CHAPTER 2 THEORETICAL AND BACKGROUND

The theories, that are related to this research, have been divided into 6 sections; such as governing equation, Brusselator system, constructing shape functions, test function, numerical method and finally, temporal discretization. The governing equation and Brusselator system are presented for the formulations and the meaning of the parameters. The constructing shape functions, that are used for space discretrization, are the moving Kriging approximation and the radial point interpolation method. In this thesis, a fundamental solution and a Heaviside step function are used as the test function. In a part of the numerical method are presented the domain and boundary integrals such as Trapezoidal, Simpson and midpoint rules and the Gauss-Legendre quadrature method. Moreover, the Euler, Runge-Kutta and Crank-Nicolson methods are presented for temporal discretization.

2.1 Governing Equation

The numerical simulations of the coupled pair of nonlinear partial differential equation (Shirzadi, 2013) are as follow:

$$\begin{aligned} \frac{\partial u}{\partial t} &= D_1 \nabla^2 u + \alpha_1 u + A(u,v) + f_1(\boldsymbol{x},t), \\ \frac{\partial v}{\partial t} &= D_2 \nabla^2 v + \alpha_2 v + \beta u + B(u,v) + f_2(\boldsymbol{x},t) \end{aligned} \tag{2.1}$$

given initial and Dirichlet and/or Neumann's boundary conditions in the two-dimensional region Ω, where $D_1, D_2, \alpha_1, \alpha_2$ and β are given constants, A and B are functions of the field variables u and v, f_1 and f_2 are assumed to be prescribed sources. In the case of a two-component reaction system, $u(\boldsymbol{x},t)$ and $v(\boldsymbol{x},t)$ stand for the concentrations and D_1, D_2 for the diffusion coefficients of the chemical species**.**

2.2 Brusselator System

Let $u = u(x,y,t)$ and $v = v(x,y,t)$ represent the concentrations of two reaction products at time t, A and B are constant concentrations of two input reagents, and D (a constant) represent the reactor length. Then, the partial differential equations associated with the "Brusselator" system are given by (Adomian, 1995)

$$\frac{\partial u}{\partial t} = B + u^2 v - (A+1)u + D\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right), \quad 0 < x, y < L, t > 0, \tag{2.2}$$

$$\frac{\partial v}{\partial t} = Au - u^2 v + D\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right), \qquad 0 < x, y < L, t > 0, \tag{2.3}$$

subject to Neumann boundary conditions on the boundary $\partial\Omega$ of the square Ω defined by the lines $x = 0, y = 0, x = L, y = L$, given by

$$\frac{\partial u(0,y,t)}{\partial x} = \frac{\partial u(L,y,t)}{\partial x} = \frac{\partial u(x,0,t)}{\partial y} = \frac{\partial u(x,L,t)}{\partial y} = 0, \qquad t \geq 0, \quad (2.4)$$

$$\frac{\partial v(0,y,t)}{\partial x} = \frac{\partial v(L,y,t)}{\partial x} = \frac{\partial v(x,0,t)}{\partial y} = \frac{\partial v(x,L,t)}{\partial y} = 0, \qquad t \geq 0, \quad (2.5)$$

and initial conditions

$$u(x,y,0) = f(x,y), \quad (x,y) \in \Omega \cup \partial\Omega, \qquad (2.6)$$

$$v(x,y,0) = g(x,y), \quad (x,y) \in \Omega \cup \partial\Omega,$$

where f and g are given continuous functions of x and y.

Consider the diffusion-free "Brusselator" system given by Eqs.(2.2) and (2.3) with $\alpha = 0$

$$\frac{\partial u}{\partial t} = f_1(u,v) = B + u^2 v - (A+1)u, \quad t > 0, \quad u(0) = U^0, \qquad (2.7)$$

$$\frac{\partial v}{\partial t} = f_2(u,v) = Au - u^2 v, \quad t > 0, \quad v(0) = V^0, \qquad (2.8)$$

in which $u = u(t), v = v(t)$ and A and B are positive real constants. It can be shown that the only critical point of the ordinary differential equation (ODE) system is $(u^*, v^*) = (B, A/B)$. The Jacobian, J^*, at the critical point (u^*, v^*) is given by

$$J^* = \begin{bmatrix} A-1 & B^2 \\ -A & -B^2 \end{bmatrix}$$

and its eigenvalues $\lambda_{1,2}$ satisfy the characteristic equation

$$\lambda^2 + (1 - A + B^2)\lambda + B^2 = 0$$

The roots of this equation, the eigenvalues of J^*, clearly depend on $1 - A + B^2$ and on the quantity $\Delta \equiv (1 - A + B^2)^2 - 4B^2$.

For small values of the diffusion coefficientα, if $1 - A + B^2 > 0$ then the numerical solution of the Brusselator system converges to an equilibrium points $(B, A/B)$ (Twizell,1999)

2.3 Constructing Shape Function

In this thesis, There are two methods for approximating The regular nodal shape functions such as the moving kriging approximation (MKA) method and radial point interpolation method (RPIM).

2.3.1 Moving Kriging Approximation

The kriging approximation is a well-known geostatic technique for spatial interpolation in geology and mining (Lei, 2003). The formulation of the construction of meshless

shape function by moving kriging approximation (MKA) is introduced briefly in the following.Similar to the MLS approximation, Consider the function $u(\boldsymbol{x})$ defined in the domain Ω discretized by a set of properly scattered nodes $\boldsymbol{x}_i$, $(i = 1,2,\dots,n)$, where n is the total number of nodes in the whole domain. It is assumed that only N nodes surrounding point $\boldsymbol{x}$ have the effect on $u(\boldsymbol{x})$.The sub-domain $\Omega_{\boldsymbol{x}}$ that encompasses these surrounding nodes is called the interpolation domain of point $\boldsymbol{x}$. The MKA $u^h(\boldsymbol{x})$ at point $\boldsymbol{x}$ is defined as presented in (Lei, 2003; Chen, 2011). Therefore the formulation of the meshless shape function using MKA is given by

$$u^h(\boldsymbol{x}) = \sum_{I=1}^{N} \phi_I(\boldsymbol{x}) u_I = \boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{u} \qquad , \boldsymbol{x} \in \Omega_{\boldsymbol{x}} \tag{2.9}$$

where $\boldsymbol{u} = [u(\boldsymbol{x}_1)\ u(\boldsymbol{x}_2) \cdots u(\boldsymbol{x}_N)]^T$ is a vector value of the function in the domain Ω. $\boldsymbol{\Phi}(\boldsymbol{x})$ is a $1 \times N$ vector of shape functions, expressed as:

$$\boldsymbol{\Phi}(\boldsymbol{x}) = \boldsymbol{p}^T(\boldsymbol{x})\boldsymbol{A} + \boldsymbol{r}^T(\boldsymbol{x})\boldsymbol{B}, \tag{2.10}$$

where matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are defined as:

$$\begin{aligned} \boldsymbol{A} &= (\boldsymbol{P}^T\boldsymbol{R}^{-1}\boldsymbol{P})^{-1}\boldsymbol{P}^T\boldsymbol{R}^{-1}, \\ \boldsymbol{B} &= \boldsymbol{R}^{-1}(\boldsymbol{I} - \boldsymbol{P}\boldsymbol{A}). \end{aligned} \tag{2.11}$$

In which $\boldsymbol{I}$ is a unit matrix of size $N \times N$, and vector $\boldsymbol{p}(\boldsymbol{x})$ is:

$$\boldsymbol{p}^T(\boldsymbol{x}) = [p_1(\boldsymbol{x}_1) \quad \cdots \quad p_m(\boldsymbol{x}_N)]. \tag{2.12}$$

In general, a linear basis in two-dimensional space is:

$$\boldsymbol{p}^T(\boldsymbol{x}) = (1, x, y), \ \ m = 3\ , \tag{2.13}$$

a quadratic basis is given as

$$\boldsymbol{p}^T(\boldsymbol{x}) = (1, x, y, x^2, xy, y^2), \ \ m = 6, \tag{2.14}$$

and a cubic basis is

$$\boldsymbol{p}^T(\boldsymbol{x}) = (1, x, y, x^2, xy, y^2, x^3, x^2y, xy^2, y^3), \ \ m = 10. \tag{2.15}$$

For matrix $\boldsymbol{P}$ with the size $N \times m$, values of the polynomial basis functions (2.12) at the given set of nodes are collected:

$$\boldsymbol{P} = \begin{bmatrix} p_1(\boldsymbol{x}_1) & \cdots & p_m(\boldsymbol{x}_1) \\ \cdots & \cdots & \cdots \\ p_1(\boldsymbol{x}_N) & \cdots & p_m(\boldsymbol{x}_N) \end{bmatrix}. \tag{2.16}$$

Matrices $\boldsymbol{R}$ and vector $\boldsymbol{r}(\boldsymbol{x})$ are defined by the following equations:

$$\boldsymbol{R} = \begin{bmatrix} \gamma(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & \gamma(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ \cdots & \cdots & \cdots \\ \gamma(\boldsymbol{x}_N, \boldsymbol{x}_1) & \cdots & \gamma(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{bmatrix}, \tag{2.17}$$

$$\boldsymbol{r}^T(\boldsymbol{x}) = \begin{bmatrix} \gamma(\boldsymbol{x}, \boldsymbol{x}_1) & \cdots & \gamma(\boldsymbol{x}, \boldsymbol{x}_N) \end{bmatrix}, \tag{2.18}$$

where $\gamma(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the correlation function between any pair of nodes located at $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, representing the covariance of the field value $u(\boldsymbol{x})$, i.e.

$$\gamma(\boldsymbol{x}_i, \boldsymbol{x}_j) = E\left[u(\boldsymbol{x}_i)\, u(\boldsymbol{x}_j)\right], \tag{2.19}$$

Similarly, the covariance $E\left[u(\boldsymbol{x}_i)\, u(\boldsymbol{x}_j)\right]$ can be replaced by $\gamma(\boldsymbol{x}, \boldsymbol{x}_j)$. It can be seen from the foregoing formulations that the values of matrices $\boldsymbol{R}$ and $\boldsymbol{r}$ play important roles in the computation. A simple and frequently- used correlation function is a Gaussian function:

$$\gamma(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\theta r_{ij}^2}, \tag{2.20}$$

where $r_{ij} = \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|$ and $\theta > 0$ are the correlation parameters used to fit the model and are assumed to be given.

The first-order partial derivatives of the shape function $\boldsymbol{\Phi}(\boldsymbol{x})$ against the coordinates $\boldsymbol{x}_i, i = 1,2$ can be easily obtained from Eq. (2.3)

$$\boldsymbol{\Phi}_{,i}(\boldsymbol{x}) = \boldsymbol{p}_{,i}^T(\boldsymbol{x})\boldsymbol{A} + \boldsymbol{r}_{,i}^T(\boldsymbol{x})\boldsymbol{B}, \tag{2.21}$$

where $(\cdot)_{,i}$ denotes $\partial(\cdot)/\partial x^i$.

The shape function obtained from the moving kriging approximation possesses the following delta function property:

$$\Phi_I(\boldsymbol{x}_J) = \delta_{IJ} = \begin{cases} 1, (I = J, I = 1,2, \dots, N) \\ 0, (I \neq J, I, J = 1,2, \dots, N) \end{cases} \tag{2.22}$$

The moving kriging approach is an exact interpolator, and its shape functions can exactly reproduce any function included in the basis. In particular, if all constants and linear terms are included, it reproduces a general linear polynomial exactly,

$$\left.\begin{aligned}\sum_{I=1}^{N}\Phi_I(\boldsymbol{x}) &= 1\\ \sum_{I=1}^{N}\Phi_I(\boldsymbol{x})\,x_I &= x\\ \sum_{I=1}^{N}\Phi_I(\boldsymbol{x})\,y_I &= y\end{aligned}\right\}. \tag{2.23}$$

2.3.2 Radial Point Interpolation Method

The radial point interpolation method (RPIM) $u^h(\boldsymbol{x})$ at point $\boldsymbol{x}$ is defined as presented in (Chen, 2010). Consider the function $u(\boldsymbol{x})$ defined in the domain Ω discretized by a set of properly scattered nodes $\boldsymbol{x}_i, (i = 1,2,\dots,n)$, where n is the total number of nodes in the whole domain. It is assumed that only N nodes surrounding point $\boldsymbol{x}$ have the effect on $u(\boldsymbol{x})$. The sub-domain $\Omega_{\boldsymbol{x}}$ that encompasses these surrounding nodes is called the interpolation domain of point $\boldsymbol{x}$. The $u^h(\boldsymbol{x})$ at $\boldsymbol{x}_q$ is approximated in the form of

$$u^h(\boldsymbol{x},\boldsymbol{x}_q) = \sum_{i=1}^{N} R_i(\boldsymbol{x})a_i + \sum_{j=1}^{m} P_j(\boldsymbol{x})b_j = \boldsymbol{R}^T(\boldsymbol{x})\boldsymbol{a} + \boldsymbol{P}^T(\boldsymbol{x})\boldsymbol{b}, \tag{2.24}$$

where $R_i(\boldsymbol{x})$ is a radial basis function, $P_j(\boldsymbol{x})$ is a monomial in the space coordinates $\boldsymbol{x}_i = (x_i, y_i)^T, m$ is the number of polynomial basis functions. The polynomial basis function is defined as the MKA method. Coefficients a_i and b_j are constants yet to be determined. We also choose $m < N$ to have better stability of the interpolating function. For give $\boldsymbol{x}$, we have

$$\boldsymbol{a} = [a_1, a_2, \dots, a_N]^T, \qquad \boldsymbol{b} = [b_1, b_2, \dots, b_m]^T$$

$$\boldsymbol{R}^T(\boldsymbol{x}) = [R_1(\boldsymbol{x}), R_2(\boldsymbol{x}), \dots, R_N(\boldsymbol{x})], \qquad \boldsymbol{P}^T(\boldsymbol{x}) = [P_1(\boldsymbol{x}), P_2(\boldsymbol{x}), \dots, P_m(\boldsymbol{x})]$$

Typically, in two-dimensional problems

$$\boldsymbol{R}_i(\boldsymbol{x}) = \boldsymbol{R}_i(r_i) = R_i(x, y) \tag{2.25}$$

$$\boldsymbol{r}_i(\boldsymbol{x}) = \sqrt{(x - x_i)^2 + (y - y_i)^2} \tag{2.26}$$

Now enforcing equation (2.24) to be satisfied at nodes to determine the coefficients a_i and b_i. The matrix form is:

$$\boldsymbol{U}_s = \boldsymbol{R}_Q\boldsymbol{a} + \boldsymbol{P}_m\boldsymbol{b} \tag{2.27}$$

where $\boldsymbol{U}_s = [u_1, u_2, \dots . u_N]^T$. Matrix $\boldsymbol{R}_Q$ is given by

$$\boldsymbol{R}_Q = \begin{bmatrix} R_1(\boldsymbol{x}_1) & R_2(\boldsymbol{x}_1) & \cdots & R_N(\boldsymbol{x}_1) \\ R_1(\boldsymbol{x}_2) & R_2(\boldsymbol{x}_2) & \cdots & R_N(\boldsymbol{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ R_1(\boldsymbol{x}_N) & R_2(\boldsymbol{x}_N) & \cdots & R_N(\boldsymbol{x}_N) \end{bmatrix}_{N\times N}. \tag{2.28}$$

The matrix $\boldsymbol{P}_m$ is a $N \times m$ matrix given by

$$\boldsymbol{P}_m = \begin{bmatrix} P_1(\boldsymbol{x}_1) & P_2(\boldsymbol{x}_1) & \cdots & P_m(\boldsymbol{x}_1) \\ P_1(\boldsymbol{x}_2) & P_2(\boldsymbol{x}_2) & \cdots & P_m(\boldsymbol{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ P_1(\boldsymbol{x}_N) & P_2(\boldsymbol{x}_N) & \cdots & P_m(\boldsymbol{x}_N) \end{bmatrix}_{N\times m}. \tag{2.29}$$

However, there are $N + m$ variables in equation (2.27), but only have N equations, so it is an undetermined equations, solving the above equation (2.27) needs to impose a constraint equation

$$\boldsymbol{P}_m^T \boldsymbol{a} = 0. \tag{2.30}$$

Combing equations (2.20) and (2.23), the matrix form becomes

$$\begin{bmatrix} \boldsymbol{R}_Q & \boldsymbol{P}_m \\ \boldsymbol{P}_m^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix} = \begin{bmatrix} \boldsymbol{U}_s \\ 0 \end{bmatrix}. \tag{2.31}$$

Solving equation (2.24), we can obtain

$$\boldsymbol{b} = \boldsymbol{S}_b \boldsymbol{U}_s, \tag{2.32}$$

$$\boldsymbol{a} = \boldsymbol{S}_a \boldsymbol{U}_{s,} \tag{2.33}$$

where $\boldsymbol{S}_b = \left[\boldsymbol{P}_m^T \boldsymbol{R}_Q^{-1} \boldsymbol{P}_m\right]^{-1} \boldsymbol{P}_m \boldsymbol{R}_Q^{-1}$, $\boldsymbol{S}_a = \boldsymbol{R}_Q^{-1} - \boldsymbol{R}_Q^{-1} \boldsymbol{P}_m \boldsymbol{S}_b$.

Substituting $\boldsymbol{a}, \boldsymbol{b}$ back into equation (2.17), we obtain

$$\boldsymbol{U}^h(\boldsymbol{x}, \boldsymbol{x}_q) = [\boldsymbol{R}^T \boldsymbol{S}_a + \boldsymbol{P}^T \boldsymbol{S}_b] \boldsymbol{U}_s = \sum_{i=1}^{N} \boldsymbol{\Phi}_i(\boldsymbol{x})\, u_i = \boldsymbol{\Phi}(\boldsymbol{x}) \boldsymbol{U}_s\,, \tag{2.34}$$

where shape function $\boldsymbol{\Phi}(\boldsymbol{x})$ with the size $1 \times N$ is given by

$$\boldsymbol{\Phi}(\boldsymbol{x}) = \left[\boldsymbol{\Phi}_1(\boldsymbol{x}), \boldsymbol{\Phi}_2(\boldsymbol{x}), \ldots, \boldsymbol{\Phi}_N(\boldsymbol{x})\right]. \tag{2.35}$$

A radial basis function is defined as: Gaussian (EXP):

$$R_i(\boldsymbol{x}) = e^{(-cr^2)} = e^{\left(-c[(x-x_i)^2 + (y-y_i)^2]\right)}, \tag{2.36}$$

where c is all called shape-parameter.

$$\frac{\partial \Phi_k}{\partial x} = \sum_{i=1}^{N} \frac{\partial R_i}{\partial x} S_{ik}^a + \sum_{j=1}^{N} \frac{\partial P_j}{\partial x} S_{jk}^b ,$$
$$\frac{\partial \Phi_k}{\partial y} = \sum_{i=1}^{N} \frac{\partial R_i}{\partial y} S_{ik}^a + \sum_{j=1}^{N} \frac{\partial P_j}{\partial y} S_{jk}^b . \quad (2.37)$$

2.4 Test Function

2.4.1 The fundamental solution

A fundamental solution u^* in infinite space, to the following differential equation (Atluri, 2004):

$$u^*_{,kk}(\boldsymbol{x}, \boldsymbol{y}) + \delta(\boldsymbol{x} - \boldsymbol{y}) = 0, \quad (2.38)$$

defind in an infinite space, where $\delta(\boldsymbol{x} - \boldsymbol{y})$ is the Dirac delta function; $\boldsymbol{x}$ is usually called the generic or field point; $\boldsymbol{y}$ is the source point; and $(\cdot)_{,ii}$ is the second order derivative by k. Note that the linear differential operator in the differential operator in the differential equation governing u^* is the same as in that governing the trial function u.For the 2-D Laplace problem, the fundamental solution u^* is known to be:

$$u^*(\boldsymbol{x}, \boldsymbol{y}) = -\frac{1}{2\pi} \ln r, \qquad r = \|\boldsymbol{x} - \boldsymbol{y}\|,$$

If we consider the solution u of the Laplacian differential equation $u_{,ii} = 0$ as the potential distribution, then the fundamental solution may be physically interpreted as the potential in an infinite space due to a source of unit strength at the source point $\boldsymbol{y}$. The fundamental solution is sometimes called the singular solution due to the fact that it is singular at $\boldsymbol{x} = \boldsymbol{y}$.

The trial solution (or the companion solution) $\tilde{u}$ is associated with the fundamental solution u^* and is defined as the solution of the following Dirichlet problem over the sub-domain Ω_s',

$$\nabla^2 \tilde{u} = 0, \text{ on } \Omega_s' \quad \text{and} \quad \tilde{u} = u^*(\boldsymbol{x}, \boldsymbol{y}) \quad \text{on } \partial\Omega_s' \quad (2.39)$$

where $\Omega_s' \subseteq \Omega_s$ such that $\Omega_s' = \Omega_s$ for an interior source point $\boldsymbol{y}$; and Ω_s' is the extended whole sphere which encloses $\partial\Omega_s$, a part of the sphere, for a boundary source point $\boldsymbol{y}$ (see Fig.2.1).

Upon solving the companion solution $\tilde{u}^*(\boldsymbol{x}, \boldsymbol{y})$, we can solve the nonlinear problem by using a numerical discretization technique. Over the regular shaped sphere Ω_s' and Ω_s, The companion solution $\tilde{u}$ can be easily and analytically obtained for most differential operators for which the fundamental solution are available. For the current 2-D potential problem, the sub domain Ω_s is a circle of radius r_0.

The companion solution to Eq.(2.39) on the circle is a constant and given by

$$\tilde{u} = -\frac{1}{2\pi} lnr_0.$$

Therefore, the modified test function becomes

$$\tilde{u}^* = u^* - \tilde{u} = \frac{1}{2\pi} ln\left(\frac{r_0}{r}\right). \tag{2.40}$$

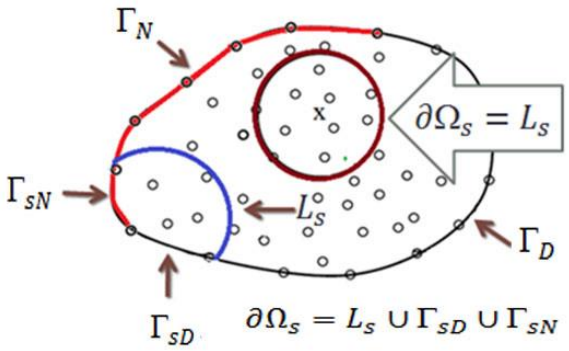

Figure 2.1 Local boundaries, the supports of nodes , the domain of definition of the MLS approximation for the trial function at a point.

2.4.2 The Heaviside Step Function

The following piecewise constant function or Heaviside step functions, which is used as test function, is given by

$$w(\boldsymbol{x}) = \begin{cases} 1, & \boldsymbol{x} \in \Omega_s^i, \\ 0, & \boldsymbol{x} \notin \Omega_s^i. \end{cases} \tag{2.41}$$

2.5 Numerical Method

2.5.1 Trapezoidal Rule

Trapezoidal Rule is based on the Newton-Cotes formula that states if one can approximate the integrand as n[th] order polynomial

$$\boldsymbol{I} = \int_a^b f(\boldsymbol{x})\, d\boldsymbol{x},$$

where $f(\boldsymbol{x}) \approx f_{\boldsymbol{n}}(\boldsymbol{x})$ and $f_n(\boldsymbol{x}) = a_0 + a_0 x + \cdots + a_{n-1}x^{n-1} + a_n x^n$.
Then the integral of that function is approximated by the integral of n^{th} order polynomial.

$$\int_{\boldsymbol{a}}^{\boldsymbol{b}} f(\boldsymbol{x})\, d\boldsymbol{x} \approx \int_{\boldsymbol{a}}^{\boldsymbol{b}} f_n(\boldsymbol{x})\, d\boldsymbol{x} \tag{2.42}$$

Trapezoidal Rule assumes $n = 1$, that is, the area under the linear polynomial,

$$\int_{a}^{b} f(x)\,dx = (b-a)\left[\frac{f(a)+f(b)}{2}\right]. \tag{2.43}$$

The area under the curve is a trapezoid. The integral

$$\int_{a}^{b} f(x)\,dx \approx Area\ of\ trapezoid$$

$$= \frac{1}{2}(Sum\ of\ parallel\ sides)(height)$$

$$= \frac{1}{2}\big(f(b)+f(a)\big)(b-a)$$

$$= \frac{1}{2}\big(f(b)+f(a)\big)\left[\frac{f(a)+f(b)}{2}\right]$$

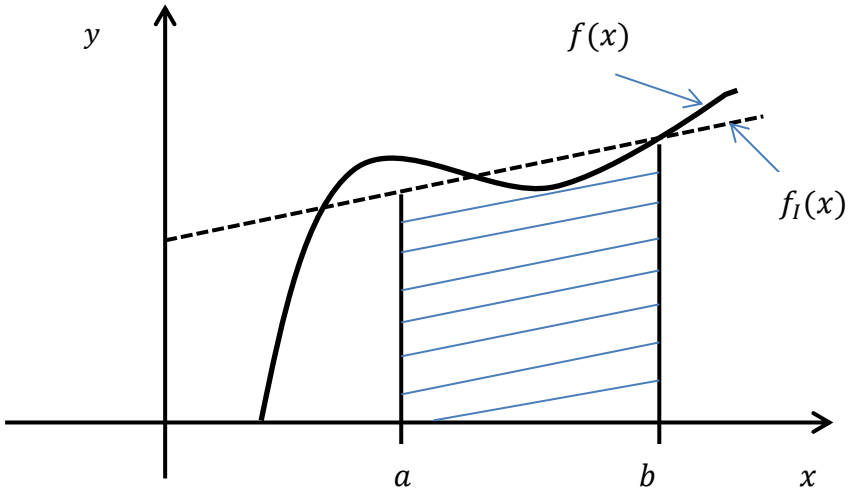

Figure. 2.2 Geometric representation.

2.5.2 Simpson Rule

Simpson's rule is a method for numerical integration, the numerical approximation of definite integrals. Specifically, it is the following approximation:

$$\int_{a}^{b} f(x)\,dx \approx \frac{b-a}{6}\left[f(a)+4f\left(\frac{a+b}{2}\right)+f(b)\right].$$

Simpson's rule also corresponds to the three-point Newton-Cotes quadrature rule. The method is credited to the mathematician Thomas Simpson (1710–1761) of

Leicestershire, England. Kepler used similar formulas over 100 years prior. In German, the method is sometimes called Keplersche Fassregel for this reason. Simpson's rule is a staple of scientific data analysis and engineering. Simpson's rule can be derived in various ways.

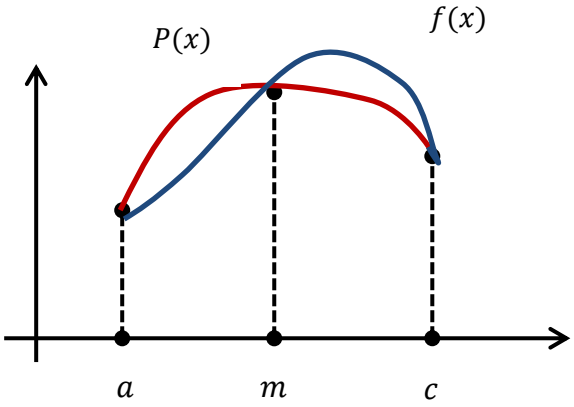

Figure2.3 Simpson's rule can be derived by approximating the integrand $f(x)$ by the quadratic interpolant $P(x)$.

Quadratic Interpolation

One derivation replaces the integrand $f(x)$ by the quadratic polynomial (i.e. parabola) $p(x)$ which takes the same values as $f(x)$ at the end points a and b and the midpoint $m = (a + b)/2$. One can use Lagrange polynomial interpolation to find an expression for this polynomial,

$$P(x) = f(a)\frac{(x-m)(x-b)}{(a-m)(a-b)} + f(m)\frac{(x-a)(x-b)}{(m-a)(m-b)}$$

$$+f(b)\frac{(x-a)(x-m)}{(b-a)(b-m)},$$

$$\int_a^b P(x)\,dx \approx \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right].$$

This calculation can be carried out more easily if one first observes that (by scaling) there is no loss of generality in assuming that $a = -1$ and $b = 1$.

2.4.3 Midpoint Rule

Midpoint Rule

$$\int_a^b f(x)\,dx \approx M_n = \Delta x[f(\bar{x}_1) + f(\bar{x}_2) + \cdots + f(\bar{x}_n)]$$

where

$$\Delta x = \frac{b-a}{n}$$

and $\bar{x}_1 = \frac{1}{2}(x_{i-1} + x_i) =$ midpoint of $[x_{i-1}, x_i]$.

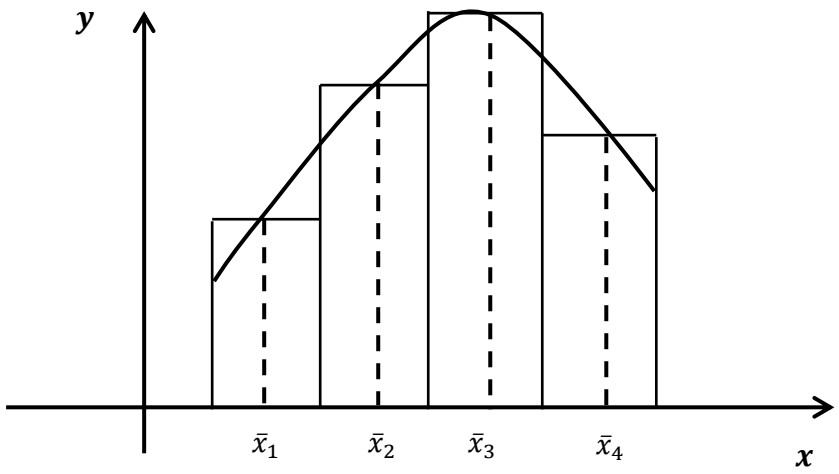

Figure2.4 Midpoint approximation.

2.5.4 Gauss-Legendre Quadrature Method

For 1-D, Let x_i be nodes and w_i be weights. The quadrature techniques formulation is following as (Abbott, 2005):

$$I_1 = \int_a^b f(x)dx, \tag{2.44}$$

where $f(x)$ be a polynomial of order 2n-1. Let [a,b] be [-1,1] can be accomplished by scaling.

$$\int_{-1}^{1} f(x)dx = \int_{-1}^{1} g(\xi)d\xi \approx \sum_{i=1}^{n} w_i\, g_i \tag{2.45}$$

where ξ be transformation of variable x and $g(\xi)$ be transformation of variable x.

For 2-D, Let x_i and y_i, $i = 1,2,\dots,n$ are nodes and w_i and w_j are weights.

The quadrature techniques formulation is following as :

$$I_2 = \int_a^b f(x,y)d\Omega = \int_{-1}^{1}\int_{-1}^{1} g(\xi,\eta)d\,\xi d\eta \approx \int_{-1}^{1} I_1\, d\eta$$

$$= \int_{-1}^{1} \sum_{i=1}^{n_\xi} w_i\, g_i d\eta \approx \sum_{j=1}^{n_\eta} w_j \left(\sum_{i=1}^{n_\xi} w_i\, g_{ij} \right) = \sum_{j=1}^{n_\eta}\sum_{i=1}^{n_\xi} w_i w_j\, g_{ij}, \tag{2.46}$$

where ξ_i, η_j are transformation of variables and g_{ij} is transformation function.

Calculating for weight

If we change the boundary condition form [-1,1]. We defined

$$x = C_0 + C_1\lambda,$$

where C_0, C_1 are unknown constants.

$$a = C_0 + C_1(-1),$$

$$b = C_0 + C_1(-1),$$

$$C_0 = \frac{\begin{vmatrix} a & -1 \\ b & 1 \end{vmatrix}}{\begin{vmatrix} 1 & -1 \\ 1 & 1 \end{vmatrix}} = \frac{a+b}{1+1} = \frac{a+b}{2},$$

$$C_1 = \frac{\begin{vmatrix} 1 & a \\ 1 & b \end{vmatrix}}{\begin{vmatrix} 1 & -1 \\ 1 & 1 \end{vmatrix}} = \frac{b-a}{1+1} = \frac{b-a}{2},$$

Hence

$$x = \left(\frac{a+b}{2}\right) + \left(\frac{b-a}{2}\right)\lambda$$

$$dx = \left(\frac{b-a}{2}\right)d\lambda$$

$$I = \int_a^b f(x)dx = \int_{-1}^{1} f(\lambda)\left(\frac{b-a}{2}\right)d\lambda \approx \left(\frac{b-a}{2}\right)\sum_{i=1}^{n} w_i\, f(\lambda_i)$$

For 2-D, we used Gauss-Legendre polynomial $(1, \lambda, \lambda^2, \lambda^3)$ then

$$w_1 f(\lambda_1) + w_2 f(\lambda_2) = \int_{-1}^{1} 1 d\lambda = 2,$$

$$w_1 f(\lambda_1) + w_2 f(\lambda_2) = \int_{-1}^{1} \lambda d\lambda = 0,$$

$$w_1 f(\lambda_1) + w_2 f(\lambda_2) = \int_{-1}^{1} \lambda^2 d\lambda = \frac{2}{3},$$

$$w_1 f(\lambda_1) + w_2 f(\lambda_2) = \int_{-1}^{1} \lambda^3 d\lambda = 0.$$

Hence

$$\begin{aligned} w_1 f(1) + w_2 f(1) &= 2, \\ w_1 f(\lambda_1) + w_2 f(\lambda_2) &= 0 \\ w_1 f\left(\lambda_1^2\right) + w_2 f\left(\lambda_2^2\right) &= \frac{2}{3} \\ w_1 f\left(\lambda_1^3\right) + w_2 f\left(\lambda_2^3\right) &= 0. \end{aligned} \tag{2.47}$$

The answers of equation (2.47) are $w_1 = w_2 = 1$, $\lambda_1 = -\frac{\sqrt{3}}{2}$ and $\lambda_2 = \frac{\sqrt{3}}{2}$. Some low-order rules for solving the integration problem are listed table1.

Table 2.1 Abscissas and weights for Gaussian quadrature.

Number of points, *n*	Points, x_i	Weights, w_i
1	0	2
2	$\pm\frac{\sqrt{3}}{2}$	1
3	0	$\frac{8}{9}$
	$\pm\sqrt{\frac{3}{5}}$	$\frac{5}{9}$
4	$\pm\sqrt{\frac{\left(3-2\sqrt{6/5}\right)}{7}}$	$\frac{18+\sqrt{30}}{36}$
	$\pm\sqrt{\frac{\left(3+2\sqrt{6/5}\right)}{7}}$	$\frac{18-\sqrt{30}}{36}$
5	0	$\frac{128}{225}$
	$\pm\frac{1}{3}\sqrt{5-2\sqrt{10/7}}$	$\frac{322+13\sqrt{70}}{900}$
	$\pm\frac{1}{3}\sqrt{5-2\sqrt{10/7}}$	$\frac{322-13\sqrt{70}}{900}$

2.6 Temporal Discretization

2.6.1 Euler Method

Often it is not possible or desirable to solve a differential equation,

$$\frac{dP}{dt} = f(P),$$

analytically, and one turns to numerical or computational methods. A numerical method seeks to approximate the solution to the equation at discrete times. Time is subdivided into intervals of length Δt , so that $t_n = n\Delta t$, and then the method approximates the solution at those times, $P_n \approx P(t_n)$. One of the oldest ideas for doing this is the Euler method. Since $\frac{dP}{dt} \approx \frac{\Delta P}{\Delta t}$ one may write

$$\frac{\Delta P}{\Delta t} = \frac{P(t_n + \Delta t) - P(t_n)}{\Delta t} = \frac{P_{n+1} - P_n}{\Delta t} \approx \frac{dP}{dt} = f(P_n).$$

Solving this expression for P_{n+1} you end up with a discrete equation which predicts a future value of P, P_{n+1}, in terms of a past value:

$$P_{n+1} = P_n + \Delta t f(P_n).$$

This can be used to approximate solutions to the differential equation.

2.6.2 Runge-Kutta Method

One member of the family of Runge–Kutta methods is often referred to as "RK4", "classical Runge–Kutta method" or simply as "the Runge–Kutta method". Let an initial value problem be specified as follows.

$$\dot{y} = f(t, y), \qquad y(t_0) = y_0.$$

Here, y is an unknown function (scalar or vector) of time t which we would like to approximate; we are told that $\dot{y}$, the rate at which y changes, is a function of t and of y itself. At the initial time t_0 the corresponding y-value is y_0. The function f and the data $t_0, y_0$are given.

Now pick a step-size $h > 0$ and define

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$t_{n+1} = t_n + h$$

for n = 0, 1, 2, 3, . . . , using

$$k_1 = f(t_n, y_n),$$

$$k_2 = f\left(t_n + \tfrac{h}{2}, y_n + \tfrac{h}{2}k_1\right),$$

$$k_3 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right),$$

$$k_4 = f(t_n + h, y_n + hk_3),$$

Here y_{n+1} is the RK4 approximation of (t_{n+1}) , and the next value (y_{n+1}) is determined by the present value (y_n) plus the weighted average of four increments, where each increment is the product of the size of the interval, h, and an estimated slope specified by function f on the right-hand side of the differential equation.

1. k_1 is the increment based on the slope at the beginning of the interval, using $\dot{y}$, (Euler's method) ;

2. k_2 is the increment based on the slope at the midpoint of the interval, using $\dot{y} + \frac{h}{2}k_1$;

3. k_3 is again the increment based on the slope at the midpoint, but now using $\dot{y} + \frac{h}{2}k_2$;

4. k_4 is the increment based on the slope at the end of the interval, using $\dot{y} + hk_3$.

2.6.3 Crank-Nicolson Method

The Crank–Nicolson method is based on the trapezoidal rule, giving second-order convergence in time. For example, in one dimension, if the partial differential equation is

$$\frac{\partial u}{\partial t} = F\left(u, x, t, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}\right),$$

then, letting $u(i\Delta x, n\Delta t) = u_i^n$, the equation for Crank–Nicolson method is a combination of the forward Euler method at n and the backward Euler method at $n + 1$ (note, however, that the method itself is not simply the average of those two methods, as the equation has an implicit dependence on the solution):

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = F_i^n\left(u, x, t, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}\right), \tag{2.48}$$

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = F_i^{n+1}\left(u, x, t, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}\right), \tag{2.49}$$

Hence (Eq.(2.48)+Eq.(2.49))/2 reveals in Eq.(2.50),

$$\frac{u_i^{n+1}-u_i^n}{\Delta t}=\frac{1}{2}\left[F_i^{n+1}\left(u,x,t,\frac{\partial u}{\partial x},\frac{\partial^2 u}{\partial x^2}\right)+F_i^n\left(u,x,t,\frac{\partial u}{\partial x},\frac{\partial^2 u}{\partial x^2}\right)\right], \quad (2.50)$$

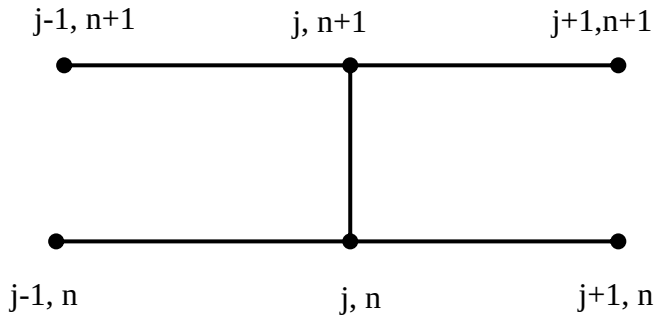

Figure2.5 The Crank–Nicolson stencil for a 1-D problem.