

Share



in partial fraction no matter its complex or other give the short cut numerical technique to find A,B,C,D,E and so on easily by limiting infinity 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

)

$R(x) =$

$Q(x)$

$P(x)$

where

P

(

x

)

$P(x)$ and

Q

(

x

)

Perform Long Division: If the degree of

P

(

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

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

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

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

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
1

(
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

→

r

d

n

-

1

d

x

n

-

1

[
 (
 x
 $-$
 r
 $)$
 n
 R
 (
 x
 $)$
 $]$
 $A =$
 $(n-1)!$
 1

$x \rightarrow r$
 \lim

dx
 $n-1$

d
 $n-1$

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

Numerical Evaluation: For numerical approximation, you can use:

Direct Substitution: Substitute values close to the roots into

R
 (
 x
 $)$

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

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

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

, you'd identify:

Setup:

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

Find Coefficients:

For
A:
A
=

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

)
 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)
```



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



```
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



```
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

 Copy code

```
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.