



# ***Meshless Methods***

***Soheil Mohammadi***

*Professor of Computational Mechanics*

*School of Civil Engineering*

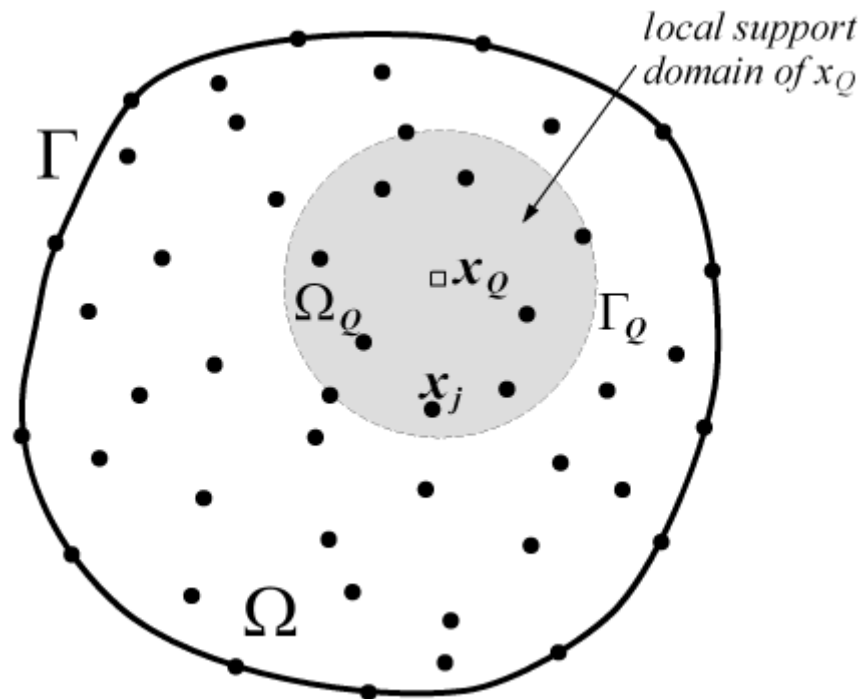
*University of Tehran*

Tel: +98-21-6111 2258, Fax: +98-21-6640 3808, Cell: +98-912-5030 392

Email: [smoham@ut.ac.ir](mailto:smoham@ut.ac.ir), <http://eng.ut.ac.ir/civ/smohammadi>

# Point Interpolation Method (PIM):

- PIM allows for the approximation function to pass through the function value at each scattered node within the support domain.
- MLS is capable of producing an approximation with the desired order of consistency.



# Point Interpolation Method (PIM):

- Assume a point of interest  $x_Q$
- PIM interpolates  $u(x)$  using the nodal values at the nodes of the support domain of  $x_Q$

$$u^h(x, x_Q) = \sum_{j=1}^n B_j(x) a_j(x_Q)$$

- where

$B_j(x)$  basis function corresponding to  $x_Q$

$a_j(x_Q)$  coefficient for  $B_j(x)$

## Polynomial PIM (PPIM):

- The basis function in PPIM is a complete polynomial function:

$$u^h(x, x_Q) = P^T(x) a(x_Q) = \sum_{j=1}^n p_j(x) a_j(x_Q)$$

- All polynomial terms should be selected symmetrically.

$$P^T(x) = \{1, x, x^2, x^3, \dots\}$$

$$P^T(x) = \{1, x, y, x^2, xy, y^2, \dots\}$$

- The number of polynomial terms is the same as the number of nodes within the support domain.

## Polynomial PIM (PPIM):

- The coefficients  $a_j$  can be determined by enforcing that the PPIM approximation is satisfied at the  $n$  nodes in the support domain of  $x_Q$ .
- At a node  $i$ :

$$u_i = P^T(x_i)a, \quad i = 1, \dots, n$$

$$u_i = u^h(x_i)$$

- or

$$U_s = P_Q a$$

$$U_s^T = \{u_1, u_2, \dots, u_n\}$$

## Polynomial PIM (PPIM):

$$P_Q^T = \{p^T(x_1), p^T(x_2), \dots, p^T(x_n)\}$$

$$P_Q = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix} : 1D$$

$$P_Q = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & x_1 y_1 & y_1^2 & \cdots \\ 1 & x_2 & y_2 & x_2^2 & x_2 y_2 & y_2^2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\ 1 & x_n & y_n & x_n^2 & x_n y_n & y_n^2 & \cdots \end{bmatrix} : 2D$$

## Polynomial PIM (PPIM):

- Solving  $U_s = P_Q a$  for  $a$

$$a = P_Q^{-1} U_s$$

- And the final approximation,

$$u^h(x) = \phi(x) U_s$$

$$\phi(x) = p^T(x) P_Q^{-1}$$

$$\phi(x) = [\phi_1(x), \phi_2(x), \dots, \phi_n(x)]$$

- The  $j^{th}$  derivative of the PPIM shape function is simply computed:

$$\phi^{(j)}(x) = [p^{(j)}(x)]^T P_Q^{-1}$$

## Polynomial PIM (PPIM):

- PPIM shape functions  $\phi_i(x)$  are linearly independent in the support domain.
- $\phi_i(x)$  possess the Kronecker delta function property. This is important for imposing essential boundary conditions.

$$\phi_i(x_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

- $\phi_i(x)$  is the partition of unity if the constant term is always included in the basis function:

$$\sum_{i=1}^n \phi_i(x) = 1$$

- This is not a requirement for  $0 \leq \phi_i(x) \leq 1$

## Polynomial PIM (PPIM):

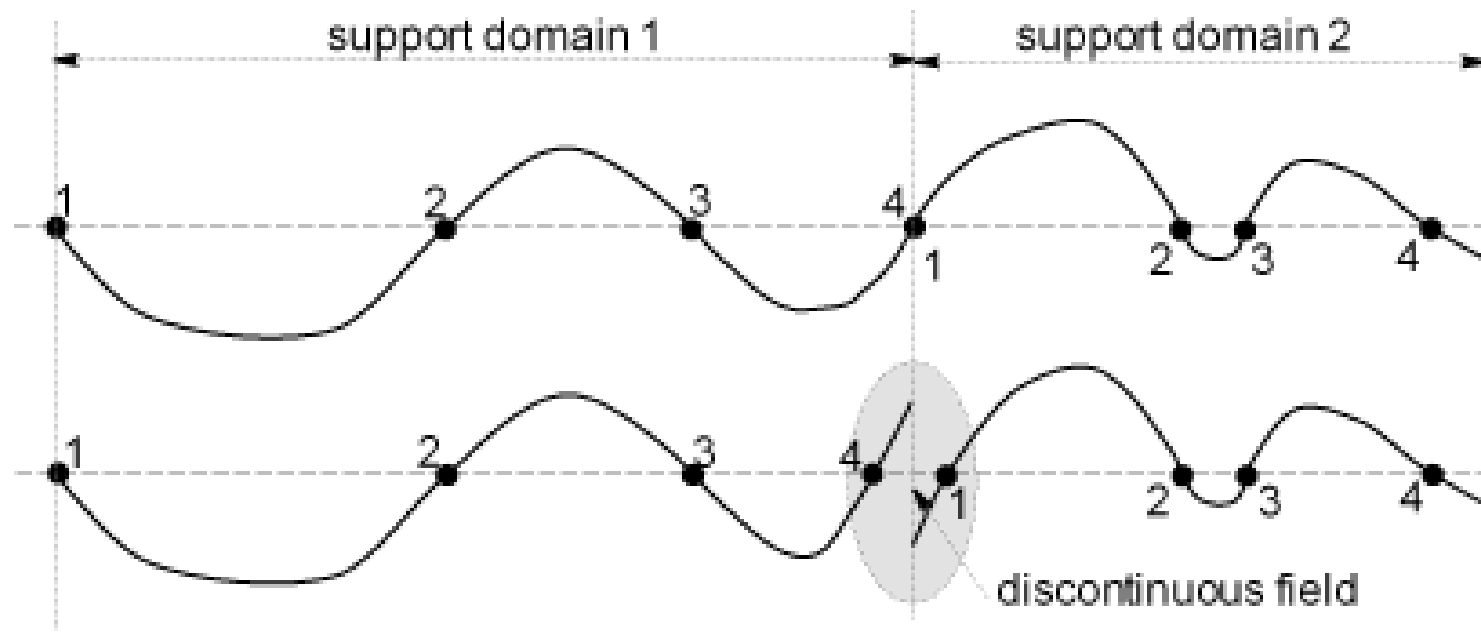
- If the first order monomials is included in the basis  $\phi_i(x)$  , possesses linear reproducing properties, essential for passing the patch test:

$$\sum_{i=1}^n \phi_i(x) x_i = x$$

- Derivatives of  $\phi_i(x)$  have simple polynomial forms.
- No weight function is required.
- PPIM can produce a shape function of any order by increasing the number of nodes for interpolation.
- The coefficients in PPIM are constants.

## Polynomial PIM (PPIM):

- More oscillations are expected around the exact solution, if higher order basis functions are included.
- PPIM shape functions are not compatible. As a result, the PPIM approximation can be discontinuous when  $x_Q$  moves and the support domain updates its nodes.



## Polynomial PIM (PPIM):

- A main drawback of PPIM is that  $P_Q^{-1}$  may not exist due to singularity of  $P_Q$ .
- This problem is more frequent in PPIM because it does not contain the smoothing effect of the weight functions.
- Available remedies are:
  - Method of moving nodes
  - Coordinate transformation
  - Matrix triangulation
  - Radial basis function (RBF)
  - Combination of RBF and PPIM

# Radial Point Interpolation (RPIM):

- Multiquads (MQ) were developed by Hardy in 1971 for the study of the gravitational field of the earth.
- MQ was re-visited by a mathematician (Franke) in 1982.
- Kansa re-discovered the method for computational mechanics in 1990. The method is sometimes called the Kansa method.
- Wang and Liu (2000) proposed the RPIM as a remedy for original PIM deficiencies.

# Radial Point Interpolation (RPIM):

- RPIM uses the radial basis function (RBF) instead of the polynomial basis:

$$u^h(x, x_Q) = \sum_{j=1}^n R_j(x) a_j(x_Q) = R^T(x) a(x_Q)$$

$$R^T(r) = \{R_1(r_1), R_2(r_2), \dots, R_n(r_n)\}$$

$$r_i = \left[ (x - x_i)^2 - (y - y_i)^2 \right]^{1/2}$$

- RPIM always works and guarantees the existence of the inverse of the moment matrix. (provided that the guidelines of choosing the shape parameters of the radial basis functions are followed).

# Radial Point Interpolation (RPIM):

- Frequently used Radial Basis Functions (RBF):

- Multiquadrics (MQ)

$$R_i(x, y) = (r_i^2 + c^2)^q = \left[ (x - x_i)^2 + (y - y_i)^2 + c^2 \right]^q$$

- Gaussian (EXP)

$$R_i(x, y) = \exp(-cr_i^2) = \exp\left[-c\left((x - x_i)^2 + (y - y_i)^2\right)\right]$$

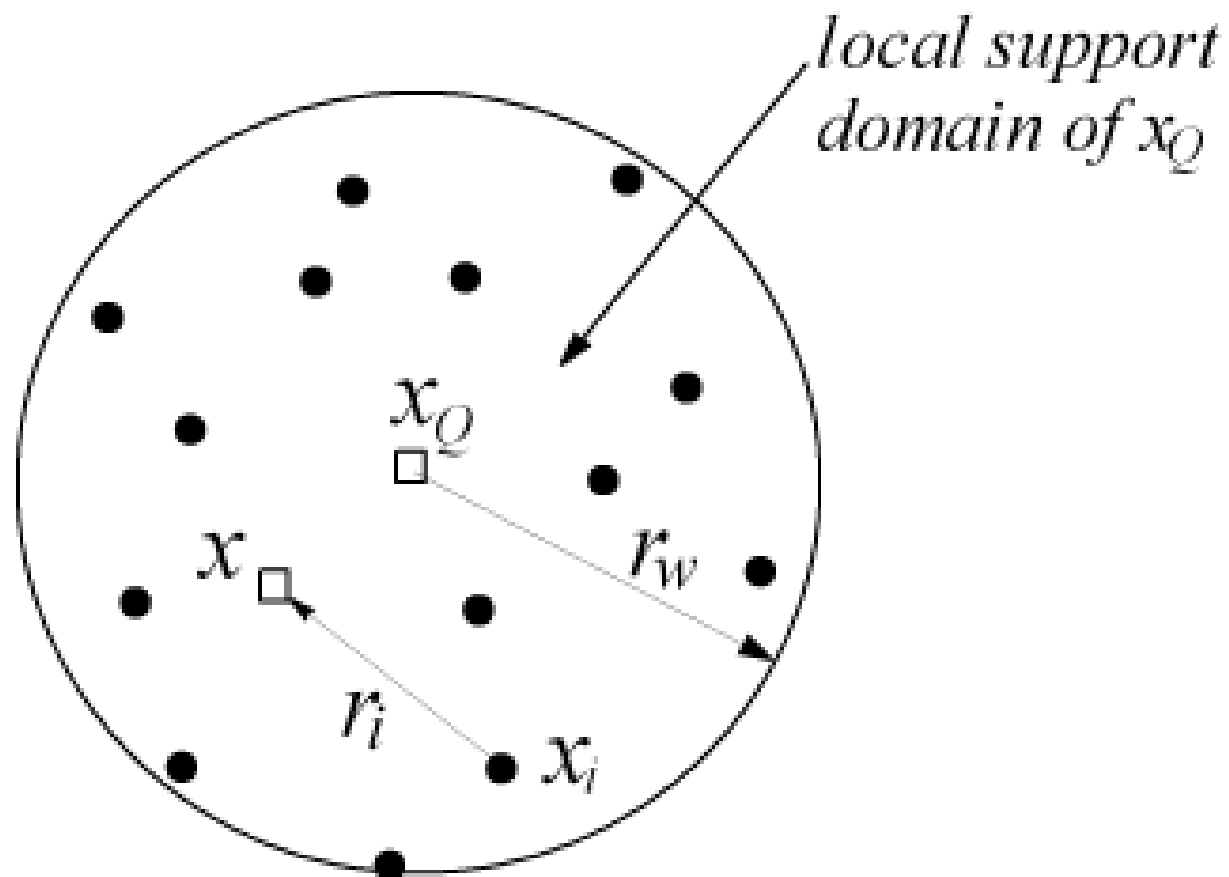
- TPS

$$R_i(x, y) = r_i^\eta = \left( (x - x_i)^2 + (y - y_i)^2 \right)^\eta$$

- Logarithmic

$$R_i(r_i) = r_i^\eta \log r_i$$

## Radial Point Interpolation (RPIM):



# Radial Point Interpolation (RPIM):

- RPIM uses the radial basis function (RBF) instead of the polynomial basis.
- The structure of approximation in RPIM is different from the PIM:
  - The PIM is based on the definition of a support domain for the point of interest  $x_Q$  and all points (including the nodes  $x_j$ ) are treated with respect to  $x_Q$ ,
  - The RPIM, however, builds the approximation on the basis of the distance of each point from the nodes  $x_j$ .
- RPIM always works and guarantees the existence of the inverse of the moment matrix. (provided that the guidelines of choosing the shape parameters of the radial basis functions are followed).

# Radial Point Interpolation (RPIM):

- The RPIM interpolation:

$$u^h(x, x_Q) = \sum_{j=1}^n R_j(x) a_j(x_Q) = R^T(x) a(x_Q)$$

- Similar to PIM, the interpolation is enforced to pass through all nodes within the support domain:

$$U_s = R_Q a$$

and

$$a = R_Q^{-1} U_s, \quad R_Q = \begin{bmatrix} R_1(r_1) & R_2(r_1) & \cdots & R_n(r_1) \\ R_1(r_2) & R_2(r_2) & \cdots & R_n(r_2) \\ \vdots & \vdots & \ddots & \vdots \\ R_1(r_n) & R_2(r_n) & \cdots & R_n(r_n) \end{bmatrix}$$

$$r_k = \left[ (x_k - x_i)^2 + (y_k - y_i)^2 \right]^{1/2}$$

# Radial Point Interpolation (RPIM):

- $R_Q$  is symmetric, because the distance is directionless:

$$R_i(r_j) = R_j(r_i)$$

- The final solution is then obtained as:

$$u^h(x) = R^T(x) R_Q^{-1} U_s$$

$$u^h(x) = \phi(x) U_s$$

$$\phi(x) = R^T(x) R_Q^{-1}$$

$$\phi_k(x) = \sum_{i=1}^n R_i(x) S_{ik} \quad , S = R_Q^{-1}$$

$S_{ik}$  is the  $ik^{\text{th}}$  element of  $R_Q^{-1}$ ; a constant matrix for given locations of  $n$  nodes in the support domain.

# Radial Point Interpolation (RPIM):

- Derivatives of RPIM shape functions are obtained in terms of derivatives of RBF functions:

$$\phi_k(x) = \sum_{i=1}^n R_i(x) S_{ik}$$

$$\phi_{k,x}(x) = \sum_{i=1}^n R_{i,x}(x) S_{ik}$$

$$\phi_{k,y}(x) = \sum_{i=1}^n R_{i,y}(x) S_{ik}$$

- For instance, derivatives of MQ basis functions are:

$$R_{i,x} = 2q(r_i^2 + c^2)^{q-1}(x - x_i)$$

$$R_{i,y} = 2q(r_i^2 + c^2)^{q-1}(y - y_i)$$

# Radial Point Interpolation (RPIM):

- The radial moment matrix  $R_Q$  is always convertible for arbitrary scattered nodes if some specific known RBF shape parameters are avoided.
- $R_Q$  is symmetric.
- RPIM is not consistent. A remedy is to add necessary polynomial terms.
- RPIM will not pass the standard patch test.
- Approximation of any continuous function using RBFs converge.
- RBF shape parameters must be tuned for better performance of the RPIM. Improper parameters will lead to poor results.

## Radial/Polynomial PIM (RPPIM):

- RPIM fails to reconstruct a linear field exactly. It is not consistent and can not pass the standard patch test.
- Adding polynomials into the radial basis functions, ensures the consistency of RPIM shape functions.
- $C^1$  consistency may be obtained by adding polynomial terms up to the linear order.
- RPPIM provides much more flexibility in choosing RBF shape parameters.

## Radial/Polynomial PIM (RPPIM):

- RPPIM approximation for a field function  $u(x)$  over the support domain of an interest point including  $n$  nodes:

$$u^h(x) = \sum_{i=1}^n R_i(x)a_i + \sum_{j=1}^m p_j(x)b_j$$

$$u^h(x) = R^T(x)a + p^T(x)b$$

with

$$a^T = \{a_1, a_2, \dots, a_n\}$$

$$b^T = \{b_1, b_2, \dots, b_m\}$$

$$R^T(x) = \{R_1(x), R_2(x), \dots, R_n(x)\}$$

$$p^T(x) = \{p_1(x), p_2(x), \dots, p_m(x)\}$$

## Radial/Polynomial PIM (RPPIM):

- The coefficients  $a$  and  $b$  are determined by enforcing that the interpolation passes through all  $n$  nodes within the support domain. For the point  $k$ :

$$u_k = u^h(x_k) = \sum_{i=1}^n R_i(x_k) a_i + \sum_{j=1}^m p_j(x_k) b_j, \quad k = 1, \dots, n$$

or

$$U_s = R_Q a + P_m b$$

- An extra requirement that guarantees a unique approximation is required:

$$\sum_{i=1}^n p_j(x_i) a_i = 0, \quad j = 1, \dots, m$$

or

$$P_m^T a = 0$$

# Radial/Polynomial PIM (RPPIM):

- The final set of simultaneous equations:

$$\begin{bmatrix} R_Q & P_m \\ P_m^T & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} U_s \\ 0 \end{bmatrix}$$

where

$$R_Q = \begin{bmatrix} R_1(r_1) & R_2(r_1) & \cdots & R_n(r_1) \\ R_1(r_2) & R_2(r_2) & \cdots & R_n(r_2) \\ \vdots & \vdots & \ddots & \vdots \\ R_1(r_n) & R_2(r_n) & \cdots & R_n(r_n) \end{bmatrix}_{n \times n}$$
$$P_m = \begin{bmatrix} p_1(x_1) & p_2(x_1) & \cdots & p_n(x_1) \\ p_1(x_2) & p_2(x_2) & \cdots & p_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(x_n) & p_2(x_n) & \cdots & p_n(x_n) \end{bmatrix}_{n \times m}$$

## Radial/Polynomial PIM (RPPIM):

- The solution can be obtained as:

$$a = S^a U_s$$

$$b = S^b U_s$$

$$S^a = R_Q^{-1} [1 - P_m S^b]$$

$$S^b = \underbrace{\left[ P_m^T R_Q^{-1} P_m \right]^{-1}}_{\text{a transformed moment matrix}} \underbrace{P_m^T R_Q^{-1}}_{\text{computed only once}}$$

$$u^h(x) = \left[ R^T(x) S^a + p^T(x) S^b \right] U_s = \phi(x) U_s$$

- The transformed matrix  $P_m^T R_Q^{-1} P_m$  is symmetric, and practically invertible as long as  $n \gg m$ .

# Radial/Polynomial PIM (RPPIM):

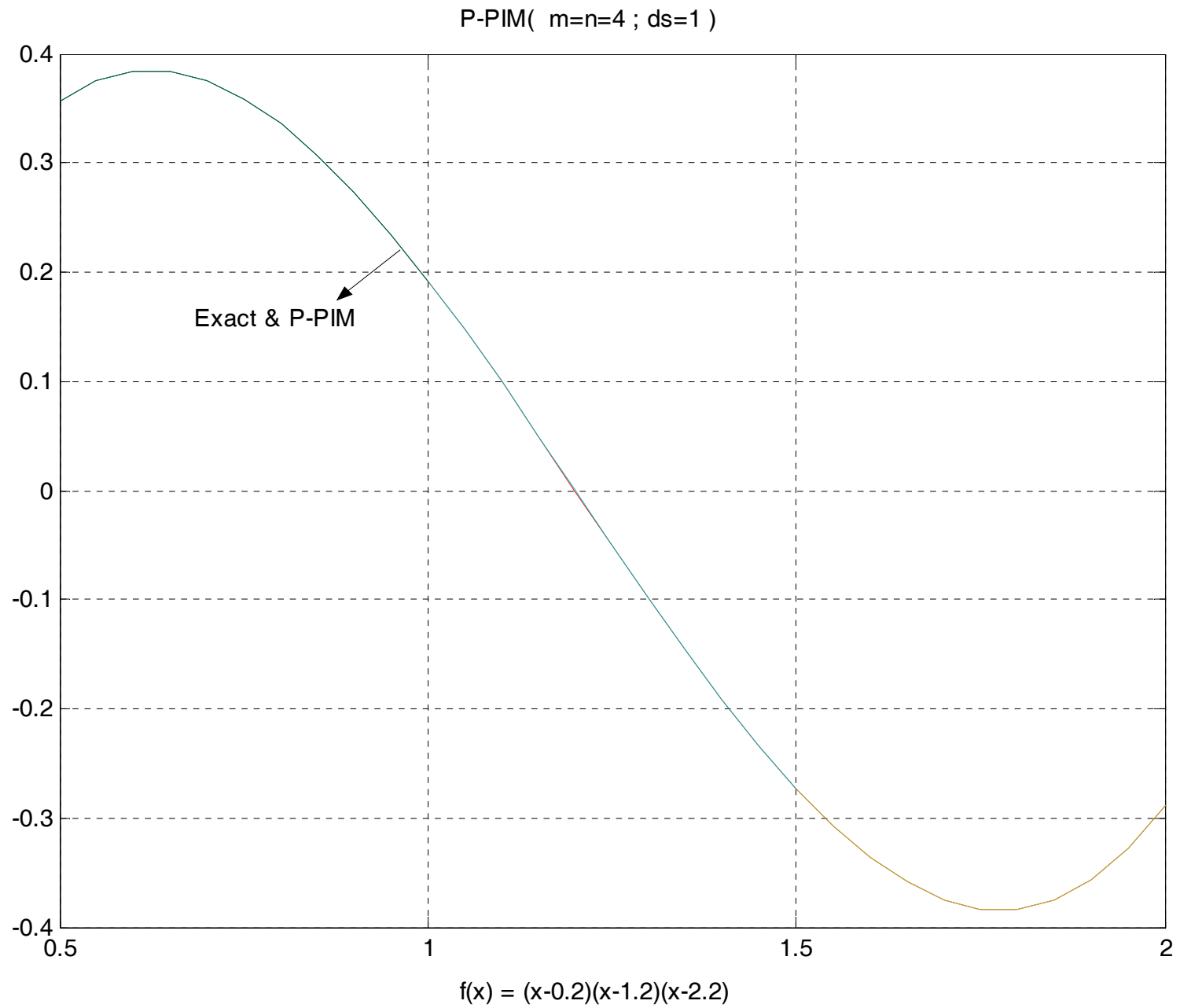
- The RPPIM shape function:

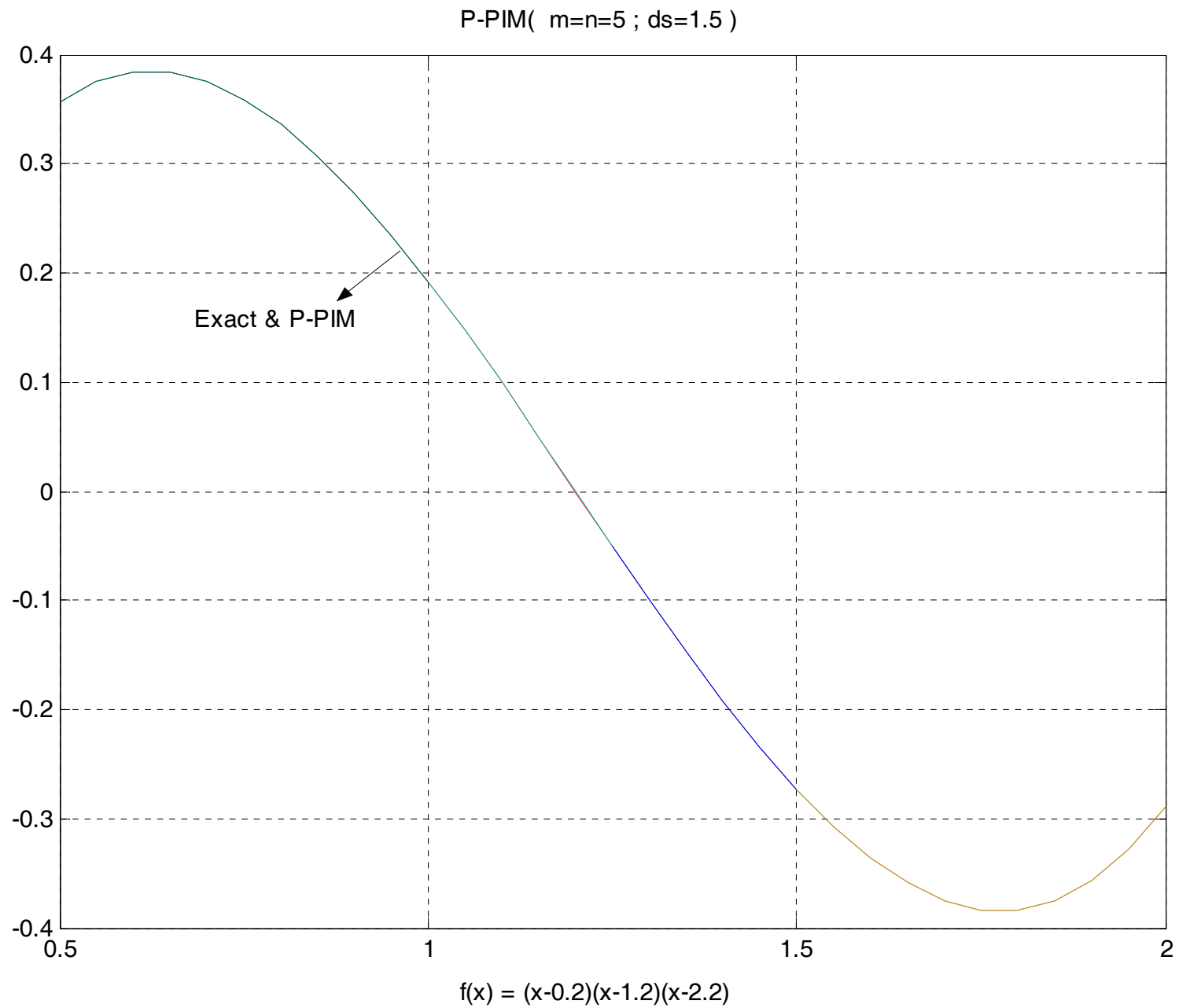
$$\phi(x) = R^T(x)S^a + p^T(x)S^b$$

$$\phi_k(x) = \sum_{i=1}^n R_i(x)S_{ik}^a + \sum_{j=1}^m p_j(x)S_{jk}^b$$

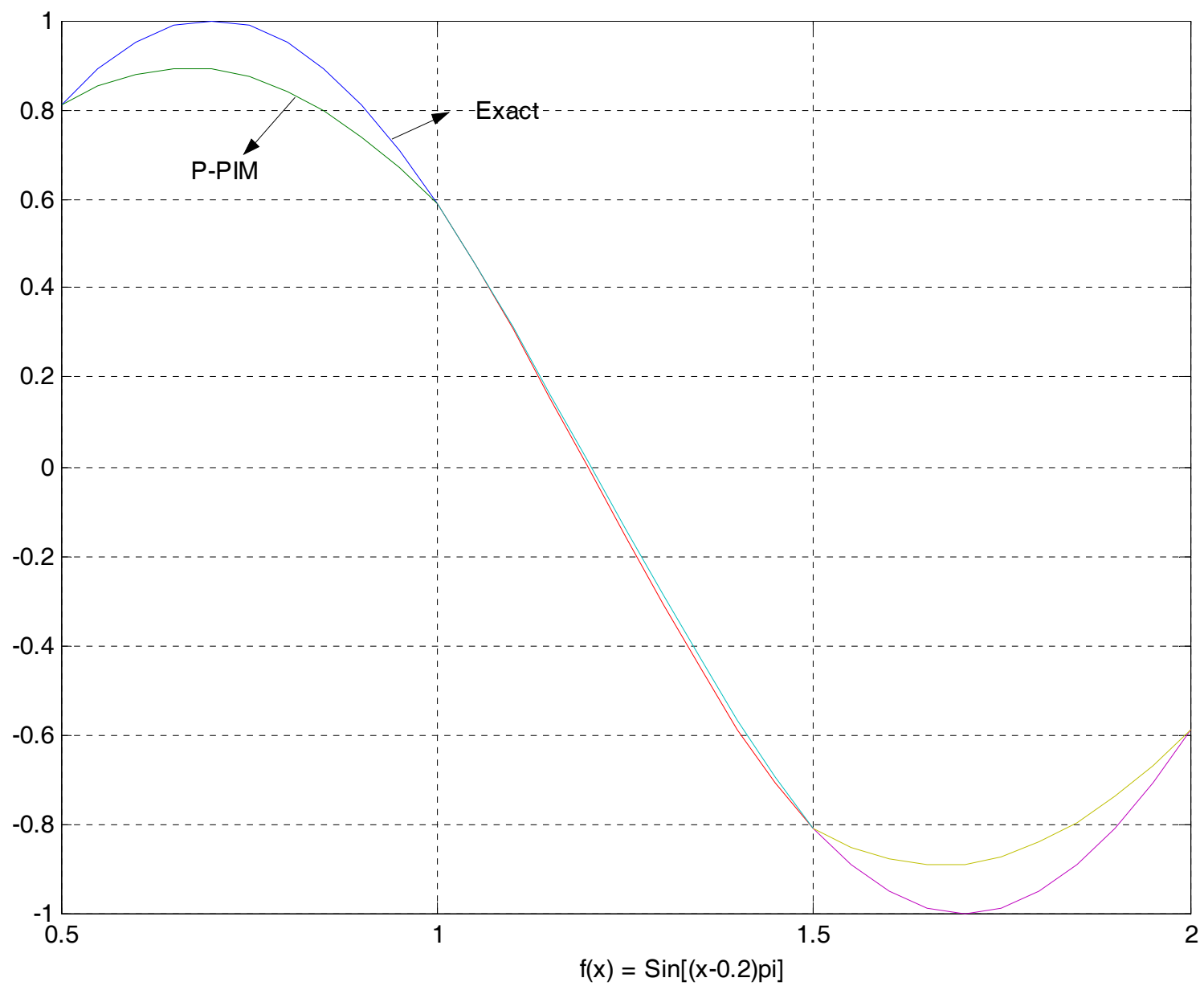
- and the shape function derivatives:

$$\phi_{k,x}(x) = \sum_{i=1}^n R_{i,x}(x)S_{ik}^a + \sum_{j=1}^m p_{j,x}(x)S_{jk}^b$$

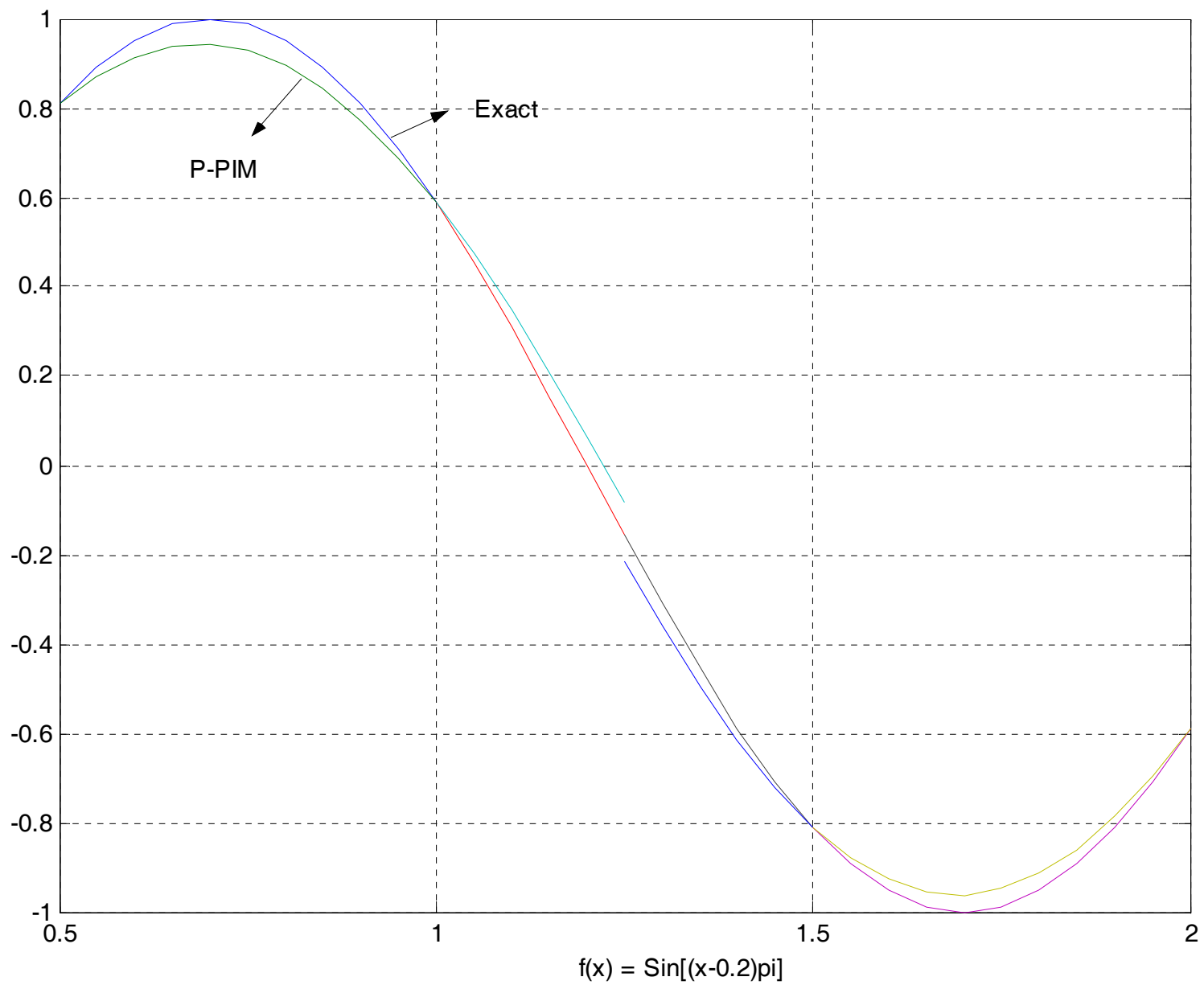


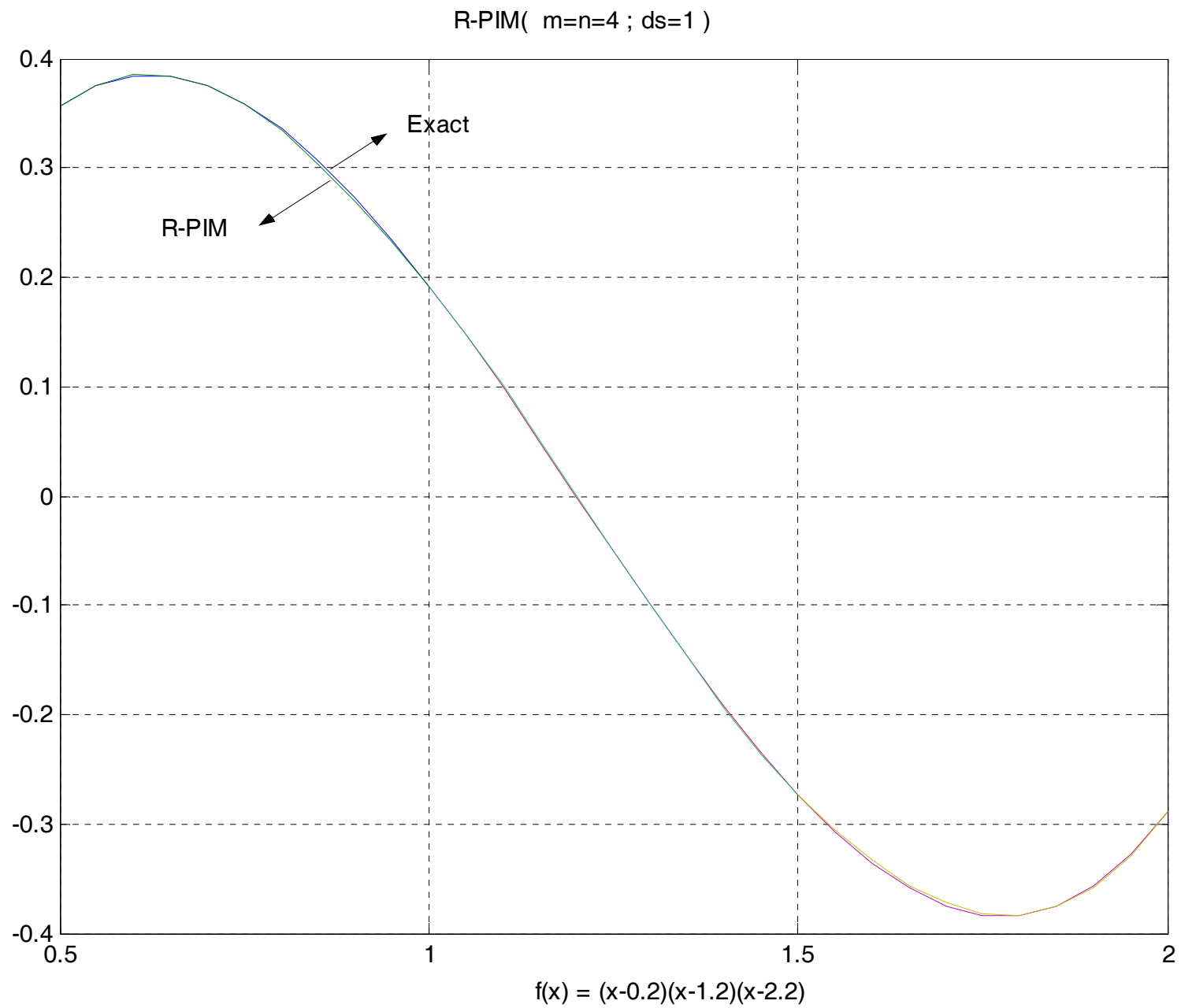


P-PIM( m=n=4 ; ds=1 )

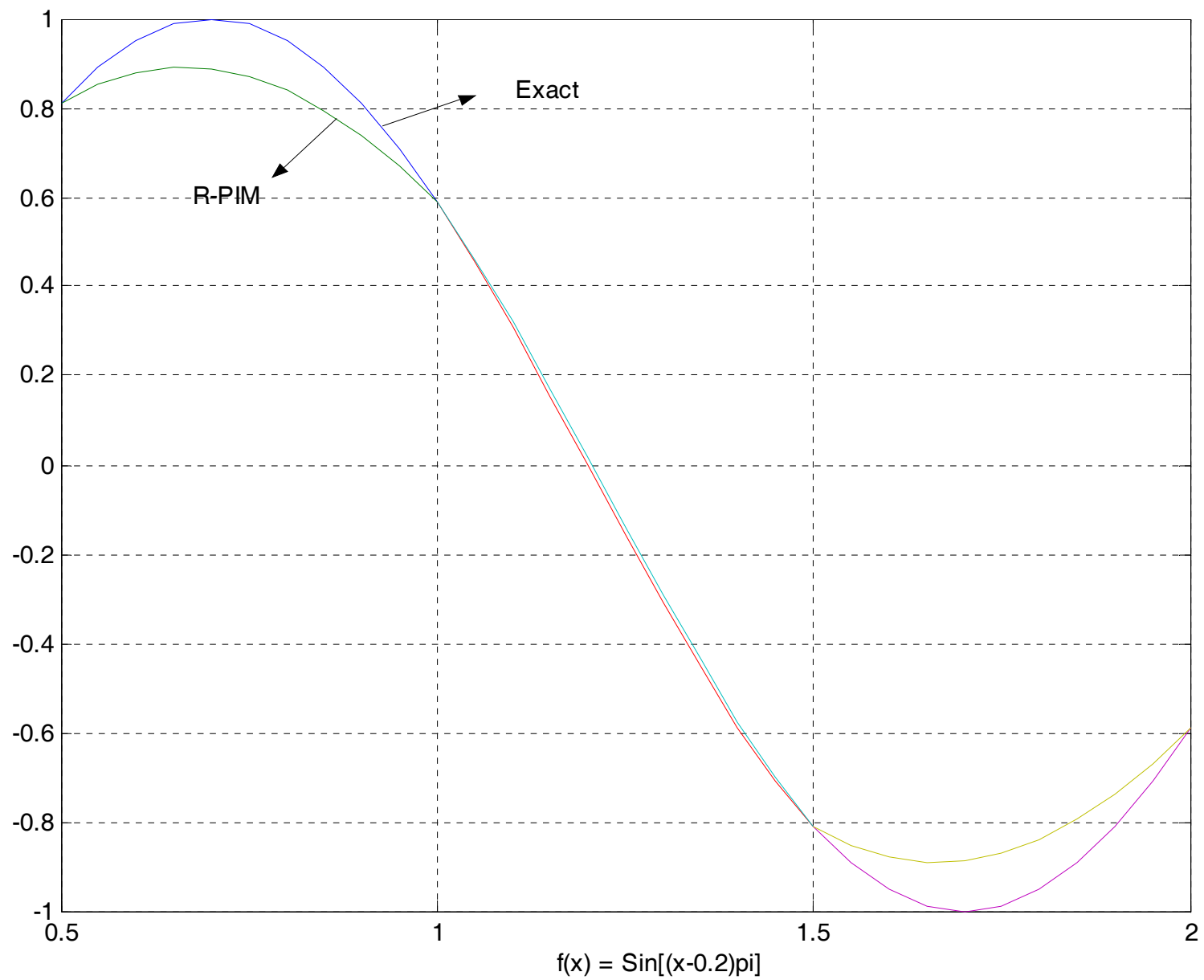


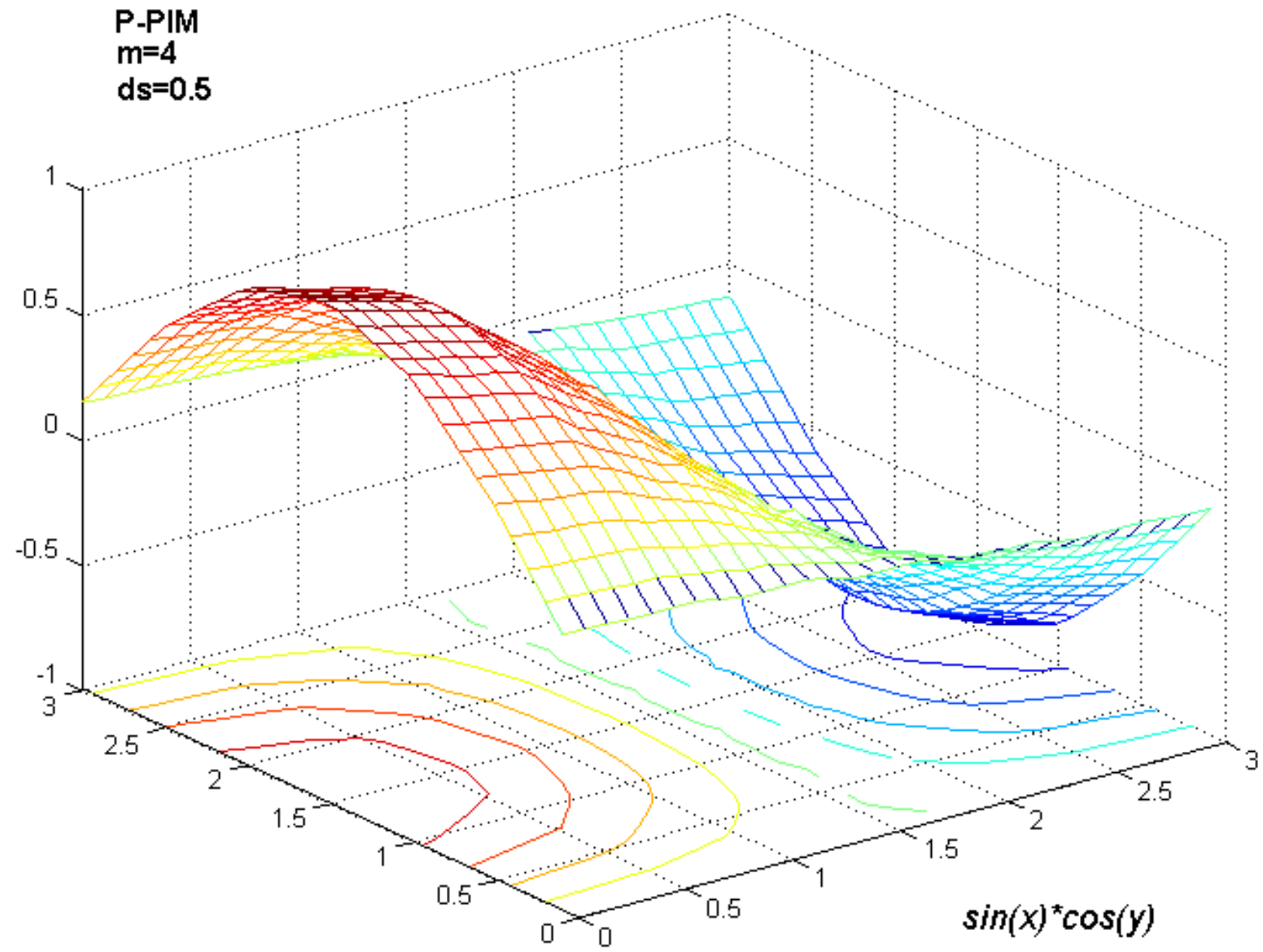
P-PIM( m=n=5 ; ds=1.5 )



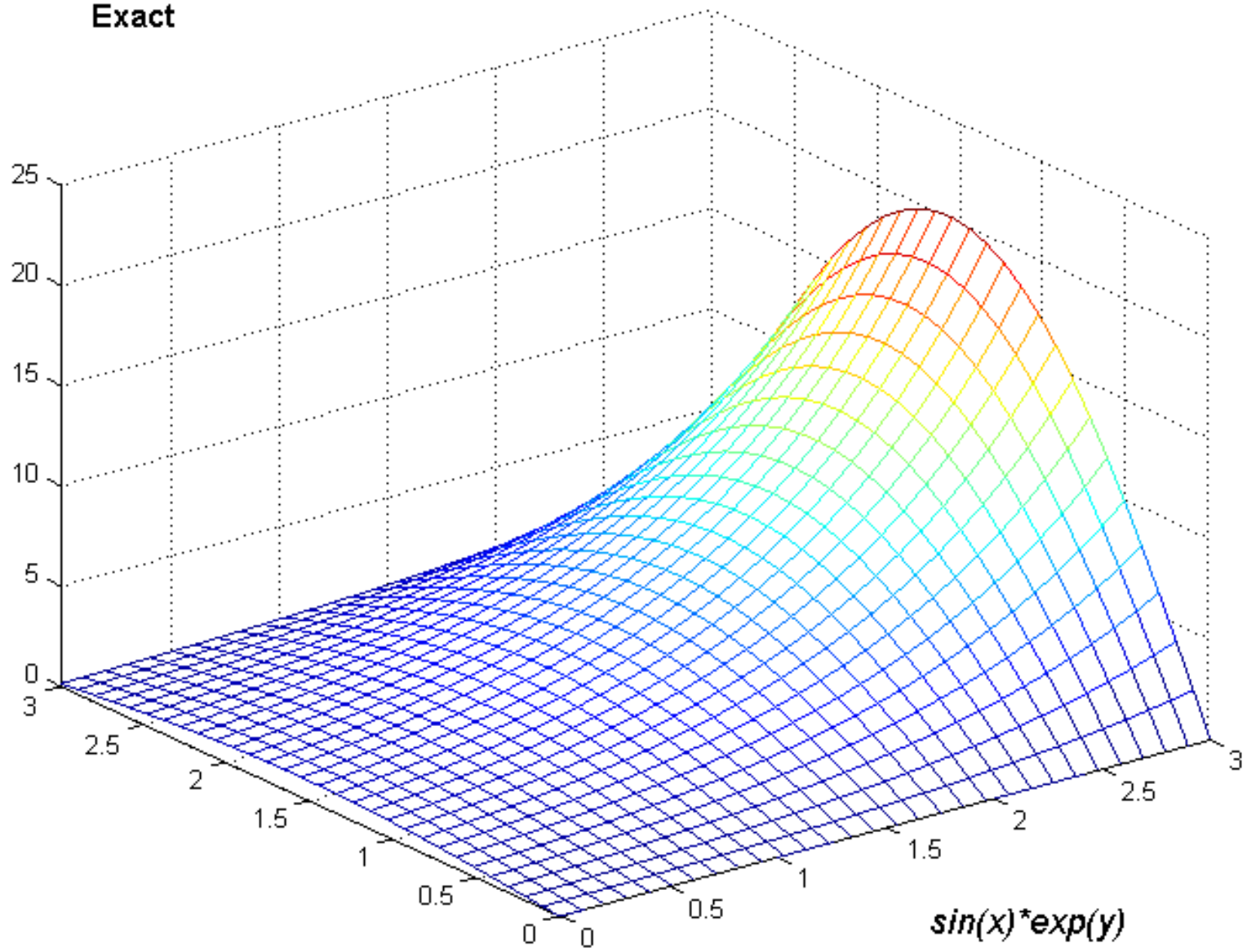


R-PIM( m=n=4 ; ds=1 )





Exact



P-PIM  
m=4  
ds=0.5

