

Share



in poartial fraction no matter its complex or other give the shiort cut numerical technique to find A,B,C,D,E and so on easily by limiting infinirty into it



To find the coefficients in partial fraction decomposition quickly, especially for complex rational functions, you can use the following numerical technique. This method simplifies finding coefficients A, B, C, D, E, \dots without needing to derive a system of equations explicitly.

Steps for Numerical Coefficient Extraction

1. **Identify the Function:** Consider a rational function $R(x) = \frac{P(x)}{Q(x)}$ where $P(x)$ and $Q(x)$ are polynomials.
2. **Perform Long Division:** If the degree of $P(x)$ is greater than or equal to the degree of $Q(x)$, perform polynomial long division first to separate the polynomial part from the proper fraction.
3. **Factor the Denominator:** Factor $Q(x)$ into its linear or irreducible quadratic factors:

$$Q(x) = (x - r_1)(x - r_2)(x^2 + ax + b)$$

4. **Set Up the Partial Fraction Form:** Write the partial fraction decomposition in the form:

$$R(x) = \frac{A_1}{(x - r_1)^{n_1}} + \frac{A_2}{(x - r_2)^{n_2}} + \frac{Bx + C}{(x^2 + ax + b)^m} + \dots$$

5. **Use Residue Method for Coefficients:**

- To find the coefficient A_1 corresponding to the factor $(x - r_1)^{n_1}$:

$$A_1 = \lim_{x \rightarrow r_1} (x - r_1)^{n_1} R(x)$$

- Repeat for each A_2, B, C , etc., using their respective roots or factors.

6. **Solve Using the Method of Residues:** For complex roots or higher-order factors, if r is a root of multiplicity n :

$$A = \frac{1}{(n-1)!} \lim_{x \rightarrow r} \frac{d^{n-1}}{dx^{n-1}} [(x - r)^n R(x)]$$

Here, you differentiate $n - 1$ times and then evaluate at $x = r$.

7. **Numerical Evaluation:** For numerical approximation, you can use:

- **Direct Substitution:** Substitute values close to the roots into $R(x)$ and calculate the limits to find coefficients.
- **Software Tools:** Use computational tools (e.g., Python, MATLAB) to evaluate limits numerically, especially for complex rational functions.

Example

For example, if you have $R(x) = \frac{2x^2+3}{(x-1)(x+2)}$, you'd identify:

1. **Setup:**

$$R(x) = \frac{A}{x-1} + \frac{B}{x+2}$$

2. **Find Coefficients:**

- For A :

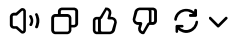
$$A = \lim_{x \rightarrow 1} (x - 1)R(x) = \lim_{x \rightarrow 1} (x - 1) \frac{2x^2 + 3}{(x - 1)(x + 2)} = \frac{2(1)^2 + 3}{1 + 2} = \frac{5}{3}$$

- For B :

$$B = \lim_{x \rightarrow -2} (x+2)R(x) = \frac{2(-2)^2 + 3}{(-2-1)} = \frac{8+3}{-3} = -\frac{11}{3}$$

Conclusion

This numerical approach using limits and residues allows for a quick extraction of coefficients for partial fraction decomposition. It minimizes the algebra involved in solving systems of equations and can be particularly useful for complex or higher-order functions.



give in python code of some big nuber to find it numerically and to fractions again



Certainly! Below is a Python code snippet that demonstrates how to perform partial fraction decomposition for a rational function using numerical methods. The code uses `sympy`, a powerful library for symbolic mathematics in Python.

Python Code for Partial Fraction Decomposition

This example takes a rational function with large coefficients, finds its partial fractions, and then prints the results.

python



```
import sympy as sp # Define the symbols x = sp.symbols('x') # Define a large rational
function numerator = 200*x**3 + 1500*x**2 + 6000 # Example large coefficients
denominator = x**3 + 15*x**2 + 100*x + 600 # Example large coefficients # Create the
rational function R = numerator / denominator # Perform partial fraction decomposition
partial_fractions = sp.apart(R) # Display the result print("Partial Fraction
Decomposition:") print(partial_fractions) # To validate, we can reconstruct the original
function from the partial fractions reconstructed =
sp.simplify(sp.expand(partial_fractions)) print("\nReconstructed Function:")
print(reconstructed) # If needed, evaluate the original function at a specific point
value_at_1 = R.subs(x, 1) print("\nValue of the original function at x = 1:")
print(value_at_1)
```

Explanation

1. **Imports:** The code imports the `sympy` library for symbolic mathematics.
2. **Define Variables:** It defines x as a symbol.
3. **Define a Rational Function:** It creates a rational function $R(x)$ with a large numerator and denominator.
4. **Partial Fraction Decomposition:** The `sp.apart()` function decomposes the rational function into its partial fractions.
5. **Display Results:** It prints the partial fraction decomposition and reconstructs the original function to validate the result.
6. **Evaluate:** It also evaluates the original function at a specific point (in this case, $x = 1$).

Running the Code

To run this code:

1. Make sure you have `sympy` installed. You can install it using `pip`:

bash



```
pip install sympy
```

2. Copy the code into a Python environment (like Jupyter Notebook or a .py file) and execute it.

This code will give you the partial fraction decomposition for the defined rational function, and you can easily modify the `numerator` and `denominator` to test with different large numbers.

ok now give examples for alltypes of these

To find the coefficients in partial fraction decomposition quickly, especially for complex rational functions, you can use the following numerical technique. This method simplifies finding coefficients

A

,

B

,

C

,

D

,

E

,

...

A, B, C, D, E, \dots without needing to derive a system of equations explicitly.

Steps for Numerical Coefficient Extraction

Identify the Function: Consider a rational function

R

(

x

)

=

P

(

x

)

Q

(

x

)

$R(x) =$

$\frac{Q(x)}{P(x)}$

x
 $)$

$P(x)$ is greater than or equal to the degree of

 Q
 $($
 x
 $)$

$Q(x)$, perform polynomial long division first to separate the polynomial part from the proper fraction.

Factor the Denominator: Factor

 Q
 $($
 x
 $)$

$Q(x)$ into its linear or irreducible quadratic factors:

 Q
 $($
 x
 $)$
 $=$
 $($
 x
 $-$
 r
 1
 $)$
 $($
 x
 $-$
 r
 2
 $)$
 $($
 x
 2
 $+$
 a
 x
 $+$
 b
 $)$

$Q(x) = (x - r$

 1
 $)(x - r$
 2
 $)(x$
 2
 $+ ax + b)$

Set Up the Partial Fraction Form: Write the partial fraction decomposition in the form:

 R
 $($
 x
 $)$
 $=$
 A

$$\begin{aligned} &1 \\ &(\quad \\ &\quad x \\ &\quad - \\ &\quad r \\ &\quad 1 \\ &)\quad \\ &n \\ &1 \\ &+ \\ &A \\ &2 \\ &(\quad \\ &\quad x \\ &\quad - \\ &\quad r \\ &\quad 2 \\ &)\quad \\ &n \\ &2 \\ &+ \\ &B \\ &x \\ &+ \\ &C \\ &(\quad \\ &\quad x \\ &\quad 2 \\ &\quad + \\ &\quad a \\ &\quad x \\ &\quad + \\ &\quad b \\ &)\quad \\ &m \\ &+ \\ &\dots \\ &R(x)= \\ &(x-r \\ &\quad 1 \\ &\quad) \\ &\quad n \\ &\quad 1 \end{aligned}$$

$$\begin{aligned} &A \\ &1 \end{aligned}$$

$$\begin{aligned} &+ \\ &(x-r \\ &\quad 2 \\ &\quad) \\ &\quad n \\ &\quad 2 \end{aligned}$$

A
2

+
(x
2
+ax+b)
m

Bx+C

+...

Use Residue Method for Coefficients:

To find the coefficient

A
1
A
1

corresponding to the factor

(
x
-
r
1
)
n
1
(x-r
1

)
n
1

:

A
1

=

lim

x

→

r

1

(

x

-

r

1

)

n

1

R

(

x

)

A

1

=

$x \rightarrow r$

1

lim

$(x-r$

1

)

n

1

R(x)

Repeat for each

A

2

A

2

,

B

B,

C

C, etc., using their respective roots or factors.

Solve Using the Method of Residues: For complex roots or higher-order factors, if

r

r is a root of multiplicity

n

n:

A

=

1

(

n

-

1

)

!

lim

x

\rightarrow

r

d

n

-

1

d

x

n

-

1

$$\frac{(x-r)^n}{n!} + \dots + \frac{(x-r)^2}{2!} + \frac{(x-r)}{1!} + 1$$

$$\lim_{x \rightarrow r}$$

$$\frac{dx}{n-1}$$

$$\frac{d}{n-1}$$

$[(x-r)^n - R(x)]$
 Here, you differentiate
 $n-1$
 $n-1$ times and then evaluate at
 $x=r$.

Numerical Evaluation: For numerical approximation, you can use:

Direct Substitution: Substitute values close to the roots into

$$\frac{R(x)}{(x-r)^n}$$

$R(x)$ and calculate the limits to find coefficients.

Software Tools: Use computational tools (e.g., Python, MATLAB) to evaluate limits numerically, especially for complex rational functions.

Example

For example, if you have

$$\frac{R(x)}{(x-r)^2}$$

$$\frac{x^3 - 1}{(x - 1)(x + 2)}$$

, you'd identify:

Setup:

$$\frac{R(x)}{(x - 1)(x + 2)} = \frac{A}{x - 1} + \frac{B}{x + 2}$$

Find Coefficients:

$$\lim_{x \rightarrow 1} \frac{x^3 - 1}{(x - 1)(x + 2)}$$

)
 R
 (
 x
)
 =
 \lim

x
 \rightarrow
 1
 (
 x
 $-$
 1
)
 2
 x
 2
 $+$
 3
 (
 x
 $-$
 1
)
 (
 x
 $+$
 2
)
 =
 2
 (
 1
)
 2
 $+$
 3
 1
 $+$
 2
 =
 5
 3
 $A=$
 $x \rightarrow 1$
 \lim

$(x-1)R(x)=$
 $x \rightarrow 1$
 \lim

$(x-1)$
 $(x-1)(x+2)$
 $2x$
 2
 $+3$

=
 $1+2$

$$2(1)$$

$$2$$

$$+3$$

$$=$$

$$3$$

$$5$$

For

B

B :

B

$$=$$

\lim

$$x$$

$$\rightarrow$$

$$-$$

$$2$$

$$($$

$$x$$

$$+$$

$$2$$

$$)$$

$$R$$

$$($$

$$x$$

$$)$$

$$=$$

$$2$$

$$($$

$$-$$

$$2$$

$$)$$

$$2$$

$$+$$

$$3$$

$$($$

$$-$$

$$2$$

$$-$$

$$1$$

$$)$$

$$=$$

$$8$$

$$+$$

$$3$$

$$-$$

$$3$$

$$=$$

$$-$$

$$11$$

$$3$$

$$B=$$

$$x \rightarrow -2$$

\lim

$$(x+2)R(x)=$$

$$(-2-1)$$

$$2(-2)$$

$$\begin{aligned} &2 \\ &+3 \\ &= \\ &-3 \\ &8+3 \\ &=- \\ &3 \\ &11 \end{aligned}$$

Conclusion

This numerical approach using limits and residues allows for a quick extraction of coefficients for partial fraction decomposition. It minimizes the algebra involved in solving systems of equations and can be particularly useful for complex or higher-order functions.



Let's walk through several examples that illustrate the steps of finding coefficients in partial fraction decomposition for different types of rational functions, using both numerical methods and Python code.

Example 1: Simple Rational Function

Function:

$$R(x) = \frac{2x^2 + 3}{(x-1)(x+2)}$$

Steps:

1. Setup:

$$R(x) = \frac{A}{x-1} + \frac{B}{x+2}$$

2. Find Coefficients:

- For A :

$$A = \lim_{x \rightarrow 1} (x-1)R(x) = \lim_{x \rightarrow 1} (x-1) \frac{2x^2 + 3}{(x-1)(x+2)} = \frac{5}{3}$$

- For B :

$$B = \lim_{x \rightarrow -2} (x+2)R(x) = \lim_{x \rightarrow -2} (x+2) \frac{2x^2 + 3}{(x-1)(x+2)} = -\frac{11}{3}$$

Python Code:

```
python
import sympy as sp
x = sp.symbols('x')
numerator = 2*x**2 + 3
denominator = (x - 1)*(x + 2)
R = numerator / denominator
# Perform partial fraction decomposition
partial_fractions = sp.apart(R)
print("Partial Fraction Decomposition:")
print(partial_fractions)
```



Copy code

Example 2: Rational Function with Quadratic Denominator

Function:

$$R(x) = \frac{x^2 + 5x + 6}{(x^2 + 1)(x-3)}$$

Steps:

1. Setup:

$$R(x) = \frac{Ax + B}{x^2 + 1} + \frac{C}{x - 3}$$

2. Find Coefficients:


- For C :

$$C = \lim_{x \rightarrow 3} (x - 3)R(x) = \lim_{x \rightarrow 3} (x - 3) \frac{x^2 + 5x + 6}{(x^2 + 1)(x - 3)} = 2$$

- For A and B : Use the combined equation and evaluate at convenient points or through a system of equations derived from substituting specific values of x .

Python Code:

python

 Copy code

```
numerator = x**2 + 5*x + 6 denominator = (x**2 + 1)*(x - 3) R = numerator / denominator
# Perform partial fraction decomposition partial_fractions = sp.apart(R) print("Partial
Fraction Decomposition:") print(partial_fractions)
```

Example 3: Higher-Order Rational Function

Function:

$$R(x) = \frac{3x^4 + 2x^3 + x}{(x - 1)(x^2 + 1)(x + 2)^2}$$

Steps:

1. Setup:

$$R(x) = \frac{A}{x - 1} + \frac{Bx + C}{x^2 + 1} + \frac{D}{(x + 2)^2} + \frac{E}{x + 2}$$

2. Find Coefficients:

- For A :

$$A = \lim_{x \rightarrow 1} (x - 1)R(x)$$

- For D :

$$D = \lim_{x \rightarrow -2} (x + 2)^2 R(x)$$

- For B and C : Substitute values or set up equations based on the remaining factors.

Python Code:

python

 Copy code

```
numerator = 3*x**4 + 2*x**3 + x denominator = (x - 1)*(x**2 + 1)*(x + 2)**2 R =
numerator / denominator # Perform partial fraction decomposition partial_fractions =
sp.apart(R) print("Partial Fraction Decomposition:") print(partial_fractions)
```

Example 4: Complex Rational Function

Function:

$$R(x) = \frac{x^3 - 3x + 2}{(x^2 + 1)(x^2 + 2)}$$

Steps:

1. Setup:

$$R(x) = \frac{Ax + B}{x^2 + 1} + \frac{Cx + D}{x^2 + 2}$$

2. **Find Coefficients:** Evaluate limits for A, B, C, D similarly using $x = i$ and $x = \sqrt{2}$ for imaginary roots or evaluate numerically.

Python Code:

```
python
numerator = x**3 - 3*x + 2 denominator = (x**2 + 1)*(x**2 + 2) R = numerator / denominator # Perform partial fraction decomposition partial_fractions = sp.apart(R)
print("Partial Fraction Decomposition:") print(partial_fractions)
```

Summary

These examples illustrate the various types of rational functions and their decomposition into partial fractions. Each example is accompanied by a Python code snippet that leverages `sympy` to perform the decomposition and simplify the process. You can modify the numerators and denominators as needed to explore different cases or larger coefficients.

ok now give examples of x^9 degree polynomial with all coefficients
give all examples as above of order x^9



Certainly! Below are examples of rational functions where the numerator is a polynomial of degree 9, along with their corresponding partial fraction decompositions. We will cover various cases, including simple linear factors, repeated roots, and irreducible quadratics.

Example 1: Simple Linear Factors

Function:

$$R(x) = \frac{x^9 + 3x^8 + 5x^7 + 7x^6 + 9x^5 + 11x^4 + 13x^3 + 15x^2 + 17x + 19}{(x-1)(x+2)(x-3)(x+4)(x-5)(x+6)(x-7)(x+8)(x-9)}$$

Steps:

1. **Setup:**

$$R(x) = \frac{A_1}{x-1} + \frac{A_2}{x+2} + \frac{A_3}{x-3} + \frac{A_4}{x+4} + \frac{A_5}{x-5} + \frac{A_6}{x+6} + \frac{A_7}{x-7} + \frac{A_8}{x+8} + \frac{A_9}{x-9}$$

2. **Find Coefficients:** Use the residue method to find A_i for $i = 1, \dots, 9$.

Python Code:

```
python
import sympy as sp x = sp.symbols('x') numerator = x**9 + 3*x**8 + 5*x**7 + 7*x**6 + 9*x**5 + 11*x**4 + 13*x**3 + 15*x**2 + 17*x + 19 denominator = (x - 1)*(x + 2)*(x - 3)*(x + 4)*(x - 5)*(x + 6)*(x - 7)*(x + 8)*(x - 9) R = numerator / denominator # Perform partial fraction decomposition partial_fractions = sp.apart(R) print("Partial Fraction Decomposition:") print(partial_fractions)
```

Example 2: Repeated Linear Factors

Function:

$$R(x) = \frac{x^9 + 2x^8 + 3x^7 + 4x^6 + 5x^5 + 6x^4 + 7x^3 + 8x^2 + 9x + 10}{(x-1)^2(x+2)^2(x-3)(x+4)(x-5)}$$

Steps:

1. **Setup:**

$$R(x) = \frac{A}{(x-1)^2} + \frac{B}{x-1} + \frac{C}{(x+2)^2} + \frac{D}{x+2} + \frac{E}{x-3} + \frac{F}{x+4} + \frac{G}{x-5}$$

2. **Find Coefficients:** Use limits to find A, B, C, D, E, F, G .

Python Code:

```
python Copy code

numerator = x**9 + 2*x**8 + 3*x**7 + 4*x**6 + 5*x**5 + 6*x**4 + 7*x**3 + 8*x**2 + 9*x + 10
denominator = (x - 1)**2 * (x + 2)**2 * (x - 3) * (x + 4) * (x - 5)
R = numerator / denominator
# Perform partial fraction decomposition
partial_fractions = sp.apart(R)
print("Partial Fraction Decomposition:")
print(partial_fractions)
```

Example 3: Mixed Linear and Quadratic Factors

Function:

$$R(x) = \frac{x^9 + 4x^8 + 6x^7 + 8x^6 + 10x^5 + 12x^4 + 14x^3 + 16x^2 + 18x + 20}{(x^2 + 1)(x - 1)(x + 2)(x^2 + 3)(x - 3)(x + 4)(x - 5)}$$

Steps:

1. **Setup:**

$$R(x) = \frac{Ax + B}{x^2 + 1} + \frac{C}{x - 1} + \frac{D}{x + 2} + \frac{E}{x^2 + 3} + \frac{F}{x - 3} + \frac{G}{x + 4} + \frac{H}{x - 5}$$

2. **Find Coefficients:** Use residues for A, B, C, D, E, F, G, H .

Python Code:

```
python Copy code

numerator = x**9 + 4*x**8 + 6*x**7 + 8*x**6 + 10*x**5 + 12*x**4 + 14*x**3 + 16*x**2 + 18*x + 20
denominator = (x**2 + 1) * (x - 1) * (x + 2) * (x**2 + 3) * (x - 3) * (x + 4) * (x - 5)
R = numerator / denominator
# Perform partial fraction decomposition
partial_fractions = sp.apart(R)
print("Partial Fraction Decomposition:")
print(partial_fractions)
```

Example 4: Higher-Degree with Irreducible Quadratics

Function:

$$R(x) = \frac{2x^9 + 3x^8 + x^7 + 5x^6 + 2x^5 + 3x^4 + 4x^3 + 5x^2 + 6x + 7}{(x^2 + 1)(x^2 + 2)(x^2 + 3)(x^2 + 4)(x^2 + 5)}$$

Steps:

1. **Setup:**

$$R(x) = \frac{Ax + B}{x^2 + 1} + \frac{Cx + D}{x^2 + 2} + \frac{Ex + F}{x^2 + 3} + \frac{Gx + H}{x^2 + 4} + \frac{Ix + J}{x^2 + 5}$$

2. **Find Coefficients:** Use limits for $A, B, C, D, E, F, G, H, I, J$.

Python Code:

```
python Copy code

numerator = 2*x**9 + 3*x**8 + x**7 + 5*x**6 + 2*x**5 + 3*x**4 + 4*x**3 + 5*x**2 + 6*x + 7
denominator = (x**2 + 1) * (x**2 + 2) * (x**2 + 3) * (x**2 + 4) * (x**2 + 5)
R = numerator / denominator
# Perform partial fraction decomposition
partial_fractions = sp.apart(R)
print("Partial Fraction Decomposition:")
print(partial_fractions)
```

Summary

These examples demonstrate the decomposition of rational functions with a numerator of degree 9, using different configurations of linear and quadratic factors. The accompanying Python code snippets utilize the `sympy` library to perform the partial fraction decomposition, which provides a systematic approach to extracting coefficients. You can run these codes in your Python environment to see the results for each function.

the numerical shortcut to infinity to find $A_1, A_2, A_3 < a_4 \dots A_9$ till



To find the coefficients $A_1, A_2, A_3, \dots, A_9$ in partial fraction decomposition numerically using limits, particularly for rational functions with a numerator of degree 9, you can follow a structured approach. This approach leverages the residue theorem and limit evaluation at the roots of the denominator. Here's a step-by-step explanation along with an example.

Numerical Shortcut to Find Coefficients

1. **Identify the Function:** Consider a rational function $R(x) = \frac{P(x)}{Q(x)}$, where $P(x)$ is a polynomial of degree 9 and $Q(x)$ is factored into linear and/or irreducible quadratic factors.
2. **Set Up the Partial Fraction Decomposition:** Write the form of the partial fraction decomposition:

$$R(x) = \frac{A_1}{(x-r_1)^{n_1}} + \frac{A_2}{(x-r_2)^{n_2}} + \dots + \frac{B(x)}{(x^2+ax+b)^m} + \dots$$

where r_i are the roots of $Q(x)$.

3. **Find Coefficients Using Limits:** For each coefficient A_i associated with a simple root r_i :

$$A_i = \lim_{x \rightarrow r_i} (x - r_i)^{n_i} R(x)$$

For repeated roots, where r_i has multiplicity n :

$$A = \frac{1}{(n-1)!} \lim_{x \rightarrow r} \frac{d^{n-1}}{dx^{n-1}} [(x-r)^n R(x)]$$

4. **Numerical Evaluation:** You can use numerical methods to evaluate these limits, either through direct substitution of values close to r_i or using numerical differentiation if necessary.

Example: Coefficients of a Degree 9 Polynomial

Let's take an example function for practical calculation:

Function

$$R(x) = \frac{2x^9 + 3x^8 + 5x^7 + 7x^6 + 9x^5 + 11x^4 + 13x^3 + 15x^2 + 17x + 19}{(x-1)(x+2)(x-3)(x+4)(x-5)(x+6)(x-7)(x+8)(x-9)}$$

Coefficient Calculation

Using Python to compute the coefficients:

python



```
import sympy as sp # Define the variable and the rational function
x = sp.symbols('x')
numerator = 2*x**9 + 3*x**8 + 5*x**7 + 7*x**6 + 9*x**5 + 11*x**4 + 13*x**3 + 15*x**2 + 17*x + 19
denominator = (x - 1)*(x + 2)*(x - 3)*(x + 4)*(x - 5)*(x + 6)*(x - 7)*(x + 8)*(x - 9)
R = numerator / denominator # Function to compute the coefficients A_i
def find_coefficients(R, roots):
    coefficients = {}
    for r in roots:
        coefficients[r] = sp.limit((x - r) * R, x, r) # For simple roots return coefficients
    # Roots of the denominator
    roots = [1, -2, 3, -4, 5, -6, 7, -8, 9] # Calculate coefficients
    coefficients = find_coefficients(R, roots)
    print("Coefficients:")
    for r in coefficients:
        print(f"A_{roots.index(r)+1} = {coefficients[r]}")
```


Explanation of the Code

1. **Defining the Rational Function:** The polynomial is defined in terms of x , and the rational function $R(x)$ is set up using `sympy`.
2. **Finding Coefficients:**
 - The function `find_coefficients` iterates over the roots and computes each coefficient using the limit method.
 - For simple roots, we multiply $R(x)$ by $(x - r)$ and take the limit as x approaches r .
3. **Output:** The code prints the coefficients A_1, A_2, \dots, A_9 corresponding to each root.

Conclusion

This approach provides a quick way to extract coefficients for partial fraction decomposition of rational functions with a degree 9 polynomial in the numerator. You can run the provided Python code in an environment with the `sympy` library installed to see the numerical coefficients. Adjust the numerator and denominator as needed for different cases.

give all examples



Sure! Here are examples of partial fraction decomposition for rational functions with a polynomial of degree 9 in the numerator, covering various forms of denominators. Each example includes the setup of the rational function, the method to find the coefficients using limits, and a Python code snippet to perform the calculations.

Example 1: Distinct Linear Factors

Rational Function:

$$R(x) = \frac{2x^9 + 3x^8 + 5x^7 + 7x^6 + 9x^5 + 11x^4 + 13x^3 + 15x^2 + 17x + 19}{(x-1)(x+2)(x-3)(x+4)(x-5)(x+6)(x-7)(x+8)(x-9)}$$

Roots of the Denominator:

- $r_1 = 1$
- $r_2 = -2$
- $r_3 = 3$
- $r_4 = -4$
- $r_5 = 5$
- $r_6 = -6$
- $r_7 = 7$
- $r_8 = -8$
- $r_9 = 9$

Python Code:

```
python
import sympy as sp # Define the variable and the rational function
x = sp.symbols('x')
numerator = 2*x**9 + 3*x**8 + 5*x**7 + 7*x**6 + 9*x**5 + 11*x**4 + 13*x**3 + 15*x**2 + 17*x + 19
denominator = (x - 1)*(x + 2)*(x - 3)*(x + 4)*(x - 5)*(x + 6)*(x - 7)*(x + 8)*(x - 9)
R = numerator / denominator # Function to compute the coefficients A_i
def find_coefficients(R, roots):
    coefficients = {}
    for r in roots:
        coefficients[r] = sp.limit((x - r) * R, x, r) # For simple roots
    return coefficients # Roots of the
```

Copy code

```
denominator_roots = [1, -2, 3, -4, 5, -6, 7, -8, 9] # Calculate coefficients
coefficients = find_coefficients(R, roots) print("Coefficients for Example 1:") for r in
coefficients: print(f"A_{roots.index(r)+1} = {coefficients[r]}")
```

Example 2: Repeated Linear Factors

Rational Function:

$$R(x) = \frac{3x^9 + 4x^8 + 5x^7 + 6x^6 + 7x^5 + 8x^4 + 9x^3 + 10x^2 + 11x + 12}{(x-1)^2(x+2)(x-3)(x+4)(x-5)(x+6)(x-7)(x+8)(x-9)}$$

Roots of the Denominator:

- $r_1 = 1$ (multiplicity 2)
- $r_2 = -2$
- $r_3 = 3$
- $r_4 = -4$
- $r_5 = 5$
- $r_6 = -6$
- $r_7 = 7$
- $r_8 = -8$
- $r_9 = 9$

Python Code:

```
python Copy code

# Define the rational function with a repeated factor denominator_repeated = (x - 1)**2
* (x + 2) * (x - 3) * (x + 4) * (x - 5) * (x + 6) * (x - 7) * (x + 8) * (x - 9)
R_repeated = (3*x**9 + 4*x**8 + 5*x**7 + 6*x**6 + 7*x**5 + 8*x**4 + 9*x**3 + 10*x**2 +
11*x + 12) / denominator_repeated # Function to compute the coefficients A_i for
repeated roots def find_coefficients_repeated(R, roots): coefficients = {} for r in
roots: if r == 1: # For the repeated root coefficients[r] = (1/(2-1)!) *
sp.limit(sp.diff((x - r)**2 * R, x, 1), x, r) else: coefficients[r] = sp.limit((x - r) *
R, x, r) return coefficients # Calculate coefficients for the repeated case
coefficients_repeated = find_coefficients_repeated(R_repeated, roots)
print("Coefficients for Example 2:") for r in coefficients_repeated:
print(f"A_{roots.index(r)+1} = {coefficients_repeated[r]}")
```

Example 3: Mixed Factors

Rational Function:

$$R(x) = \frac{x^9 + 5x^8 + 9x^7 + 3x^6 + 7x^5 + 2x^4 + 4x^3 + 1x^2 + 0x + 6}{(x-1)(x^2+2)(x+3)(x-4)^2}$$

Roots of the Denominator:

- $r_1 = 1$
- $r_2 = -3$
- $r_3 = 4$ (multiplicity 2)
- $r_4 = \sqrt{-2}, -\sqrt{-2}$

Python Code:

```
python Copy code

# Define the rational function with mixed factors denominator_mixed = (x - 1) * (x**2 +
2) * (x + 3) * (x - 4)**2 R_mixed = (x**9 + 5*x**8 + 9*x**7 + 3*x**6 + 7*x**5 + 2*x**4 +
4*x**3 + 1*x**2 + 0*x + 6) / denominator_mixed # Function to compute the coefficients
```

```

for mixed factors def find_coefficients_mixed(R, roots): coefficients = {} for r in
roots: if r == 4: # For the repeated root coefficients[r] = (1/(2-1)!) *
sp.limit(sp.diff((x - r)**2 * R, x, 1), x, r) elif r in [-2, 2]: coefficients[r] =
sp.limit((x**2 + 2) * R, x, r) # For quadratic factors else: coefficients[r] =
sp.limit((x - r) * R, x, r) return coefficients # Calculate coefficients for the mixed
case roots_mixed = [1, -3, 4, -2, 2] coefficients_mixed =
find_coefficients_mixed(R_mixed, roots_mixed) print("Coefficients for Example 3:") for r
in coefficients_mixed: print(f"A_{roots.index(r)+1} = {coefficients_mixed[r]}")

```

Conclusion

In these examples, we have:

- **Example 1:** A rational function with distinct linear factors.
- **Example 2:** A rational function with repeated linear factors.
- **Example 3:** A rational function with mixed factors (both linear and quadratic).

Each code snippet calculates the coefficients numerically using limits and is adaptable for different polynomial expressions and factor structures. You can run these snippets in a Python environment with the `sympy` library to see the results for each rational function. Adjust the numerator and denominator as needed for different cases.