

Question: Using Feedforward Neural Network to solve the ODE below:

$$\begin{cases} \frac{dy}{dx} + \left(x + \frac{1+3x^2}{1+x+x^3}\right)y = x^3 + 2x + x^2 \frac{1+3x^2}{1+x+x^3}, \\ y(0) = 1. \end{cases}$$

Using SR1, DFP, BFGS algorithms to program and compare their performance.

(1) Using a 1-2-1 FNN, terminal condition is when $\varepsilon = 1.5$. different algorithms have performance below:

		SR1	BFGS	DFP
n	m	k	k	k
2	5	197	13	16
3	5	159	12	16

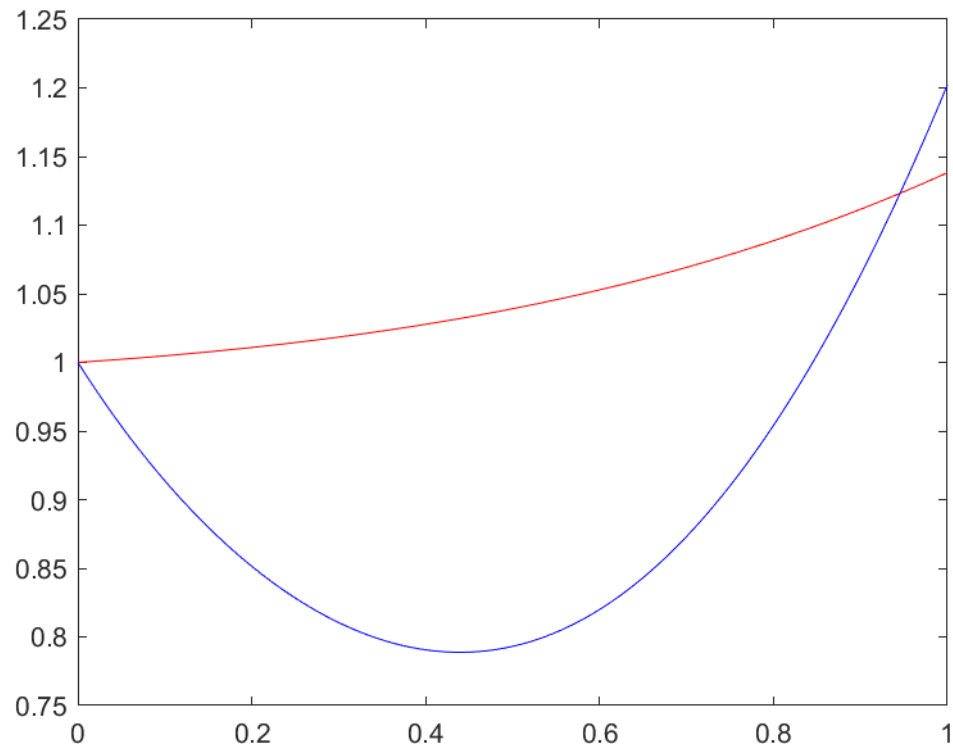
<p>p1 =</p> <p>1. 0519977381824998805660968241549</p> <p>1. 0309067803441490469519696131196</p> <p>0. 2033337468101811398914339616662</p> <p>0. 16231415075499506920923702010479</p> <p>1. 7340299708480378617138488021832</p> <p>1. 7245623340768081705855333933509</p> <p>k =</p>	<p>p1 =</p> <p>1. 0626812958385182506413710750134</p> <p>1. 0626812958385182506413710750134</p> <p>1. 0430761396714543190476296312649</p> <p>0. 13912733829903658213748206362324</p> <p>0. 13912733829903658213748206362324</p> <p>0. 095405037358874685905298683061474</p> <p>1. 7532256169340930779374862917815</p> <p>1. 7532256169340930779374862917815</p> <p>1. 7415830303665471214156565831185</p> <p>k =</p>
---	--

<p>p1 =</p> <ol style="list-style-type: none"> 1. 2490752400956025773505877478824 1. 2028778562790010301318194002249 0. 32299018143998163165826060445508 0. 27677737090672437845887268534028 2. 4841172047504958114234518550196 2. 485642291069597374827409246787 <p>k =</p>	<p>p1 =</p> <ol style="list-style-type: none"> 1. 2571287710439851779228543924437 1. 2571287710439851779228543924437 1. 2079343192371441441484832480615 0. 22906019729187628433505814446551 0. 22906019729187628433505814446551 0. 17873552848649815975570356218702 2. 5877112801329659005801660216117 2. 5877112801329659005801660216117 2. 5884302848811580226676272866235 <p>k =</p>
--	--

<pre> p1 = 1. 3683947354356152197911475829492 1. 3175045610139466854109772723502 0. 36985960364199162136434285894798 0. 31964463649674651253067413208129 2. 7273232225278680301557569721121 2. 7295275453537327080689935715928 k = 13 </pre>	<pre> p1 = 1. 3830580817066468301501629624689 1. 3830580817066468301501629624689 1. 3282092427082968880154435058915 0. 27496739613815455944646697169874 0. 27496739613815455944646697169874 0. 22011791184117567627369499766868 2. 8608954327516607236725168643937 2. 8608954327516607236725168643937 2. 8626161596944108478682207558152 k = 12 </pre>
---	---

(2) It is easy to see that BFGS is the best performing algorithm, and we compare it to the exact solution graph (red line is the numerical solution, blue line is the exact solution).:

n=2,m=5 :



n=3,m=5 :

