A Parametric Identity for the Sum of Three Cubes and Its Special Cases

Abstract

This paper presents a master polynomial identity with 11 variables (**a, b, c, d, e, f, g, h, x, y, z**) that decomposes the sum of three cubes into a product of three factors. In addition, a special case obtained via a dedicated substitution and parameter permutation is provided, along with several standalone sums of three cubes featuring a similar structural property. All identities have been symbolically verified using the SymPy library in Python.

1. Master Parametric Identity

For arbitrary values of variables and parameters, the following identity holds:

$$
\begin{aligned}
&a*((a*d^3+b*e^3+c*f^3)*((d-h*x)*(a*x^3+b*y^3+c*z^3)+g*x)+g*h*(3*(b*d*e*y^2\\
&\quad+c*d*f*z^2-b*e^2*x*y-c*f^2*x*z)+g*(d-h*x)-d*h*(a*x^3+b*y^3+c\\
&\quad*z^3)))^3\\
&+b*((a*d^3+b*e^3+c*f^3)*((e-h*y)*(a*x^3+b*y^3+c*z^3)+g*y)+g*h*(3*(a*d*e*x^2\\
&\quad+c*e*f*z^2-a*d^2*x*y-c*f^2*y*z)+g*(e-h*y)-e*h*(a*x^3+b*y^3+c\\
&\quad*z^3)))^3\\
&+c*((a*d^3+b*e^3+c*f^3)*((f-h*z)*(a*x^3+b*y^3+c*z^3)+g*z)+g*h*(3*(a*d*f*x^2\\
&\quad+b*e*f*y^2-a*d^2*x*z-b*e^2*y*z)+g*(f-h*z)-f*h*(a*x^3+b*y^3+c\\
&\quad*z^3)))^3\\
&=(a*(d-h*x)^3+b*(e-h*y)^3+c*(f-h*z)^3)*\\
&((a*x^3+b*y^3+c*z^3)^2*(a*d^3+b*e^3+c*f^3)+3*g*(a*x^3+b*y^3+c*z^3)*(a*d^2*x+b*e^2\\
&\quad*y+c*f^2*z)+3*g^2*(a*d*x^2+b*e*y^2+c*f*z^2)+g^3)*\\
&((a*x^3+b*y^3+c*z^3)*(a*d^3+b*e^3+c*f^3)^2+3*g*h*(a*d^3+b*e^3+c*f^3)*(a*d*x^2+b\\
&\quad*e*y^2+c*f*z^2)+3*g^2*h^2*(a*d^2*x+b*e^2*y+c*f^2*z)+g^3*h^3)
\end{aligned}
$$

Identity No. 1 is symmetric under simultaneous permutations of the triples **(a, d, x)**, **(b, e, y)**, and **(c, f, z)**. The correctness of this identity has been verified via symbolic computations in SymPy (see Listing 1).

2. Special Case

Under a specialized choice and permutation of parameters, the master identity reduces to the following form (Identity No. 2):

$$
\begin{aligned}
&\Big((a-b)*(2*a^3*c^4-2*a^3*c^3*e-a^3*c*d*e^2+2*a^2*b*c^4+2*a^2*b*c^3*d-2*a^2*b*c^3\\
&\quad*e+a^2*b*c*d^2*e+2*a^2*b*c*d*e^2-a^2*b*d^2*e^2+2*a*b^2*c^3*d-2*a\\
&\quad*b^2*c*d^2*e-a*b^2*c*d*e^2+2*a*b^2*d^2*e^2+b^3*c*d^2*e-b^3*d^2*e^2)\Big)^3\\
&-\frac{a}{b}*(2*a^4*c^3*e+4*a^3*b*c^4-2*a^3*b*c^3*d-2*a^3*b*c*d*e^2+a^3*b*d^2*e^2+4*a^2\\
&\quad*b^2*c^4-2*a^2*b^2*c^3*e+2*a^2*b^2*c*d^2*e+4*a^2*b^2*c*d*e^2-3*a^2*b^2\\
&\quad*d^2*e^2+2*a*b^3*c^3*d-4*a*b^3*c*d^2*e-2*a*b^3*c*d*e^2+3*a*b^3*d^2\\
&\quad*e^2+2*b^4*c*d^2*e-b^4*d^2*e^2)^3\\
&+(2*a^4*c^4+2*a^4*c^3*e-a^4*c*d*e^2+4*a^3*b*c^4-2*a^3*b*c^3*d+a^3*b*c*d^2*e+a^3\\
&\quad*b*c*d*e^2+a^3*b*d^2*e^2+2*a^2*b^2*c^4-2*a^2*b^2*c^3*e-a^2*b^2*c*d^2*e\\
&\quad+a^2*b^2*c*d*e^2-3*a^2*b^2*d^2*e^2+2*a*b^3*c^3*d-a*b^3*c*d^2*e-a*b^3\\
&\quad*c*d*e^2+3*a*b^3*d^2*e^2+b^4*c*d^2*e-b^4*d^2*e^2)^3\\
&=-\frac{a}{b}*(a-b)*(a^2*e^3-2*a*b*c^3-2*a*b*e^3-2*b^2*c^3+b^2*e^3)\\
&\quad*(2*a^4*c^3+2*a^3*b*c^3+a^2*b^2*d^3-2*a*b^3*d^3+b^4*d^3)\\
&\quad*\big(4*a^5*c^6+8*a^4*b*c^6+a^4*b*d^3*e^3+4*a^3*b^2*c^6-4*a^3*b^2*d^3*e^3+6\\
&\quad*a^2*b^3*d^3*e^3-4*a*b^4*d^3*e^3+b^5*d^3*e^3\big)
\end{aligned}
$$

The correctness of this identity has been verified via symbolic computations in SymPy (see Listing 2).

Below are additional autonomous families of parametric solutions. These structures exhibit different degrees of polynomial homogeneity and are not derived from Identity No. 1 via direct parameter substitution, representing independent branches of cubic forms.

3. Identity No. 3:

$$(a^4 + a^3*d - a^3*e + a*b^3 + 3*a*b*d*e + a*c^3 + 3*a*c*d*e + 3*a*d^2*e - 3*a*d*e^2 + b^3*d - b^3*e - 3*b^2*d*e + c^3*d - c^3*e - 3*c^2*d*e - 3*d^2*e^2)^3$$
$$+(a^3*b + a^3*d - a^3*e - 3*a^2*d*e + 3*a*b*d*e + b^4 + b^3*d - b^3*e + b*c^3 + 3*b*c*d*e + 3*b*d^2*e - 3*b*d*e^2 + c^3*d - c^3*e - 3*c^2*d*e - 3*d^2*e^2)^3$$
$$+(a^3*c + a^3*d - a^3*e - 3*a^2*d*e + 3*a*c*d*e + b^3*c + b^3*d - b^3*e - 3*b^2*d*e + 3*b*c*d*e + c^4 + c^3*d - c^3*e + 3*c*d^2*e - 3*c*d*e^2 - 3*d^2*e^2)^3$$
$$= ((a+d)^3 + (b+d)^3 + (c+d)^3) * ((a-e)^3 + (b-e)^3 + (c-e)^3) *$$
$$(a^6 + 3*a^4*d*e + 2*a^3*b^3 + 3*a^3*b*d*e + 2*a^3*c^3 + 3*a^3*c*d*e + 9*a^2*d^2*e^2 + 3*a*b^3*d*e + 3*a*c^3*d*e + b^6 + 3*b^4*d*e + 2*b^3*c^3 + 3*b^3*c*d*e + 9*b^2*d^2*e^2 + 3*b*c^3*d*e + c^6 + 3*c^4*d*e + 9*c^2*d^2*e^2 + 9*d^3*e^3)$$

The correctness of this identity has been verified via symbolic computations in SymPy (see Listing 3).

4. Identity No. 4:

$$(a^4 + a^3*d - a^3*e + a*b^3 + 3*a*b*d*e + a*c^3 + 3*a*c*d*e + a*d^2*e - a*d*e^2 + b^3*d - b^3*e + 3*b^2*d*e + c^3*d - c^3*e - 3*c^2*d*e - d^2*e^2)^3$$
$$+(a^3*b - a^3*d + a^3*e + 3*a^2*d*e + 3*a*b*d*e + b^4 - b^3*d + b^3*e + b*c^3 + 3*b*c*d*e + b*d^2*e - b*d*e^2 - c^3*d + c^3*e + 3*c^2*d*e + d^2*e^2)^3$$
$$+(a^3*c + a^3*d - a^3*e - 3*a^2*d*e + 3*a*c*d*e + b^3*c + b^3*d - b^3*e + 3*b^2*d*e + 3*b*c*d*e + c^4 + c^3*d - c^3*e + c*d^2*e - c*d*e^2 - d^2*e^2)^3$$
$$=((a+d)^3 + (b-d)^3 + (c+d)^3) * ((a-e)^3 + (b+e)^3 + (c-e)^3) *$$
$$(a^6 + 3*a^4*d*e + 2*a^3*b^3 + 3*a^3*b*d*e + 2*a^3*c^3 + 3*a^3*c*d*e + 3*a^2*d^2*e^2 + 3*a*b^3*d*e + 3*a*c^3*d*e + b^6 + 3*b^4*d*e + 2*b^3*c^3 + 3*b^3*c*d*e - 3*b^2*d^2*e^2 + 3*b*c^3*d*e + c^6 + 3*c^4*d*e + 3*c^2*d^2*e^2 + d^3*e^3)$$

The correctness of this identity has been verified via symbolic computations in SymPy (see Listing 4).

5. Identity No. 5:

$$(a^4 + a^3*d - a^3*e + a*b^3 + a*c^3 + 3*a*c*d*e + 2*a*d^2*e - 2*a*d*e^2 + b^3*d - b^3*e + c^3*d - c^3*e - 3*c^2*d*e - 2*d^2*e^2)^3$$
$$+ b^3*(a^3 + 3*a*d*e + b^3 + c^3 + 3*c*d*e + 2*d^2*e - 2*d*e^2)^3$$
$$+ (a^3*c + a^3*d - a^3*e - 3*a^2*d*e + 3*a*c*d*e + b^3*c + b^3*d - b^3*e + c^4 + c^3*d - c^3*e + 2*c*d^2*e - 2*c*d*e^2 - 2*d^2*e^2)^3$$
$$= ((a+d)^3 + b^3 + (c+d)^3) * ((a-e)^3 + b^3 + (c-e)^3) *$$
$$(a^6 + 3*a^4*d*e + 2*a^3*b^3 + 2*a^3*c^3 + 3*a^3*c*d*e + 6*a^2*d^2*e^2 + 3*a*b^3*d*e + 3*a*c^3*d*e + b^6 + 2*b^3*c^3 + 3*b^3*c*d*e + c^6 + 3*c^4*d*e + 6*c^2*d^2*e^2 + 4*d^3*e^3)$$

The correctness of this identity has been verified via symbolic computations in SymPy (see Listing 5).

6. Identity No. 6:

$$(a^3 - 3*a*b*d + a*d*e*f + b^3 - 3*b^2*d + c^3 + d^2*e*f)^3$$
$$+(-a^3 - 3*a^2*d - 3*a*b*d - b^3 + b*d*e*f - c^3 - d^2*e*f)^3$$
$$+ (c*d*(-3*a - 3*b + e*f))^3$$
$$= d*(-3*a - 3*b + e*f) * ((a+d)^3 + (b-d)^3 + c^3) * (3*a^4 + 3*a^3*b + 3*a^2*d*e*f + 3*a*b^3 + 3*a*c^3 + 3*b^4 - 3*b^2*d*e*f + 3*b*c^3 + d^2*e^2*f^2)$$

The correctness of this identity has been verified via symbolic computations in SymPy (see Listing 6).

Keywords
Diophantine equations, algebraic identities, sum of three cubes, parametric solutions, number theory, symbolic computation.

Conclusion

For each identity (Nos. 1–6), the difference between the left-hand side and the right-hand side simplifies identically to zero.

These identities may provide useful algebraic forms for further investigations of cubic surfaces and rational points.

Appendix A. Verification Scripts (SymPy)

Listing 1

```
import sympy as sp

# 1. Variables
a, b, c, d, e, f, g, h, x, y, z = sp.symbols('a b c d e f g h x y z')

# 2. Left-hand side (LHS)
lhs_term1 = a * ((a*d**3+ b*e**3+ c*f**3)*((d - h*x)*(a*x**3+ b*y**3+ c*z**3)+ g*x)
            + g*h*(3*(b*d*e*y**2+ c*d*f*z**2 - b*e**2*x*y - c*f**2*x*z)
                + g*(d - h*x) - d*h*(a*x**3+ b*y**3+ c*z**3)))**3

lhs_term2 = b * ((a*d**3+ b*e**3+ c*f**3)*((e - h*y)*(a*x**3+ b*y**3+ c*z**3)+ g*y)
            + g*h*(3*(a*d*e*x**2+ c*e*f*z**2 - a*d**2*x*y - c*f**2*y*z)
                + g*(e - h*y) - e*h*(a*x**3+ b*y**3+ c*z**3)))**3

lhs_term3 = c * ((a*d**3+ b*e**3+ c*f**3)*((f - h*z)*(a*x**3+ b*y**3+ c*z**3)+ g*z)
            + g*h*(3*(a*d*f*x**2+ b*e*f*y**2 - a*d**2*x*z - b*e**2*y*z)
                + g*(f - h*z) - f*h*(a*x**3+ b*y**3+ c*z**3)))**3

LHS = lhs_term1+ lhs_term2+ lhs_term3

# 3. Right-hand side (RHS)
rhs_factor1 = a*(d - h*x)**3+ b*(e - h*y)**3+ c*(f - h*z)**3

rhs_factor2 = ((a*x**3+ b*y**3+ c*z**3)**2 * (a*d**3+ b*e**3+ c*f**3)
            + 3*g*(a*x**3+ b*y**3+ c*z**3)*(a*d**2*x+ b*e**2*y+ c*f**2*z)
            + 3*g**2*(a*d*x**2+ b*e*y**2+ c*f*z**2)+ g**3)

rhs_factor3 = ((a*x**3+ b*y**3+ c*z**3)*(a*d**3+ b*e**3+ c*f**3)**2
            + 3*g*h*(a*d**3+ b*e**3+ c*f**3)*(a*d*x**2+ b*e*y**2+ c*f*z**2)
            + 3*g**2*h**2*(a*d**2*x+ b*e**2*y+ c*f**2*z)+ g**3*h**3)

RHS = rhs_factor1 * rhs_factor2 * rhs_factor3

# 4. Check
difference = sp.expand(LHS - RHS)
print("Difference:", difference)
print("Identity holds:", difference == 0)
```

Listing 2

```
import sympy as sp

a, b, c, d, e = sp.symbols('a b c d e')

# Inner terms of the three cubes
term1 = (a - b)*(
   2*a**3*c**4 - 2*a**3*c**3*e - a**3*c*d*e**2
  + 2*a**2*b*c**4+ 2*a**2*b*c**3*d - 2*a**2*b*c**3*e+ a**2*b*c*d**2*e
  + 2*a**2*b*c*d*e**2 - a**2*b*d**2*e**2
  + 2*a*b**2*c**3*d - 2*a*b**2*c*d**2*e - a*b**2*c*d*e**2+ 2*a*b**2*d**2*e**2
  + b**3*c*d**2*e - b**3*d**2*e**2
)

term2 = (
   2*a**4*c**3*e+ 4*a**3*b*c**4 - 2*a**3*b*c**3*d - 2*a**3*b*c*d*e**2+ a**3*b*d**2*e**2
  + 4*a**2*b**2*c**4 - 2*a**2*b**2*c**3*e+ 2*a**2*b**2*c*d**2*e+ 4*a**2*b**2*c*d*e**2 -
3*a**2*b**2*d**2*e**2
  + 2*a*b**3*c**3*d - 4*a*b**3*c*d**2*e - 2*a*b**3*c*d*e**2+ 3*a*b**3*d**2*e**2
  + 2*b**4*c*d**2*e - b**4*d**2*e**2
)

term3 = (
   2*a**4*c**4+ 2*a**4*c**3*e - a**4*c*d*e**2
  + 4*a**3*b*c**4 - 2*a**3*b*c**3*d+ a**3*b*c*d**2*e+ a**3*b*c*d*e**2+ a**3*b*d**2*e**2
  + 2*a**2*b**2*c**4 - 2*a**2*b**2*c**3*e - a**2*b**2*c*d**2*e+ a**2*b**2*c*d*e**2 -
3*a**2*b**2*d**2*e**2
  + 2*a*b**3*c**3*d - a*b**3*c*d**2*e - a*b**3*c*d*e**2+ 3*a*b**3*d**2*e**2
  + b**4*c*d**2*e - b**4*d**2*e**2
)

LeS = term1**3 - (a/b)*term2**3+ term3**3

# Right-hand side
ReS = (
   -(a/b)*(a - b)
   * (a**2*e**3 - 2*a*b*c**3 - 2*a*b*e**3 - 2*b**2*c**3+ b**2*e**3)
   * (2*a**4*c**3+ 2*a**3*b*c**3+ a**2*b**2*d**3 - 2*a*b**3*d**3+ b**4*d**3)
   * (4*a**5*c**6+ 8*a**4*b*c**6+ a**4*b*d**3*e**3+ 4*a**3*b**2*c**6
     - 4*a**3*b**2*d**3*e**3+ 6*a**2*b**3*d**3*e**3 - 4*a*b**4*d**3*e**3+ b**5*d**3*e**3)
)

difference = sp.simplify(LeS - ReS)
print("Specialized Identity difference:", difference)
print("Identity holds:", difference == 0)
```

Listing 3

```
import sympy as sp

a, b, c, d, e = sp.symbols('a b c d e')

# Three terms on the left-hand side
term1 = (
   a**4+ a**3*d - a**3*e+ a*b**3+ 3*a*b*d*e+ a*c**3+ 3*a*c*d*e
  + 3*a*d**2*e - 3*a*d*e**2+ b**3*d - b**3*e - 3*b**2*d*e
  + c**3*d - c**3*e - 3*c**2*d*e - 3*d**2*e**2
)

term2 = (
   a**3*b+ a**3*d - a**3*e - 3*a**2*d*e+ 3*a*b*d*e+ b**4+ b**3*d - b**3*e
  + b*c**3+ 3*b*c*d*e+ 3*b*d**2*e - 3*b*d*e**2+ c**3*d - c**3*e
   - 3*c**2*d*e - 3*d**2*e**2
)

term3 = (
   a**3*c+ a**3*d - a**3*e - 3*a**2*d*e+ 3*a*c*d*e+ b**3*c+ b**3*d - b**3*e
   - 3*b**2*d*e+ 3*b*c*d*e+ c**4+ c**3*d - c**3*e+ 3*c*d**2*e - 3*c*d*e**2
   - 3*d**2*e**2
)

LHS = term1**3+ term2**3+ term3**3

# Right-hand side
factor1 = (a+ d)**3+ (b+ d)**3+ (c+ d)**3
factor2 = (a - e)**3+ (b - e)**3+ (c - e)**3

factor3 = (
   a**6+ 3*a**4*d*e+ 2*a**3*b**3+ 3*a**3*b*d*e+ 2*a**3*c**3+ 3*a**3*c*d*e
  + 9*a**2*d**2*e**2+ 3*a*b**3*d*e+ 3*a*c**3*d*e+ b**6+ 3*b**4*d*e
  + 2*b**3*c**3+ 3*b**3*c*d*e+ 9*b**2*d**2*e**2+ 3*b*c**3*d*e+ c**6
  + 3*c**4*d*e+ 9*c**2*d**2*e**2+ 9*d**3*e**3
)

RHS = factor1 * factor2 * factor3

difference = sp.expand(LHS - RHS)
print("Identity difference:", difference)
print("Identity holds:", difference == 0)
```

Listing 4

```
import sympy as sp

a, b, c, d, e = sp.symbols('a b c d e')

# LHS
term1 = (
   a**4 + a**3*d - a**3*e + a*b**3 + 3*a*b*d*e + a*c**3 + 3*a*c*d*e
   + a*d**2*e - a*d*e**2 + b**3*d - b**3*e + 3*b**2*d*e
   + c**3*d - c**3*e - 3*c**2*d*e - d**2*e**2
)

term2 = (
   a**3*b - a**3*d + a**3*e + 3*a**2*d*e + 3*a*b*d*e + b**4 - b**3*d + b**3*e
   + b*c**3 + 3*b*c*d*e + b*d**2*e - b*d*e**2 - c**3*d + c**3*e
   + 3*c**2*d*e + d**2*e**2
)

term3 = (
   a**3*c + a**3*d - a**3*e - 3*a**2*d*e + 3*a*c*d*e + b**3*c + b**3*d - b**3*e
   + 3*b**2*d*e + 3*b*c*d*e + c**4 + c**3*d - c**3*e + c*d**2*e - c*d*e**2
   - d**2*e**2
)

LHS = term1**3 + term2**3 + term3**3

# RHS
factor1 = (a + d)**3 + (b - d)**3 + (c + d)**3
factor2 = (a - e)**3 + (b + e)**3 + (c - e)**3
factor3 = (
   a**6 + 3*a**4*d*e + 2*a**3*b**3 + 3*a**3*b*d*e + 2*a**3*c**3 + 3*a**3*c*d*e
   + 3*a**2*d**2*e**2 + 3*a*b**3*d*e + 3*a*c**3*d*e + b**6 + 3*b**4*d*e
   + 2*b**3*c**3 + 3*b**3*c*d*e - 3*b**2*d**2*e**2 + 3*b*c**3*d*e + c**6
   + 3*c**4*d*e + 3*c**2*d**2*e**2 + d**3*e**3
)

RHS = factor1 * factor2 * factor3

difference = sp.expand(LHS - RHS)
print("Identity difference:", difference)
print("Identity holds:", difference == 0)
```

Listing 5

```
import sympy as sp

a, b, c, d, e = sp.symbols('a b c d e')

term1 = (
   a**4 + a**3*d - a**3*e + a*b**3 + a*c**3 + 3*a*c*d*e
   + 2*a*d**2*e - 2*a*d*e**2 + b**3*d - b**3*e + c**3*d - c**3*e
   - 3*c**2*d*e - 2*d**2*e**2
)

term2 = (
   b*(a**3 + 3*a*d*e + b**3 + c**3 + 3*c*d*e + 2*d**2*e - 2*d*e**2)
)

term3 = (
   a**3*c + a**3*d - a**3*e - 3*a**2*d*e + 3*a*c*d*e + b**3*c + b**3*d - b**3*e
   + c**4 + c**3*d - c**3*e + 2*c*d**2*e - 2*c*d*e**2 - 2*d**2*e**2
)

LHS = term1**3 + term2**3 + term3**3

factor1 = (a + d)**3 + b**3 + (c + d)**3
factor2 = (a - e)**3 + b**3 + (c - e)**3
factor3 = (
   a**6 + 3*a**4*d*e + 2*a**3*b**3 + 2*a**3*c**3 + 3*a**3*c*d*e
   + 6*a**2*d**2*e**2 + 3*a*b**3*d*e + 3*a*c**3*d*e + b**6 + 2*b**3*c**3
   + 3*b**3*c*d*e + c**6 + 3*c**4*d*e + 6*c**2*d**2*e**2 + 4*d**3*e**3
)

RHS = factor1 * factor2 * factor3

difference = sp.expand(LHS - RHS)
print("Identity difference:", difference)
print("Identity holds:", difference == 0)
```

Listing 6

```
import sympy as sp

a, b, c, d, e, f = sp.symbols('a b c d e f')

term1 = a**3 - 3*a*b*d + a*d*e*f + b**3 - 3*b**2*d + c**3 + d**2*e*f
term2 = -a**3 - 3*a**2*d - 3*a*b*d - b**3 + b*d*e*f - c**3 - d**2*e*f
term3 = c*d* (-3*a - 3*b + e*f)

LHS = term1**3 + term2**3 + term3**3

RHS = (
   d * (-3*a - 3*b + e*f)
   * ((a + d)**3 + (b - d)**3 + c**3)
   * (3*a**4 + 3*a**3*b + 3*a**2*d*e*f + 3*a*b**3 + 3*a*c**3 + 3*b**4 - 3*b**2*d*e*f + 3*b*c**3 +
d**2*e**2*f**2)
)

difference = sp.expand(LHS - RHS)
print("Identity difference:", difference)
print("Identity holds:", difference == 0)
```