Chapter V

Numerical Solution of Fuzzy Differential Equations by the RKGL Method
Chapter 5

Numerical Solution of Fuzzy Differential Equations by the RKGL Method

5.1 Introduction

Allahviranloo and Otadi [12] investigated the Gaussian quadratures for approximation of fuzzy integrals. Subsequently they studied the Gaussian quadratures for approximation of fuzzy multiple integrals in [13]. Efficient s-stage Runge-Kutta methods are used for the numerical solution of the fuzzy differential equations and the convergence of the methods is proved by Palligkinis et al. [72]. The solution of fuzzy differential equations by differential transformation method is established in [10]. Abbasbandy and Allahviranloo [3] studied the numerical solution of fuzzy differential equations by using the Runge-Kutta method. Recently Pederson and Sambandham [82] discussed the hybrid fuzzy differential equations by initial value problems by a characterization theorem. In this chapter we study the numerical solution of fuzzy differential equations by the method of RKGL. Further we obtain the numerical solutions of hybrid fuzzy differential equations by using the above method.
5.2 Preliminaries

Runge-Kutta (RK) Method:

A very popular and common method to solve the fuzzy initial value problem is Runge-Kutta method. The general form of RK method is

$$y_{i+1}^\alpha = y_i^\alpha + h_i F(t_i, y_i),$$

$$\bar{y}_{i+1}^\alpha = \bar{y}_i^\alpha + h_i \bar{F}(t_i, y_i),$$

where $h_i = x_{i+1} - x_i$ is the step-size, $F(t_i, y_i), \bar{F}(t_i, y_i)$ are the functions associated with the particular RK method. The global and local errors for Runge-Kutta method of order $r$ are given below.

**Global error:**

$$\Delta_i^\alpha = y_i^\alpha - Y_i^\alpha,$$

$$\Delta_i^\alpha = \bar{y}_i^\alpha - \bar{Y}_i^\alpha,$$

**Local error:**

$$\xi_{i+1}^\alpha = [Y_i^\alpha + h_i F^\alpha (t_i, Y_i)] - Y_{i+1},$$

$$\xi_{i+1}^\alpha = [\bar{Y}_i^\alpha + h_i \bar{F}^\alpha (t_i, Y_i)] - \bar{Y}_{i+1}.$$ 

Here $Y_i$'s denote the exact values.

Gauss-Legendre (GL) Quadrature:

The Gauss-Legendre quadrature formula on $[-1,1]$ is

$$\int_{-1}^{1} f^\alpha (t, y) dt = \sum_{i=1}^{m} W_i f^\alpha (\tilde{t}_i, y_i) + E^\alpha (f),$$

$$\int_{-1}^{1} \bar{f}^\alpha (t, y) dt = \sum_{i=1}^{m} W_i \bar{f}^\alpha (\tilde{t}_i, y_i) + \bar{E}^\alpha (\tilde{f}), \quad 0 \leq \alpha \leq 1.$$ 

The weights $W_i, i = 1, \ldots, m$, are real numbers with the property $\sum_{i=1}^{m} W_i = 2$. This follows (5.2.2) when applied to $f^\alpha (t, y) = \bar{f}^\alpha (t, y) = 1$.

By using the transformation $t_i = \frac{u + v + (v - u) \tilde{t}_i}{2}$, where $\tilde{t}_i$'s are the nodes of $[-1,1]$ and $t_i$'s are the nodes of $[u, v]$, we transform the finite interval $[u, v]$ into $[-1,1]$. 

On an arbitrary interval \([u, v]\), Gauss Legendre quadrature is
\[
\int_u^v f^\alpha(t, y)\,dt \approx \frac{v - u}{2} \sum_{i=1}^{m} W_i f^\alpha(t_i, y_i) + E^\alpha(f) \\
= h \sum_{i=1}^{m} A_i f^\alpha(t_i, y_i) + E^\alpha(f),
\]
(5.2.3)
\[
\int_u^v \bar{f}^\alpha(t, y)\,dt \approx \frac{v - u}{2} \sum_{i=1}^{m} A_i \bar{f}^\alpha(t_i, y_i) + E^\alpha(\bar{f}) \\
= h \sum_{i=1}^{m} A_i \bar{f}^\alpha(t_i, y_i) + E^\alpha(\bar{f}), \quad 0 \leq \alpha \leq 1,
\]
where \(A_i = (m + 1)W_i/2\), and \(h\) denotes the average length of the subintervals into which \([u, v]\) is subdivided by the nodes \(t_i\).

It can be shown that the approximation error may be expressed as follows:
\[
E^\alpha(f) = \frac{f^{2m}(\eta)}{(2m)!} \int_u^v q_m^2(t)\,dt, \quad \eta \in (u, v),
\]
(5.2.4)
\[
E^\alpha(\bar{f}) = \frac{\bar{f}^{2m}(\bar{\eta})}{(2m)!} \int_u^v \bar{q}_m^2(t)\,dt, \quad \bar{\eta} \in (u, v), \quad 0 \leq \alpha \leq 1,
\]
where \(q_m(t) = \prod_{i=1}^{m} (t - t_i)\) with \(t_i, i = 0, \ldots, m\) as the roots of Legendre polynomials.

Assume that \(t_i = u + \sigma_i h\) where \(\sigma_i\) is an appropriate constant associated with each \(t_i\). Also assume that \(t = u + sh\) where \(s\) is a continuous variable and
\[
q_m(s) = h^m \prod_{i=1}^{m} (s - \sigma_i)
\]
(5.2.5)
and
\[
E^\alpha(f) = \frac{f^{2m}(\eta)}{(2m)!} \int_0^{m+1} h^2 q_m^2(s)\,hds, \quad \eta \in (u, v),
\]
(5.2.6)
\[
E^\alpha(\bar{f}) = \frac{\bar{f}^{2m}(\bar{\eta})}{(2m)!} \int_0^{m+1} h^2 \bar{q}_m^2(s)\,hds, \quad \bar{\eta} \in (u, v), \quad 0 \leq \alpha \leq 1.
\]

Therefore the error in Gauss-Legendre quadrature is, \(\Delta(t) = \bar{\Delta}(t) = O(h^{2m+1})\).
Theorem 5.2.1. For arbitrary fixed $\alpha$: $0 \leq \alpha \leq 1$, the Runge-Kutta approximations of (3.2.2) converge to the exact solutions $\tilde{Y}^\alpha(t), \bar{Y}^\alpha(t)$ uniformly in $t$ for $\tilde{Y}, \bar{Y} \in C^4[a,b]$.

5.3 The RKGL Method for the Fuzzy Initial Value Problem

Consider the fuzzy differential equations of the form

$$
\begin{align*}
\tilde{y}'(t) & = \tilde{f}(t, y(t)), \quad t \in [a, b], \\
\tilde{y}(a) & = \tilde{y}_0,
\end{align*}
$$

where $\tilde{f} \in C[\mathbb{R}^+ \times E, E]$. First of all, we divide the interval $[a, b]$ into $N$ subintervals. We replace each subinterval by a set of $p+1$ discrete grid points. We choose the grid points such that they are consistent with the positions of the roots of the $m$th degree Legendre polynomial on the particular subinterval.

Let us start with the subinterval $[a, b]$ on which discrete nodes $a = t_0, t_1, \ldots, t_m$ are defined. At the nodes $t_0, t_1, \ldots, t_m$, we use the RK method to obtain the solution and at the node $t_p = t_{m+1}$ we use Gauss-Legendre method to obtain the solution

$$
\begin{align*}
\tilde{y}^\alpha_p & = \tilde{y}^\alpha_{m+1} = \tilde{y}^\alpha_0 + h \sum_{i=1}^{m} A_i \tilde{f}^\alpha(t_i, y_i), \\
\tilde{y}^\alpha_p & = \bar{y}^\alpha_{m+1} = \bar{y}^\alpha_0 + h \sum_{i=1}^{m} A_i \bar{f}^\alpha(t_i, y_i),
\end{align*}
$$

where the sum is a quadrature formula, and $A_i$'s are the appropriate weights.

Similarly, for the next subinterval $[t_p, t_{2p}]$, we use the RK method to find the values $y_{m+2}, y_{m+3}, \ldots, y_{2m+1}$ and

$$
\begin{align*}
y^\alpha_{2p} & = \tilde{y}^\alpha_p + h \sum_{i=m+2}^{2m+1} A_i \tilde{f}^\alpha(t_i, y_i), \\
\bar{y}^\alpha_{2p} & = \bar{y}^\alpha_p + h \sum_{i=m+2}^{2m+1} A_i \bar{f}^\alpha(t_i, y_i).
\end{align*}
$$

We follow the same procedure for all subintervals.
Algorithm:

Step 1: Divide the interval $[a, b]$ into $N$ subintervals such that

$$[a, b] = \bigcup_{i=1}^{N} [t_{(i-1)p}, t_{ip}]$$  \hspace{1cm} (5.3.4)

Step 2: Let $i = 1$. The corresponding interval is $[t_{(i-1)p}, t_{ip}]$.

Step 3: We choose the grid points such that they are consistent with the positions of the roots of the $m$th degree Legendre polynomial on the particular subinterval.

Step 4: Use the RK method to find the solutions $y_{(i-1)p+1}, y_{(i-1)p+2}, \ldots, y_{(i-1)p+m}$.

Step 5: Use the following Gauss-Legendre method to find the solution at $y_{ip}$

$$y_{ip}^{\alpha} = y_{(i-1)p}^{\alpha} + h \sum_{j=1}^{m} A_{(i-1)p+j} A_{(i-1)p+j} f^\alpha(t_{(i-1)p+j}, y_{(i-1)p+j}),$$

$$\overline{y}_{ip}^{\alpha} = \overline{y}_{(i-1)p}^{\alpha} + h \sum_{j=1}^{m} A_{(i-1)p+j} A_{(i-1)p+j} f^\alpha(t_{(i-1)p+j}, y_{(i-1)p+j}).$$ \hspace{1cm} (5.3.5)

Step 6: $i = i + 1$.

Step 7: If $i \leq N$, go to step 3.

Step 8: Proceed until an approximate solution at $b$ is obtained.

The RKrGLm method is the fusion of Runge-Kutta (order $r$) method and Gauss-Legendre (order $m$) method. In this method, there is no need to evaluate $f(t, y)$ at GL nodes. It result in the evaluation of $f(t, y)$ at the numerous stages of the RK method. This is the most significant contribution to the RK methods. Also the RKrGLm method is used to reduce the computational effort and global error.

### 5.3.1 Error Analysis

A tilde is placed over a symbol to denote a fuzzy set like $\tilde{y}, \tilde{Y}, \ldots$ and a tilde is placed over a boldface symbol to denote a fuzzy $2 \times 2$ matrix like $\tilde{F}, \tilde{X}, \ldots$.
Theorem 5.3.1. For RKGL, we have
\[
\tilde{\Delta}_{np} = \sum_{j=1}^{n} (\tilde{e}_{jp} + h \tilde{B}_{(j-1)p+1,jp-1} + h \tilde{C}_{jp} \tilde{\Delta}_{(j-1)p}),
\]
where
\[
\tilde{\psi}_{(j-1)p+i} = \tilde{I} + (1 - \delta_{i,p-1}) \sum_{t=i}^{p-2} \left( \prod_{k=(j-1)p+i}^{(j-1)p+t} \tilde{\beta}_k \right), \quad i = 1, 2, \ldots, p - 1,
\]
where \(\delta_{i,p-1}\) is the Kronecker delta and the underarrow means that the product is right-to-left with increasing \(k\). This order of multiplication is important because the \(\tilde{\beta}\)'s are matrices and
\[
\tilde{\gamma}_{(j-1)p+i} = A_{(j-1)p+i} \tilde{\psi}_{(j-1)p+i}, \quad i = 1, 2, \ldots, p - 1,
\]
\[
\tilde{B}_{(j-1)p+1,jp-1} = \sum_{i=1}^{p-1} \tilde{\gamma}_{(j-1)p+i} \tilde{e}_i,
\]
\[
\tilde{C}_{jp} = \sum_{t=1}^{p-1} A_{(j-1)p+t} \tilde{\psi}_{(j-1)p+t}, \quad \tilde{\gamma}_{(j-1)p+t} \left( \prod_{k=(j-1)p+t}^{(j-1)p+t-1} \tilde{\beta}_k \right).
\]
Proof. Without loss of generality, we assume that \(\tilde{\Delta}_0 = 0\).

Consider \(n = 1\). Then, for the first subinterval \([t_0, t_1]\), we have
\[
\tilde{\Delta}_1 = \tilde{y}_1 - \tilde{Y}_1
\]
\[
= \tilde{Y}_0 + \tilde{\Delta}_0 + h_0 \tilde{f}(t_0, \tilde{Y}_0 + \tilde{\Delta}_0) - \tilde{Y}_1
\]
\[
= [\tilde{Y}_0 + h_0 \tilde{f}(t_0, \tilde{Y}_0) - \tilde{Y}_1] + [\tilde{I} + h_0 \tilde{f}_Y(t_0, \xi_0)] \tilde{\Delta}_0
\]
\[
= \tilde{e}_1 + \tilde{\beta}_0 \tilde{\Delta}_0,
\]
where \(\tilde{\beta}_0 = [\tilde{I} + h_0 \tilde{f}_Y(t_0, \xi_0)]\) and \(\xi_0\) is an appropriate set of constants such that \(\tilde{f}_Y(t_0, \xi_0) \tilde{\Delta}_0\) is the residual term in the first-order Taylor expansion of \(\tilde{f}(t_0, \tilde{Y}_0 + \tilde{\Delta}_0)\) and \(\tilde{f}_Y\) is the Jacobian matrix
\[
\tilde{f}_Y = \begin{bmatrix}
\frac{\partial f}{\partial \tilde{Y}} & \frac{\partial f}{\partial \tilde{Y}} \\
\frac{\partial \tilde{f}}{\partial \tilde{Y}} & \frac{\partial \tilde{f}}{\partial \tilde{Y}}
\end{bmatrix},
\]
and $\tilde{I}$ is the identity matrix.

Next

$$\tilde{\Delta}_2 = \tilde{y}_2 - \tilde{Y}_2$$
$$= \tilde{Y}_1 + \tilde{\Delta}_1 + h_1 f(t_1, \tilde{Y}_1 + \tilde{\Delta}_1) - \tilde{Y}_2$$
$$= [\tilde{Y}_1 + h_1 f(t_1, \tilde{Y}_1) - \tilde{Y}_2] + [\tilde{I} + h_1 f_Y(t_1, \tilde{\xi}_1)]\tilde{\Delta}_1$$
$$= \tilde{\varepsilon}_2 + \tilde{\beta}_1 \tilde{\Delta}_1$$
$$= \tilde{\varepsilon}_2 + \tilde{\beta}_1 \tilde{\varepsilon}_1 + \tilde{\beta}_1 \tilde{\beta}_0 \tilde{\Delta}_0. $$

Similarly we can show that

$$\tilde{\Delta}_3 = \tilde{\varepsilon}_3 + \tilde{\beta}_2 \tilde{\varepsilon}_2 + \tilde{\beta}_2 \tilde{\beta}_1 \tilde{\varepsilon}_1 + \tilde{\beta}_2 \tilde{\beta}_1 \tilde{\beta}_0 \tilde{\Delta}_0,$$

$$\tilde{\Delta}_4 = \tilde{\varepsilon}_4 + \tilde{\beta}_3 \tilde{\varepsilon}_3 + \tilde{\beta}_3 \tilde{\beta}_2 \tilde{\varepsilon}_2 + \tilde{\beta}_3 \tilde{\beta}_2 \tilde{\beta}_1 \tilde{\varepsilon}_1 + \tilde{\beta}_3 \tilde{\beta}_2 \tilde{\beta}_1 \tilde{\beta}_0 \tilde{\Delta}_0,$$

and

$$\tilde{\Delta}_m = \tilde{\varepsilon}_m + \tilde{\beta}_{m-1} \tilde{\varepsilon}_{m-1} + \cdots + \tilde{\beta}_{m-1} \tilde{\varepsilon}_{m-2} \cdots \tilde{\beta}_2 \tilde{\beta}_1 \tilde{\varepsilon}_1 + \tilde{\beta}_{m-1} \tilde{\varepsilon}_{m-2} \cdots \tilde{\beta}_2 \tilde{\beta}_1 \tilde{\beta}_0 \tilde{\Delta}_0,$$

$$\tilde{\beta}_n = [\tilde{I} + h_n f_Y(t_n, \tilde{\xi}_n)], \ n = 0, 1, \ldots, m, \text{ and } \tilde{\xi}_n \text{ is an appropriate set of constants.}$$

Finally we get

$$\tilde{\Delta}_p = \tilde{y}_p - \tilde{Y}_p$$

$$= \tilde{y}_0 + h \sum_{j=1}^{p-1} A_j f(t_j, \tilde{y}_j) - \tilde{Y}_p$$

$$= \tilde{Y}_0 + h \sum_{j=1}^{p-1} A_j f(t_j, \tilde{Y}_j + \tilde{\Delta}_j) - \tilde{Y}_p$$

$$= \tilde{Y}_0 + h \sum_{j=1}^{p-1} A_j f(t_j, \tilde{Y}_j) + h \sum_{j=1}^{p-1} A_j f_Y(t_j, \tilde{\xi}_j) \tilde{\Delta}_j - \tilde{Y}_p$$

$$= [\tilde{Y}_0 + h \sum_{j=1}^{p-1} A_j f(t_j, \tilde{Y}_j) - \tilde{Y}_p] + h \sum_{j=1}^{p-1} \tilde{y}_j \tilde{\varepsilon}_j + h \tilde{C}_p \tilde{\Delta}_0$$

$$= \tilde{\varepsilon}_p + h \tilde{B}_{1,p-1} + h \tilde{C}_p \tilde{\Delta}_0.$$
On the second subinterval, \([t_p, t_{2p}]\), we have

\[
\tilde{\Delta}_{p+1} = \tilde{\epsilon}_{p+1} + \tilde{\beta}_p \tilde{\Delta}_p,
\]

\[
\tilde{\Delta}_{p+2} = \tilde{\epsilon}_{p+2} + \tilde{\beta}_{p+1} \tilde{\epsilon}_{p+1} + \tilde{\beta}_{p+1} \tilde{\beta}_p \tilde{\Delta}_p,
\]

\[
\vdots
\]

\[
\tilde{\Delta}_{2p-1} = \tilde{\epsilon}_{2p-1} + \tilde{\beta}_{2p-2} \tilde{\epsilon}_{2p-2} + \cdots + \tilde{\beta}_{2p-2} \tilde{\beta}_{2p-3} \cdots \tilde{\beta}_{p+1} \tilde{\epsilon}_{p+1} + \tilde{\beta}_{2p-2} \tilde{\beta}_{2p-3} \cdots \tilde{\beta}_{p+1} \tilde{\beta}_p \tilde{\Delta}_p,
\]

so that

\[
\tilde{\Delta}_{2p} = \tilde{y}_{2p} - \tilde{Y}_{2p}
\]

\[
= \tilde{Y}_p + \tilde{\Delta}_p + h \sum_{j=p+1}^{2p-1} A_j \tilde{f}(t_j, \tilde{y}_j) - \tilde{Y}_{2p}
\]

\[
= \tilde{Y}_p + \tilde{\Delta}_p + h \sum_{j=p+1}^{2p-1} A_j \tilde{f}(t_j, \tilde{Y}_j) + \tilde{\Delta}_j - \tilde{Y}_{2p}
\]

\[
= \tilde{Y}_p + \tilde{\Delta}_p + h \sum_{j=p+1}^{2p-1} A_j \tilde{f}(t_j, \tilde{Y}_j) + h \sum_{j=p+1}^{2p-1} A_j \tilde{f}_y(t_j, \tilde{\xi}_j) \tilde{\Delta}_j - \tilde{Y}_{2p}
\]

\[
= \tilde{Y}_p + \tilde{\Delta}_p + h \sum_{j=p+1}^{2p-1} A_j \tilde{f}(t_j, \tilde{Y}_j) + h \tilde{B}_{p+1,2p-1} + h \tilde{C}_{2p} \tilde{\Delta}_p - \tilde{Y}_{2p}
\]

\[
= [\tilde{Y}_p + h \sum_{j=p+1}^{2p-1} A_j \tilde{f}(t_j, \tilde{Y}_j) - \tilde{Y}_{2p}] + h \tilde{B}_{p+1,2p-1} + h \tilde{C}_{2p} \tilde{\Delta}_p + \tilde{\Delta}_p
\]

\[
= \tilde{\epsilon}_{2p} + h \tilde{B}_{p+1,2p-1} + h \tilde{C}_{2p} \tilde{\Delta}_p + \tilde{\epsilon}_p + h \tilde{B}_{1,p-1} + h \tilde{C}_p \tilde{\Delta}_0
\]

\[
= (\tilde{\epsilon}_{2p} + \tilde{\epsilon}_p) + (h \tilde{B}_{p+1,2p-1} + h \tilde{B}_{1,p-1}) + (h \tilde{C}_{2p} \tilde{\Delta}_p + h \tilde{C}_p \tilde{\Delta}_0 \tilde{C}_{ip} \tilde{\Delta}_{(i-1)p})
\]

\[
= \sum_{i=1}^{2} (\tilde{\epsilon}_{ip} + h \tilde{B}_{(i-1)p+1,ip-1} + h \tilde{C}_{ip} \tilde{\Delta}_{(i-1)p}).
\]

Now assume that for the subinterval \([t_{(N-1)p}, t_{Np}]\),

\[
\tilde{\Delta}_{Np} = \sum_{i=1}^{N} (\tilde{\epsilon}_{ip} + h \tilde{B}_{(i-1)p+1,ip-1} + h \tilde{C}_{ip} \tilde{\Delta}_{(i-1)p}).
\]
Then, for the subinterval \([t_{Np}, t_{(N+1)p}]\),

\[
\tilde{\Delta}_{(N+1)p} = \tilde{y}_{(N+1)p} - \tilde{Y}_{(N+1)p} = \tilde{Y}_{Np} + \tilde{\Delta}_{Np} + h \sum_{j=Np+1}^{(N+1)p-1} A_j \tilde{f}(t_j, \tilde{y}_j + \tilde{\Delta}_j) - \tilde{Y}_{(N+1)p} = [\tilde{Y}_{Np} + h \sum_{j=Np+1}^{(N+1)p-1} A_j \tilde{f}(t_j, \tilde{y}_j) - \tilde{Y}_{(N+1)p}] + h\tilde{B}_{Np+1,(N+1)p-1} + h\tilde{C}_{(N+1)p}\tilde{\Delta}_{Np} + \tilde{\Delta}_{Np} + h\tilde{C}_{(N+1)p}\tilde{\Delta}_{Np} + \tilde{\Delta}_{Np}
\]

\[
= \tilde{\epsilon}_{(N+1)p} + h\tilde{B}_{Np+1,(N+1)p-1} + h\tilde{C}_{(N+1)p}\tilde{\Delta}_{Np} + \sum_{i=1}^{N} (\tilde{\epsilon}_{ip} + h\tilde{B}_{(i-1)p+1,ip-1} + h\tilde{C}_{ip}\tilde{\Delta}_{(i-1)p}) = \sum_{i=1}^{N+1} (\tilde{\epsilon}_{ip} + h\tilde{B}_{(i-1)p+1,ip-1} + h\tilde{C}_{ip}\tilde{\Delta}_{(i-1)p}).
\]

Thus

\[
\tilde{\Delta}_{np} = \sum_{i=1}^{n} (\tilde{\epsilon}_{ip} + h\tilde{B}_{(i-1)p+1,ip-1} + h\tilde{C}_{ip}\tilde{\Delta}_{(i-1)p}). \quad (5.3.6)
\]

In (5.3.6), \(\sum_{i=1}^{n} \tilde{\epsilon}_{ip}\) is the sum of the local errors at the GL nodes and \(\sum_{i=1}^{n} h\tilde{B}_{(i-1)p+1,ip-1}\) is the sum of local errors at the RK nodes.

Therefore

\[
\sum_{i=1}^{n} \tilde{\epsilon}_{ip} \propto \sum_{i=1}^{n} h^{2m+1} = nh^{2m+1}
\]
\[
\sum_{i=1}^{n} h\tilde{B}_{(i-1)p+1,ip-1} \propto \sum_{i=1}^{n} h^{r+2} = nh^{r+2}
\]
\[
\sum_{i=1}^{n} h\tilde{C}_{ip}\tilde{\Delta}_{(i-1)p} \propto \sum_{i=1}^{n} h^{r+2} = nh^{r+2}.
\]
And

\[
\tilde{\Delta}_{np} = Q_1 nh^{2m + 1} + Q_2 nh^{r+2}
\]

\[
= \left( \frac{Q_1}{p+1} \right) h^{2m(p+1)} + \left( \frac{Q_2}{p+10} \right) h^{r+1(p+1)} n h
\]

\[
= \left( \frac{Q_1(b-a)}{p+1} \right) h^{2m} + \left( \frac{Q_2(b-a)}{p+1} \right) h^{r+1}
\]

\[
= O(h^{-\min(2m, r+1)}),
\]

where \( Q_1 \) and \( Q_2 \) are vectors of appropriate coefficients. Hence, if we choose \( r \) and \( m \) such that \( 2m \geq r + 1 \), then \( \tilde{\Delta}_{np} = O(h^{r+1}) \). This means that the global error of RKGL is better than that of the underlying RK method. \( \square \)

### 5.3.2 Consistency, Convergence and Stability

**Theorem 5.3.2.** The RKGL method is consistent, convergent and stable.

**Proof.** RK method is consistent, convergent and strongly stable, so that RKGL method is also consistent, convergent and strongly stable at every RK node. The global error at every GL node is \( O(h^{-\min(2m, r+1)}) \) and it tends to zero as \( h \) tends to zero. Therefore RKGL is convergent at the GL nodes. From (5.3.5),

\[
\begin{align*}
\hat{y}^\alpha_{ip} & = y^\alpha_{(i-1)p} + h \sum_{j=1}^{m} A(i-1)p+j \bar{f}^\alpha(t(i-1)p+j, y(i-1)p+j), \\
\bar{y}^\alpha_{ip} & = \bar{y}^\alpha_{(i-1)p} + h \sum_{j=1}^{m} A(i-1)p+j \bar{f}^\alpha(t(i-1)p+j, y(i-1)p+j).
\end{align*}
\]

(5.3.7)

We use the RK method to find the solutions at \( t(i-1)p+j \) for \( j = 1, 2, \ldots, m \). We have

\[
\begin{align*}
y^\alpha_{(i-1)p+1} & = y^\alpha_{(i-1)p} + h F(t(i-1)p, y(i-1)p), \\
\bar{y}^\alpha_{(i-1)p+1} & = \bar{y}^\alpha_{(i-1)p} + h \bar{F}(t(i-1)p, y(i-1)p), \\
y^\alpha_{(i-1)p+2} & = y^\alpha_{(i-1)p+1} + h F(t(i-1)p+1, y(i-1)p+1),
\end{align*}
\]
\[
\bar{y}_{(i-1)p+2} = \bar{y}_{(i-1)p+1} + hF(t_{(i-1)p+1}, y_{(i-1)p+1}),
\]
\[
\vdots
\]
\[
\bar{y}_{(i-1)p+m} = \bar{y}_{(i-1)p+m-1} + hF(t_{(i-1)p+m-1}, y_{(i-1)p+m-1}),
\]
\[
\bar{y}_{(i-1)p+m} = \bar{y}_{(i-1)p+m-1} + h\bar{F}(t_{(i-1)p+m-1}, y_{(i-1)p+m-1}).
\]

This gives
\[
y_{(i-1)p}^\alpha = y_{(i-1)p+m}^\alpha - \sum_{j=0}^{m-1} hF(t_{(i-1)p+j}, y_{(i-1)p+j}),
\]
\[
\bar{y}_{(i-1)p}^\alpha = \bar{y}_{(i-1)p+m}^\alpha - \sum_{j=0}^{m-1} h\bar{F}(t_{(i-1)p+j}, y_{(i-1)p+j}).
\]

We obtain
\[
y_{(i-1)p+m+1}^\alpha = y_{(i-1)p+m}^\alpha + h(M - L),
\]
\[
\bar{y}_{(i-1)p+m+1}^\alpha = \bar{y}_{(i-1)p+m}^\alpha + h(M - \bar{L}),
\]
(5.3.8)

where
\[
L = \sum_{j=0}^{m-1} hF(t_{(i-1)p+j}, y_{(i-1)p+j}),
\]
\[
\bar{L} = \sum_{j=0}^{m-1} h\bar{F}(t_{(i-1)p+j}, y_{(i-1)p+j}),
\]
\[
M = \sum_{j=1}^{m} A_{(i-1)p+j} f^\alpha(t_{(i-1)p+j}, y_{(i-1)p+j}),
\]
\[
\bar{M} = \sum_{j=1}^{m} A_{(i-1)p+j} \bar{f}^\alpha(t_{(i-1)p+j}, y_{(i-1)p+j}).
\]

(5.3.8) becomes
\[
Y_{(i-1)p+m+1} = Y_{(i-1)p+m} + \Delta_{(i-1)p+m} + h(M - L),
\]
\[
\bar{Y}_{(i-1)p+m+1} = \bar{Y}_{(i-1)p+m} + \bar{\Delta}_{(i-1)p+m} + h(M - \bar{L}),
\]
where $\Delta$’s are global errors.

\[
\frac{\Delta^{\alpha}_{(i-1)p+m+1} - \Delta^{\alpha}_{(i-1)p+m}}{h} = \frac{Y^{\alpha}_{(i-1)p+m} - Y^{\alpha}_{(i-1)p+m+1}}{h} + \frac{h(M - L)}{h},
\]

\[
\frac{\Delta^{\alpha}_{(i-1)p+m+1} - \Delta^{\alpha}_{(i-1)p+m}}{h} = \frac{\bar{Y}^{\alpha}_{(i-1)p+m} - \bar{Y}^{\alpha}_{(i-1)p+m+1}}{h} + \frac{h(M - \bar{L})}{h}.
\]

Since $\Delta_i = O(h^{r+1}) + O(h^{2})$ at any node $t_i$,

\[
\frac{\Delta^{\alpha}_{(i-1)p+m+1} - \Delta^{\alpha}_{(i-1)p+m}}{h} \to 0, \quad \frac{\Delta^{\alpha}_{(i-1)p+m+1} - \Delta^{\alpha}_{(i-1)p+m}}{h} \to 0 \text{ as } h \to 0.
\]

Therefore as $h \to 0$,

\[
-Y'(t_{(i-1)p+m}) + (M - L) = 0,
\]

\[
-\bar{Y}'(t_{(i-1)p+m}) + (M - \bar{L}) = 0.
\]

Hence RKGL method is consistent. For (5.3.7), there exists only one characteristic polynomial $p(\lambda) = \lambda - 1$ and it is clear that it satisfies the root condition. Then, by Theorem 1.5.1, the RKGL method is stable at GL nodes. \hfill \Box

### 5.3.3 Numerical Examples

**Example 5.3.1** Consider the fuzzy initial value problem

\[
\begin{align*}
\tilde{y}'(t) &= \tilde{y}(t), \quad t \in [0, 1], \\
\tilde{y}(0) &= [0.75, 1, 1.125].
\end{align*}
\] (5.3.9)

The exact solution of (5.3.9) is

\[
\tilde{y}(t) = (0.75e^t, e^t, 1.125e^t).
\] (5.3.10)

We have used RK4 and RK4GL3 method to solve the above fuzzy initial value problem. Here we choose the $m$ and $r$ values such that $2m \geq r + 1$ so that the global order of RK4GL3 is increased by one which is same as the local order of RK4. The comparison of global errors of RK4GL3 and RK4 is shown in the following Figure 5.1.
Example 5.3.2 Consider the fuzzy initial value problem

\[
\tilde{y}'(t) = -\tilde{y} + t + 1, \quad t \in [0, 1],
\]
\[
\tilde{y}(0) = [0.96, 1, 1.01].
\] (5.3.11)

The exact solution of (5.3.11) is

\[
\tilde{y}(t) = (t - 0.025e^t + 0.985e^{-t}, t + e^{-t}, t + 0.025e^t + 0.985e^{-t}).
\] (5.3.12)
The comparison of global errors of RK4GL3 and RK4 is shown in the following Figure 5.2.

Figure 5.2: -o- RK4, -*- RK4GL3.
5.4 The RKGL Method for the Hybrid Fuzzy Differential System

Consider the hybrid fuzzy differential system of the form

\[
\begin{align*}
\ddot{y}(t) &= \ddot{f}(t, y(t), \lambda_k(y_k)), \quad t \in [t_k, t_{k+1}], \\
\ddot{y}(t_k) &= \ddot{y}_k,
\end{align*}
\]  \tag{5.4.1}

where \(0 \leq t_0 < t_1 < \cdots < t_k < \cdots, t_k \to \infty, \ddot{f} \in C(\mathbb{R}^+ \times E \times E, E), \lambda_k \in C(E, E)\).

For the hybrid fuzzy differential equation (5.4.1), we develop the RKGL method via an application of the RKGL method for fuzzy differential equations when \(\ddot{f}\) and \(\lambda_k\) in (5.4.1) are obtained via the Zadeh extension principle from \(f \in C(\mathbb{R}^+ \times \mathbb{R} \times \mathbb{R}, \mathbb{R})\) and \(\lambda_k \in C(\mathbb{R}, \mathbb{R})\).

For a fixed \(\alpha\)-level set and \(K \in \mathbb{Z}^+\), we divide each interval \([t_k, t_{k+1}]\) into \(N_k\) subintervals. Next we replace each subinterval by \(p + 1\) discrete grid points (including the endpoints). Here we choose the grid points \(\{t_{k,(i-1)p+1}, t_{k,(i-1)p+2}, \ldots, t_{k,(i-1)p+m}\}\) such that they are consistent with the positions of the roots of the \(m\)th degree Legendre polynomial on the particular subinterval \([t_{k,(i-1)p}, t_{k,i}p]\).

Algorithm:

**Step 1:** Fix \(K \in \mathbb{Z}^+\). Divide each interval \([t_k, t_{k+1}]\) into \(N_k\) subintervals such that

\[
[t_k, t_{k+1}] = \bigcup_{i=1}^{N}[t_{k,(i-1)p}, t_{k,i}p]
\]

**Step 2:** Let \(i = 1\). The corresponding interval is \([t_{k,(i-1)p}, t_{k,i}p]\).

**Step 3:** We choose the grid points such that they are consistent with the positions of the roots of the \(m\)th degree Legendre polynomial on the particular subinterval.

**Step 4:** Use the RK method to find the solutions \(y_{k,(i-1)p+1}, y_{k,(i-1)p+2}, \ldots, y_{k,(i-1)p+m}\).
Step 5: Use the following Gauss-Legendre method to find the solution at $y_{k,i_p}$:
\[
\begin{align*}
\bar{y}_{k,i_p}^\alpha &= \bar{y}_{k,(i-1)p}^\alpha + h \sum_{j=1}^{m} A_{k,(i-1)p+j} f^\alpha (t_{k,(i-1)p+j}, y_{k,(i-1)p+j}), \\
\bar{y}_{k,i_p}^\alpha &= \bar{y}_{k,(i-1)p}^\alpha + h \sum_{j=1}^{m} A_{k,(i-1)p+j} \bar{f}^\alpha (t_{k,(i-1)p+j}, y_{k,(i-1)p+j}).
\end{align*}
\] (5.4.2)

Step 6: $i = i + 1$.

Step 7: If $i \leq N_k$, go to step 3.

Step 8: Proceed until an approximate solution at $t_{k+1}$ is obtained.

The RKrGLm method is the fusion of Runge-Kutta(order r) method and Gauss-Legendre(order m) method. In this method, there is no need to evaluate $f(t, y)$ at GL nodes. It results in the evaluation of $f(t, y)$ at the numerous stages of the RK method. This is the most significant contribution to the RK methods. The global error of RKGL is one order better than that of the underlying RK method.

**Theorem 5.4.1.** For arbitrary fixed $\alpha$: $0 \leq \alpha \leq 1$ and $k \in \mathbb{Z}^+$, the RKGL method of (5.4.2) converges to the exact solutions $\bar{Y}^\alpha(t_{k+1}), \bar{Y}^\alpha(t_{k+1})$ that is, for all $i$, $j$
\[
\lim_{h_0, \ldots, j_{k-1} \to 0} \bar{y}_{k,N_k}^\alpha = \bar{Y}^\alpha(t_{k+1}), \quad \lim_{h_0, \ldots, j_{k-1} \to 0} \bar{y}_{k,N_k}^\alpha = \bar{Y}^\alpha(t_{k+1}).
\]

**Proof.** See [82].

5.4.1 Numerical Examples

**Example 5.4.1** Consider the hybrid fuzzy initial value problem
\[
\begin{align*}
\vec{y}'(t) &= \vec{y}(t) + m(t)\lambda_k(\vec{y}(t)), & t \in [t_k, t_{k+1}], & t_k = k, & k = 0, 1, 2, \ldots, \\
\vec{y}(0) &= [0.75, 1, 1.125],
\end{align*}
\] (5.4.3)

where
\[
m(t) = \begin{cases} 
2(t \mod 1), & \text{if } t \mod 1 \leq 0.5, \\
2(1 - t \mod 1), & \text{if } t \mod 1 > 0.5,
\end{cases}
\] (5.4.4)
and

\[ \lambda_k(\mu) = \begin{cases} 
\hat{0}, & \text{if } k = 0, \\
\mu, & k \in \{1, 2, \ldots\}. 
\end{cases} \quad (5.4.5) \]

The hybrid fuzzy initial value problem (5.4.3) is equivalent to the following system of fuzzy initial value problems:

\[
\begin{align*}
\tilde{y}_0(t) &= \tilde{y}_0(t), & t \in [0, 1], \\
\tilde{y}_0(0) &= [0.75, 1, 1.125], \\
\tilde{y}_k(t) &= \tilde{y}_k(t) + m(t)\tilde{y}_k(t_k), & t \in [t_k, t_{k+1}], \\
\tilde{y}_k(t_k) &= \tilde{y}_{k-1}(t_k), & i = 1, 2, \ldots.
\end{align*}
\]

In (5.4.3), \( \tilde{y}(t) + m(t)\lambda_k(\tilde{y}(t_k)) \) is a continuous function of \( t, \tilde{y} \) and \( \lambda_k(\tilde{y}(t_k)) \). Therefore, for each \( k=0,1,2,\ldots \), the hybrid fuzzy initial value problem

\[
\begin{align*}
\tilde{y}(t) &= \tilde{y}(t) + m(t)\lambda_k(y(t_k)), & t \in [t_k, t_{k+1}], & t_k = k, \\
\tilde{y}(t_k) &= \tilde{y}_{k},
\end{align*}
\]

has a unique solution on \([t_k, t_{k+1}].\)

For \([0,1]\), the exact solution of (5.4.3) is

\[ \tilde{y}(t) = [0.75e^t, e^t, 1.125e^t]. \]

For \([1, 1.5]\), the exact solution of (5.4.3) is

\[ \tilde{y}(t) = \tilde{y}(1)(3e^{t-1} - 2t). \]

For \([1.5, 2]\), the exact solution of (5.4.3) is

\[ \tilde{y}(t) = \tilde{y}(1)(2t - 2 + e^{t-1.5}(3\sqrt{e} - 4)). \]

The results of Example 5.4.1 on \([0,2]\) are shown in Figure 5.3.
Figure 5.3: *- RK4, - RK4GL3.
Example 5.4.2 Consider the hybrid fuzzy initial value problem
\[
\begin{aligned}
\begin{cases}
\tilde{y}'(t) = -\tilde{y}(t) + m(t)\lambda_k(y(t_k)), & t \in [t_k, t_{k+1}], \quad t_k = k, \quad k = 0, 1, 2, \ldots \\
\tilde{y}(0) = [0.75, 1, 1.125],
\end{cases}
\end{aligned}
\] (5.4.10)

where
\[
m(t) = |\sin(\pi t)|, \quad k = 0, 1, 2, \ldots, \quad (5.4.11)
\]
and
\[
\lambda_k(\mu) = \begin{cases}
\tilde{0}, & \text{if } k = 0, \\
\mu, & k \in 1, 2 \ldots .
\end{cases} \quad (5.4.12)
\]

The hybrid fuzzy initial value problem (5.4.10) is equivalent to the following system :
\[
\begin{aligned}
\begin{cases}
\tilde{y}_0'(t) = -\tilde{y}_0(t), \\
\tilde{y}_0'(t) = -\tilde{y}_0(t), \\
\tilde{y}_0'(t) = -\tilde{y}_0(t), \quad t \in [0, 1], \\
\tilde{y}_0(0) = [0.75, 1, 1.125], \\
\tilde{y}_k'(t) = -\tilde{y}_k(t) + m(t)y_k(t_k), \\
\tilde{y}_k'(t) = -\tilde{y}_k(t) + m(t)y_k(t_k), \quad t \in [t_k, t_{k+1}], \\
\tilde{y}_k(t_k) = \tilde{y}_{k-1}(t_k), \quad k = 1, 2, \ldots.
\end{cases}
\end{aligned}
\] (5.4.13)

For \([0, 1]\), the exact solution of (5.4.10) is
\[
\tilde{y}(t) = [-0.1875e^t + 0.9375e^{-t}, e^{-t}, 0.1875e^t + 0.9375e^{-t}]. \quad (5.4.14)
\]

For \([1, 2]\), the exact solution of (5.4.10) is
\[
\tilde{y}(t) = \begin{pmatrix}
-0.1875[e^t + \frac{1}{1+\pi^2}(\sin(\pi t) + \Pi \cos(\pi t)) + \pi e^t] \\
+0.9375[e^{-t} - \frac{1}{1+\pi^2}(\sin(\pi t) - \Pi \cos(\pi t)) - \pi e^{-t}] \\
0.1875[e^t + \frac{1}{1+\pi^2}(\sin(\pi t) + \Pi \cos(\pi t)) + \pi e^t] \\
+0.9375[e^{-t} - \frac{1}{1+\pi^2}(\sin(\pi t) - \Pi \cos(\pi t)) - \pi e^{-t}]
\end{pmatrix}
\]
The results of Example 5.4.2 on \([0,2]\) are shown in Figure 5.4.

Figure 5.4: *- RK4, - RK4GL3.