algorithms. This main program turns out to be capable of filling almost all of the conjectured domain for (d,g). The few remaining gaps are then removed by a secondary program, which implements a certain search-algorithm. But first about half of the remaining gaps may be removed by inspection, utilizing a trivial special case of this final algorithm.

Selecting the list of curve generating algorithms was not immediate. This process involved a certain amount of experimentation, which would have been impossible without the computer. Interesting as this experimental phase was, we will not go into this at all here, but only present the appropriate algorithms as they now stand. Part of the experiments are described in [Ho1], to which the reader is referred. A different path which was attempted, but discarded for this particular project, was to study non singular curves on certain projective surfaces with singularities.

The reason for stopping at d = 100 is basically one of economy. The main program works by going through all possible cases, and in doing so a given g turns out to be realized a large number of times as the genus of non-equivalent curves. Since the CPU time grows rapidly with d, further computations along these lines seem impracticable.[1)] Instead a certain type of stocastic algorithm could be used, particularly because of the "overkill" encountered in the main program referred to above. We might return to this in [Ho2], which constitutes a continuation of the work initiated in this article.

When explicitely giving algorithms, we adhere to the format established by D. Knuth (cf. [Kn]), even though an ALGOL-like notation might have been more transparent in the present cases.

I would like to thank Robin Hartshorne for calling my attention to the methods of §1.2, and Vinjar Wærenskjold as well as Dag-Olav Bjorøy of the University of Bergen for helping me getting started on the EXEC-8 operating system.

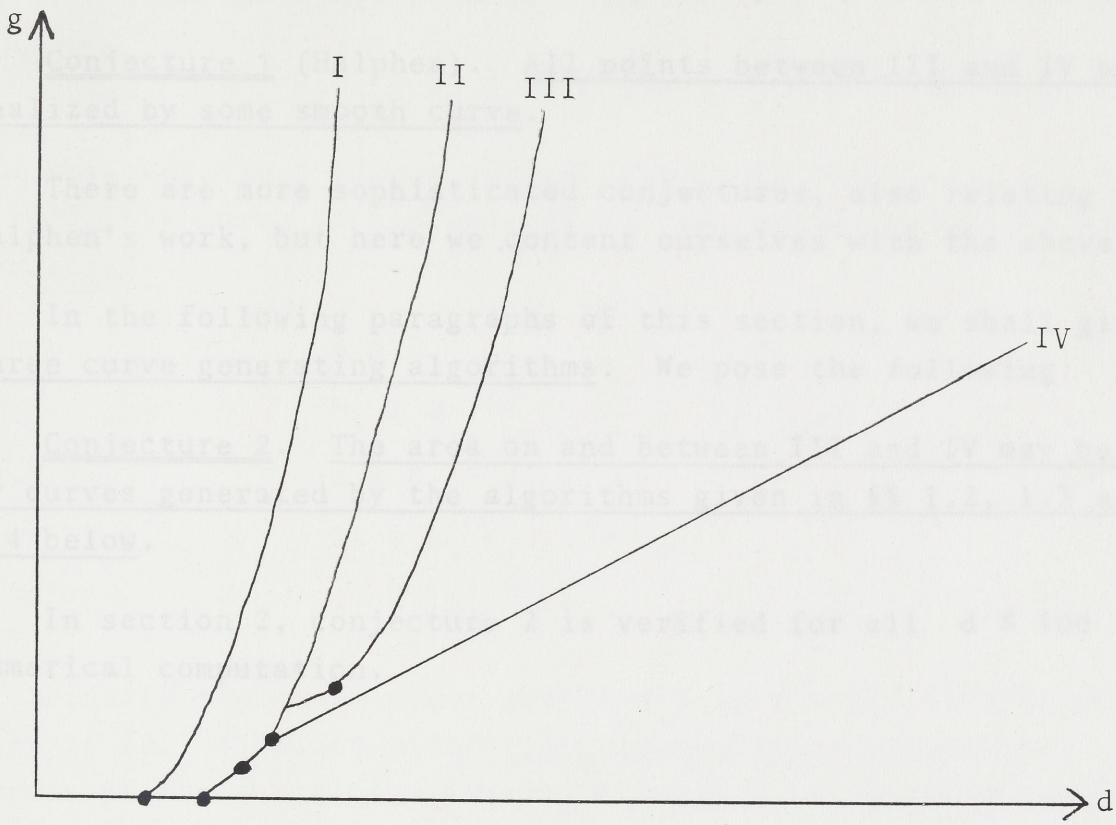
1. Curve generating algorithms.

1.1. Existence of smooth curves in $\mathbb{P}^3$. We consider the following question: Given a pair (d,g) of non negative integers. When does there exist a smooth curve in $\mathbb{P}^3$ of degree d and genus g?

Here $\mathbb{P}^3$ denotes projective 3-space over some algebraically closed field k.

1) See note added at the end of this paper.

The situation may be summarized in the following diagram:



I-IV are given as follows:

I: $g = \frac{1}{2}(d-1)(d-2)$

II: $g = \left[\frac{1}{4}d^2 - d + 1\right]$

III: $g = \left[\frac{1}{6}d(d-3)+1\right]$

IV: $g = d - 3$

Clearly there are no curves to the left of I. On I we find the plane curves, while again there are no curves between I and II. Between II and III, inclusive of the upper limit, there are some curves. They are all contained in the non singular quadric surface, and their (d,g) values are easily listed. On and below IV all points are realized by suitable curves. See [Har1] and [Har2] for details on this, which is actually quite elementary. All curves on III are realized (see below), and the open question thus is:

Conjecture 1 (Halphen). All points between III and IV are realized by some smooth curve.

There are more sophisticated conjectures, also relating to Halphen's work, but here we content ourselves with the above.

In the following paragraphs of this section, we shall give three curve generating algorithms. We pose the following:

Conjecture 2. The area on and between III and IV may be filled by curves generated by the algorithms given in §§ 1.2, 1.3 and 1.4 below.

In section 2, conjecture 2 is verified for all $d \leq 100$ by numerical computation.

1.2. Curves on the non singular cubic surface in $\mathbb{P}^3$. Following [Har1] and [Har2] we first recall below the explicit description of degrees and genera of the non singular curves which lie on the non singular cubic surface.

A non singular cubic surface X in $\mathbb{P}^3$ is isomorphic to $\mathbb{P}^2$ with six points $P_1,\dots,P_6$ blown up. Let ℓ denote the total transform of a line in $\mathbb{P}^2$, and e_i the exceptional divisors corresponding to P_i, $i=1,\dots,6$. Then $\mathrm{Pic}(X)$ is the free abelian group of rank 7 generated by $\ell, e_1,\dots,e_6$.

Let Y be an irreducible curve on X of degree $d \geq 3$. Then one can write the linear equivalence class of Y as

(1.2.1) $$a\ell - \sum_{i=1}^{6} b_i e_i$$

with $a > 0$ and $b_i \geq 0$. The degree d and the genus g of Y is given as follows:

$$d = 3a - \sum b_i$$

(1.2.2)

$$g = \binom{a-1}{2} - \sum\binom{b_i}{2}$$

We normalize the above data by assuming

(1.2.3) $$b_1 \geq b_2 \geq \ldots \geq b_6$$

Moreover, this linear equivalence class contains a non singular irreducible curve if and only if the following conditions are satisfied:

(1.2.4)
$$a > 0$$
$$b_6 \geq 0$$
$$a \geq b_1 + b_2$$
$$2a \geq b_1 + \ldots + b_5$$
$$a^2 > \sum b_i^2$$

Finally the above class contains a very ample divisor if and only if (2.1.4) holds with <u>strict inequalities</u> everywhere. For proofs of the above, see [Har1], Chapter V, Exersise 4.8 and Corollary 4.13.

For a given value of d there is only a finite number of strings $a, b_1, \ldots, b_6$ such that (1.2.2)-(1.2.4) holds. In fact, since

$$2a \geq b_1 + \ldots + b_5 = 3a - d - b_6$$

and

$$b_6 \leq \frac{1}{6}(3a-d)$$

we get

$$2a \geq 3a - d - \frac{1}{6}(3a-d)$$

Hence

$$a \leq \frac{5}{3}d\ .$$

Thus Hartshorne's description above yields a method which in principle makes it possible to compute all values of g for curves on the non singular cubic of a fixed degree d.

To make this into a reasonably efficient algorithm, however, one first of all needs to find the best possible upper bound for a. Here we have the following

<u>Proposition 1.2.5</u>. <u>All possible values of g are attained for $a \leq d$</u>.

<u>Proof</u>. In fact, we show that the data above are symmetrical around $a = d$. So let

$$a = d + \alpha , \quad a \geq \alpha \geq 0$$

Let $b_1, \ldots, b_6$ be such that (1.2.2) holds. Then

$$\sum b_i = d + 3\alpha$$

$$g = \frac{1}{2}\left\{d^2 + (2\alpha - 3)d + \alpha^2 - 3\alpha + 2 - \sum b_i(b_i - 1)\right\}$$

Now put

$$\tilde{a} = d - \alpha , \quad \tilde{b}_i = b_i - \alpha$$

Since

$$2a \geq 3a - d - b_6$$

we have in general that

$$b_6 \geq a - d$$

Hence $b_6 \geq \alpha$ and thus $\tilde{b}_6 \geq 0$. Further, since

$$d + \alpha \geq b_1 + b_2$$

we get

$$\tilde{a} \geq \tilde{b}_1 + \tilde{b}_2$$

Similarly one verifies that

$$2\tilde{a} \geq \tilde{b}_1 + \ldots + \tilde{b}_5$$

Finally we also have

$$\tilde{a}^2 > \sum \tilde{b}_i^2$$

since one verifies that

$$\tilde{a}^2 - \sum \tilde{b}_i^2 = a^2 - \sum b_i^2 > 0 \; .$$

If $\tilde{d}$ and $\tilde{g}$ are defined by (1.2.2) using $\tilde{a}, \tilde{b}_1, \ldots, \tilde{b}_6$, we now get

$$\tilde{d} = 3\tilde{a} - \sum \tilde{b}_i = 3(a-2\alpha) - \sum b_i + 6\alpha = d \; .$$

Observing that

(1.2.6) $$g = \frac{1}{2}\left\{a^2 - \sum b_i^2 - d + 2\right\}$$

we also get

$$\tilde{g} = g$$

by what is already verified above.

Using the above description, one can get some general information on the occurring values of g. For instance, it is easily seen that all (d,g) lie on or below III and that all point on III are realized: If $d = 3\nu$, take

$$b_1 = \text{---} = b_6 = \nu \; , \quad a = 3\nu$$

Then all inequalities required hold, and by (1.2.6)

$$g = \frac{1}{2}\left(9\nu^2 - 6\nu^2 - 3\nu + 2\right)$$

$$= \frac{1}{2}\left(\frac{1}{3}d^2 - d + 2\right) = \frac{1}{6}d(d-3) + 1$$

For $d = 3\nu \pm 1$ the argument is modified in the obvious way.

We are now ready to formulate the first of our three algorithms. This algorithm is the fundamental one, which by itself fills a large portion of the conjectured (d,g)-range. However, it does leave certain gaps, and as d increases the number of gaps grows rapidly. See §2.1. for more on this.

Hartshorne's algorithm (Curves on the non singular cubic in $\mathbb{P}^3$). Given an integer d, we let gen_i, $0 \le i \le \left[\frac{1}{6}d(d-3)\right]+1$ be integers initialized to zero. a and $b_1,\dots,b_6$ are integral variables. The algorithm terminates with $gen_g = 1$ if g is the genus of a curve of degree d on the non singular cubic and $gen_g = 0$ otherwise.

Init a : Set $a:=d$

Init b_1: Set $b_1:=3a-d$

Init b_2: Set $b_2:=\mathrm{Min}\{3a-d-b_1, b_1\}$

H1: Test. If $a<b_1+b_2$ then go to Loop b_2

Init b_3: Set $b_3:=\mathrm{Min}\{3a-d-b_1-b_2, b_2\}$

Init b_4: Set $b_4:=\mathrm{Min}\{3a-d-b_1-b_2-b_3, b_3\}$

Init b_5: Set $b_5:=\mathrm{Min}\{3a-d-b_1-b_2-b_3-b_4, b_4\}$

H1′: Test. If $2a<b_1+\dots+b_5$ then go to Loop b_5

H2: Set b_6 and test. Set $b_6:=3a-d-b_1-\dots-b_5$. If $b_6>b_5$ or $a^2\le b_1^2+\dots+b_6^2$ then go to Loop b_5

H3: Set gen_g. Set $g:=\frac{1}{2}\Big((a-1)(a-2)-\sum b_i(b_i-1)\Big)$. Set $gen_g:=1$.

Loop b_5: If $b_5>0$, then $b_5:=b_5-1$. Go to H1.

Loop b_4: If $b_4>0$, then $b_4:=b_4-1$. Go to Init b_5

Loop b_3: If $b_3>0$, then $b_3:=b_3-1$. Go to Init b_4

Loop b_2: If $b_2>0$, then $b_2:=b_2-1$. Go to Init b_3

Loop b_1: If $b_1>0$, then $b_1:=b_1-1$. Go to Init b_2

Loop a : If $a>0$, then $a:=a-1$. Go to Init b_1

E: End. The algorithm terminates.

1.3. Type (a,b)-curves. In this section we describe a class of curves which is generated in an analogous manner to the curves on the non singular quadric surface in $\mathbb{P}^3$. In fact, we have the following observation:

Theorem 1.3.1. Let d_i, g_i, $i=1,2$ be integers such that there exist non singular curves X_i in $\mathbb{P}^3$ of degree d_i and genus g_i for $i=1,2$. Let

$$a_i \ge \begin{cases} g_i+3 & \text{if } g_i \ge 2 \\ 3 & \text{if } g_i = 1 \\ 1 & \text{if } g_i = 0 \end{cases}$$

<u>Then there exists a non singular irreducible curve in</u> P^3 <u>of degree</u>

$$d = d_2 a_1 + d_1 a_2$$

and genus

$$g = g_1 + g_2 - g_1 g_2 + (a_1+g_1-1)(a_2+g_2-1) \ .$$

<u>Proof</u>. We need the following result, which is due to G. Halphen. For a proof, one may consult [Har1] Chapter IV §6.

<u>Lemma</u>. <u>A curve</u> X <u>of genus</u> $g \geq 2$ <u>has a non-special very ample divisor of degree</u> d <u>if and only if</u> $d \geq g + 3$.

<u>Remark</u>. By this lemma, all points on and below IV in §1.1 are realized as the (d,g) of some non-singular curve in P^3.

In particular it follows that there exists a very ample divisor D_i on X_i of degree a_i. Letting

$$Q = X_1 \times X_2$$

we get that

$$O_Q(D_1,D_2) = pr_1^* O_{X_1}(D_1) \otimes pr_2^* O_{X_2}(D_2)$$

is a very ample sheaf on Q. Hence the corresponding linear equivalence class of divisors contains non singular, irreducible curves by Bertini's Theorem. Let Y denote one of these. Then $J_Y = O_Q(-Y) = O_Q(-D_1,-D_2)$ is the Ideal sheaf of Y on Q. We thus have the exact sequence

$$0 \to O_Q(-D_1,-D_2) \to O_Q \to O_Y \to 0$$

The long exact cohomology sequence becomes

$$\begin{aligned} 0 &\to H^0(O_Q(-D_1,-D_2),Q) \to H^0(O_Q,Q) \to H(Y,O_Y) \\ &\to H^1(O_Q(-D_1,-D_2),Q) \to H^1(O_Q,Q) \to H^1(Y,O_Y) \\ &\to H^2(O_Q(-D_1,-D_2),Q) \to H^2(O_Q,Q) \to 0 \end{aligned}$$

By the Künneth-formula we get

$$H^0(O_Q(-D_1,-D_2),Q) = H^0(O_{X_1}(-D_1)) \otimes H^0(O_{X_2}(-D_2)) = 0$$

and similarly

$$H^1(O_Q(-D_1,-D_2),Q) = H^0(O_{X_2}(-D_1),X_1) \otimes H^1(O_{X_2}(-D_2),X_2) \ .$$

$$\oplus H^1(O_{X_1}(-D_1),X_1) \otimes H^0(O_{X_2}(-D_2),X_2) = 0$$

while

$$H^2(O_Q(-D_1,-D_2),Q)) \simeq H^1(O_{X_1}(-D_1),X_1)\otimes H^1(O_{X_2}(-D_2),X_2)$$

We also get, in the same way

$$H^1(O_Q) \simeq H^1(O_{X_1})\oplus H^1(O_{X_2})$$

$$H^2(O_Q) \simeq H^1(O_{X_1})\otimes H^1(O_{X_2})$$

By Riemann-Roch's theorem for the curves X_1 and X_2 we get

$$\dim H^0(O_{X_i}(-D_i),X_i) - \dim H^1(O_{X_i}(-D_i),X_i) = -a_i - g_i + 1$$

so that

$$\dim H^1(O_{X_i}(-D_i),X_i) = a_i + g_i - 1$$

Hence

$$\dim H^2(O_Q(-D_1,-D_2),Q) = (a_1+g_1-1)(a_2+g_2-1)$$

Moreover,

$$\dim H^1(O_Q,Q) = g_1 + g_2$$

$$\dim H^2(O_Q,Q) = g_1 g_2$$

Thus the long exact cohomology sequence yields

$$g_1 + g_2 - g(Y) + (a_1+g_1-1)(a_2+g_2-1) - g_1 g_2 = 0$$

so that the claimed formula for $g(Y)$ follows.

In order to show the formula for d, note that the projective embedding of Q corresponding to the linear system $[Y]$ is the composition

$$X_1 \times X_2 \hookrightarrow \mathbb{P}^3 \times \mathbb{P}^3 \hookrightarrow \mathbb{P}^{15}$$

where the first embedding is the product of the two given embeddings, and the last is the Segre-embedding. Let s,t,τ denote, respectively, the pullbacks of the hyperplane classes of $\mathbb{P}^3$ via the first and the second projection, and the hyperplane class of $\mathbb{P}^{15}$. Then we have

$$[Y] = [D_1 \times X_2] + [X_1 \times D_2] \in A(\mathbb{P}^3 \times \mathbb{P}^3)$$

so that

$$[Y] = d_2 a_1 s^3 t^2 + d_1 a_2 s^2 t^3$$

Since s^3t^2 and s^2t^3 are both mapped to τ^{14}, we are done.

Remark. Of course we may assume that

$$d - 1 \geq d_1 \geq d_2$$

The last inequality is by symmetry, and the first follows from the formula for d.

Curves of the above type will be referred to as type (a,b)-curves. The type (a,b)-curves form a more sparse family than those generated by Hartshorne's algorithm, but is still sufficiently rich not only to fill all but a handful of the gaps left by Hartshorne's algorithm, but also to realistically test some of the conjectures related to general postulation problems for curves in $\mathbb{P}^3$. We hope to return to this in a forthcoming paper [Ho 2].

The algorithm generating these curves is given below.

Algorithm for type (a,b)-curves. Given an integer d, we let gen_g, $0 \leq g \leq \frac{1}{2}(d-1)(d-2)$ be integers initialized to zero. $aa_i := i$, $ab_i := d-i$ for $1 \leq i \leq d$ correspond to $d_2 a_1$ and $d_1 a_2$ above. g_1, g_2, j and k are integers. The integral variable $Minab_i$ is defined for $0 \leq i \leq \frac{1}{2}(d-2)(d-3)$, and set as follows: $Minab_0 := 1$, $Minab_1 := 3$, $Minab_i := i+3$ for $i \geq 2$. The boolean variable $Truegenus_{i,j}$ is defined for $1 \leq i \leq d-1$, $0 \leq j \leq \frac{1}{2}(i-1)(i-2)$. It is set to TRUE if j is known to be the genus of some smooth curve in $\mathbb{P}^3$ of degree i, and to FALSE otherwise. The algorithm terminates with $gen_g = 1$ if (d,g) is realized by a curve of type (a,b), and $gen_g = 0$ otherwise.

If, as in this case, the purpose is to verify Halphen's conjecture, then we may set Truegenus under the assumption that the conjecture holds up to $d-1$. On the other hand it is more convenient in practice to content oneself with fewer cases where Truegenus is set to True, as we shall see in §2.2. Also, in the actual computer programs one should limit the use of Truegenus, thereby speeding up the execution of the program.

Init j : Set $j:=1$.
Init i : Set $i:=1$.
Init k : Set $k:=1$.
AB 1. Test. If aa_k/i or ab_k/j is not an integer, then go to Loop k.
Init g_1: Set $g_1:=0$.
Init g_2: Set $g_2:=0$.
AB 2. Test and compute g. With notation as in the theorem, we now have:

$$d_2=i,\ a_1=aa_k/i$$
$$d_1=j,\ a_2=ab_k/j$$

If $\text{Truegenus}_{j,g_1}$ and $\text{Truegenus}_{i,g_2}$ are true, and if

$$aa_k/i \geq \text{Minab}_{g_1}$$
$$ab_k/j \geq \text{Minab}_{g_2}$$

then set

$$g:=g_1+g_2-g_1g_2+(aa_k/i+g_1-1)(ab_k/j+g_2-1)$$
$$\text{gen}_g:=1$$

Loop g_2: If $g_2<\frac{1}{2}(i-1)(i-2)$ then $g_2:=g_2+1$. Go to AB 2.
Loop g_1: If $g_1<\frac{1}{2}(j-1)(j-2)$ then $g_1:=g_1+1$. Go to Init g_2.
Loop k : If $k<d$ then $k:=k+1$. Go to AB 1.
Loop i : If $i<j$ then $i:=i+1$. Go to Init k.
Loop j : If $j<d-1$ then $j:=j+1$. Go to Init i.
E: End. The algorithm terminates.

1.4. Curves on re-embeddings of the cubic surface.
In §1.2 all the very ample linear systems of the non singular cubic surface is described. By a trivial modification of Hartshorne's algorithm we may generate the parameters $(\alpha, \beta_1,\ldots,\beta_6)$ of these, and we find in particular that the surface may be re-embedded into $\mathbb{P}^5$ as a surface of degree 5 in 3 ways, and one of degree 6 in 6 ways, 7 in 12 ways, 8 in 18 ways, 9 in 39 ways and 10 in 54 ways. See the table at the end of this paper for the parameters of those embeddings.

Clearly this large number of possible re-embeddings places at our disposal a powerful tool for the generation of curves in $\mathbb{P}^3$.

In fact, suppose that the linear system is given by the parameters

$$\alpha,\ \beta_1,\ldots,\beta_6$$

and a curve on the cubic surface by

$$a,\ b_1,\ldots,b_6$$

Then by the new embedding it gets the degree

$$d = \alpha a - \Sigma\beta_i b_i$$

and since it may be projected isomorphically onto a curve in $\mathbb{P}^3$, we obtain the following:

Theorem 1.4.1. With the above notation, assume that the parameters satisfy all the requirements of §1.2. Let σ be a permutation of $\{1,2,\ldots,6\}$. Let

$$g = \tfrac{1}{2}((a-1)(a-2) - \Sigma b_i(b_i-1))$$

$$d = \alpha a - \Sigma\beta_{\sigma(i)} b_i$$

Then there exists a smooth curve in $\mathbb{P}^3$ of degree d and genus g.

We immediately observe that the gap-filling procedure of twisting some curve on the cubic surface, which a priori appears as an ad hoc method, is a special case of this. Namely, take

$$(\alpha,\ \beta_1,\ldots,\beta_6) = (3\nu,\ \nu,\ldots,\nu)\ .$$

The algorithm which returns all curves of this type for some given d will not be made explicit here. The strength of the theorem is in no way fully utilized in the computations of the next section, but we will return to this class of curves in the forthcoming case [Ho2]. For the time being, we have only used the theorem to device a simple search procedure to fill the gaps remaining after the algorithms in §§1.2, 1.3; see §2.3.

§2. The computations.

2.1. Implementing Hartshorne's algorithm. We now determine all possible values of degree d and genus g for non singular curves on the non singular cubic surface in $\mathbb{P}^3$.

The following SIMULA program uses Hartshorne's algorithm described in section 1.2. It takes as input integers $min \le max$, and evaluates all possible values of g for $min \le d \le max$.

The program gives as output all gaps in the g-sequence for the various values of d.

GAPS

```
BEGIN INTEGER MIN,MAX,GMAX,GGMAX,D,A,G,S1,S2,
B1,B2,B3,B4,B5,B6,BM2,BM3,BM4,BM5$
REF(OUTFILE) NAVN$
NAVN :- NEW OUTFILE(@TABELL@)$
NAVN.OPEN(BLANKS(70))$
  MIN := ININT$ MAX := ININT$
  GMAX := MAX*(MAX-3)/6 + 1$
  BEGIN INTEGER ARRAY GAPS(0:GMAX)$
   NAVN.OUTTEXT(@        CURVES ON THE NON SING. CUBIC SURFACE:@)$
      NAVN.OUTIMAGE$
  FOR D := MIN STEP 1 UNTIL MAX DO
   BEGIN NAVN.OUTIMAGE$ NAVN.OUTTEXT(@D = @)$ NAVN.OUTINT(D,5)$
   GGMAX := D*(D-3)/6$
   IF MOD(D,3) =  0 THEN GGMAX := GGMAX + 1$
   NAVN.OUTTEXT(@      GMAX = @)$
   NAVN.OUTINT(GGMAX,5)$ NAVN.OUTTEXT(@      GAPS:@)$
   FOR A := D      STEP -1 UNTIL 0 DO
     BEGIN
       FOR B1 := 3*A - D   STEP -1 UNTIL 0 DO
       BEGIN BM2 := IF  B1 GT 3*A-D-B1 THEN
                      3*A-D-B1 ELSE B1$
         FOR B2 := BM2 STEP -1 UNTIL 0 DO
         BEGIN IF A LT B1+B2 THEN GOTO OUT2$
           BM3 := IF B2 GT 3*A-D-B1-B2 THEN
                        3*A-D-B1-B2 ELSE B2$
           FOR B3:= BM3 STEP -1 UNTIL 0 DO
           BEGIN BM4 := IF B3 GT 3*A-D-B1-B2-B3 THEN
                              3*A-D-B1-B2-B3 ELSE B3$
             FOR B4 := BM4 STEP -1 UNTIL 0 DO
             BEGIN BM5 := IF B4 GT 3*A-D-B1-B2-B3-B4 THEN
                                  3*A-D-B1-B2-B3-B4 ELSE B4$
               FOR B5 := BM5 STEP -1 UNTIL 0 DO
               BEGIN S1 := B1+B2+B3+B4+B5$
                 IF 2*A LT S1 THEN GOTO OUT5$
                 B6 := 3*A - D - S1$
                 IF B6 GT B5 THEN GOTO OUT5$
                 IF B6 LT 0 THEN GOTO OUT5$
                 S2 := B1**2+B2**2+B3**2+B4**2+
                             B5**2+B6**2$
                 IF A**2 LE S2 THEN GOTO OUT5$
                 G := ((A-1)*(A-2)+(S1+B6)-S2)/2$
                 GAPS(G) := -1$
OUT5:          END$
             END$
           END$
OUT2:    END$
       END$
     END$
    FOR A := 1 STEP 1 UNTIL GGMAX  DO
    BEGIN
    IF GAPS(A) = 0 THEN NAVN.OUTINT(A,5)$
    GAPS(A) := 0$
    END$ GAPS(D*(D-3)/6 + 1) := 0$
    END$
  END$
  NAVN.CLOSE$
END$
```

The program produced the following output;

```
   CURVES ON THE NON SING. CUBIC SURFACE:

D =    1     GMAX =    0     GAPS:
D =    2     GMAX =    0     GAPS:
D =    3     GMAX =    1     GAPS:
D =    4     GMAX =    1     GAPS:
D =    5     GMAX =    2     GAPS:
D =    6     GMAX =    4     GAPS:
D =    7     GMAX =    5     GAPS:
D =    8     GMAX =    7     GAPS:
D =    9     GMAX =   10     GAPS:
D =   10     GMAX =   12     GAPS:   1
D =   11     GMAX =   15     GAPS:   1   2
D =   12     GMAX =   19     GAPS:   1   2
D =   13     GMAX =   22     GAPS:   1   2   3
D =   14     GMAX =   26     GAPS:   1   2   3
D =   15     GMAX =   31     GAPS:   1   2   3   4
D =   16     GMAX =   35     GAPS:   1   2   3   4
D =   17     GMAX =   40     GAPS:   1   2   3   4   5
D =   18     GMAX =   46     GAPS:   1   2   3   4   5   9
D =   19     GMAX =   51     GAPS:   1   2   3   4   5   6
  10
D =   20     GMAX =   57     GAPS:   1   2   3   4   5   6
  10  11
D =   21     GMAX =   64     GAPS:   1   2   3   4   5   6
   7  11  12  19
D =   22     GMAX =   70     GAPS:   1   2   3   4   5   6
   7  11  12  13
D =   23     GMAX =   77     GAPS:   1   2   3   4   5   6
   7   8  12  13  14
D =   24     GMAX =   83     GAPS:   1   2   3   4   5   6
   7   8  12  13  14  15  22
D =   25     GMAX =   92     GAPS:   1   2   3   4   5   6
   7   8   9  13  14  15  16  23  25
D =   26     GMAX =  100     GAPS:   1   2   3   4   5   6
   7   8   9  13  14  15  16  17  25
D =   27     GMAX =  109     GAPS:   1   2   3   4   5   6
   7   8   9  10  14  15  16  17  18  25  26
D =   28     GMAX =  117     GAPS:   1   2   3   4   5   6

   7   8   9  10  14  15  16  17  18  19  26  28
D =   29     GMAX =  126     GAPS:   1   2   3   4   5   6
   7   8   9  10  11  15  16  17  18  19  20  27  28  29
  31
D =   30     GMAX =  136     GAPS:   1   2   3   4   5   6
   7   8   9  10  11  15  16  17  18  19  20  21  28  29
  31
D =   31     GMAX =  145     GAPS:   1   2   3   4   5   6
   7   8   9  10  11  12  16  17  18  19  20  21  22  29
  30  31  32  34
D =   32     GMAX =  155     GAPS:   1   2   3   4   5   6
   7   8   9  10  11  12  16  17  18  19  20  21  22  23
  30  31  32  34
D =   33     GMAX =  166     GAPS:   1   2   3   4   5   6
   7   8   9  10  11  12  13  17  18  19  20  21  22  23
  24  31  32  33  34  35  37
D =   34     GMAX =  176     GAPS:   1   2   3   4   5   6
   7   8   9  10  11  12  13  17  18  19  20  21  22  23
  24  25  32  33  34  35  37
```

```
D =    35     GMAX =   187     GAPS:    1    2    3    4    5    6
   7    8    9   10   11   12   13   14   18   19   20   21   22   23
  24   25   26   33   34   35   36   37   38   40
D =    36     GMAX =   199     GAPS:    1    2    3    4    5    6
   7    8    9   10   11   12   13   14   18   19   20   21   22   23
  24   25   26   27   34   35   36   37   38   40
D =    37     GMAX =   210     GAPS:    1    2    3    4    5    6
   7    8    9   10   11   12   13   14   15   19   20   21   22   23
  24   25   26   27   28   35   36   37   38   39   40   41   43   51
D =    38     GMAX =   222     GAPS:    1    2    3    4    5    6
   7    8    9   10   11   12   13   14   15   19   20   21   22   23
  24   25   26   27   28   29   36   37   38   39   40   41   43   53
D =    39     GMAX =   255     GAPS:    1    2    3    4    5    6
   7    8    9   10   11   12   13   14   15   16   20   21   22   23
  24   25   26   27   28   29   30   37   38   39   40   41   42   43
  44   46   55
D =    40     GMAX =   247     GAPS:    1    2    3    4    5    6
   7    8    9   10   11   12   13   14   15   16   20   21   22   23
  24   25   26   27   28   29   30   31   38   39   40   41   42   43
  44   46   57
D =    41     GMAX =   260     GAPS:    1    2    3    4    5    6
   7    8    9   10   11   12   13   14   15   16   17   21   22   23
  24   25   26   27   28   29   30   31   32   39   40   41   42   43
  44   45   46   47   49   57   59
D =    42     GMAX =   274     GAPS:    1    2    3    4    5    6
   7    8    9   10   11   12   13   14   15   16   17   21   22   23
  24   25   26   27   28   29   30   31   32   33   40   41   42   43
  44   45   46   47   49   59   61   76
D =    43     GMAX =   287     GAPS:    1    2    3    4    5    6
   7    8    9   10   11   12   13   14   15   16   17   18   22   23
  24   25   26   27   28   29   30   31   32   33   34   41   42   43
  44   45   46   47   48   49   50   52   60   61   63   78
D =    44     GMAX =   301     GAPS:    1    2    3    4    5    6
   7    8    9   10   11   12   13   14   15   16   17   18   22   23
  24   25   26   27   28   29   30   31   32   33   34   35   42   43
  44   45   46   47   48   49   50   52   62   63   65   81
D =    45     GMAX =   316     GAPS:    1    2    3    4    5    6
   7    8    9   10   11   12   13   14   15   16   17   18   19   23
  24   25   26   27   28   29   30   31   32   33   34   35   36   43
  44   45   46   47   48   49   50   51   52   53   55   63   64   65
  67   83
D =    46     GMAX =   330     GAPS:    1    2    3    4    5    6
   7    8    9   10   11   12   13   14   15   16   17   18   19   23
  24   25   26   27   28   29   30   31   32   33   34   35   36   37
  44   45   46   47   48   49   50   51   52   53   55   65   66   67
  69   86
D =    47     GMAX =   345     GAPS:    1    2    3    4    5    6
   7    8    9   10   11   12   13   14   15   16   17   18   19   20
  24   25   26   27   28   29   30   31   32   33   34   35   36   37
  38   45   46   47   48   49   50   51   52   53   54   55   56   58
  66   67   68   69   71   86   88
D =    48     GMAX =   361     GAPS:    1    2    3    4    5    6
   7    8    9   10   11   12   13   14   15   16   17   18   19   20
  24   25   26   27   28   29   30   31   32   33   34   35   36   37
  38   39   46   47   48   49   50   51   52   53   54   55   56   58
  68   69   70   71   73   88   91
D =    49     GMAX =   376     GAPS:    1    2    3    4    5    6
   7    8    9   10   11   12   13   14   15   16   17   18   19   20
  21   25   26   27   28   29   30   31   32   33   34   35   36   37
  38   39   40   47   48   49   50   51   52   53   54   55   56   57
  58   59   61   69   70   71   72   73   75   91   93
D =    50     GMAX =   392     GAPS:    1    2    3    4    5    6
   7    8    9   10   11   12   13   14   15   16   17   18   19   20
  21   25   26   27   28   29   30   31   32   33   34   35   36   37
  38   39   40   41   48   49   50   51   52   53   54   55   56   57
  58   59   61   71   72   73   74   75   77   92   93   96
```

One notes that $gmax = \left[\frac{1}{6}d(d-3)\right] + 1$ is printed out.

Looking at $d = 28$ in the table above, we find Hartshorne's estimate in [Har2], page 12 confirmed. Hartshorne also conjecture that in general all gaps occur for $g \leqslant g_0(d) \approx \frac{1}{3}d^{3/2}$. For $1 \leqslant d \leqslant 51$ we get the following table for $\left[\frac{1}{3}d^{3/2}\right]$:

d	1	2	3	4	5	6	7	8	9	10	11	12
$\left[\frac{1}{3}d^{3/2}\right]$	0	0	1	2	3	4	6	7	9	10	12	13

13	14	15	16	17	18	19	20	21	22	23	24	25
15	17	19	21	23	25	27	29	32	34	36	39	41

26	27	28	29	30	31	32	33	34	35	36	37	38
44	46	49	52	54	57	60	63	66	69	72	75	78

39	40	41	42	43	44	45	46	47	48	49	50	51
81	84	87	90	93	97	100	103	107	110	114	117	121

Thus Hartshorne's conjecture holds for these values of d .

2.2. The Main Program. The program given below finds curves in the area covered by the conjecture as follows:

Integers $min \leqslant max$ are given as input, and d runs from min to max . For fixed d , Hartshorne's algorithm is first used with a=d, d-1. Then some type (a,b)-curves not on the quadric surface are generated. The idea here is to keep the values of g_1, and g_2 relatively small, since experimental evidence have suggested that it is such values which contribute most in filling the gaps left by Hartshorne's algorithm.

Indeed, it turns out that as d approaches 100, only very occa- tional gaps are left unfilled in this way.

It is clear from the output given in §2.2, that Halphen's conjecture holds for all $d \leqslant 24$. This is built into the program.

GAPS

```
BEGIN INTEGER MIN,MAX,GMIN,GMAX,D,A,G,S1,S2,S,B,I,J,K,G1,G2,
B1,B2,B3,B4,B5,B6,BM2,BM3,BM4,BM5$BOOLEAN FOUND$
BOOLEAN ARRAY TRUEGENUS(1:6,0:10)$
  MIN := ININT$ MAX := ININT$
  FOR I := 1 STEP 1 UNTIL 6 DO
    TRUEGENUS(I,0) := TRUE$
  FOR I := 3 STEP 1 UNTIL 6 DO
    TRUEGENUS(I,1) := TRUE$
  TRUEGENUS(5,2) := TRUEGENUS(6,2) :=
  TRUEGENUS(6,3) := TRUEGENUS(6,4) :=
  TRUEGENUS(6,10) := TRUEGENUS(5,6) := TRUE$
  IF MIN LT 24 THEN MIN := 24$
  FOR D := MIN STEP 1 UNTIL MAX DO
  BEGIN GMAX := D*(D-3)/6$
    IF MOD(D,3) = 0 THEN GMAX := GMAX + 1$
      GMIN := D-2$
    BEGIN INTEGER ARRAY GAPS(GMIN:GMAX),MINAB(0:GMAX),
      AA,BA,MAXG(1:D)$
      FOR A := D     STEP -1 UNTIL D-1 DO
        BEGIN
          FOR B1 := A  STEP -1 UNTIL 0 DO
          BEGIN BM2 := IF B1 GT 3*A-D-B1 THEN
                        3*A-D-B1 ELSE B1$
            FOR B2 := BM2 STEP -1 UNTIL 0 DO
            BEGIN IF A LT B1+B2 THEN GOTO OUT2$
              BM3 := IF B2 GT 3*A-D-B1-B2 THEN
                         3*A-D-B1-B2 ELSE B2$
              FOR B3 := BM3 STEP -1 UNTIL 0 DO
              BEGIN BM4 := IF B3 GT 3*A-D-B1-B2-B3 THEN
                              3*A-D-B1-B2-B3 ELSE B3$
                FOR B4 := BM4 STEP -1 UNTIL 0 DO
                BEGIN BM5 := IF B4 GT 3*A-D-B1-B2-B3-B4 THEN
                                3*A-D-B1-B2-B3-B4 ELSE B4$
                  FOR B5 := BM5 STEP -1 UNTIL 0 DO
                  BEGIN S1 := B1+B2+B3+B4+B5$
                    IF 2*A LT S1 THEN GOTO OUT5$
                    B6 := 3*A - D - S1$
                    IF B6 GT B5 THEN GOTO OUT5$
                    IF B6 LT 0 THEN GOTO OUT5$
                    S2 := B1**2+B2**2+B3**2+B4**2+
                          B5**2+B6**2$
                    IF A**2 LE S2 THEN GOTO OUT5$
                    G := ((A-1)*(A-2)+(S1+B6)-S2)/2$
                    IF G LT GMIN THEN GOTO OUT $
                    GAPS(G) := -1$
OUT5$             END$
                END$
              END$
OUT5$       END$
          END$
        END$
```

```
      FOR I := 1 STEP 1 UNTIL D DO
      BEGIN IF I LE 24 THEN MAXG(I) := I*(I-3)/6$
        IF MOD(I,3) = 0 THEN MAXG(I) := MAXG(I)+1$
      IF I GT 24 THEN MAXG(I) := I-3$
      END$
        FOR I := 1 STEP 1 UNTIL 6 DO
          MAXG(I) := 10$
      MAXG(7) := 6$MAXG(8) := 9$
      MAXG(14) := 27$ MAXG(19) := 52$
        MAXG(21) := 64$ MAXG(24) := 85$
      FOR I := 2 STEP 1 UNTIL GMAX DO
        MINAB(I) := I+3$
      MINAB(0) := 1$ MINAB(1) := 3$
      FOR I := 2 STEP 1 UNTIL D  DO
      BEGIN AA(I) := I$
        BA(I) := D-I$
      END$
      FOR I := 2 STEP 1 UNTIL D -1 DO
      BEGIN FOR J := 1 STEP 1 UNTIL I DO
        BEGIN FOR K :=1 STEP 1 UNTIL D  DO
          BEGIN IF MOD(AA(K),I) NE 0 THEN GOTO OUT$
            IF MOD(BA(K),J) NE 0 THEN GOTO OUT$
            FOR G1 := 0 STEP 1 UNTIL MAXG(J) DO
              BEGIN FOR G2 := 0 STEP 1 UNTIL MAXG(I) DO
                BEGIN G := G1+G2-G1*G2+(AA(K)/I+G1-1)*
                                (BA(K)/J+G2-1)$
                  IF G LT GMIN  THEN GOTO OUUT$
                  IF G GT GMAX THEN GOTO OUUT$
          IF AA(K)/I LT MINAB(G1) THEN GOTO OUUT$
          IF BA(K)/J LT MINAB(G2) THEN GOTO OUUT$
                  IF I LE 6 THEN
                  BEGIN
                    IF NOT TRUEGENUS(J,G1) THEN GOTO OUUT$
                    IF NOT TRUEGENUS(I,G2) THEN GOTO OUUT$
                  END$
                  GAPS(G) := -1$
OUUT:           END$
              END$
OUT:        END$
          END$
        END$
    OUTIMAGE$ OUTIMAGE$
    OUTTEXT(@D = @)$ OUTINT(D,5)$
    OUTTEXT(@      POSSIBLE GAPS ARE: @)$
    FOR G := GMIN STEP 1 UNTIL GMAX DO
      IF GAPS(G) = 0 THEN
      BEGIN OUTINT(G,5)$
        FOUND := TRUE$
      END$
     IF NOT FOUND THEN OUTTEXT(@    NONE.@)$
     OUTTEXT(@    HALPHENS BOUND =@)$ OUTINT(GMAX,5)$
      FOUND := FALSE$
  END$
END$
END$
```

The output for $25 \leq d \leq 100$ is given in the table below. $H(d) = \left[\frac{1}{6}d(d-3)\right] + 1$.

d	Gaps	H(d)	d	Gaps	H(d)	d	Gaps	H(d)
25	23	92	51	None	409	77	None	950
26	None	100	52	None	425	78	None	976
27	None	109	53	None	442	79	None	1001
28	None	117	54	None	460	80	None	1027
29	27	126	55	59,80,83	477	81	None	1054
30	None	136	56	84	495	82	None	1080
31	29	145	57	107	514	83	None	1107
32	None	155	58	None	532	84	174,198	1135
33	35	166	59	63	551	85	None	1162
34	None	176	60	68,87	571	86	None	1190
35	None	187	61	119	590	87	None	1219
36	38	199	62	122	610	88	None	1247
37	35	210	63	None	631	89	None	1276
38	None	222	64	148	651	90	None	1306
39	39	235	65	None	672	91	None	1335
40	44	247	66	128	694	92	None	1365
41	39	260	67	None	714	93	None	1396
42	None	274	68	None	737	94	268	1424
43	47	287	69	143	760	95	None	1457
44	None	301	70	164	782	96	None	1489
45	47,63,83	316	71	None	805	97	None	1520
46	86	330	72	140,147,170	829	98	282	1552
47	None	345	73	None	852	99	None	1585
48	50	361	74	None	876	100	None	1617
49	59	376	75	147	901			
50	72	392	76	None	925			

Some of the possible gaps encountered here may be filled immediately, simply by taking the ν-uple embedding of some curve of lower degree, for some $\nu = 2,3,\ldots$. We get the table below.

When listed, $(d/\nu;\ g_1,\ldots,g_r)$ indicates that there exist curves of degree d/ν and genera $g_1,\ldots,g_r$, so that those gaps may be filled by twisting.

d	Gaps	(d/ν;g)	d	Gaps	(d/ν;g)
25	23		57	107	
29	27		59	63	
31	29		60	68,87	(30;68,87)
33	35		61	119	
36	38	(18;38)	62	122	(31;122)
37	35		64	148	(32;148)
39	39		66	128	(33;128)
40	44	(20;44)	69	143	
41	39		70	164	(35;164)
43	47		72	140,147,170	(36;140,147,170)
45	47,63,83		75	147	
46	86		84	174,198	(42;174,198)
48	50	(24;50)	94	268	(47;268)
49	59		98	282	(49;282)
50	72	(25;72)			
55	59,80,83				
56	84	(28;84)			

2.3. The Search Procedure. The remaining gaps are now filled by re-embedding the cubic surface. It turns out that it is not too difficult to encounter linear systems which when used to re-embedd the cubic surface, transforms some known curve into one which fills a gap. No systematic attempts were made to minimize the required data, however. Instead we first generated the parameters of all very ample linear systems of degree between 3 and 10, and stored them on the file VADATAFILE. The program implements an obvious modification of Hartshorne's algorithm, and is omitted here. Then our initial approach was to store all gap-values on another file, GADATAFILE., and generate the parameters of all non singular curves of ordinary degree between integers

Min and Max, storing those parameter sets for which the corresponding g appeared on GADATAFILE on the file DATAFILE. This being done, the program MAKELIST below would produce a list of all possible curves obtainable by re-embedding the ones on DATAFILE by the linear systems on VADATAFILE. The list is stored on the OUTPUTFIL in the format

(New degree, Genus)

The program is the following:

MAKELIST

```
BEGIN INTEGER I,D,COUNT$
INTEGER ARRAY B,PB(0:7)$
REF(INFILE) PAR,VAPAR$
REF(OUTFILE) LIST$
  PAR :- NEW INFILE(@DATAFILE@)$
  VAPAR :- NEW INFILE(@VADATAFILE@)$
  LIST :- NEW OUTFILE(@OUTPUTFIL@)$
  PAR.OPEN(BLANKS(60))$
  VAPAR.OPEN(BLANKS(60))$
  LIST.OPEN(BLANKS(110))$
  WHILE NOT PAR.LASTITEM DO
     BEGIN
    FOR I := 0 STEP 1 UNTIL 7 DO
      B(I) := PAR.ININT$
      PAR.INIMAGE$
      WHILE NOT VAPAR.LASTITEM DO
        BEGIN
        FOR I := 0 STEP 1 UNTIL 6 DO
        PB(I) := VAPAR.ININT$
        VAPAR.INIMAGE$
        D := B(0)*PB(0)-B(1)*PB(1)-B(2)*PB(2)-B(3)*PB(3)
            -B(4)*PB(4)-B(5)*PB(5)-B(6)*PB(6)$
        LIST.OUTINT(D,3)$LIST.OUTINT(B(7),4)$
        LIST.OUTTEXT(@    @)$
        COUNT := COUNT + 1$
        IF COUNT = 380000 THEN GOTO E$
        END$
      VAPAR.CLOSE$
      VAPAR.OPEN(BLANKS(60))$
    END$
E:  PAR.CLOSE$
  VAPAR.CLOSE$
  LIST.CLOSE$
OUTTEXT(@NUMBER OF ENTRIES ON OUTPUTFIL:     @)$
OUTINT(COUNT,20)$ OUTIMAGE$
END$
```

The final program scans the list produced above, and gives as output a list of all appearing values of g for Min $\leq$ New degree $\leq$ Max.

SEARCH

```
BEGIN INTEGER I,D,LD,G,MIN,MAX,MING$
BOOLEAN FOUND$
REF(INFILE) LIST$
LIST :- NEW INFILE(@OUTPUTFIL@)$
LIST.OPEN(BLANKS(110))$
OUTTEXT(@               THE FOLLOWING GENERA ARE VERIFIED:  @)$
MIN := ININT$ MAX := ININT$
 FOR D := MIN STEP 1 UNTIL MAX DO
  BEGIN INTEGER ARRAY GAPS(D-2:D*(D-3)/6)$
  WHILE NOT LIST.LASTITEM DO
  BEGIN LD := LIST.ININT$ G:= LIST.ININT$
    IF LD = D AND G LE D*(D-3)/6 AND G GE D-2 THEN
      GAPS(G) := -1$
  END$
  LIST.CLOSE$ LIST.OPEN(BLANKS(110))$
  OUTIMAGE$ OUTIMAGE$
  OUTTEXT(@D = @)$ OUTINT(D,5)$
OUTTEXT(@     :  @)$
  FOR G := D-2 STEP 1 UNTIL D*(D-3)/6 DO
    IF GAPS(G) = -1 THEN
    BEGIN OUTINT(G,5)$
       FOUND := TRUE$
     END$
    IF NOT FOUND THEN OUTTEXT(@    NONE.@)$
  FOUND := FALSE$
  END$
LIST.CLOSE$
END$
```

Initial experimentation with these programs revealed, however, that it was surprisingly easy to find curves which filled the gaps when re-embedded in this way. Therefore we proceeded in a somewhat different manner.

First, a list of curve parameters was selected more or less at radnom, with some educated guesses. Of course, all the wanted genus-values were represented on the list, which is reproduced in the table of Appendix 2. With this list on DATAFILE, the programs MAKELIST and SEARCH were run, producing the output given in Appendix 3 for $25 \leq d \leq 75$.

All unfilled gaps in the table at the end of §2.3 now appear in the appropriate line in the output listing of Appendix 3. Thus our Conjecture 2 is verified for all $d \leq 100$, and in particular Halphen's conjecture holds in this range.

Of course the list of parameters in Appendix 2 is not minimal, no effort was made to produce such a list. This is easily done, but would serve no purpose in our situation.

Appendix 1. The first 123 very ample linear systems on the cubic surface.

α	β_1	β_2	β_3	β_4	β_5	β_6	d
3	1	1	1	1	1	1	3
6	3	2	2	2	2	2	5
5	2	2	2	2	1	1	5
4	2	1	1	1	1	1	5
8	3	3	3	3	3	3	6
7	3	3	3	2	2	2	6
6	3	2	2	2	2	1	6
6	2	2	2	2	2	2	6
5	2	2	2	1	1	1	6
4	1	1	1	1	1	1	6
9	5	3	3	3	3	3	7
9	4	4	3	3	3	3	7
8	4	3	3	3	2	2	7
8	3	3	3	3	3	2	7
7	4	2	2	2	2	2	7
7	3	3	3	3	1	1	7
7	3	3	3	2	2	1	7
7	3	3	2	2	2	2	7
6	3	2	2	2	1	1	7
6	2	2	2	2	2	1	7
5	3	1	1	1	1	1	7
5	2	2	1	1	1	1	7
11	5	4	4	4	4	4	8
10	5	4	4	3	3	3	8
10	4	4	4	4	3	3	8
9	5	3	3	3	3	2	8
9	4	4	4	3	2	2	8
9	4	4	3	3	3	2	8
9	4	3	3	3	3	3	8
8	4	3	3	3	2	1	8
8	4	3	3	2	2	2	8
8	3	3	3	3	3	1	8
8	3	3	3	3	2	2	8
7	4	2	2	2	2	1	8
7	3	3	3	2	1	1	8
7	3	3	2	2	2	1	8
7	3	2	2	2	2	2	8
6	3	2	2	1	1	1	8
6	2	2	2	2	1	1	8
5	2	1	1	1	1	1	8
13	5	5	5	5	5	5	9
12	7	4	4	4	4	4	9
12	6	5	4	4	4	4	9
12	5	5	5	4	4	4	9
11	6	4	4	4	3	3	9
11	5	5	5	3	3	3	9
11	5	5	4	4	3	3	9
11	5	4	4	4	4	3	9
11	4	4	4	4	4	4	9
10	6	4	4	4	4	4	9
10	5	4	4	4	2	2	9
10	5	4	4	3	3	2	9
10	5	4	3	3	3	3	9

α	β_1	β_2	β_3	β_4	β_5	β_6	d
10	4	4	4	4	3	2	9
10	4	4	4	3	3	3	9
9	5	3	3	3	3	1	9
9	5	3	3	3	2	2	9
9	4	4	4	4	1	1	9
9	4	4	4	3	2	1	9
9	4	4	4	2	2	2	9
9	4	4	3	3	3	1	9
9	4	4	3	3	2	2	9
9	4	3	3	3	3	2	9
9	3	3	3	3	3	3	9
8	5	2	2	2	2	2	9
8	4	3	4	4	1	1	9
8	4	3	3	2	2	1	9
8	4	3	2	2	2	2	9
8	3	3	3	3	2	1	9
8	3	3	3	2	2	2	9
7	4	2	2	2	1	1	9
7	3	3	3	1	1	1	9
7	3	3	2	2	1	1	9
7	3	2	2	2	2	1	9
7	2	2	2	2	2	2	9
6	4	1	1	1	1	1	9
6	3	2	1	1	1	1	9
6	2	2	2	1	1	1	9
5	1	1	1	1	1	1	9
14	7	5	5	5	5	5	10
14	6	6	5	5	5	5	10
13	7	5	5	4	4	4	10
13	6	6	5	4	4	4	10
13	6	5	5	5	4	4	10
13	5	5	5	5	5	4	10
12	7	4	4	4	4	3	10
12	6	5	5	4	3	3	10
12	6	5	4	4	4	3	10
12	6	4	4	4	4	4	10
12	5	5	5	5	3	3	10
12	5	5	5	4	4	3	10
12	5	5	4	4	4	4	10
11	6	4	4	4	3	2	10
11	6	4	4	3	3	3	10
11	5	5	5	4	2	2	10
11	5	5	5	3	3	2	10
11	5	5	4	4	3	2	10
11	5	5	4	3	3	3	10
11	5	4	4	4	4	2	10
11	5	4	4	4	3	3	10
11	4	4	4	4	4	3	10
10	6	3	3	3	3	2	10
10	5	4	4	4	2	1	10
10	5	4	4	3	3	1	10
10	5	4	4	3	2	2	10
10	5	4	3	3	3	2	10

CONT.:

α	β_1	β_2	β_3	β_4	β_5	β_6	d
10	5	3	3	3	3	3	10
10	4	4	4	4	3	1	10
10	4	4	4	4	2	2	10
10	4	4	4	3	3	2	10
10	4	4	3	3	3	3	10
9	5	3	3	3	2	1	10
9	5	3	3	2	2	2	10
9	4	4	4	3	1	1	10
9	4	4	4	2	2	1	10
9	4	4	3	3	2	1	10
9	4	4	3	2	2	2	10
9	4	3	3	3	3	1	10
9	4	3	3	3	2	2	10
9	3	3	3	3	3	2	10
8	5	2	2	2	2	1	10
8	4	3	3	2	1	1	10
8	4	3	2	2	2	1	10
8	4	2	2	2	2	2	10
8	3	3	3	3	1	1	10
8	3	3	3	2	2	1	10
8	3	3	2	2	2	2	10
7	4	2	2	1	1	1	10
7	3	3	2	1	1	1	10
7	3	2	2	2	1	1	10
7	2	2	2	2	2	1	10
6	3	1	1	1	1	1	10
6	2	2	1	1	1	1	10

Appendix 2. A list of selected curve parameters.

a	b_1	b_2	b_3	b_4	b_5	b_6	d
24	12	11	9	8	4	3	59
14	6	6	5	4	4	3	23
13	6	5	4	4	3	3	23
13	5	5	5	4	4	2	23
15	6	6	6	5	4	3	27
14	6	6	4	4	4	3	27
14	6	5	5	5	3	3	27
17	7	7	7	5	4	4	35
17	7	7	6	6	4	3	35
17	7	8	8	5	5	5	39
17	6	6	6	6	6	4	39
16	7	7	5	4	4	4	35
16	7	6	6	5	4	3	35
16	6	6	5	5	5	4	39
14	5	5	5	5	5	2	27
19	8	8	8	6	6	1	39
18	7	7	7	5	5	4	47
21	12	7	7	6	5	5	47
25	15	10	7	7	6	5	59
25	15	9	7	7	6	6	63
25	12	9	8	7	7	7	83
25	12	8	8	8	8	6	83
25	9	9	9	9	9	5	86
24	9	9	9	8	7	5	86
22	8	8	8	8	6	5	80
20	15	10	9	9	8	7	107
35	20	15	12	10	7	6	119
35	18	16	10	9	9	8	143
32	12	12	12	11	11	5	147
15	6	6	5	5	4	4	29
14	6	5	4	4	4	4	29
15	5	5	5	5	4	3	29
25	12	10	8	7	7	6	80
25	12	9	9	8	6	6	80
25	12	9	8	8	8	5	80
25	11	11	8	8	6	6	80
25	11	10	10	7	6	6	80
25	10	10	10	9	6	5	80
25	10	10	9	9	8	4	80
24	12	8	8	7	6	6	80
24	11	10	8	6	6	6	80
24	11	10	7	7	7	5	80
24	11	9	9	7	6	5	80
24	11	8	8	8	8	4	80
24	10	10	8	8	7	4	80
24	10	9	9	9	5	5	80
23	11	9	6	6	6	6	80
23	11	8	8	6	6	5	80
23	10	10	7	6	6	5	80
23	10	9	8	7	6	4	80
23	9	8	8	8	8	3	80

Appendix 3. Output from SEARCH.

THE FOLLOWING GENERA ARE VERIFIED:

D = 25	:	23	27	29	35	59	63	80	83	86
D = 26	:	35								
D = 27	:	27	35	39						
D = 28	:	27	29	39						
D = 29	:	27	29	47	107					
D = 30	:	29	35	47						
D = 31	:	29	35	39	47					
D = 32	:	39								
D = 33	:	35	39	59	147					
D = 34	:	35	39	47						
D = 35	:	35	39	47	59	63	119	143		
D = 36	:	35	39	59	80					
D = 37	:	35	39	47	59	63	80			
D = 38	:	39	47	49	80	83	86			
D = 39	:	39	47	80	83	86				
D = 40	:	39	47	59	80	86				
D = 41	:	39	47	59	63	80	86			
D = 42	:	47	59	80						
D = 43	:	47	59	80	107					
D = 44	:	47	63	80	83	107				
D = 45	:	47	59	63	80	83				
D = 46	:	47	59	80	83	86				
D = 47	:	47	59	63	80	83	86			
D = 48	:	47	59	80	86	119				
D = 49	:	47	59	63	80	86				
D = 50	:	59	63	80	83	86	107	119	147	
D = 51	:	59	63	80	83	86				
D = 52	:	59	80	83	86	143				
D = 53	:	59	63	80	83	86	107	147		
D = 54	:	59	63	80	83	86	147			
D = 55	:	59	63	80	83	86				
D = 56	:	59	63	80	83	86	119			
D = 57	:	59	63	80	83	86	107			
D = 58	:	59	80	83	107	119				
D = 59	:	59	63	80	83	86	107	147		
D = 60	:	59	63	80	83	86	107	143		
D = 61	:	59	63	80	83	86	119	143	147	
D = 62	:	63	80	83	86	107	147			
D = 63	:	63	80	83	86	119				
D = 64	:	63	80	83	86	107	119			
D = 65	:	63	80	83	86	107	119	147		
D = 66	:	80	83	86	147					
D = 67	:	80	83	86	107	147				
D = 68	:	80	83	86	107					
D = 69	:	80	83	86	107	119	143			
D = 70	:	80	83	86	119	143	147			
D = 71	:	80	83	86	107	119	143	147		
D = 72	:	80	83	86	107	147				
D = 73	:	80	83	86	107	119	147			
D = 74	:	80	83	86	107	119	147			
D = 75	:	80	83	86	107	147				

R E F E R E N C E S

[G-P] Gruson, L. and Peskine, C.: "Genre de courbes de l'espace projectiv." In Algebraic Geometry. Tromsø, Norway 1977. Springer Lecture Notes in Mathematics 687.

[Hal] Halphen, G.: "Mémoire sur la classification de courbes gauches algébriques," J.Ec.Polyt. 52, (1882), pp. 1-200.

[Har1] Hartshorne, R.: Algebraic Geometry. Graduate texts in Mathematics. Springer-Verlag 1977.

[Har2] Hartshorne, R.: "On the classification of algebraic space curves." Preprint.

[Ho1] Holme, A.: "Classification in projective algebraic geometry." In the Proceedings of the 18th Scandinavian Congress of Mathematicians, 1980. To appear in Progress in Mathematics, Birkhäuser Verlag.

[Ho2] Holme, A.: Classification of curves in P^3 by a Monte Carlo method." To appear.

[Kn] Knuth, D.: The Art of Computer Programming. Vol. 2. Seminumerical Algorithms. Addison-Wesley Publ. Co., 1969.

[No1] Noether, M.: "Zur Theorie der algebraischen Raumkurven." Journal für die reine und angewandte Mathematik. 33 (1882), pp. 271, 318.

[No2] Noether, M.: Zur Grundlegung der Theorie der algebraischen Raumkurven. Verlag der Königlichen Akademie der Wissenschaften, Berlin (1883).

Note added in proof.

After these computations were completed in 1980, Peskine and Gruson have proved Halphen's conjecture. One of the key steps in their proof is a confirmation of Hartshorne's conjecture for an upper bound of the gaps on the non singular cubic surface. I understand they are now in the process of writing up their work.

My program implementing Hartshornes's Algorithm has been made more efficient by Svein Mossige of the University of Bergen. Unfortunately it still runs too slowly to be of any use beyond say, d = 200 .

The results obtained are listed in the table given below. Upper and lower limits of the gap intervals are given, the reader may compare the output format to that of the table in §2.

14
1 3
15
1 4
16
1 4
17
1 5
18
1 5 9 9
19
1 6 10 10
20
1 6 10 11
21
1 7 11 12 19 19
22
1 7 11 13
23
1 8 12 14
24
1 8 12 15 22 22
25
1 9 13 16 23 25 25 25
26
1 9 13 17 23 25
27
1 10 14 18 25 26
28
1 10 14 19 26 26 28 28
29
1 11 15 20 27 29 31 31
30
1 11 15 21 28 29 31 31
31
1 12 16 22 29 32 34 34
32
1 12 16 23 30 32 34 34
33
1 13 17 24 31 35 37 37
34
1 13 17 25 32 35 37 37
35
1 14 18 26 33 38 40 40

36
1 14 18 27 34 38 40 40
37
1 15 19 28 35 41 43 43 51 51
38
1 15 19 29 36 41 43 43 53 53
39
1 16 20 30 37 44 46 46 55 55
40
1 16 20 31 38 44 46 46 57 57
41
1 17 21 32 39 47 49 49 57 57 59 59
42
1 17 21 33 40 47 49 49 59 59 61 61 76 76
43
1 18 22 34 41 50 52 52 60 61 63 63 78 78
44
1 18 22 35 42 50 52 52 62 63 65 65 81 81
45
1 19 23 36 43 53 55 55 63 65 67 67 83 83
46
1 19 23 37 44 53 55 55 65 67 69 69 86 86
47
1 20 24 38 45 56 58 58 66 69 71 71 86 86 88 88
48
1 20 24 39 46 56 58 58 68 71 73 73 88 88 91 91
49
1 21 25 40 47 59 61 61 69 73 75 75 91 91 93 93
50
1 21 25 41 48 59 61 61 71 75 77 77 92 93 96 96
51
1 22 26 42 49 62 64 64 72 77 79 79 94 94 96 96 98 98
52
1 22 26 43 50 62 64 64 74 79 81 81 96 98 101 101
53
1 23 27 44 51 65 67 67 75 81 83 83 98 99 101 101 103 103
54
1 23 27 45 52 65 67 67 77 83 85 85 101 103 106 106
55
1 24 28 46 53 68 70 70 78 85 87 87 102 104 106 106 108 108
56
1 24 28 47 54 68 70 70 80 87 89 89 104 104 106 108 111 111
57
1 25 29 48 55 71 73 73 81 89 91 91 106 109 111 111 113 113

58
1 25 29 49 50 71 73 73 85 91 95 95 108 109 111 113 116 116 133 133
59
1 26 30 50 57 74 76 76 84 95 95 95 110 114 116 116 118 118 136 136
60
1 26 30 51 58 74 76 76 86 95 97 97 112 114 116 118 121 121 139 139
61
1 27 31 52 59 77 79 79 87 97 99 99 114 119 121 121 123 123 142 142
62
1 27 31 53 60 77 79 79 89 99 101 101 116 119 121 123 126 126 145 145
63
1 28 32 54 61 80 82 82 90 101 103 103 118 124 126 126 128 128 148 148
64
1 28 32 55 62 80 82 82 92 103 105 105 120 124 126 128 131 131 148 148
151 151
65
1 29 33 56 63 85 85 85 95 105 107 107 122 129 131 131 133 133 151 151
154 154
66
1 29 33 57 64 85 85 85 95 107 109 109 124 129 131 133 136 136 154 154
157 157
67
1 30 34 58 65 86 88 88 96 109 111 111 126 134 136 136 138 138 157 157
160 160 184 184
68
1 30 34 59 66 86 88 88 98 111 113 113 128 134 136 138 141 141 158 158
160 160 163 163
69
1 31 35 60 67 89 91 91 99 113 115 115 130 139 141 141 143 143 161 161
163 163 166 166 191 191
70
1 31 35 61 68 89 91 91 101 115 117 117 132 139 141 143 146 146 163 164
166 166 169 169
71
1 32 36 62 69 92 94 94 102 117 119 119 134 144 146 146 148 148 166 167
169 169 172 172 198 198
72
1 32 36 63 70 92 94 94 104 119 121 121 136 144 146 148 151 151 169 170
172 172 175 175 199 199
73
1 33 37 64 71 95 97 97 105 121 123 123 138 149 151 151 153 153 172 173
175 175 178 178 205 205
74
1 33 37 65 72 95 97 97 107 123 125 125 140 149 151 153 156 156 173 173
175 176 178 178 181 181 205 206
75
1 34 38 66 73 98 100 100 108 125 127 127 142 154 156 156 158 158 176 176
178 179 181 181 184 184 208 208 212 212
76
1 34 38 67 74 98 100 100 110 127 129 129 144 154 156 158 161 161 178 179
181 182 184 184 187 187 212 215
77
1 35 39 68 75 101 103 103 111 129 131 131 146 159 161 161 163 163 181 182
184 185 187 187 190 190 215 215 219 219
78
1 35 39 69 76 101 103 103 113 131 133 133 148 159 161 163 166 166 183 185
187 188 190 190 193 193 219 220
79
1 36 40 70 77 104 106 106 114 133 135 135 150 164 166 166 168 168 186 188
190 191 193 193 196 196 220 220 222 222 226 226
80
1 36 40 71 78 104 106 106 116 135 137 137 152 164 166 168 171 171 188 191
193 194 196 196 199 199 226 227
81
1 37 41 72 79 107 109 109 117 137 139 139 154 169 171 171 173 173 191 194
196 197 199 199 202 202 226 227 229 229 233 233
82
1 37 41 73 80 107 109 109 119 139 141 141 156 169 171 173 176 176 193 197
199 200 202 202 205 205 229 229 233 234
83
1 38 42 74 81 110 112 112 120 141 143 143 158 174 176 176 178 178 196 200
202 203 205 205 208 208 232 234 236 236 240 240
84
1 38 42 75 82 110 112 112 122 143 145 145 160 174 176 178 181 181 198 203
205 206 208 208 211 211 235 236 240 241
85
1 39 43 76 83 113 115 115 125 145 147 147 162 179 181 181 183 183 201 206
208 209 211 211 214 214 239 241 243 243 247 247
86
1 39 43 77 84 113 115 115 125 147 149 149 164 179 181 183 186 186 203 209
211 212 214 214 217 217 241 243 247 248
87
1 40 44 78 85 116 118 118 126 149 151 151 166 184 186 186 188 188 206 212
214 215 217 217 220 220 244 244 246 248 250 250 254 254
88
1 40 44 79 86 116 118 118 128 151 153 153 168 184 186 188 191 191 208 215
217 218 220 220 223 223 247 250 254 255
89

1 41 43 80 87 119 121 121 129 133 133 155 170 189 191 191 193 193 211 218
220 221 223 223 226 226 230 231 233 233 237 237 261 261

90
1 41 43 81 88 119 121 121 131 155 157 157 172 189 191 193 196 196 213 221
223 224 226 226 229 229 233 257 261 262

91
1 42 46 82 89 122 124 124 132 157 159 159 174 194 196 196 198 198 216 224
226 227 229 229 232 232 236 238 260 262 264 264 268 268

92
1 42 46 83 90 122 124 124 134 139 161 161 176 194 196 198 201 201 218 227
229 230 232 232 233 233 260 264 268 269

93
1 43 47 84 91 125 127 127 133 161 163 163 178 199 201 201 203 203 221 230
232 233 233 233 238 238 262 265 267 269 271 271 273 275 343 343

94
1 43 47 83 [illegible] 123 127 127 137 163 163 165 180 199 201 203 206 206 223 233
233 236 238 2[illegible]8 241 241 263 265 267 271 273 276

95
1 44 48 86 93 123 130 130 138 133 167 167 182 204 203 206 208 208 226 236
238 239 241 241 244 244 263 272 274 276 278 278 282 282 352 352

96
1 44 48 87 94 128 130 130 140 167 169 169 184 204 206 208 211 211 228 239
241 242 244 244 247 247 271 272 274 278 282 283 355 355

97
1 45 49 88 95 131 133 133 141 169 171 171 186 209 211 211 213 213 231 242
244 245 247 247 250 250 274 279 281 283 285 285 289 289 361 361

98
1 45 49 89 96 131 133 133 143 171 173 173 188 209 211 213 216 216 233 245
247 248 250 250 253 253 277 279 281 285 289 290 364 364

99
1 46 50 90 97 134 136 136 144 173 175 175 190 214 216 216 218 218 236 248
250 251 253 253 256 256 280 286 288 290 292 292 296 296 370 370

100
1 46 50 91 98 134 136 136 146 175 177 177 192 214 216 218 221 221 238 251
253 254 256 256 259 259 283 286 288 292 296 297 373 373

101
1 47 51 92 99 137 139 139 147 177 179 179 194 219 221 221 223 223 241 254
256 257 259 259 262 262 280 293 295 297 299 299 303 303 379 379

102
1 47 51 93 100 137 139 139 149 179 181 181 196 219 221 223 226 226 243 257
259 260 262 262 265 265 289 293 295 299 303 304 379 379 382 382

103
1 48 52 94 101 140 142 142 150 181 183 183 198 224 226 226 228 228 246 260
262 263 263 265 268 268 292 300 302 304 306 306 310 310 338 338 388 388

104
1 48 52 95 102 140 142 142 152 183 185 185 200 224 226 228 231 231 248 263
265 266 268 268 271 271 295 300 302 306 310 311 342 342 388 388 391 391

105
1 49 53 96 103 143 145 145 153 185 187 187 202 229 231 231 233 233 251 266
268 269 271 271 274 274 298 307 309 311 313 313 317 317 345 346 391 391 397 397

106
1 49 53 97 104 143 145 145 155 187 189 189 204 229 231 233 236 236 253 269
271 272 274 274 277 277 301 307 309 313 317 318 349 350 397 397 400 400

107
1 50 54 98 105 146 148 148 156 189 191 191 206 234 236 236 238 238 256 272
274 275 277 277 280 280 304 314 316 318 320 320 324 324 353 354 400 400 406 406

108
1 50 54 99 106 146 148 148 158 191 193 193 208 234 236 238 241 241 258 275
277 278 280 280 283 283 307 314 316 320 324 325 357 358 406 406 409 09

109
1 51 55 100 107 149 151 151 159 193 195 195 210 239 241 241 243 243 261 278
280 281 283 283 286 286 310 321 323 325 327 327 331 331 359 359 361 362 407 407
409 409 415 415

110
1 51 53 101 108 149 151 151 161 195 197 197 212 239 241 243 246 246 263 281
283 284 286 286 289 289 313 321 323 327 331 332 363 363 365 366 415 415 418 418

111
1 52 56 102 109 152 154 154 162 197 199 199 214 244 246 246 248 248 266 284
286 287 289 289 292 292 316 328 330 332 334 334 338 338 366 367 369 370 415 416
41[illegible] 418 4[illegible]4 424

112
1 52 56 103 110 152 154 154 164 199 201 201 216 244 246 248 251 251 268 287
289 290 292 292 295 295 319 328 330 334 338 339 370 371 373 374 419 419 424 424
427 427

113
1 53 57 104 111 155 157 157 165 201 203 203 218 249 251 251 253 253 271 290
292 293 293 295 298 298 322 335 337 339 341 341 345 345 373 375 377 378 424 425
427 427 433 433

114																			
1	53	57	105	112	155	157	157	167	203	205	205	220	249	251	253	256	256	273	293
295	296	298	298	301	301	325	335	337	341	343	346	377	379	381	382	427	423	433	433
436	436	2001																	
115																			
1	54	58	106	113	158	160	160	168	205	207	207	222	254	256	256	258	258	276	296
298	299	301	301	304	304	328	342	344	346	348	348	352	352	381	383	385	386	433	434
436	436	443	442																
116																			
1	54	58	107	114	158	160	160	170	207	209	209	224	254	256	258	261	261	278	299
301	302	304	304	307	307	331	342	344	348	352	353	385	387	389	390	436	437	442	442
445	445																		
117																			
1	55	59	108	115	161	163	163	171	209	211	211	226	259	261	261	263	263	281	302
304	305	307	307	310	310	334	349	351	353	355	355	359	359	387	387	389	391	393	394
439	439	442	445	445	445	451	451												
118																			
1	55	59	109	116	161	163	163	173	211	213	213	228	259	261	263	266	266	283	305
307	308	310	310	313	313	337	349	351	355	359	360	391	391	393	395	397	398	445	445
445	446	451	451	454	454														
119																			
1	56	60	110	117	164	166	166	174	213	215	215	230	264	266	266	268	268	286	308
310	311	313	313	316	316	340	356	358	360	362	362	366	366	394	395	397	399	401	402
448	448	451	452	454	454	460	460												
120																			
1	56	60	111	118	164	166	166	176	215	217	217	232	264	266	268	271	271	288	311
313	314	316	316	319	319	343	356	358	362	366	367	398	399	401	403	405	406	451	452
454	455	460	460	463	463														
121																			
1	57	61	112	119	167	169	169	177	217	219	219	234	269	271	271	273	273	291	314
316	317	319	319	322	322	346	363	365	367	369	369	373	373	401	403	405	407	409	410
455	455	457	457	460	461	463	463	469	[illegible]										
122																			
1	57	61	113	120	167	169	169	179	219	221	221	236	269	271	273	276	276	293	317
319	320	322	322	325	325	349	363	365	369	373	374	405	407	409	411	413	414	460	461
463	464	469	469	472	472														
123																			
1	58	62	114	121	170	172	172	180	221	223	223	238	274	276	276	278	278	296	320
322	323	325	325	328	328	352	370	372	374	376	376	380	380	408	411	413	415	417	418
463	464	466	466	469	470	472	472	478	478										
124																			
1	58	62	115	122	170	172	172	182	223	225	225	240	274	276	278	281	281	298	323
325	326	328	328	331	331	355	370	372	376	380	381	412	415	417	419	421	422	469	470
472	473	478	478	481	481														
125																			
1	59	63	116	123	173	175	175	183	225	227	227	242	279	281	281	283	283	301	326
328	329	331	331	334	334	358	377	379	381	383	385	387	387	415	419	421	423	425	426
471	473	475	475	478	479	481	481	487	487										
126																			
1	59	63	117	124	173	175	175	185	227	229	229	244	279	281	283	286	286	303	329
331	332	334	334	337	337	361	377	379	383	387	388	419	425	425	427	429	430	475	475
478	479	481	482	487	487	490	490												
127																			
1	60	64	118	125	176	178	178	186	229	231	231	246	284	286	286	288	288	306	332
334	335	337	337	340	340	364	384	386	388	390	390	394	394	422	427	429	431	433	434
479	482	484	484	487	488	490	490	496	496										
128																			
1	60	64	119	126	176	178	178	188	231	233	233	248	284	286	288	291	291	308	335
337	338	340	340	343	343	367	384	386	390	394	395	426	431	433	435	437	438	483	484
487	488	490	491	496	496	499	499												
129																			
1	61	65	120	127	179	181	181	189	233	235	235	250	289	291	291	293	293	311	338
340	341	343	343	346	346	370	391	393	395	397	397	401	401	429	435	437	439	441	442
487	491	493	495	496	497	499	499	503	503										
130																			
1	61	65	121	128	179	181	181	191	235	237	237	252	289	291	293	296	296	313	341
343	344	346	346	349	349	373	391	393	397	401	402	433	439	441	443	445	446	491	493
496	497	499	500	505	505	508	508												
131																			
1	62	66	122	129	182	184	184	192	237	239	239	254	294	296	296	298	298	316	344
346	347	349	349	352	352	376	398	400	402	404	404	408	408	436	443	445	447	449	450
496	500	502	502	505	506	508	508	514	514										
132																			
1	62	66	123	130	182	184	184	194	239	241	241	256	294	296	298	301	301	318	347
349	350	352	352	355	355	379	398	400	404	408	409	440	447	449	451	453	454	499	502
505	506	508	509	514	514	517	517												
133																			
1	63	67	124	131	185	187	187	195	241	243	243	258	299	301	301	303	303	321	350
352	353	355	355	358	358	382	405	407	409	411	411	415	415	443	451	453	455	457	458
503	503	505	509	511	511	514	515	517	517	523	523								

134																			
1	63	67	125	132	135	187	187	197	245	245	245	260	299	301	303	306	306	325	333
355	356	358	358	361	361	385	405	407	411	415	416	447	455	457	459	461	462	507	511
514	515	517	518	525	525	526	526												
135																			
1	64	68	126	133	136	190	190	198	245	247	247	262	304	306	306	308	308	326	335
358	359	361	361	364	364	366	412	414	415	415	416	422	422	430	459	461	465	435	406
511	512	514	516	520	520	523	524	526	526	532	532								
136																			
1	64	68	127	134	188	190	190	200	247	249	249	264	304	306	308	311	311	325	339
361	362	364	364	367	367	391	412	414	418	422	425	454	465	465	467	469	470	515	520
523	524	526	527	532	532	535	535	576	576										
137																			
1	65	69	128	135	191	195	195	201	249	251	251	266	309	311	311	315	315	351	362
364	365	367	367	370	370	394	419	421	425	425	425	429	429	457	467	469	471	475	474
519	521	525	527	529	529	532	535	535	535	541	541	581	581	641	641				
138																			
1	65	69	129	136	191	193	193	203	251	253	253	268	309	311	313	316	316	333	365
367	368	370	370	373	373	397	419	421	425	429	430	461	471	473	475	477	478	523	529
532	533	535	536	541	541	544	544	586	586										
139																			
1	65	70	130	137	194	196	196	204	253	255	255	270	314	316	316	318	318	336	368
370	371	373	373	376	376	400	420	426	430	432	432	436	435	464	473	477	479	481	482
527	530	532	536	538	538	541	542	544	544	550	550	591	591	652	652				
140																			
1	66	70	131	138	194	195	196	200	255	257	257	272	314	316	318	321	321	338	371
373	374	376	376	379	379	403	420	428	432	436	437	466	479	481	483	485	486	532	538
541	542	544	545	550	550	555	555	596	596										
141																			
1	67	71	132	139	197	199	199	207	257	259	259	274	319	321	321	323	323	341	374
376	377	379	379	382	382	406	433	435	437	439	439	443	443	471	483	485	487	489	490
535	539	541	545	547	547	550	551	553	553	557	559	601	601	663	663				
142																			
1	67	71	133	140	197	199	199	209	259	261	261	276	319	321	323	326	326	343	377
379	380	382	382	385	385	409	433	435	439	443	444	475	487	489	491	493	494	539	539
541	547	550	551	553	554	559	559	562	562	605	605	606	606						
143																			
1	68	72	134	141	200	202	202	210	261	263	263	278	324	326	326	328	328	346	380
382	383	385	385	388	388	412	440	442	444	446	446	450	450	478	491	493	495	497	498
545	548	550	554	556	556	559	560	562	562	568	568	608	608	611	611	674	674		
144																			
1	68	72	135	142	200	202	202	212	263	265	265	280	324	326	328	331	331	348	383
385	386	388	388	391	391	413	440	442	446	450	451	482	495	497	499	501	502	547	548
550	556	559	560	562	565	568	568	571	571	613	613	616	616						
145																			
1	69	73	136	143	203	205	205	213	265	267	267	282	329	331	331	333	333	351	386
388	389	391	391	394	394	418	447	449	451	453	453	457	457	485	499	501	503	505	506
551	557	559	563	565	565	568	569	571	571	577	577	618	618	621	621	685	685		
146																			
1	69	73	137	144	203	205	205	215	267	269	269	284	329	331	333	336	336	353	389
391	392	394	394	397	397	421	447	449	453	457	458	489	503	505	507	509	510	555	557
559	565	568	569	571	572	577	577	580	580	621	621	623	623	626	626				
147																			
1	70	74	138	145	206	208	208	216	269	271	271	286	334	336	336	338	338	356	392
394	395	397	397	400	400	424	454	456	458	460	460	464	464	492	507	509	511	513	514
550	566	568	572	574	574	577	578	580	580	586	586	626	626	628	628	651	651	696	696
148																			
1	70	74	139	146	206	208	208	218	271	273	273	288	334	336	338	341	341	358	395
397	398	400	400	403	403	427	454	456	460	464	465	496	511	513	515	517	518	563	566
568	574	577	578	580	581	586	586	589	589	631	631	633	633	636	636	696	696		
149																			
1	71	75	140	147	209	211	211	219	273	275	275	290	339	341	341	343	343	361	398
400	401	403	403	406	406	430	461	463	465	467	467	471	471	499	515	517	519	521	522
567	575	577	581	583	583	586	587	589	589	595	595	636	636	638	638	641	641	707	707
150																			
1	71	75	141	148	209	211	211	221	275	277	277	292	339	341	343	346	346	363	401
403	404	406	406	409	409	433	461	463	467	471	472	503	519	521	523	525	526	571	575
577	583	586	587	589	590	595	595	598	598	641	641	643	643	646	646	706	707		
151																			
1	72	76	142	149	212	214	214	222	277	279	279	294	344	346	346	348	348	366	404
406	407	409	409	412	412	436	468	470	472	474	474	478	478	506	523	525	527	529	530
575	584	586	590	592	592	595	596	598	598	604	604	646	646	648	648	651	651	718	718