

# Function approximation

wiki: [https://en.wikipedia.org/wiki/Function\\_approximation](https://en.wikipedia.org/wiki/Function_approximation)

definition: **select a function among a well-defined class that closely matches ("approximates") a target function in a task-specific way**

## Taylor series

wiki: [https://en.wikipedia.org/wiki/Taylor\\_series](https://en.wikipedia.org/wiki/Taylor_series)

A special case of Pade approximation.

## Pade approximation

wiki: [https://en.wikipedia.org/wiki/Pad%C3%A9\\_approximant](https://en.wikipedia.org/wiki/Pad%C3%A9_approximant)

target function:  $f(x)$

function collection:  $\frac{P_L(x)}{Q_M(x)} \approx f(x)$

1.  $P_L(x) = p_0 + p_1x + \dots + p_Lx^L$
2.  $Q_M(x) = 1 + q_1x + \dots + q_Mx^M$

process

1. Taylor expansion:  $f(x) \approx \sum_{n=0}^{n=L+M} A_n x^n$
2. then:  $Q_M(x) \sum_{n=0}^{n=L+M} A_n x^n \approx P_L(x)$

$$\left\{ \begin{array}{rcl} p_0 & = & A_0 \\ p_1 - A_0 q_1 & = & A_1 \\ p_2 - A_1 q_1 - A_0 q_2 & = & A_2 \\ & \dots & \\ p_L - A_{L-1} q_1 - \dots - A_0 q_L & = & A_L \\ -A_L q_1 - \dots - A_{L-M+1} q_M & = & A_{L+1} \\ & \dots & \\ -A_{L+M-1} q_1 - \dots - A_L q_M & = & A_{L+M} \end{array} \right.$$

$$\begin{pmatrix} 1 & & & & 0 & & & \\ & 1 & & & -A_0 & & 0 & \\ & & 1 & & -A_1 & & -A_0 & 0 \\ & & & \ddots & \vdots & \vdots & \vdots & \vdots \\ & & & & 1 & -A_{L-1} & -A_{L-2} & -A_{L-3} \\ & & & & & \vdots & \vdots & \vdots \\ & & & & & -A_{L+M-1} & -A_{L+M-2} & -A_{L+M-3} \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_L \\ q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_M \end{pmatrix} = \begin{pmatrix} A_0 \\ A_1 \\ A_2 \\ \vdots \\ A_L \\ \vdots \\ A_{L+M} \end{pmatrix}$$

```
%symbolic toolbox needed, otherwise need to calculate high order taylor coeffcient by hand
syms x
expr1 = sin(6*x)/x;
taylor_coefficient = sym2poly(taylor(expr1,x,'order',9));

data_x = linspace(-1,1,100);
data_y = double(subs(expr1,x,data_x));

fig = figure();
ax = axes('Parent', fig, 'NextPlot', 'add', 'Box', 'on');

hline0 = plot(data_x, data_y, '.', 'MarkerSize',20);

[p,q] = my_pade_approximation(taylor_coefficient, 8);
y = polyval(p,data_x)./polyval(q,data_x);
hline1 = plot(data_x, y,'LineWidth',2);

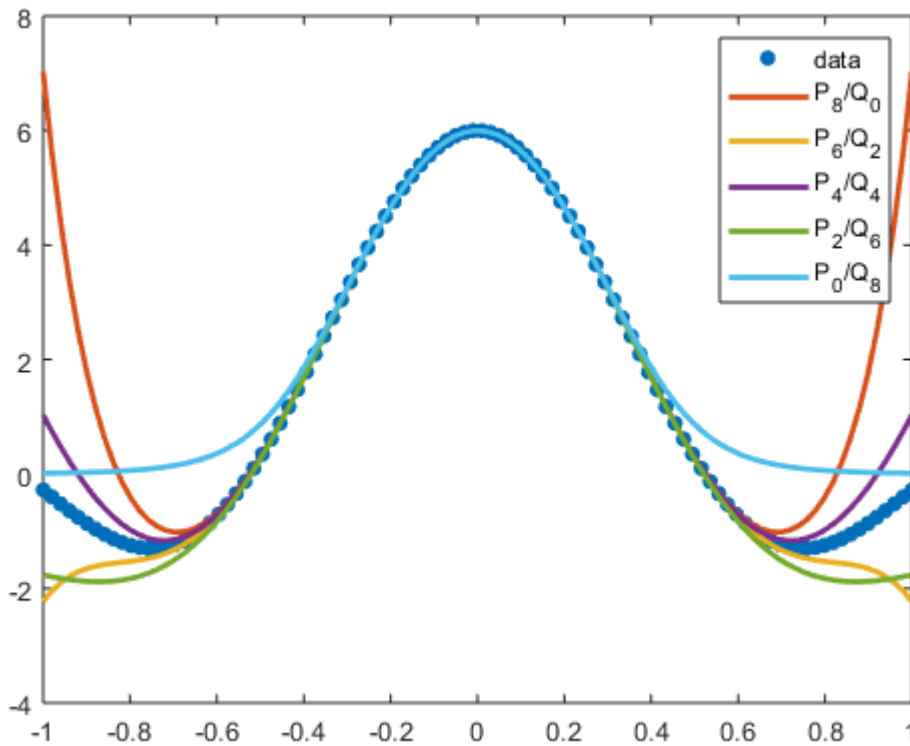
[p,q] = my_pade_approximation(taylor_coefficient, 6);
y = polyval(p,data_x)./polyval(q,data_x);
hline2 = plot(data_x, y,'LineWidth',2);

[p,q] = my_pade_approximation(taylor_coefficient, 4);
y = polyval(p,data_x)./polyval(q,data_x);
hline3 = plot(data_x, y,'LineWidth',2);

[p,q] = my_pade_approximation(taylor_coefficient, 2);
y = polyval(p,data_x)./polyval(q,data_x);
hline4 = plot(data_x, y,'LineWidth',2);

[p,q] = my_pade_approximation(taylor_coefficient, 0);
y = polyval(p,data_x)./polyval(q,data_x);
hline5 = plot(data_x, y,'LineWidth',2);

legend([hline0,hline1,hline2,hline3,hline4,hline5],{'data','P_8/Q_0','P_6/Q_2','P_4/Q_4','P_2/Q_6','P_0/Q_8'})
```



## function

```
function [p,q] = my_pade_approximation(taylor_coefficient, L)
% Pade approximation
% taylor_coefficient(int, (1,N1)): from high order to zero-order
% L(int, (1,1))
% p(float,(1,L+1)): from high order to zero-order
% q(float,(1,M+1)), from high order to zero-order, remember to add zero-order coefficient which
%
% reference:
%   wiki: https://en.wikipedia.org/wiki/Pade approximant
%   http://mathworld.wolfram.com/PadeApproximant.html
%   http://mathfaculty.fullerton.edu/mathews/n2003/pade/PadeApproximationProof.pdf
p = 0;
q = 0;
% not available until 20181025
end
```