Inverse Polynomials for Nonzero Constant Jacobian

Dhananjay P. Mehendale
Sir Parashurambhau College, Tilak Road, Pune 411030
India

Abstract

The so called Jacobian problem [1] or Jacobian conjecture [2], [3] demands the existence of inverse functions of polynomial nature when the Jacobian is a nonzero constant (=1). Bass, Connell, Wright in there paper [3] have shown that it is enough to construct (or show the existence of) such inverse polynomials for special type of cubic polynomials whose Jacobian is a nonzero constant (=1).To settle Jacobian conjecture one needs to show the existence of inverse polynomials for special type of cubic polynomials whose Jacobian is a nonzero constant (=1) for all dimensions [3]. In this paper we explicitly give these inverse polynomials for two and three dimensions, i.e. for two and three variables cases. Any higher dimensional cases are no different than these special cases and it is possible to obtain such inverse polynomials in any higher dimensional cases also.

1. **Introduction:** Given n polynomials $u = (u_1, u_2, \cdots, u_n)$ in n variables $x = (x_1, x_2, \cdots, x_n)$ and their Jacobian $J_x(u) = \det\left(\frac{\partial u_i}{\partial x_j}\right)$, $i, j = 1, 2, \cdots, n$ is a nonzero constant in the ground field k of characteristic zero. The problem called the Jacobian problem [1] or the Jacobian conjecture [2], [3] is to show that the polynomial rings are same, i.e., $k[x_1, x_2, \cdots, x_n]$ = $k[u_1, u_2, \cdots, u_n]$. In other words, one needs to show that one can construct the so called n inverse polynomials: $x = (x_1, x_2, \cdots, x_n)$ in n variables $u = (u_1, u_2, \cdots, u_n)$.

   The important result due to H. Bass, E. H. Connell, and D. Wright in [3] very much reduces the computational burden by their **important reduction** in degree achieved for the Jacobian problem. According to their result it is enough to settle the Jacobian problem for the special homogeneous polynomials $u \equiv u(x)$ of the special cubic form, namely,

$u \equiv u(x) = x - H(x)$, where $H(tx) = t^3 H(x)$ for all $t \in k$ and all $x \in k^n$, k being the ground field of characteristic zero. Their result essentially is as follows:

Theorem [BCW]: The Jacobian conjecture is true for polynomials $u(x)$ having every number of variables n, and for every degree if and only if it is true for polynomials having every number of variables n, and having cubic degree i.e. having special kind of cubic-homogeneous form: $u(x) = x - H(x)$, where $H(\alpha x) = \alpha^3 H(x)$ for every $\alpha \in k$, the ground field of characteristic zero.

□

The Jacobian conjecture can now be stated in the light of the above theorem[BCW] as follows:

Jacobian Conjecture: Given n polynomials, each one of having special kind of cubic-homogeneous form: $u(x) = x - H(x)$, where $H(\alpha x) = \alpha^3 H(x)$ for every $\alpha \in k$, the ground field of characteristic zero. Thus, $u = (u_1, u_2, \cdots, u_n)$ are polynomials of special type mentioned above in n variables $x = (x_1, x_2, \cdots, x_n)$ and their Jacobian $J_x(u) = \det\left(\frac{\partial u_i}{\partial x_j}\right)$, $i, j = 1,2,\cdots,n$ is a nonzero constant (=1) in the ground field k of characteristic zero then we get a polynomial inverse for this system of polynomials i.e. we get each x_i as special kind of cubic-homogeneous polynomial: $x(u) = u - H(u)$, where $H(\alpha u) = \alpha^3 H(u)$ for every $\alpha \in k$, the ground field of characteristic zero polynomial in n variables $u = (u_1, u_2, \cdots, u_n)$.

□

2. The Inverse Polynomials for two dimensions: We first deal below with two dimensional case. We first take polynomials in two variables, i.e. polynomials $f = f(x, y)$ and $g = g(x, y)$ such that their Jacobian,

$$J_{(x,y)}(f,g) = \det\begin{pmatrix} f_x & f_y \\ g_x & g_y \end{pmatrix} = 1 \neq 0$$

We find the inverse polynomials $x = x(f,g)$ and $y = y(f,g)$.

Quadratic Polynomials: Let f and g be following quadratic polynomials:

$$f = a_{10}x + a_{01}y + a_{20}x^2 + a_{11}xy + a_{02}y^2$$
$$g = b_{10}x + b_{01}y + b_{20}x^2 + b_{11}xy + b_{02}y^2$$

It is easy to verify that the required inverse polynomials are also quadratic and they are:

$$x = b_{01}f - a_{01}g + \frac{1}{2}(b_{01}b_{11} - 2b_{10}b_{02})f^2 + (2a_{10}b_{02} - a_{01}b_{11})fg$$
$$+ \frac{1}{2}(a_{01}a_{11} - 2a_{10}a_{02})g^2$$
$$y = -b_{10}f + a_{10}g + \frac{1}{2}(b_{10}b_{11} - 2b_{01}b_{20})f^2 + (2a_{01}b_{20} - a_{10}b_{11})fg$$
$$+ \frac{1}{2}(a_{10}a_{11} - 2a_{01}a_{20})g^2$$

Cubic Polynomials of BCW form: As per the requirements of the theorem [BCW] stated above it is enough to consider cubic polynomials of special kind for inversion in the general case. So, let f and g be following cubic polynomials of special [BCW] type:

$$f = a_{10}x + a_{01}y + a_{30}x^3 + a_{21}x^2y + a_{12}xy^2 + a_{03}y^3$$
$$g = b_{10}x + b_{01}y + b_{30}x^3 + b_{21}x^2y + b_{12}xy^2 + b_{03}y^3$$

It is easy to verify that the required inverse polynomials also turn out to be cubic polynomials and they are also of special [BCW] type:

$$x = b_{01}f - a_{01}g + \frac{1}{3}(b_{01}^2 b_{21} - 2b_{01}b_{10}b_{12} + 3b_{10}^2 b_{03})f^3$$
$$+ (-b_{01}^2 a_{21} + 2b_{01}b_{10}a_{12} - 3b_{10}^2 a_{03})f^2 g$$
$$+ (a_{01}^2 b_{21} - 2a_{01}a_{10}b_{12} + 3a_{10}^2 b_{03})fg^2$$
$$+ \frac{1}{3}(-a_{01}^2 a_{21} + 2a_{01}a_{10}a_{12} - 3a_{10}^2 a_{03})g^3$$

$$y = -b_{10}f + a_{10}g + \frac{1}{3}(-3b_{01}^2 b_{30} + 2b_{01}b_{10}b_{21} - b_{10}^2 b_{12})f^3$$
$$+ (3b_{01}^2 a_{30} - 2b_{01}b_{10}a_{21} + b_{10}^2 a_{12})f^2 g$$
$$+ (-3a_{01}^2 b_{30} + 2a_{01}a_{10}b_{21} - a_{10}^2 b_{12})fg^2$$
$$+ \frac{1}{3}(3a_{01}^2 a_{30} - 2a_{01}a_{10}a_{21} + a_{10}^2 a_{12})g^3$$

3. **The Inverse Polynomials for three dimensions:** We now deal below with three dimensional case. We now take polynomials in three variables, i.e. $f = f(x, y, z)$, $g = g(x, y, z)$ and $h = h(x, y, z)$ such that their Jacobian,

$$J_{(x,y,z)}(f, g, h) = \det\begin{bmatrix} f_x & f_y & f_z \\ g_x & g_y & g_z \\ h_x & h_y & h_z \end{bmatrix} = 1 \neq 0$$

We find the inverse polynomials $x = x(f, g, h)$, $y = y(f, g, h)$. and $z = z(f, g, h)$.

Cubic Polynomials of BCW form: As per the requirements of the theorem [BCW] stated above it is enough to consider cubic polynomials of special kind for inversion in the general case. So, we directly deal with general case, i.e. we take f , g and h as following cubic polynomials of special [BCW] type:

Suppose we are given the following polynomials for which the Jacobian is a nonzero constant (=1) as desired.

f=a100*x+a010*y+a001*z+a300*x^3+a210*x^2*y+a201*x^2*z+a120*x*y^2+a111*x*y*z+a102*x*z^2+a030*y^3+a021*y^2*z+a012*y*z^2+a003*z^3

and

g=b100*x+b010*y+b001*z+b300*x^3+b210*x^2*y+b201*x^2*z+b120*x*y^2+b111*x*y*z+b102*x*z^2+b030*y^3+b021*y^2*z+b012*y*z^2+b003*z^3

and

h=c100*x+c010*y+c001*z+c300*x^3+c210*x^2*y+c201*x^2*z+c120*x*y^2+c111*x*y*z+c102*x*z^2+c030*y^3+c021*y^2*z+c012*y*z^2+c003*z^3

aijk, bijk, cijk $\in k$, the field of characteristic zero.

We need to show that x, y, z (which we can express as power series in variables f, g, h using the inverse function theorem) are **actually polynomials.** This claim is true and it is easy to check that the required inverse functions are actually polynomials and further they also turn out to be cubic polynomials having of special [BCW] form as given below:

x=(b010*c001- b001*c010)*f +(a001*c010-a010*c001)*g
+(a010*b001- a001*b010)*h

+1/3*[(b010^2*c001^2+b001^2*c010^2-
2*b010*b001*c010*c001)*(b010*c201+b210*c001-b001*c210-
b201*c010)+(-c100*c010*b001^2+b010*b001*c100*c001-
b100*b010*c001^2+b100*b001*c001*c010)*(b010*c111+2*b12
0*c001-2*b001*c120-b111*c010)+(-c100*c001*b010^2
+b010*b001*c010*c100 +b010*c010*c001*b100-
b001*c010^2*b100)*(2*b010*c102+b111*c001-b001*c111-
2*b102*c010)+(b001^2*c100^2-2*c100*c001*b100*b001
+c001^2*b100^2)*(b010*c021+3*b030*c001-3*b001*c030-
b021*c010)+(-c010*c001*b100^2+c100*b100*b001*c010-
c100^2*b001*b010+c100*c001*b100*b010)*(2*b010*c012+2*b
021*c001-2*b001*c021-2*b012*c010)+(b010^2*c100^2-
2*b010*c100*c010*b100+c010^2*b100^2)*(3*b010*c003+b012
*c010-b001*c012-3*b003*c010)]*f^3

+1/2*[(-2*a001*b001*c010^2+2*a001*c010*c001*b010
+2*c010*c001*b001*a010-2*c001^2*b010*a010)*(b010*c201
+b210*c001-b001*c210-b201*c010) +(a100*c001^2*b010-
a100*b001*c001*c010-c100*c001*a001*b010
+c001^2*a010*b100-c001*b100*a001*c010-
c100*c001*b001*a010+2*b001*c100*a001*c010)*(b010*c111+2
*b120*c001-2*b001*c120-b111*c010)+(-a100*b010*c010*c001
+a100*b001*c010^2+2*b010*a010*c100*c001-
b010*a001*c100*c010-a010*b100*c010*c001
+a001*b100*c010^2-a010*b001*c100*c010)*(2*b010*c102
+b111*c001-b001*c111-2*b102*c010)+(-2*a100*b100*c001^2
+2*a100*b001*c001*c100+2*c001*b100*a001*c100-
2*a001*c100^2*b001)*(b010*c021+3*b030*c001-3*b001*c030-
b021*c010)+(-a100*b010*c100*c001+2*a100*c010*c001*b100-
a100*b001*c010*c100+b010*c100^2*a001-

a010*c001*b100*c100+a010*b001*c100^2-b100*c010*a001*c100)*(2*b010*c012+2*b021*c001-2*b001*c021-2*b012*c010)+(-2*a100*b100*c010^2 +2*a100*b010*c100*c010+2*a010*b100*c100*c010-2*b010*a010*c100^2)*(3*b010*c003+b012*c010-b001*c012-3*b003*c010)]*f^2*g

+1/2*[(-2*a001*c001*b010^2+2*b010*b001*a001*c010-2*a010*b001^2*c010+2*b010*a010*b001*c001)*(b010*c201+b210*c001-b001*c210-b201*c010)+(-a100*b010*b001*c001 +a100*b001^2*c010+2*b010*c001*a001*b100-b010*b001*a001*c100-a010*b001*c001*b100-a001*b001*c010*b100+a010*b001^2*c100)*(b010*c111+2*b120*c001-2*b001*c120-b111*c010)+(a100*b010^2*c001-a100*b010*b001*c010+b010^2*a001*c100-b010*a010*b100*c001-b010*a010*b001*c100-b010*a001*b100*c010+2*a010*b100*b001*c010)*(2*b010*c102 +b111*c001-b001*c111-2*b102*c010)+(-2*a100*c100*b001^2 +2*a100*b100*b001*c001-2*b100^2*c001*a001 +2*c100*b100*b001*a001)*(b010*c021+3*b030*c001-3*b001*c030-b021*c010)+(-a100*b010*c001*b100 +2*a100*b010*c100*b001-a100*b001*c010*b100-b100*b010*c100*a001+a010*c001*b100^2+b100^2*c010*a001-a010*b100*c100*b001)*(2*b010*c012+2*b021*c001-2*b001*c021-2*b012*c010)+(2*a100*b100*b010*c010-2*a100*b010^2*c100-2*a010*b100^2*c010 +2*b010*a010*b100*c100)*(3*b010*c003+b012*c010-b001*c012-3*b003*c010)]*f^2*h

+[(c001^2*a010^2-2*a001*c010*c001*a010 +a001^2*c010^2)*(b010*c201+b210*c001-b001*c210-b201*c010)+(-a100*a010*c001^2+a100*a001*c010*c001 +a010*a001*c100*c001-c100*c010*a001^2)*(b010*c111 +2*b120*c001-2*b001*c120-b111*c010) +(a100*a010*c010*c001-a100*a001*c010^2-c100*c001*a010^2

+a010*a001*c100*c010)*(2*b010*c102+b111*c001-b001*c111-2*b102*c010)+(c001^2*a100^2-2*c100*c001*a001*a100+c100^2*a001^2)*(b010*c021+3*b030*c001-3*b001*c030-b021*c010)+(-c010*c001*a100^2+a100*a010*c100*c001+a100*a001*c100*c010-a010*a001*c100^2)*(2*b010*c012+2*b021*c001-2*b001*c021-2*b012*c010)+(c010^2*a100^2-2*c100*c010*a010*a100+c100^2*a010^2)*(3*b010*c003+b012*c010-b001*c012-3*b003*c010)]*f*g^2

+[(-2*b001*c001*a010^2+2*a010*a001*b010*c001+2*a010*a001*b001*c010-2*b010*c010*a001^2)*(b010*c201+b210*c001-b001*c210-b201*c010)+(-a100*a001*b010*c001+2*a100*a010*c001*b001-a100*a001*c010*b001+a001^2*b010*c100-a010*a001*c001*b100+c010*a001^2*b100-a010*a001*c100*b001)*(b010*c111+2*b120*c001-2*b001*c120-b111*c010)+(-a100*a010*c001*b010+2*a100*a001*c010*b010-a100*a010*b001*c010-a010*a001*c100*b010+c001*a010^2*b100+a010^2*b001*c100-a010*b100*a001*c010)*(2*b010*c102+b111*c001-b001*c111-2*b102*c010)+(-2*b001*c001*a100^2+2*a100*a001*b001*c100+2*a100*a001*b100*c001-2*a001^2*b100*c100)*(b010*c021+3*b030*c001-3*b001*c030-b021*c010)+(a100^2*b010*c001+a100^2*b001*c010-a100*a001*b010*c100-a100*a010*b100*c001-a100*a010*b001*c100-a100*a001*b100*c010+2*a010*b100*a001*c100)*(2*b010*c012+2*b021*c001-2*b001*c021-2*b012*c010)+(-2*b010*c010*a100^2+2*a100*a010*b100*c010+2*a100*a010*b010*c100-2*c100*b100*a010^2)*(3*b010*c003+b012*c010-b001*c012-3*b003*c010)]*f*g*h

+[(b001^2*a010^2-2*b010*b001*a001*a010+b010^2*a001^2)*(b010*c201+b210*c001-b001*c210-b201*c010)+(a100*b010*b001*a001-a100*a010*b001^2-

b100*b010*a001^2+a010*b100*a001*b001)*(b010*c111+2*b120*c001-2*b001*c120-b111*c010)+(-a100*a001*b010^2+a100*a010*b010*b001+a010*b100*a001*b010-b100*b001*a010^2)*(2*b010*c102+b111*c001-b001*c111-2*b102*c010)+(b001^2*a100^2-2*a001*b100*b001*a100+b100^2*a001^2)*(b010*c021+3*b030*c001-3*b001*c030-b021*c010)+(-b010*b001*a100^2+a100*a001*b100*b010+a100*a010*b100*b001-a010*a001*b100^2)*(2*b010*c012+2*b021*c001-2*b001*c021-2*b012*c010)+(b010^2*a100^2-2*b010*a010*b100*a100+b100^2*a010^2)*(3*b010*c003+b012*c010-b001*c012-3*b003*c010)]*f*h^2

+1/3*[(c001^2*a010^2-2*a001*c010*c001*a010+a001^2*c010^2)*(a001*c210+a201*c010-a010*c201-a210*c001)+(-a100*a010*c001^2+a100*a001*c010*c001+a010*a001*c100*c001-c100*c010*a001^2)*(2*a001*c120+a111*c010-a010*c111-2*a120*c001)+(a100*a010*c010*c001-a100*a001*c010^2-c100*c001*a010^2+a010*a001*c100*c010)*(a001*c111+2*a102*c010-2*a010*c102-a111*c001)+(c001^2*a100^2-2*c100*c001*a001*a100+c100^2*a001^2)*(3*a001*c030+a021*c010-a010*c021-3*a030*c001)+(-c010*c001*a100^2+a100*a010*c100*c001+a100*a001*c100*c010-a010*a001*c100^2)*(2*a001*c021+2*a012*c010-2*a010*c012-2*a021*c001)+(c010^2*a100^2-2*c100*c010*a010*a100+c100^2*a010^2)*(a001*c012+3*a003*c010-3*a010*c003-a012*c001)]*g^3

+1/2*[(-2*b001*c001*a010^2+2*a010*a001*b010*c001+2*a010*a001*b001*c010-2*b010*c010*a001^2)*(a001*c210+a201*c010-a010*c201-a210*c001)+(-a100*a001*b010*c001+2*a100*a010*c001*b001-a100*a001*c010*b001+a001^2*b010*c100-a010*a001*c001*b100+c010*a001^2*b100-a010*a001*c100*b001)*(2*a001*c120+a111*c010-a010*c111-2*a120*c001)+(-a100*a010*c001*b010

+2*a100*a001*c010*b010-a100*a010*b001*c010-
a010*a001*c100*b010+c001*a010^2*b100+a010^2*b001*c100-
a010*b100*a001*c010)*(a001*c111+2*a102*c010-2*a010*c102-
a111*c001)+(-2*b001*c001*a100^2+2*a100*a001*b001*c100
+2*a100*a001*b100*c001-2*a001^2*b100*c100)*(3*a001*c030
+a021*c010-a010*c021-3*a030*c001)+(a100^2*b010*c001
+a100^2*b001*c010-a100*a001*b010*c100-
a100*a010*b100*c001-a100*a010*b001*c100-
a100*a001*b100*c010+2*a010*b100*a001*c100)*(2*a001*c021
+2*a012*c010-2*a010*c012-2*a021*c001)
+(2*b010*c010*a100^2+2*a100*a010*b100*c010+2*a100*a010
*b010*c100-2*c100*b100*a010^2)*(a001*c012+3*a003*c010-
3*a010*c003-a012*c001)]*g^2*h

+[(b001^2*a010^2-2*b010*b001*a001*a010
+b010^2*a001^2)*(a001*c210+a201*c010-a010*c201-
a210*c001)+(a100*b010*b001*a001-a100*a010*b001^2-
b100*b010*a001^2+a010*b100*a001*b001)*(2*a001*c120+a111
*c010-a010*c111-2*a120*c001)+(-a100*a001*b010^2
+a100*a010*b010*b001+a010*b100*a001*b010-
b100*b001*a010^2)*(a001*c111+2*a102*c010-2*a010*c102-
a111*c001)+(b001^2*a100^2-2*a001*b100*b001*a100
+b100^2*a001^2)*(3*a001*c030+a021*c010-a010*c021-
3*a030*c001)+(-b010*b001*a100^2+a100*a001*b100*b010
+a100*a010*b100*b001-a010*a001*b100^2)*(2*a001*c021
+2*a012*c010-2*a010*c012-2*a021*c001)+(b010^2*a100^2-
2*b010*a010*b100*a100+b100^2*a010^2)*(a001*c012+3*a003*
c010-3*a010*c003-a012*c001)]*g*h^2

+1/3*[(b001^2*a010^2-2*b010*b001*a001*a010
+b010^2*a001^2)*(a010*b201+a210*b010-a001*b210-
a201*b010)+(a100*b010*b001*a001-a100*a010*b001^2-
b100*b010*a001^2+a010*b100*a001*b001)*(a010*b111+2*a120
*b001-2*a001*b120-a111*b010)+(-a100*a001*b010^2
+a100*a010*b010*b001+a010*b100*a001*b010-

b100*b001*a010^2)*(2*a010*b102+a111*b001-a001*b111-
2*a102*b010)+(b001^2*a100^2-2*a001*b100*b001*a100
+b100^2*a001^2)*(a010*b021+3*a030*b001-3*a001*b030-
a021*b010)+(-b010*b001*a100^2+a100*a001*b100*b010
+a100*a010*b100*b001-a010*a001*b100^2)*(2*a010*b012
+2*a021*b001-2*a001*b021-2*a012*b010)+(b010^2*a100^2-
2*b010*a010*b100*a100+b100^2*a010^2)*(3*a010*b003+a012*
b001-a001*b012-3*a003*b010)]*h^3

and

y=(b001*c100- b100*c001)*f +(a100*c001- a001*c100)*g
+(a001*b100- a100*b001)*h

+1/3*[(b010^2*c001^2+b001^2*c010^2-
2*b010*b001*c010*c001)*(3*b001*c300+b201*c100-
b100*c201-3*b300*c001)+(-c100*c010*b001^2
+b010*b001*c100*c001-b100*b010*c001^2
+b100*b001*c001*c010)*(2*b001*c210+b111*c100-b100*c111-
2*b210*c001)+(-c100*c001*b010^2+b010*b001*c010*c100
+b010*c010*c001*b100-b001*c010^2*b100)*(2*b001*c201
+2*b102*c100-2*b100*c102-2*b201*c001)+(b001^2*c100^2-
2*c100*c001*b100*b001+c001^2*b100^2)*(b001*c120+b021*c1
00-b100*c021-b120*c001)+(-c010*c001*b100^2
+c100*b100*b001*c010-c100^2*b001*b010
+c100*c001*b100*b010)*(b001*c111+2*b012*c100-
2*b100*c012-b111*c001)+(b010^2*c100^2-
2*b010*c100*c010*b100+c010^2*b100^2)*(b001*c102+3*b003
*c100-3*b100*c003-b102*c001)]*f^3

+1/2*[(-2*a001*b001*c010^2+2*a001*c010*c001*b010
+2*c010*c001*b001*a010-2*c001^2*b010*a010)*(3*b001*c300
+b201*c100-b100*c201-3*b300*c001)+(a100*c001^2*b010-
a100*b001*c001*c010-c100*c001*a001*b010
+c001^2*a010*b100-c001*b100*a001*c010-

c100*c001*b001*a010+2*b001*c100*a001*c010)*(2*b001*c210
+b111*c100-b100*c111-2*b210*c001)+(-a100*b010*c010*c001
+a100*b001*c010^2+2*b010*a010*c100*c001-
b010*a001*c100*c010-a010*b100*c010*c001
+a001*b100*c010^2-a010*b001*c100*c010)*(2*b001*c201
+2*b102*c100-2*b100*c102-2*b201*c001)
+(-2*a100*b100*c001^2+2*a100*b001*c001*c100
+2*c001*b100*a001*c100-2*a001*c100^2*b001)*(b001*c120
+b021*c100-b100*c021-b120*c001)
+(-a100*b010*c100*c001+2*a100*c010*c001*b100-
a100*b001*c010*c100+b010*c100^2*a001-
a010*c001*b100*c100+a010*b001*c100^2-
b100*c010*a001*c100)*(b001*c111+2*b012*c100-
2*b100*c012-b111*c001)+(-2*a100*b100*c010^2
+2*a100*b010*c100*c010+2*a010*b100*c100*c010-
2*b010*a010*c100^2)*(b001*c102+3*b003*c100-3*b100*c003-
b102*c001)]*f^2*g

+1/2*[(-2*a001*c001*b010^2+2*b010*b001*a001*c010-
2*a010*b001^2*c010+2*b010*a010*b001*c001)*(3*b001*c300+
b201*c100-b100*c201-3*b300*c001)+(-a100*b010*b001*c001
+a100*b001^2*c010+2*b010*c001*a001*b100-
b010*b001*a001*c100-a010*b001*c001*b100-
a001*b001*c010*b100+a010*b001^2*c100)*(2*b001*c210+b111
*c100-b100*c111-2*b210*c001)+(a100*b010^2*c001-
a100*b010*b001*c010+b010^2*a001*c100-
b010*a010*b100*c001-b010*a010*b001*c100-
b010*a001*b100*c010+2*a010*b100*b001*c010)*(2*b001*c201
+2*b102*c100-2*b100*c102-2*b201*c001)
+(-2*a100*c100*b001^2+2*a100*b100*b001*c001-
2*b100^2*c001*a001+2*c100*b100*b001*a001)*(b001*c120+b0
21*c100-b100*c021-b120*c001)+(-a100*b010*c001*b100
+2*a100*b010*c100*b001-a100*b001*c010*b100-
b100*b010*c100*a001+a010*c001*b100^2+b100^2*c010*a001-
a010*b100*c100*b001)*(b001*c111+2*b012*c100-

2*b100*c012-b111*c001)+(2*a100*b100*b010*c010-2*a100*b010^2*c100-2*a010*b100^2*c010+2*b010*a010*b100*c100)*(b001*c102+3*b003*c100-3*b100*c003-b102*c001)]*f^2*h

+[(c001^2*a010^2-2*a001*c010*c001*a010+a001^2*c010^2)*(3*b001*c300+b201*c100-b100*c201-3*b300*c001)+(-a100*a010*c001^2+a100*a001*c010*c001+a010*a001*c100*c001-c100*c010*a001^2)*(2*b001*c210+b111*c100-b100*c111-2*b210*c001)+(a100*a010*c010*c001-a100*a001*c010^2-c100*c001*a010^2+a010*a001*c100*c010)*(2*b001*c201+2*b102*c100-2*b100*c102-2*b201*c001)+(c001^2*a100^2-2*c100*c001*a001*a100+c100^2*a001^2)*(b001*c120+b021*c100-b100*c021-b120*c001)+(-c010*c001*a100^2+a100*a010*c100*c001+a100*a001*c100*c010-a010*a001*c100^2)*(b001*c111+2*b012*c100-2*b100*c012-b111*c001)+(c010^2*a100^2-2*c100*c010*a010*a100+c100^2*a010^2)*(b001*c102+3*b003*c100-3*b100*c003-b102*c001)]*f*g^2

+[(-2*b001*c001*a010^2+2*a010*a001*b010*c001+2*a010*a001*b001*c010-2*b010*c010*a001^2)*(3*b001*c300+b201*c100-b100*c201-3*b300*c001)+(-a100*a001*b010*c001+2*a100*a010*c001*b001-a100*a001*c010*b001+a001^2*b010*c100-a010*a001*c001*b100+c010*a001^2*b100-a010*a001*c100*b001)*(2*b001*c210+b111*c100-b100*c111-2*b210*c001)+(-a100*a010*c001*b010+2*a100*a001*c010*b010-a100*a010*b001*c010-a010*a001*c100*b010+c001*a010^2*b100+a010^2*b001*c100-a010*b100*a001*c010)*(2*b001*c201+2*b102*c100-2*b100*c102-2*b201*c001)+(-2*b001*c001*a100^2+2*a100*a001*b001*c100+2*a100*a001*b100*c001-2*a001^2*b100*c100)*(b001*c120+b021*c100-b100*c021-b120*c001)+(a100^2*b010*c001+a100^2*b001*c010-

a100*a001*b010*c100-a100*a010*b100*c001-
a100*a010*b001*c100-a100*a001*b100*c010
+2*a010*b100*a001*c100)*(b001*c111+2*b012*c100-
2*b100*c012-b111*c001)+(-2*b010*c010*a100^2
+2*a100*a010*b100*c010+2*a100*a010*b010*c100-
2*c100*b100*a010^2)*(b001*c102+3*b003*c100-3*b100*c003-
b102*c001)]*f*g*h

+[(b001^2*a010^2-2*b010*b001*a001*a010
+b010^2*a001^2)*(3*b001*c300+b201*c100-b100*c201-
3*b300*c001)+(a100*b010*b001*a001-a100*a010*b001^2-
b100*b010*a001^2+a010*b100*a001*b001)*(2*b001*c210+b11
1*c100-b100*c111-2*b210*c001)+(-a100*a001*b010^2
+a100*a010*b010*b001+a010*b100*a001*b010-
b100*b001*a010^2)*(2*b001*c201+2*b102*c100-2*b100*c102-
2*b201*c001)+(b001^2*a100^2-2*a001*b100*b001*a100
+b100^2*a001^2)*(b001*c120+b021*c100-b100*c021-
b120*c001)+(-b010*b001*a100^2+a100*a001*b100*b010
+a100*a010*b100*b001-a010*a001*b100^2)*(b001*c111
+2*b012*c100-2*b100*c012-b111*c001)+(b010^2*a100^2-
2*b010*a010*b100*a100+b100^2*a010^2)*(b001*c102+3*b003
*c100-3*b100*c003-b102*c001)]*f*h^2

+1/3*[(c001^2*a010^2-
2*a001*c010*c001*a010+a001^2*c010^2)*(a100*c201+3*a300*
c001-3*a001*c300-a201*c100)+(-a100*a010*c001^2
+a100*a001*c010*c001+a010*a001*c100*c001-
c100*c010*a001^2)*(a100*c111+2*a210*c001-2*a001*c210-
a111*c100)+(a100*a010*c010*c001-a100*a001*c010^2-
c100*c001*a010^2+a010*a001*c100*c010)*(2*a100*c102+2*a2
01*c001-2*a001*c201-2*a102*c100)+(c001^2*a100^2-
2*c100*c001*a001*a100+c100^2*a001^2)*(a100*c021+a120*c0
01-a001*c120-a021*c100)+(-c010*c001*a100^2
+a100*a010*c100*c001+a100*a001*c100*c010-
a010*a001*c100^2)*(2*a100*c012+a111*c001-a001*c111-

2*a012*c100)+(c010^2*a100^2-2*c100*c010*a010*a100
+c100^2*a010^2)*(3*a100*c003+a102*c001-a001*c102-
3*a003*c100)]*g^3

+1/2*[(-2*b001*c001*a010^2+2*a010*a001*b010*c001
+2*a010*a001*b001*c010-2*b010*c010*a001^2)*(a100*c201
+3*a300*c001-3*a001*c300-a201*c100)
+(-a100*a001*b010*c001+2*a100*a010*c001*b001-
a100*a001*c010*b001+a001^2*b010*c100-
a010*a001*c001*b100+c010*a001^2*b100-
a010*a001*c100*b001)*(a100*c111+2*a210*c001-2*a001*c210-
a111*c100)+(-a100*a010*c001*b010+2*a100*a001*c010*b010-
a100*a010*b001*c010-a010*a001*c100*b010
+c001*a010^2*b100+a010^2*b001*c100-
a010*b100*a001*c010)*(2*a100*c102+2*a201*c001-
2*a001*c201-2*a102*c100)+(-2*b001*c001*a100^2
+2*a100*a001*b001*c100+2*a100*a001*b100*c001-
2*a001^2*b100*c100)*(a100*c021+a120*c001-a001*c120-
a021*c100)+(a100^2*b010*c001+a100^2*b001*c010-
a100*a001*b010*c100-a100*a010*b100*c001-
a100*a010*b001*c100-a100*a001*b100*c010
+2*a010*b100*a001*c100)*(2*a100*c012+a111*c001-
a001*c111-2*a012*c100)+(-2*b010*c010*a100^2
+2*a100*a010*b100*c010+2*a100*a010*b010*c100-
2*c100*b100*a010^2)*(3*a100*c003+a102*c001-a001*c102-
3*a003*c100)]*g^2*h

+[(b001^2*a010^2-2*b010*b001*a001*a010
+b010^2*a001^2)*(a100*c201+3*a300*c001-3*a001*c300-
a201*c100)+(a100*b010*b001*a001-a100*a010*b001^2-
b100*b010*a001^2+a010*b100*a001*b001)*(a100*c111+2*a210
*c001-2*a001*c210-a111*c100)+(-a100*a001*b010^2
+a100*a010*b010*b001+a010*b100*a001*b010-
b100*b001*a010^2)*(2*a100*c102+2*a201*c001-2*a001*c201-
2*a102*c100)+(b001^2*a100^2-2*a001*b100*b001*a100

+b100^2*a001^2)*(a100*c021+a120*c001-a001*c120-
a021*c100)+(-b010*b001*a100^2+a100*a001*b100*b010
+a100*a010*b100*b001-a010*a001*b100^2)*(2*a100*c012
+a111*c001-a001*c111-2*a012*c100)+(b010^2*a100^2-
2*b010*a010*b100*a100+b100^2*a010^2)*(3*a100*c003+a102*
c001-a001*c102-3*a003*c100)]*g*h^2

+1/3*[(b001^2*a010^2-2*b010*b001*a001*a010
+b010^2*a001^2)*(3*a001*b300+a201*b100-a100*b201-
3*a300*b001)+(a100*b010*b001*a001-a100*a010*b001^2-
b100*b010*a001^2+a010*b100*a001*b001)*(2*a001*b210+a111
*b100-a100*b111-2*a210*b001)+(-a100*a001*b010^2
+a100*a010*b010*b001+a010*b100*a001*b010-
b100*b001*a010^2)*(2*a001*b201+2*a102*b100-2*a100*b102-
2*a201*b001)+(b001^2*a100^2-2*a001*b100*b001*a100
+b100^2*a001^2)*(a001*b120+a021*b100-a100*b021-
a120*b001)+(-b010*b001*a100^2+a100*a001*b100*b010
+a100*a010*b100*b001-a010*a001*b100^2)*(a001*b111
+2*a012*b100-2*a100*b012-a111*b001)+(b010^2*a100^2-
2*b010*a010*b100*a100+b100^2*a010^2)*(a001*b102+3*a003*
b100-3*a100*b003-a102*b001)]*h^3

and

z=(b100*c010- b010*c100)*f +(a010*c100- a100*c010)*g
+(a100*b010- a010*b100)*h

+1/3*[(b010^2*c001^2+b001^2*c010^2-
2*b010*b001*c010*c001)*(b100*c210+3*b300*c010-
3*b010*c300-b210*c100)+(-c100*c010*b001^2
+b010*b001*c100*c001-b100*b010*c001^2
+b100*b001*c001*c010)*(2*b100*c120+2*b210*c010-
2*b010*c210-2*b120*c100)+(-c100*c001*b010^2
+b010*b001*c010*c100+b010*c010*c001*b100-
b001*c010^2*b100)*(b100*c111+2*b201*c010-2*b010*c201-

b111*c100)+(b001^2*c100^2-2*c100*c001*b100*b001 +c001^2*b100^2)*(3*b100*c030+b120*c010-b010*c120-3*b030*c100)+(-c010*c001*b100^2+c100*b100*b001*c010-c100^2*b001*b010+c100*c001*b100*b010)*(2*b100*c021+b111*c010-b010*c111-2*b021*c100)+(b010^2*c100^2-2*b010*c100*c010*b100+c010^2*b100^2)*(b100*c012+b102*c010-b010*c102-b012*c100)]*f^3

+1/2*[(-2*a001*b001*c010^2+2*a001*c010*c001*b010 +2*c010*c001*b001*a010-2*c001^2*b010*a010)*(b100*c210 +3*b300*c010-3*b010*c300-b210*c100)+(a100*c001^2*b010-a100*b001*c001*c010-c100*c001*a001*b010 +c001^2*a010*b100-c001*b100*a001*c010-c100*c001*b001*a010+2*b001*c100*a001*c010)*(2*b100*c120 +2*b210*c010-2*b010*c210-2*b120*c100) +(a100*b010*c010*c001+a100*b001*c010^2 +2*b010*a010*c100*c001-b010*a001*c100*c010-a010*b100*c010*c001+a001*b100*c010^2-a010*b001*c100*c010)*(b100*c111+2*b201*c010-2*b010*c201-b111*c100)+(-2*a100*b100*c001^2 +2*a100*b001*c001*c100+2*c001*b100*a001*c100-2*a001*c100^2*b001)*(3*b100*c030+b120*c010-b010*c120-3*b030*c100)+(-a100*b010*c100*c001 +2*a100*c010*c001*b100-a100*b001*c010*c100 +b010*c100^2*a001-a010*c001*b100*c100+a010*b001*c100^2-b100*c010*a001*c100)*(2*b100*c021+b111*c010-b010*c111-2*b021*c100)+(-2*a100*b100*c010^2 +2*a100*b010*c100*c010+2*a010*b100*c100*c010-2*b010*a010*c100^2)*(b100*c012+b102*c010-b010*c102-b012*c100)]*f^2*g

+1/2*[(-2*a001*c001*b010^2+2*b010*b001*a001*c010-2*a010*b001^2*c010+2*b010*a010*b001*c001)*(b100*c210+3*b300*c010-3*b010*c300-b210*c100) +(a100*b010*b001*c001+a100*b001^2*c010

+2*b010*c001*a001*b100-b010*b001*a001*c100-
a010*b001*c001*b100-a001*b001*c010*b100
+a010*b001^2*c100)*(2*b100*c120+2*b210*c010-
2*b010*c210-2*b120*c100)+(a100*b010^2*c001-
a100*b010*b001*c010+b010^2*a001*c100-
b010*a010*b100*c001-b010*a010*b001*c100-
b010*a001*b100*c010+2*a010*b100*b001*c010)*(b100*c111+
2*b201*c010-2*b010*c201-b111*c100)
+(-2*a100*c100*b001^2+2*a100*b100*b001*c001-
2*b100^2*c001*a001+2*c100*b100*b001*a001)*(3*b100*c030+
b120*c010-b010*c120-3*b030*c100)
+(-a100*b010*c001*b100+2*a100*b010*c100*b001-
a100*b001*c010*b100-b100*b010*c100*a001
+a010*c001*b100^2+b100^2*c010*a001-
a010*b100*c100*b001)*(2*b100*c021+b111*c010-b010*c111-
2*b021*c100)+(2*a100*b100*b010*c010-2*a100*b010^2*c100-
2*a010*b100^2*c010+2*b010*a010*b100*c100)*(b100*c012
+b102*c010-b010*c102-b012*c100)]*f^2*h

+[(c001^2*a010^2-2*a001*c010*c001*a010
+a001^2*c010^2)*(b100*c210+3*b300*c010-3*b010*c300-
b210*c100)+(-a100*a010*c001^2
+a100*a001*c010*c001+a010*a001*c100*c001-
c100*c010*a001^2)*(2*b100*c120+2*b210*c010-2*b010*c210-
2*b120*c100)+(a100*a010*c010*c001-a100*a001*c010^2-
c100*c001*a010^2+a010*a001*c100*c010)*(b100*c111+2*b201
*c010-2*b010*c201-b111*c100)+(c001^2*a100^2-
2*c100*c001*a001*a100+c100^2*a001^2)*(3*b100*c030+b120*
c010-b010*c120-3*b030*c100) +(-c010*c001*a100^2
+a100*a010*c100*c001+a100*a001*c100*c010-
a010*a001*c100^2)*(2*b100*c021+b111*c010-b010*c111-
2*b021*c100)+(c010^2*a100^2-2*c100*c010*a010*a100
+c100^2*a010^2)*(b100*c012+b102*c010-b010*c102-
b012*c100)]*f*g^2

+[(-2*b001*c001*a010^2+2*a010*a001*b010*c001
+2*a010*a001*b001*c010-2*b010*c010*a001^2)*(b100*c210
+3*b300*c010-3*b010*c300-b210*c100)
+(-a100*a001*b010*c001+2*a100*a010*c001*b001-
a100*a001*c010*b001+a001^2*b010*c100-
a010*a001*c001*b100+c010*a001^2*b100-
a010*a001*c100*b001)*(2*b100*c120+2*b210*c010-
2*b010*c210-2*b120*c100)+(-a100*a010*c001*b010
+2*a100*a001*c010*b010-a100*a010*b001*c010-
a010*a001*c100*b010+c001*a010^2*b100+a010^2*b001*c100-
a010*b100*a001*c010)*(b100*c111+2*b201*c010-
2*b010*c201-b111*c100)+(-2*b001*c001*a100^2
+2*a100*a001*b001*c100+2*a100*a001*b100*c001-
2*a001^2*b100*c100)*(3*b100*c030+b120*c010-b010*c120-
3*b030*c100)+(a100^2*b010*c001+a100^2*b001*c010-
a100*a001*b010*c100-a100*a010*b100*c001-
a100*a010*b001*c100-a100*a001*b100*c010
+2*a010*b100*a001*c100)*(2*b100*c021+b111*c010-
b010*c111-2*b021*c100)+(-2*b010*c010*a100^2
+2*a100*a010*b100*c010+2*a100*a010*b010*c100-
2*c100*b100*a010^2)*(b100*c012+b102*c010-b010*c102-
b012*c100)]*f*g*h

+[(b001^2*a010^2-2*b010*b001*a001*a010
+b010^2*a001^2)*(b100*c210+3*b300*c010-3*b010*c300-
b210*c100)+(a100*b010*b001*a001-a100*a010*b001^2-
b100*b010*a001^2+a010*b100*a001*b001)*(2*b100*c120+2*b
210*c010-2*b010*c210-2*b120*c100)+(-a100*a001*b010^2
+a100*a010*b010*b001+a010*b100*a001*b010-
b100*b001*a010^2)*(b100*c111+2*b201*c010-2*b010*c201-
b111*c100)+(b001^2*a100^2-2*a001*b100*b001*a100
+b100^2*a001^2)*(3*b100*c030+b120*c010-b010*c120-
3*b030*c100)+(-b010*b001*a100^2
+a100*a001*b100*b010+a100*a010*b100*b001-
a010*a001*b100^2)*(2*b100*c021+b111*c010-b010*c111-

2*b021*c100)+(b010^2*a100^2-2*b010*a010*b100*a100
+b100^2*a010^2)*(b100*c012+b102*c010-b010*c102-
b012*c100)]*f*h^2

+1/3*[(c001^2*a010^2-2*a001*c010*c001*a010
+a001^2*c010^2)*(3*a010*c300+a210*c100-a100*c210-
3*a300*c010)+(-a100*a010*c001^2
+a100*a001*c010*c001+a010*a001*c100*c001-
c100*c010*a001^2)*(2*a010*c210+2*a120*c100-2*a100*c120-
2*a210*c010)+(a100*a010*c010*c001-a100*a001*c010^2-
c100*c001*a010^2+a010*a001*c100*c010)*(2*a010*c201+a111
*c100-a100*c111-2*a201*c010)+(c001^2*a100^2-
2*c100*c001*a001*a100+c100^2*a001^2)*(a010*c120+3*a030*
c100-3*a100*c030-a120*c010)+(-c010*c001*a100^2
+a100*a010*c100*c001+a100*a001*c100*c010-
a010*a001*c100^2)*(a010*c111+2*a021*c100-2*a100*c021-
a111*c010)+(c010^2*a100^2-2*c100*c010*a010*a100
+c100^2*a010^2)*(a010*c102+a012*c100-a100*c012-
a102*c010)]*g^3

+1/2*[(-2*b001*c001*a010^2+2*a010*a001*b010*c001
+2*a010*a001*b001*c010-2*b010*c010*a001^2)*(3*a010*c300
+a210*c100-a100*c210-3*a300*c010)+(-a100*a001*b010*c001
+2*a100*a010*c001*b001-a100*a001*c010*b001
+a001^2*b010*c100-a010*a001*c001*b100+c010*a001^2*b100-
a010*a001*c100*b001)*(2*a010*c210+2*a120*c100-
2*a100*c120-2*a210*c010)
+(-a100*a010*c001*b010+2*a100*a001*c010*b010-
a100*a010*b001*c010-a010*a001*c100*b010
+c001*a010^2*b100+a010^2*b001*c100-
a010*b100*a001*c010)*(2*a010*c201+a111*c100-a100*c111-
2*a201*c010)+(-2*b001*c001*a100^2
+2*a100*a001*b001*c100+2*a100*a001*b100*c001-
2*a001^2*b100*c100)*(a010*c120+3*a030*c100-3*a100*c030-
a120*c010)+(a100^2*b010*c001+a100^2*b001*c010-

a100*a001*b010*c100-a100*a010*b100*c001-
a100*a010*b001*c100-a100*a001*b100*c010
+2*a010*b100*a001*c100)*(a010*c111+2*a021*c100-
2*a100*c021-a111*c010)+(-2*b010*c010*a100^2
+2*a100*a010*b100*c010+2*a100*a010*b010*c100-
2*c100*b100*a010^2)*(a010*c102+a012*c100-a100*c012-
a102*c010)]*g^2*h

+[(b001^2*a010^2-2*b010*b001*a001*a010
+b010^2*a001^2)*(3*a010*c300+a210*c100-a100*c210-
3*a300*c010)+(a100*b010*b001*a001-a100*a010*b001^2-
b100*b010*a001^2+a010*b100*a001*b001)*(2*a010*c210+2*a1
20*c100-2*a100*c120-2*a210*c010)+(-a100*a001*b010^2
+a100*a010*b010*b001+a010*b100*a001*b010-
b100*b001*a010^2)*(2*a010*c201+a111*c100-a100*c111-
2*a201*c010)+(b001^2*a100^2-2*a001*b100*b001*a100
+b100^2*a001^2)*(a010*c120+3*a030*c100-3*a100*c030-
a120*c010)+(-b010*b001*a100^2
+a100*a001*b100*b010+a100*a010*b100*b001-
a010*a001*b100^2)*(a010*c111+2*a021*c100-2*a100*c021-
a111*c010)+(b010^2*a100^2-2*b010*a010*b100*a100
+b100^2*a010^2)*(a010*c102+a012*c100-a100*c012-
a102*c010)]*g*h^2

+1/3*[(b001^2*a010^2-2*b010*b001*a001*a010
+b010^2*a001^2)*(a100*b210+3*a300*b010-3*a010*b300-
a210*b100)+(a100*b010*b001*a001-a100*a010*b001^2-
b100*b010*a001^2+a010*b100*a001*b001)*(2*a100*b120+2*a2
10*b010-2*a010*b210-2*a120*b100)+(-a100*a001*b010^2
+a100*a010*b010*b001+a010*b100*a001*b010-
b100*b001*a010^2)*(a100*b111+2*a201*b010-2*a010*b201-
a111*b100)+(b001^2*a100^2-2*a001*b100*b001*a100
+b100^2*a001^2)*(3*a100*b030+a120*b010-a010*b120-
3*a030*b100)+(-b010*b001*a100^2
+a100*a001*b100*b010+a100*a010*b100*b001-

a010*a001*b100^2)*(2*a100*b021+a111*b010-a010*b111-2*a021*b100)+(b010^2*a100^2-2*b010*a010*b100*a100+b100^2*a010^2)*(a100*b012+a102*b010-a010*b102-a012*b100)]*h^3

4. **The Inverse Polynomials for more dimensions:** It is actually possible to construct the inverse polynomials (and thus to directly verify the fact that the inverse functions are actually polynomials and not power series) for any higher dimensional cases, i.e. for four, five, six, ... variables cases also .

References

1. S. S. Abhyankar, Expansion Techniques in Algebraic Geometry, Tata Institute of Fundamental Research, Mumbai, 1977.
2. O. H. Keller, Ganze Cremona -Transformationen, Monatsh. Math. Physik, 47, 299-306, 1939.
3. H. Bass, E. H. Connell, and D. Wright, The Jacobian Conjecture: Reduction of Degree and Formal Expansion of the Inverse. Bull. Amer. Math. Soc., 7, 287-330, 1982.