MATH 6205 - Numerical Methods for Financial Derivatives Fall 2019

Author: AkshayPatil Stud ID : 801034919 Email id : apatil17@uncc.edu

Purpose : The objective of this Python program is to compute the solution of a tridiagonal matrix using the finite differences method. WE use the heat equation to solve for explicit or implicit solution. Based on the given solution, we can discretize the linear system of difference equations at given point of time using the finite difference method. The lambda parameter is given by the user. prices of European calls and puts using

Numerical Methods: Heat equation is used to solve for explicit or implicit solution. Solving Tridiagonal system of equations. Finite Difference method to solve linear system of equations is used.

Output: Case 1

```
In [39]: runfile('/Users/AkshayPatil/Desktop/NMFD/hw8_PatilAkshay/Hw8_PatilAkshay_main.py', wdir='/
Users/AkshayPatil/Desktop/NMFD/hw8_PatilAkshay')
Reloaded modules: Hw8_PatilAkshay_EigenFunc
****************   eigenvalue-based stability analysis *********
1. Exp_Analytical and Imp_Analytical is calculated using the analytical formula
2. Exp_Func and Imp_Func using numpy.linalg.eigvals function

       dx = : 0.05,    dt =0.001,      N = 100

    Exp_Analytical  Exp_Func  Imp_Analytical  Imp_Func
0         0.999605  0.999605        1.000395  0.999605
1         0.998421  0.998421        1.001579  0.998424
2         0.996450  0.996450        1.003550  0.996462
3         0.993692  0.993692        1.006308  0.993731
4         0.990151  0.990151        1.009849  0.990247
5         0.985830  0.985830        1.014170  0.986028
6         0.980733  0.980733        1.019267  0.981098
7         0.974867  0.974867        1.025133  0.975483
8         0.968235  0.952705        1.031765  0.969213
9         0.960845 -0.599605        1.039155  0.962321
10        0.952705 -0.598421        1.047295  0.954840
11        0.943821 -0.596450        1.056179  0.946809
12        0.934204 -0.593692        1.065796  0.938266
13        0.923862  0.934204        1.076138  0.929249
14        0.912805  0.943821        1.087195  0.919798
15        0.901045  0.923862        1.098955  0.909956
16        0.888594  0.960845        1.111406  0.899761
17        0.875462 -0.590151        1.124538  0.889254
18        0.861664  0.912805        1.138336  0.878476
19        0.847214  0.901045        1.152786  0.867463
20        0.832124 -0.585830        1.167876  0.856255
21        0.816411 -0.580733        1.183589  0.844888
22        0.800089  0.888594        1.199911  0.833395
23        0.783175 -0.574867        1.216825  0.821811
24        0.765685 -0.543821        1.234315  0.810166
25        0.747638 -0.552705        1.252362  0.798491
26        0.729049 -0.560845        1.270951  0.786813
27        0.709939 -0.534204        1.290061  0.775157
28        0.690326  0.861664        1.309674  0.763549
29        0.670228  0.847214        1.329772  0.752009
..             ...       ...             ...       ...
69       -0.270228  0.001048        2.270228  0.440484
70       -0.290326 -0.047214        2.290326  0.436619
71       -0.309939  0.200000        2.309939  0.432912
72       -0.329049  0.225129        2.329049  0.429360
73       -0.347638  0.174871        2.347638  0.425960
74       -0.365685  0.025485        2.365685  0.422710
75       -0.383175  0.149768        2.383175  0.416651
76       -0.400089  0.398952        2.400089  0.419608
77       -0.416411  0.250232        2.416411  0.413837
78       -0.432124 -0.070990        2.432124  0.408628
79       -0.447214  0.374515        2.447214  0.406229
80       -0.461664 -0.163192        2.461664  0.411163
81       -0.475462  0.124713        2.475462  0.401833
82       -0.488594  0.050095        2.488594  0.399833
83       -0.501045  0.275287        2.501045  0.403965
84       -0.512805 -0.094500        2.512805  0.396218
```

Case 2

```
****************   eigenvalue-based stability analysis *********
1. Exp_Analytical and Imp_Analytical is calculated using the analytical formula
2. Exp_Func and Imp_Func using numpy.linalg.eigvals function

       dx = : 0.05,   dt =0.0015,      N = 100
```

	Exp_Analytical	Exp_Func	Imp_Analytical	Imp_Func
0	0.999408	-1.399408	1.000592	0.999408
1	0.997632	-1.397632	1.002368	0.997638
2	0.994674	-1.394674	1.005326	0.994703
3	0.990538	-1.390538	1.009462	0.990626
4	0.985226	-1.385226	1.014774	0.985441
5	0.978745	-1.378745	1.021255	0.979187
6	0.971100	-1.371100	1.028900	0.971912
7	0.962300	-1.362300	1.037700	0.963669
8	0.952352	-1.301306	1.047648	0.954519
9	0.941268	-1.315732	1.058732	0.944526
10	0.929057	-1.329057	1.070943	0.933756
11	0.915732	-1.341268	1.084268	0.922281
12	0.901306	-1.285792	1.098694	0.910171
13	0.885792	-1.269208	1.114208	0.897499
14	0.869208	-1.251568	1.130792	0.884336
15	0.851568	-1.232890	1.148432	0.870752
16	0.832890	-1.192497	1.167110	0.856818
17	0.813194	-1.170820	1.186806	0.842597
18	0.792497	-1.148186	1.207503	0.828155
19	0.770820	-1.124616	1.229180	0.813551
20	0.748186	-1.100133	1.251814	0.798841
21	0.724616	-1.352352	1.275384	0.784078
22	0.700133	-1.074762	1.299867	0.769310
23	0.674762	-1.213194	1.325238	0.754582
24	0.648528	-1.048528	1.351472	0.739934
25	0.621457	-1.021457	1.378543	0.725403
26	0.593574	0.997632	1.406426	0.711022
27	0.564909	0.999408	1.435091	0.696820
28	0.535488	0.994674	1.464512	0.682822
29	0.505342	0.990538	1.494658	0.669050
..	...	...	...	...
69	-0.905342	0.442992	2.905342	0.344194
70	-0.935488	0.410850	2.935488	0.340659
71	-0.964909	-0.275349	2.964909	0.337279
72	-0.993574	-0.237693	2.993574	0.334049
73	-1.021457	-0.200000	3.021457	0.330966
74	-1.048528	-0.162307	3.048528	0.328027
75	-1.074762	0.378104	3.074762	0.325228
76	-1.100133	-0.124651	3.100133	0.322567
77	-1.124616	-0.312930	3.124616	0.320039
78	-1.148186	0.134789	3.148186	0.317643
79	-1.170820	0.170820	3.170820	0.315376
80	-1.192497	0.098428	3.192497	0.313234
81	-1.213194	-0.676577	3.213194	0.311217
82	-1.232890	0.206486	3.232890	0.309321
83	-1.251568	0.061772	3.251568	0.307544
84	-1.269208	0.344789	3.269208	0.305884
85	-1.285792	-0.087070	3.285792	0.294169
86	-1.301306	0.564909	3.301306	0.302910
87	-1.315732	0.241749	3.315732	0.304341

Case 3

```
****************   eigenvalue-based stability analysis *********
1. Exp_Analytical and Imp_Analytical is calculated using the analytical formula
2. Exp_Func and Imp_Func using numpy.linalg.eigvals function

       dx = : 0.04,   dt =0.001,      N = 100

    Exp_Analytical  Exp_Func  Imp_Analytical  Imp_Func
0         0.999383 -1.499383        1.000617  0.999384
1         0.997533 -1.497533        1.002467  0.997539
2         0.994452 -1.494452        1.005548  0.994483
3         0.990143 -1.490143        1.009857  0.990240
4         0.984610 -1.484610        1.015390  0.984844
5         0.977859 -1.477859        1.022141  0.978339
6         0.969896 -1.469896        1.030104  0.970776
7         0.960729 -1.460729        1.039271  0.962213
8         0.950367 -1.363758        1.049633  0.952714
9         0.938821 -1.381034        1.061179  0.942348
10        0.926101 -1.397193        1.073899  0.919304
11        0.912221 -1.412221        1.087779  0.931186
12        0.897193 -1.426101        1.102807  0.906777
13        0.881034 -1.438821        1.118966  0.893682
14        0.863758 -1.345383        1.136242  0.880094
15        0.845383 -1.325928        1.154617  0.866088
16        0.825928 -1.283851        1.174072  0.851736
17        0.805410 -1.261271        1.194590  0.837107
18        0.783851 -1.237694        1.216149  0.822267
19        0.761271 -1.213142        1.238729  0.807279
20        0.737694 -1.187639        1.262306  0.792201
21        0.713142 -1.161211        1.286858  0.777086
22        0.687639 -1.450367        1.312361  0.761985
23        0.661211 -1.133883        1.338789  0.746944
24        0.633883 -1.305410        1.366117  0.732002
25        0.605684 -1.105684        1.394316  0.717197
26        0.576640 -1.016134        1.423360  0.702563
27        0.546780 -1.046780        1.453220  0.688127
28        0.516134 -0.984732        1.483866  0.673915
29        0.484732 -0.952604        1.515268  0.659949
..             ...       ...             ...       ...
69       -0.984732  0.419783        2.984732  0.335039
70       -1.016134  0.484732        3.016134  0.331550
71       -1.046780 -0.250000        3.046780  0.328215
72       -1.076640 -0.210737        3.076640  0.325030
73       -1.105684  0.098739        3.105684  0.321990
74       -1.133883 -0.289263        3.133883  0.319093
75       -1.161211  0.060862        3.161211  0.316334
76       -1.187639 -0.171512        3.187639  0.313712
77       -1.213142  0.386302        3.213142  0.308862
78       -1.237694  0.136271        3.237694  0.311222
79       -1.261271  0.173422        3.261271  0.306629
80       -1.283851  0.022679        3.283851  0.304521
81       -1.305410  0.352192        3.305410  0.302534
82       -1.325928 -0.328488        3.325928  0.300668
83       -1.345383  0.516134        3.345383  0.298919
84       -1.363758  0.210156        3.363758  0.297287
85       -1.381034 -0.132365        3.381034  0.294361
86       -1.397193 -0.015773        3.397193  0.295768
87       -1.412221 -0.367635        3.412221  0.285765
```

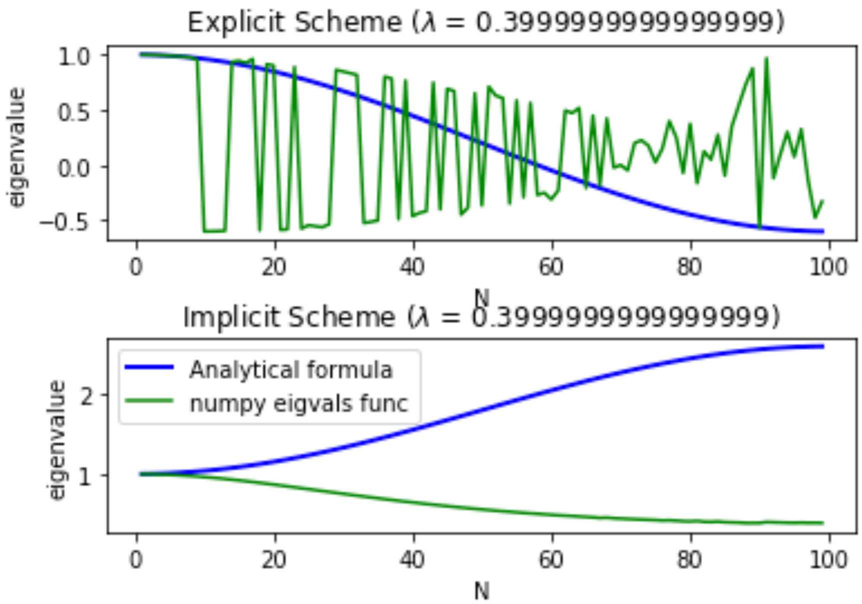
Case 4

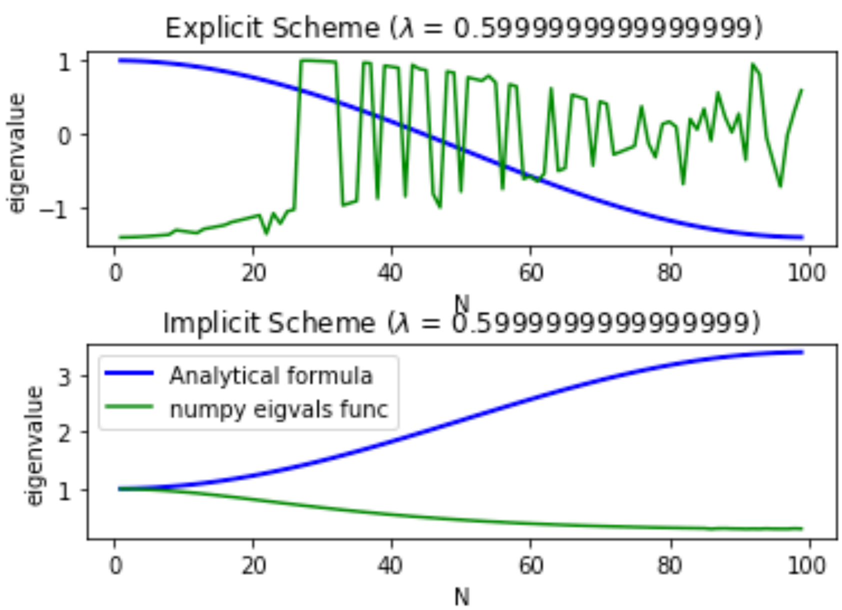
```
****************   eigenvalue-based stability analysis *********
1. Exp_Analytical and Imp_Analytical is calculated using the analytical formula
2. Exp_Func and Imp_Func using numpy.linalg.eigvals function

       dx = : 0.04,   dt =0.0015,      N = 100
```

	Exp_Analytical	Exp_Func	Imp_Analytical	Imp_Func
0	0.999075	-2.749075	1.000925	0.999076
1	0.996300	-2.746300	1.003700	0.996314
2	0.991679	-2.741679	1.008321	0.991747
3	0.985215	-2.735215	1.014785	0.985430
4	0.976916	-2.726916	1.023084	0.977437
5	0.966789	-2.716789	1.033211	0.967856
6	0.954844	-2.704844	1.045156	0.956795
7	0.941093	-2.691093	1.058907	0.944370
8	0.925551	-2.618331	1.074449	0.930709
9	0.908231	-2.639151	1.091769	0.915945
10	0.889151	-2.595790	1.110849	0.900213
11	0.868331	-2.658231	1.131669	0.883651
12	0.845790	-2.571551	1.154210	0.866393
13	0.821551	-2.545637	1.178449	0.848573
14	0.795637	-2.518075	1.204363	0.830315
15	0.768075	-2.319712	1.231925	0.811738
16	0.738891	-2.356541	1.261109	0.792953
17	0.708115	-2.488891	1.291885	0.774063
18	0.675776	-2.391907	1.324224	0.755159
19	0.641907	-2.281458	1.358093	0.736327
20	0.606541	-2.241816	1.393459	0.717638
21	0.569712	-2.200825	1.430288	0.699160
22	0.531458	-2.425776	1.468542	0.680948
23	0.491816	-2.158526	1.508184	0.663049
24	0.450825	-1.977097	1.549175	0.645505
25	0.408526	-1.928906	1.591474	0.628348
26	0.364960	-2.024201	1.635040	0.611606
27	0.320170	-1.879675	1.679830	0.595298
28	0.274201	-2.675551	1.725799	0.579442
29	0.227097	-2.070170	1.772903	0.564047
..	...	...	...	...
69	-1.977097	-0.408706	3.977097	0.251440
70	-2.024201	-0.351892	4.024201	0.248497
71	-2.070170	-0.295593	4.070170	0.245690
72	-2.114960	0.531458	4.114960	0.243016
73	-2.158526	0.491816	4.158526	0.240470
74	-2.200825	0.569712	4.200825	0.238048
75	-2.241816	0.738891	4.241816	0.235748
76	-2.281458	0.450825	4.281458	0.233565
77	-2.319712	0.178906	4.319712	0.231497
78	-2.356541	0.129675	4.356541	0.229540
79	-2.391907	-0.239866	4.391907	0.227692
80	-2.425776	0.227097	4.425776	0.225949
81	-2.458115	-0.523660	4.458115	0.224310
82	-2.488891	0.606541	4.488891	0.222772
83	-2.518075	0.079453	4.518075	0.221333
84	-2.545637	0.274201	4.545637	0.219991
85	-2.571551	0.408526	4.571551	0.218744
86	-2.595790	0.028288	4.595790	0.216528
87	-2.618331	-0.184766	4.618331	0.217590

Graph Output of Case 1 and Case 2





Graph Output of Case 3 and Case 4