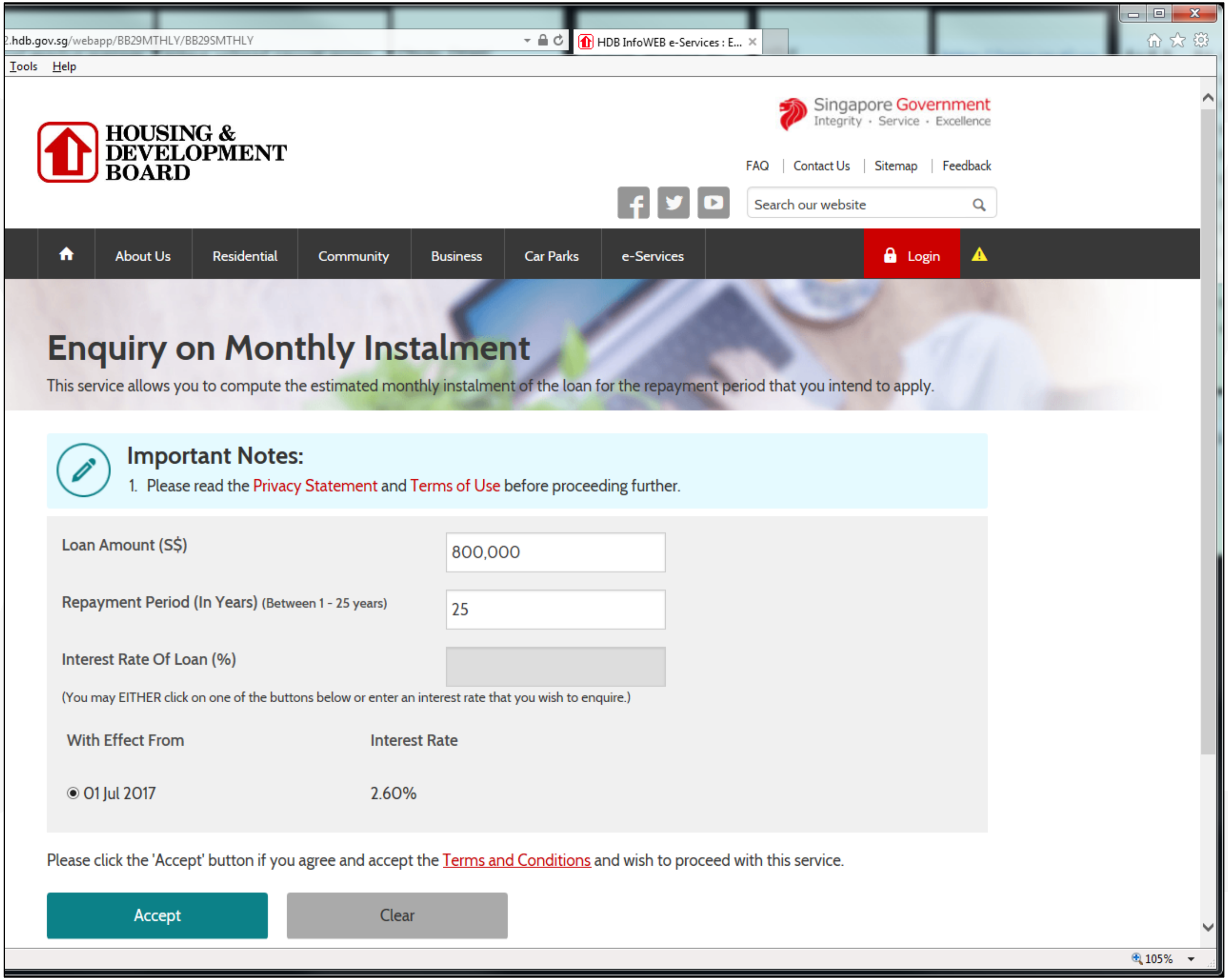
Summary of Applications (Python)

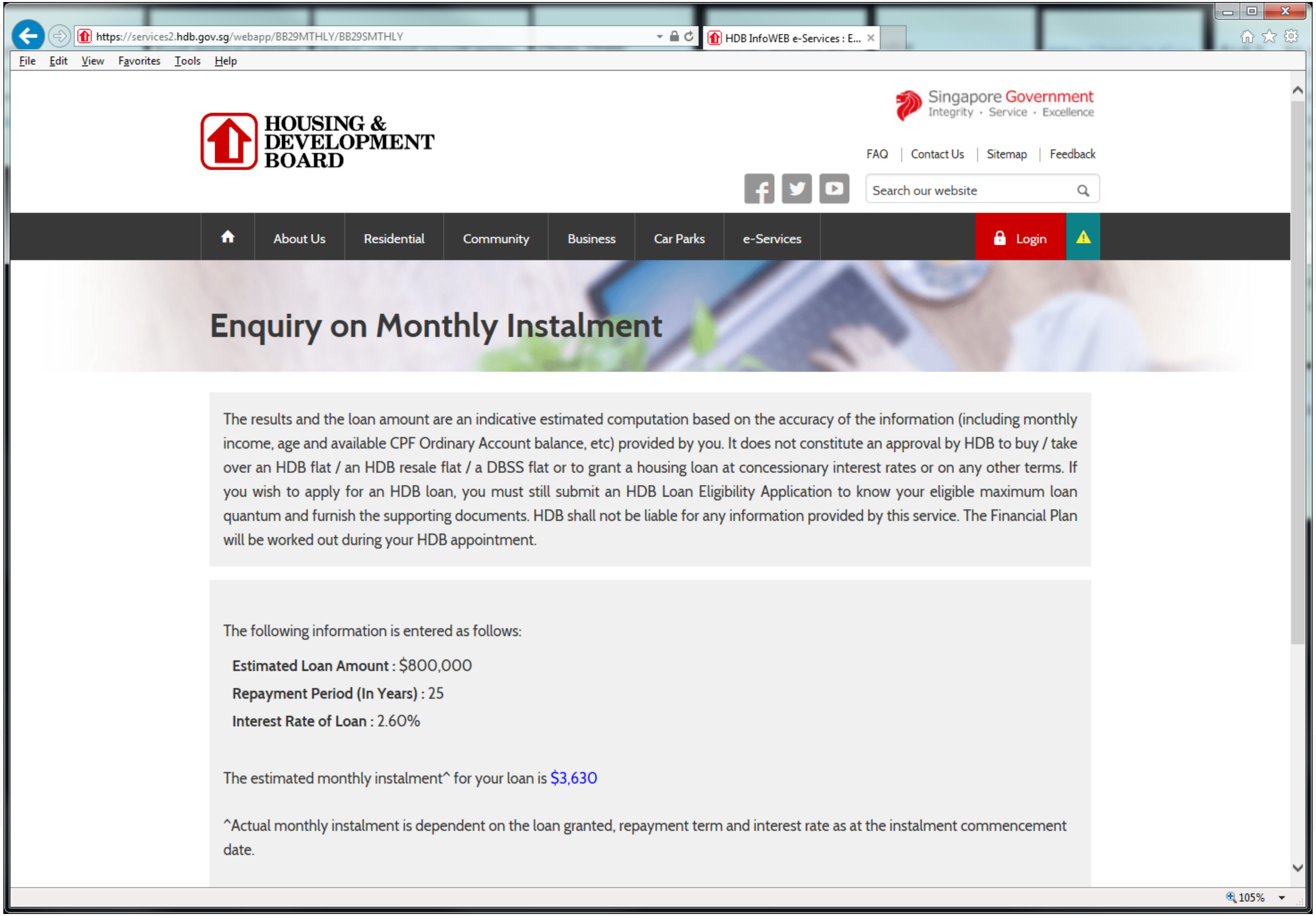
Application 1: HDB Loan Calculator

https://services2.hdb.gov.sg/webapp/BB29MTHLY/BB29SMTHLY



Application 1: HDB Loan Calculator

https://services2.hdb.gov.sg/webapp/BB29MTHLY/BB29SMTHLY


https://services2.hdb.gov.sg/webapp/BB29MTHLY/BB29SMTHLY

HDB InfoWEB e-Services : E...

File Edit View Favorites Tools Help

HOUSING & DEVELOPMENT BOARD

Singapore Government
Integrity · Service · Excellence

FAQ | Contact Us | Sitemap | Feedback

Search our website

About Us | Residential | Community | Business | Car Parks | e-Services | Login

Enquiry on Monthly Instalment

The results and the loan amount are an indicative estimated computation based on the accuracy of the information (including monthly income, age and available CPF Ordinary Account balance, etc) provided by you. It does not constitute an approval by HDB to buy / take over an HDB flat / an HDB resale flat / a DBSS flat or to grant a housing loan at concessionary interest rates or on any other terms. If you wish to apply for an HDB loan, you must still submit an HDB Loan Eligibility Application to know your eligible maximum loan quantum and furnish the supporting documents. HDB shall not be liable for any information provided by this service. The Financial Plan will be worked out during your HDB appointment.

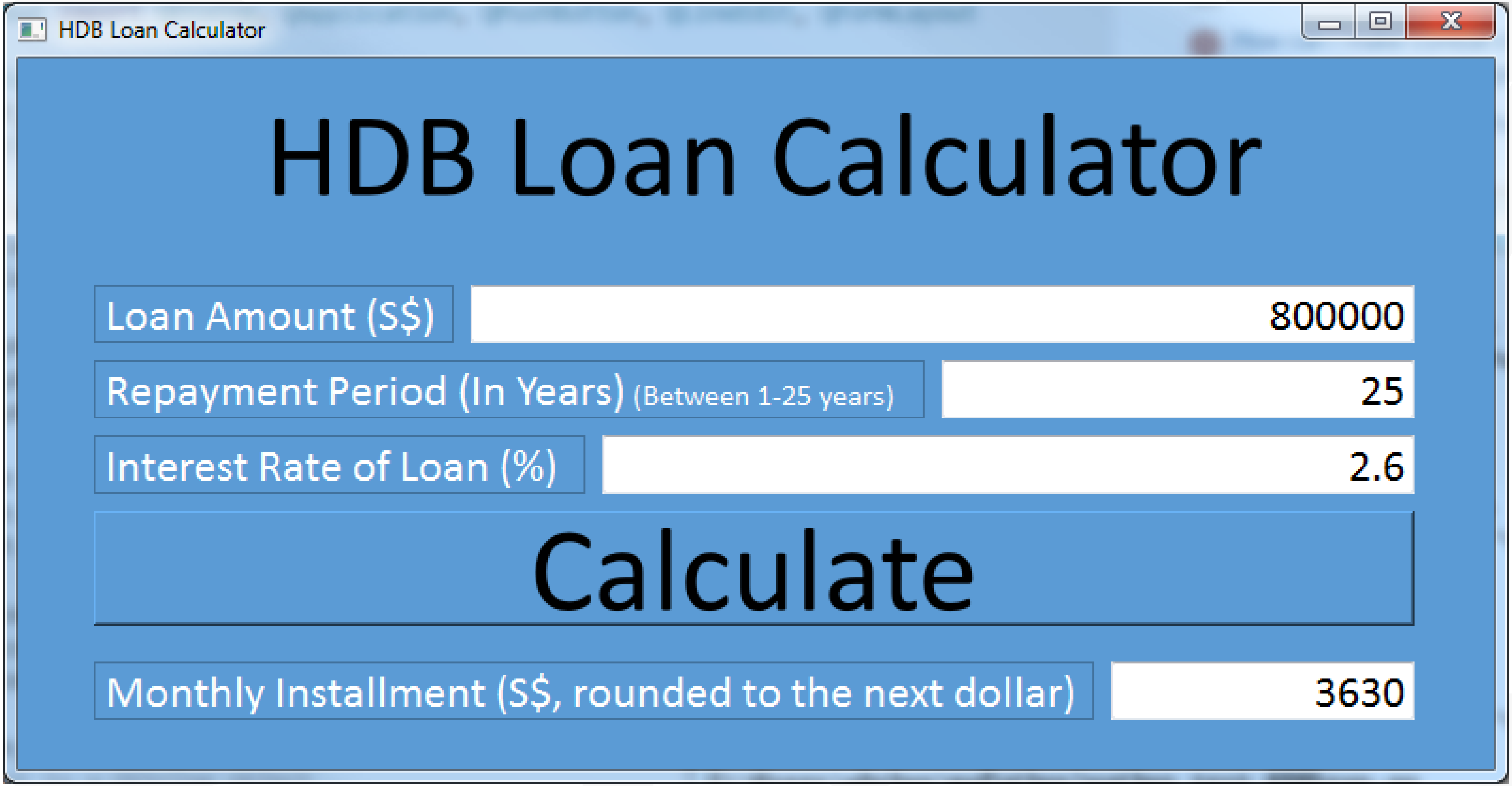
The following information is entered as follows:

Estimated Loan Amount : $800,000
Repayment Period (In Years) : 25
Interest Rate of Loan : 2.60%

The estimated monthly instalment^ for your loan is $3,630

^Actual monthly instalment is dependent on the loan granted, repayment term and interest rate as at the instalment commencement date.

105%


HDB Loan Calculator
HDB Loan Calculator
Loan Amount (S$)
800000
Repayment Period (In Years) (Between 1-25 years)
25
Interest Rate of Loan (%)
2.6
Calculate
Monthly Installment (S$, rounded to the next dollar)
3630

Algorithm:

Given PV, r and t, compute P using

$$P = \frac{\frac{r}{12}(PV)}{1 - \left(1 + \frac{r}{12}\right)^{-12t}}$$

Application 1a: HDB Loan Calculator (★☆☆☆☆)

1. ☐ Numeric Literals
2. ☐ Arithmetic Operators

- ☐ (optional) String Literals
- ☐ (optional) Variables
- ☐ (optional) Assignment Statements
- ☐ (optional) Built-in Function `input`, `float`
- ☐ (optional) Built-in Function `print`
- ☐ (optional) Built-in Function `round`
- ☐ (optional) `math` Function `ceil`

Hint: About 4 lines

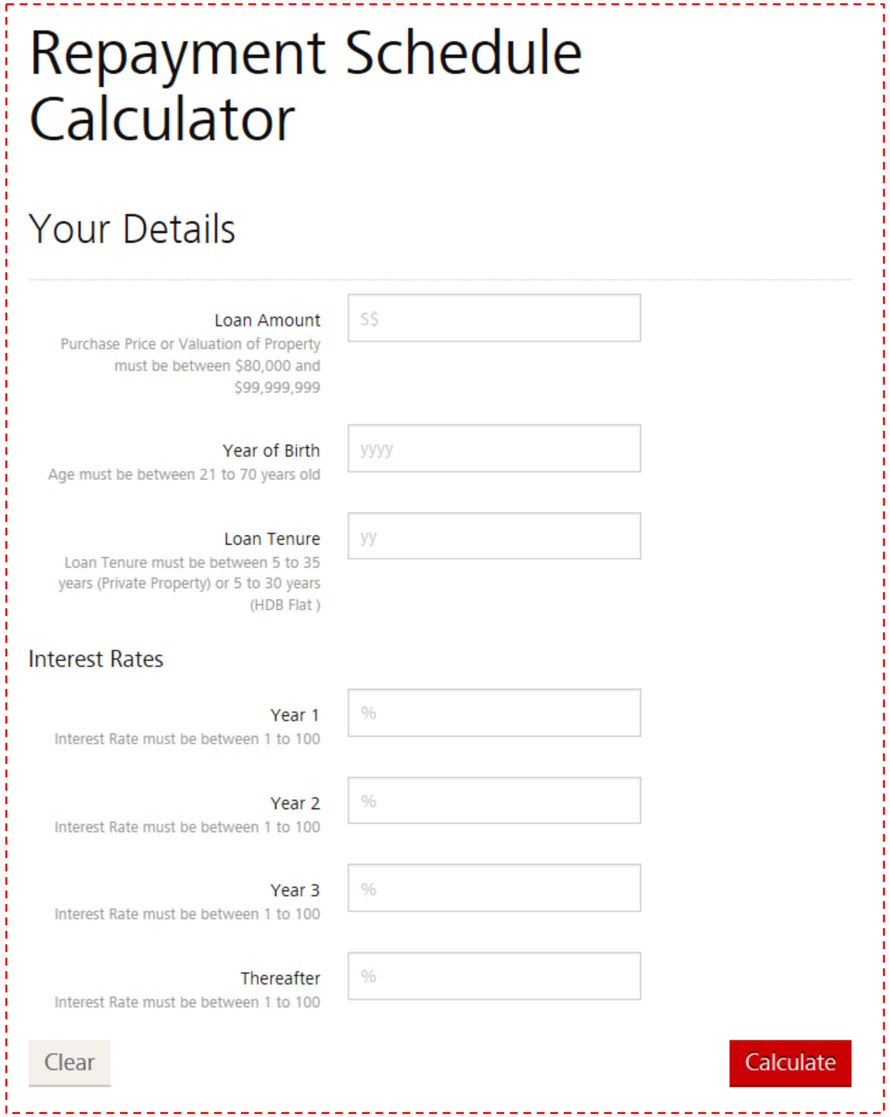
Application 1b: HDB Loan Calculator GUI (★★☆☆☆)

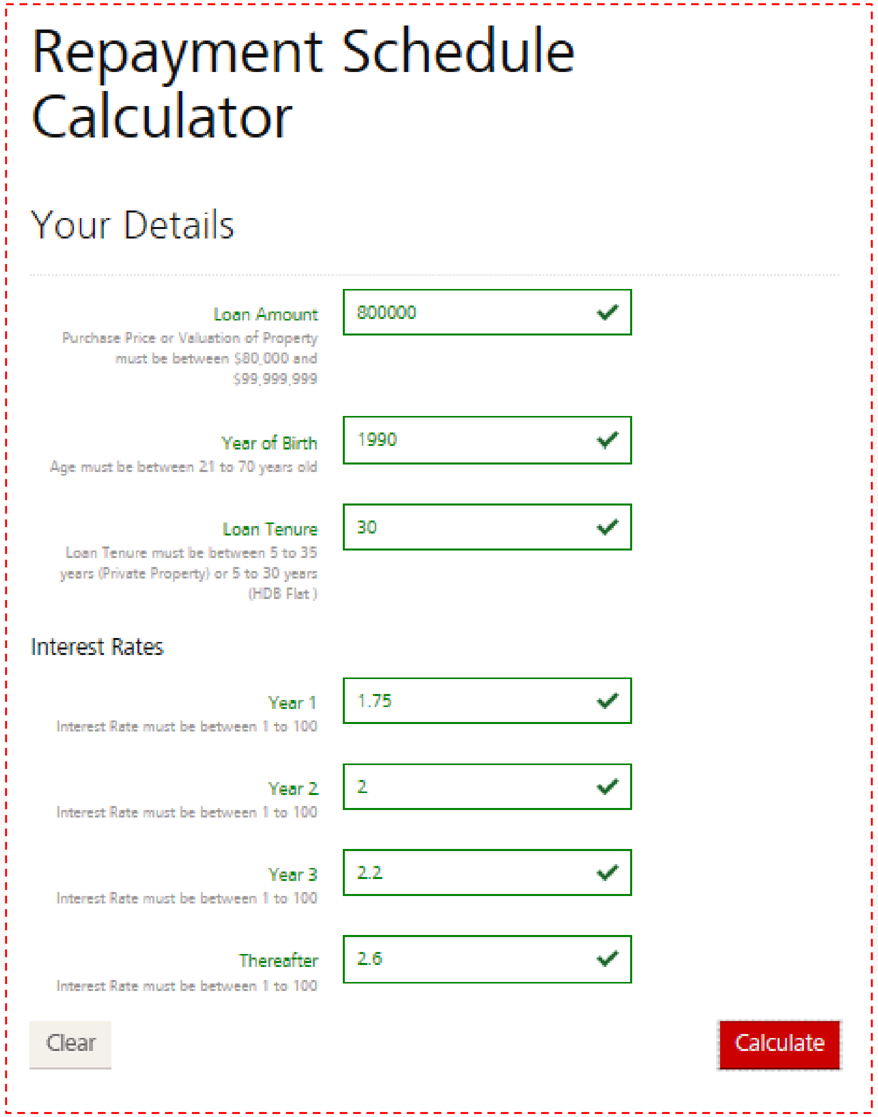
3. ☐ Qt Designer + PyQt5 Template (Label, Push Button, Line Edit)
   a) Functions
   b) Modules
   c) Classes
4. ☐ String Literals
5. ☐ Variables
6. ☐ Assignment Statements
7. ☐ Built-in Function `float` (convert to number)
8. ☐ Built-in Function `str` (convert to string)
9. ☐ Built-in Function `round` (round numbers)
10. ☐ `math` Function `ceil` (round to the next integer)

Hint: About 30 lines + .ui

Application 1c: Bank Loan Calculator

https://www.dbs.com.sg/personal/calculators/homeloans-calculators-repayment-schedule.page





Result

Your monthly repayment schedule is as follow

Month	Interest Rate	Beginning Principal	Monthly Installment	Interest Paid	Principal Paid	Ending Principal
1-12 (Yr 1)	1.75%	$800,000.00	$2,857.95	$13,836.42	$20,458.95	$779,541.05
13-24 (Yr 2)	2.00%	$779,541.05	$2,953.94	$15,407.79	$20,039.49	$759,501.56
25-36 (Yr 3)	2.20%	$759,501.56	$3,029.67	$16,509.71	$19,846.30	$739,655.26
37-48 (Yr 4)	2.60%	$739,655.26	$3,179.54	$19,003.90	$19,150.62	$720,504.63
49-60 (Yr 5)	2.60%	$720,504.63	$3,179.54	$18,500.00	$19,654.52	$700,850.12
61-72 (Yr 6)	2.60%	$700,850.12	$3,179.54	$17,982.85	$20,171.67	$680,678.45
73-84 (Yr 7)	2.60%	$680,678.45	$3,179.54	$17,452.09	$20,702.43	$659,976.02
85-96 (Yr 8)	2.60%	$659,976.02	$3,179.54	$16,907.37	$21,247.15	$638,728.87
97-108 (Yr 9)	2.60%	$638,728.87	$3,179.54	$16,348.31	$21,806.21	$616,922.66
109-120 (Yr 10)	2.60%	$616,922.66	$3,179.54	$15,774.55	$22,379.97	$594,542.69
121-132 (Yr 11)	2.60%	$594,542.69	$3,179.54	$15,185.68	$22,968.84	$571,573.85
133-144 (Yr 12)	2.60%	$571,573.85	$3,179.54	$14,581.32	$23,573.20	$548,000.66
145-156 (Yr 13)	2.60%	$548,000.66	$3,179.54	$13,961.06	$24,193.46	$523,807.20
157-168 (Yr 14)	2.60%	$523,807.20	$3,179.54	$13,324.48	$24,830.04	$498,977.16
157-168 (Yr 14)	2.60%	$523,807.20	$3,179.54	$13,324.48	$24,830.04	$498,977.16
169-180 (Yr 15)	2.60%	$498,977.16	$3,179.54	$12,671.15	$25,483.37	$473,493.80
181-192 (Yr 16)	2.60%	$473,493.80	$3,179.54	$12,000.63	$26,153.89	$447,339.91
193-204 (Yr 17)	2.60%	$447,339.91	$3,179.54	$11,312.47	$26,842.05	$420,497.86
205-216 (Yr 18)	2.60%	$420,497.86	$3,179.54	$10,606.20	$27,548.32	$392,949.54
217-228 (Yr 19)	2.60%	$392,949.54	$3,179.54	$9,881.35	$28,273.17	$364,676.37
229-240 (Yr 20)	2.60%	$364,676.37	$3,179.54	$9,137.42	$29,017.10	$335,659.27
241-252 (Yr 21)	2.60%	$335,659.27	$3,179.54	$8,373.92	$29,780.60	$305,878.67
253-264 (Yr 22)	2.60%	$305,878.67	$3,179.54	$7,590.33	$30,564.19	$275,314.48
265-276 (Yr 23)	2.60%	$275,314.48	$3,179.54	$6,786.12	$31,368.40	$243,946.09
277-288 (Yr 24)	2.60%	$243,946.09	$3,179.54	$5,960.75	$32,193.76	$211,752.32
289-300 (Yr 25)	2.60%	$211,752.32	$3,179.54	$5,113.67	$33,040.85	$178,711.47
301-312 (Yr 26)	2.60%	$178,711.47	$3,179.54	$4,244.30	$33,910.22	$144,801.25
313-324 (Yr 27)	2.60%	$144,801.25	$3,179.54	$3,352.05	$34,802.47	$109,998.78
325-336 (Yr 28)	2.60%	$109,998.78	$3,179.54	$2,436.32	$35,718.20	$74,280.58
337-348 (Yr 29)	2.60%	$74,280.58	$3,179.54	$1,496.50	$36,658.02	$37,622.57
349-360 (Yr 30)	2.60%	$37,622.57	$3,179.54	$531.95	$37,622.57	$0.00

Print Results

Result

Your monthly repayment schedule is as follow

Given → To compute

Month	Interest Rate	Beginning Principal	Monthly Installment	Interest Paid	Principal Paid	Ending Principal
1-12 (Yr 1)	1.75%	$800,000.00	$2,857.95	$13,836.42	$20,458.95	$779,541.05
13-24 (Yr 2)	2.00%	$779,541.05	$2,953.94	$15,407.79	$20,039.49	$759,501.56

Given →

1. Given "Interest Rate", "Beginning Principal", we compute "Monthly Instalment", "Interest Paid", "Principal Paid" and "Ending Principal".
2. Repeat 1 for the next year with a new "Interest Rate" and use the "Ending Principal" of the previous year as the "Beginning Principal" of the current year, till the end of the repayment period.

Formulas:

(yearly)

r: Interest Rate,
bp: Beginning Principal,
n: remaining periods (in years)
m: Monthly Instalment,
ip: Interest Paid,
pp: Principal Paid,
ep: Ending Principal.

$$m = \frac{\frac{r}{12}(bp)}{1 - \left(1 + \frac{r}{12}\right)^{-12n}}$$

$$ep = bp\left(1 + \frac{r}{12}\right)^{12} + \frac{m\left[1 - \left(1 + \frac{r}{12}\right)^{12}\right]}{\frac{r}{12}}$$

$$pp = bp - ep$$

$$ip = 12m - pp$$

Formulas:

(monthly)

r: Interest Rate,
bp: Beginning Principal,
n: remaining periods (in months)
m: Monthly Instalment,
ip: Interest Paid,
pp: Principal Paid,
ep: Ending Principal.

$$m = \frac{\frac{r}{12}(bp)}{1 - \left(1 + \frac{r}{12}\right)^{-n}}$$

$$ep = bp\left(1 + \frac{r}{12}\right) - m$$

$$pp = bp - ep$$

$$ip = m - pp$$

MainWindow

Loan Amount: 800000

Loan Tenue: 30

Calculate

Interest Rates

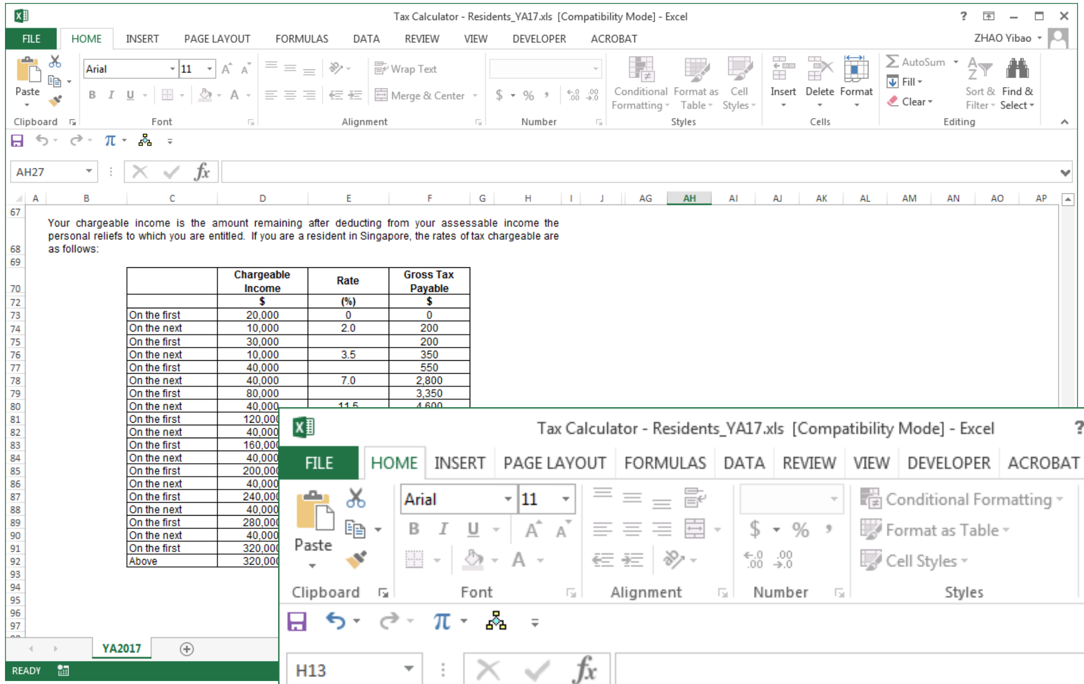
Year 1: 1.75 | Year 2: 2 | Year 3: 2.2 | Thereafter: 2.6

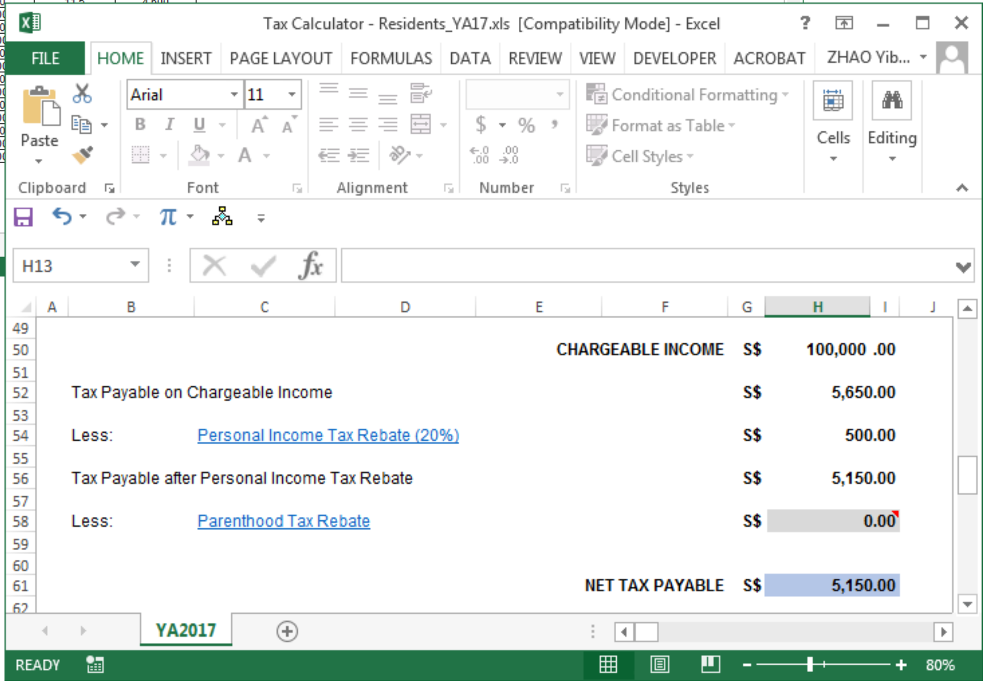
Month	Interest Rate	Beginning Principal	Monthly Installment	Interest Paid	Principal Paid	Ending Principal
1-12 (Yr 1)	1.75%	$800,000.00	$2,857.95	$13,836.42	$20,458.95	$779,541.05
1	1.75%	$800,000.00	$2,857.95	$1,166.67	$1,691.28	$798,308.72
2	1.75%	$798,308.72	$2,857.95	$1,164.20	$1,693.75	$796,614.97
3	1.75%	$796,614.97	$2,857.95	$1,161.73	$1,696.22	$794,918.75
4	1.75%	$794,918.75	$2,857.95	$1,159.26	$1,698.69	$793,220.06
5	1.75%	$793,220.06	$2,857.95	$1,156.78	$1,701.17	$791,518.89
6	1.75%	$791,518.89	$2,857.95	$1,154.30	$1,703.65	$789,815.25
7	1.75%	$789,815.25	$2,857.95	$1,151.81	$1,706.13	$788,109.11
8	1.75%	$788,109.11	$2,857.95	$1,149.33	$1,708.62	$786,400.49
9	1.75%	$786,400.49	$2,857.95	$1,146.83	$1,711.11	$784,689.38
10	1.75%	$784,689.38	$2,857.95	$1,144.34	$1,713.61	$782,975.77
11	1.75%	$782,975.77	$2,857.95	$1,141.84	$1,716.11	$781,259.66
12	1.75%	$781,259.66	$2,857.95	$1,139.34	$1,718.61	$779,541.05
13-24 (Yr 2)	2.00%	$779,541.05	$2,953.94	$15,407.79	$20,039.49	$759,501.56
25-36 (Yr 3)	2.20%	$759,501.56	$3,029.67	$16,509.71	$19,846.30	$739,655.26
37-48 (Yr 4)	2.60%	$739,655.26	$3,179.54	$19,003.90	$19,150.62	$720,504.63
49-60 (Yr 5)	2.60%	$720,504.63	$3,179.54	$18,500.00	$19,654.52	$700,850.12
61-72 (Yr 6)	2.60%	$700,850.12	$3,179.54	$17,982.85	$20,171.67	$680,678.45

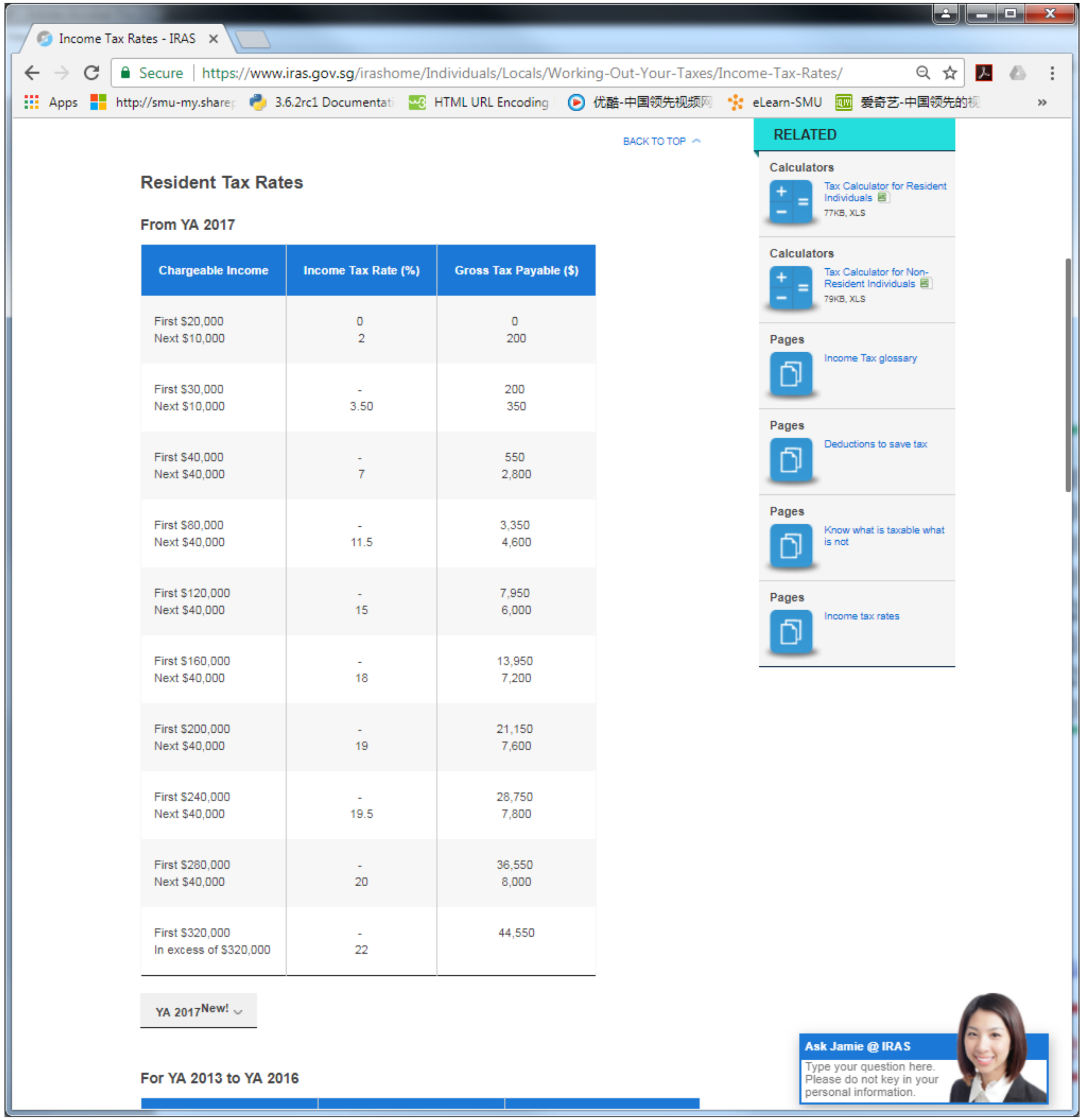
Application 2: Income Tax Calculator

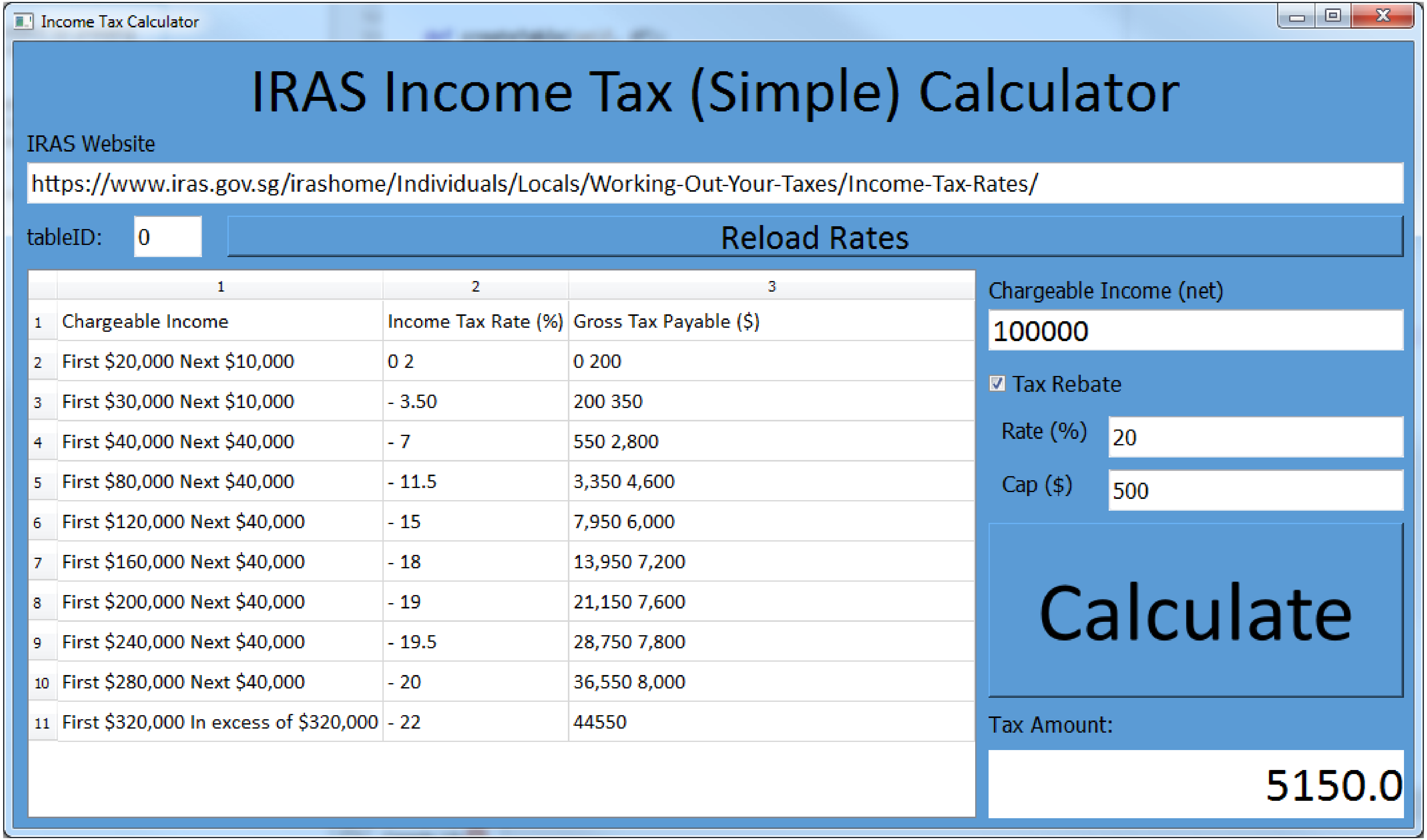
https://www.iras.gov.sg/irashome/uploadedFiles/IRASHome/Individuals/Tax%20Calculator%20-%20Residents_YA17.xls

https://www.iras.gov.sg/irashome/Individuals/Locals/Working-Out-Your-Taxes/Income-Tax-Rates/








Income Tax Calculator
IRAS Income Tax (Simple) Calculator
IRAS Website
https://www.iras.gov.sg/irashome/Individuals/Locals/Working-Out-Your-Taxes/Income-Tax-Rates/
tableID:
0
Reload Rates
1
2
3
1 Chargeable Income | Income Tax Rate (%) | Gross Tax Payable ($)
2 First $20,000 Next $10,000 | 0 2 | 0 200
3 First $30,000 Next $10,000 | - 3.50 | 200 350
4 First $40,000 Next $40,000 | - 7 | 550 2,800
5 First $80,000 Next $40,000 | - 11.5 | 3,350 4,600
6 First $120,000 Next $40,000 | - 15 | 7,950 6,000
7 First $160,000 Next $40,000 | - 18 | 13,950 7,200
8 First $200,000 Next $40,000 | - 19 | 21,150 7,600
9 First $240,000 Next $40,000 | - 19.5 | 28,750 7,800
10 First $280,000 Next $40,000 | - 20 | 36,550 8,000
11 First $320,000 In excess of $320,000 | - 22 | 44550
Chargeable Income (net)
100000
Tax Rebate
Rate (%)
20
Cap ($)
500
Calculate
Tax Amount:
5150.0

Resident Tax Rates

From YA 2017

Chargeable Income	Income Tax Rate (%)	Gross Tax Payable ($)
First $20,000 Next $10,000	0 2	0 200
First $30,000 Next $10,000	- 3.50	200 350
First $40,000 Next $40,000	- 7	550 2,800
First $80,000 Next $40,000	- 11.5	3,350 4,600
First $120,000 Next $40,000	- 15	7,950 6,000
First $160,000 Next $40,000	- 18	13,950 7,200
First $200,000 Next $40,000	- 19	21,150 7,600
First $240,000 Next $40,000	- 19.5	28,750 7,800
First $280,000 Next $40,000	- 20	36,550 8,000
First $320,000 In excess of $320,000	- 22	44,550

```
xi=[int(x.split()[1].replace('$','').replace(',','')) for x in tables[0][0][1:]]
xi
```

```
[20000, 30000, 40000, 80000, 120000, 160000, 200000, 240000, 280000, 320000]
```

```
mi=[float(x.split()[1]) for x in tables[0][1][1:]]
mi
```

```
[2.0, 3.5, 7.0, 11.5, 15.0, 18.0, 19.0, 19.5, 20.0, 22.0]
```

```
bi=[int(x.split()[0].replace(',','')) for x in tables[0][2][1:]]
bi
```

```
[0, 200, 550, 3350, 7950, 13950, 21150, 28750, 36550, 44550]
```

$$y(x) = \begin{cases} 0, & 0 \le x < 20{,}000 \\ 0 + 0.02(x - 20000), & 20{,}000 \le x < 30{,}000 \\ 200 + 0.035(x - 30000), & 30{,}000 \le x < 40{,}000 \\ 550 + 0.07(x - 40000), & 40{,}000 \le x < 80{,}000 \\ 3350 + 0.115(x - 80000), & 80{,}000 \le x < 120{,}000 \\ 7950 + 0.15(x - 120000), & 120{,}000 \le x < 160{,}000 \\ 13950 + 0.18(x - 160000), & 160{,}000 \le x < 200{,}000 \\ 21150 + 0.19(x - 200000), & 200{,}000 \le x < 240{,}000 \\ 28750 + 0.195(x - 240000), & 240{,}000 \le x < 280000 \\ 36550 + 0.20(x - 280000), & 280{,}000 \le x < 320000 \\ 44550 + 0.22(x - 320000), & 320{,}000 \le x \end{cases}$$

Application 2a: Income Tax Calculator (★★☆☆☆)

1. ☐ Numeric Literals
2. ☐ Arithmetic Operators
3. ☐ Comparison (Relational) Operators
4. ☐ Variables
5. ☐ Assignment Statements
6. ☐ Lists
7. ☐ Lists Comprehension (or `for` Statements)
8. ☐ `if` Statements
9. ☐ Built-in Functions `float`, `int`, `sum`, `max`, `print`

❖ ☐ (optional) Built-in Function `input`

Hint: About 15 lines

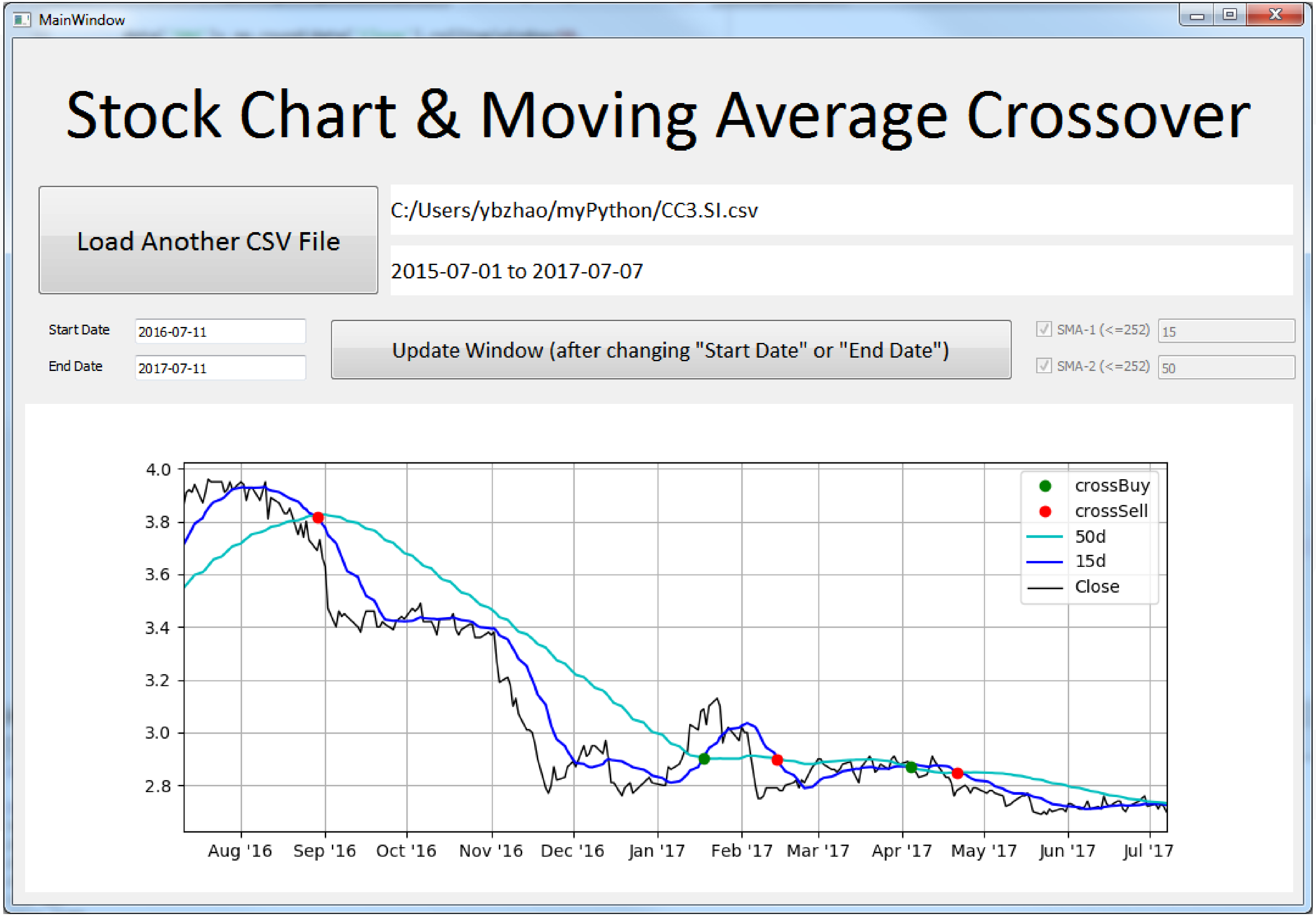
Application 2b: Income Tax Calculator GUI (★★★☆☆)

10. ☐ Qt Designer + PyQt5 Template (Label, Push Button, Line Edit, Check Box, Table)
    a) Functions, Modules and Classes
    b) `pandas` Functions `read_html`
    c) `pandas.DataFrame`
    d) String Functions `split`, `replace`

Hint: About 100 lines + .ui

Application 3: Stock Chart & Moving Average (MA) Crossover

Yahoo Finance (CC3.SI) (*created on 10/Jul/2017)
Summary > Click Chart "Full screen"
> Add Indicator > Simple Moving Average (SMA)
(window size: 1 year)


MainWindow
Stock Chart & Moving Average Crossover
Load Another CSV File
C:/Users/ybzhao/myPython/CC3.SI.csv
2015-07-01 to 2017-07-07
Start Date
2016-07-11
End Date
2017-07-11
Update Window (after changing "Start Date" or "End Date")
SMA-1 (<=252)
15
SMA-2 (<=252)
50
crossBuy
crossSell
50d
15d
Close
4.0
3.8
3.6
3.4
3.2
3.0
2.8
Aug '16
Sep '16
Oct '16
Nov '16
Dec '16
Jan '17
Feb '17
Mar '17
Apr '17
May '17
Jun '17
Jul '17

Algorithm:

Given historical daily "Close Price", $data_i$, $i = 0,1,2, \ldots, N$

1. Compute $SMA15$ and $SMA50$.
2. Compare $SMA15$ to $SMA50$ by computing $SMA15 - SMA50$. If the difference is positive, it indicates that $SMA15$ is above $SMA50$, and vice versa. We denote the earlier situation by 1, and the latter by 0. It results a sequence of 0's and 1's.
3. Compute the difference of each pair of consecutive numbers in the above mentioned sequence of 0's and 1's. It results a new sequence of values of 0 (when $0 - 0$ or $1 - 1$), 1 (when $1 - 0$) and -1 (when $0 - 1$).
4. Find the locations of 1's and -1's in the new sequence. Those will be the locations of crossovers, after shifting by 1 to the right.

Challenge: There might be missing data.

Application 3a: Stock Chart & MA Crossover (★★☆☆☆)

1. ☐ `pandas` Functions `read_csv`
2. ☐ `pandas` `DataFrame` Functions `drop`, `plot`
3. ☐ `pandas` `Series` Functions `rolling().mean()`
4. ☐ `numpy` Functions `round`, `nan`, `diff`
5. ☐ `matplotlib.pyplot` Functions `subplots`, `tight_layout`
6. ☐ Comparison (Relational) Operators
7. ☐ Variables
8. ☐ Literals (numeric and strings)
9. ☐ Assignment Statements
10. ☐ Arithmetic Operators on `pandas Series`
11. ☐ Logical Indexing

Hint: About 20 lines

Application 3b: Stock Chart & MA Crossover GUI (★★★★☆)

12. ☐ Qt Designer + PyQt5 Template (Label, Push Button, Line Edit, Check Box)
    a) Functions, Modules and Classes
    b) `datetime` Functions `datetime`, `strptime`, `date`, `date.today`, `date.year`, `date.month`, `date.day`, `strftime`,
    c) `matplotlib.dates` Function `DateFormatter`
13. ☐ Built-in Functions `enumerate`, `next`,
14. ☐ Generator Expression
15. ☐ List Comprehension
16. ☐ `pandas DataFrame` Function `copy`
17. ☐ `assert` Statement

Hint: About 200 lines + .ui

Application 4: Option Price (and Greeks) Calculator

Option pricing refers to the amount per share that an option is traded. Options are derivative contracts that give the holder (the "buyer") the right, but not the obligation, to buy or sell the underlying instrument at an agreed-upon price on or before a specified future date. Although the holder of the option is not obligated to exercise the option, the option writer (the "seller") has an obligation to buy or sell the underlying instrument if the option is exercised.

Price ⇔ Implied Volatility

Read more:

Options Pricing http://www.investopedia.com/university/options-pricing/#ixzz4mVw9p0nB

Application 4: Option Price (and Greeks) Calculator

Many option traders rely on **the "Greeks"** to evaluate option positions and to determine option sensitivity. The Greeks are a collection of statistical values that measure the risk involved in an options contract in relation to certain underlying variables. Popular Greeks include Delta, Vega, Gamma and Theta.

For example, let $V(S, K, r, q, \tau, \sigma)$ denote the option price function,

$$\text{Delta: } \Delta = \frac{\partial V}{\partial S} \approx \frac{V(S+\Delta S, K, r, q, \tau, \sigma) - V(S, K, r, q, \tau, \sigma)}{\Delta S}$$

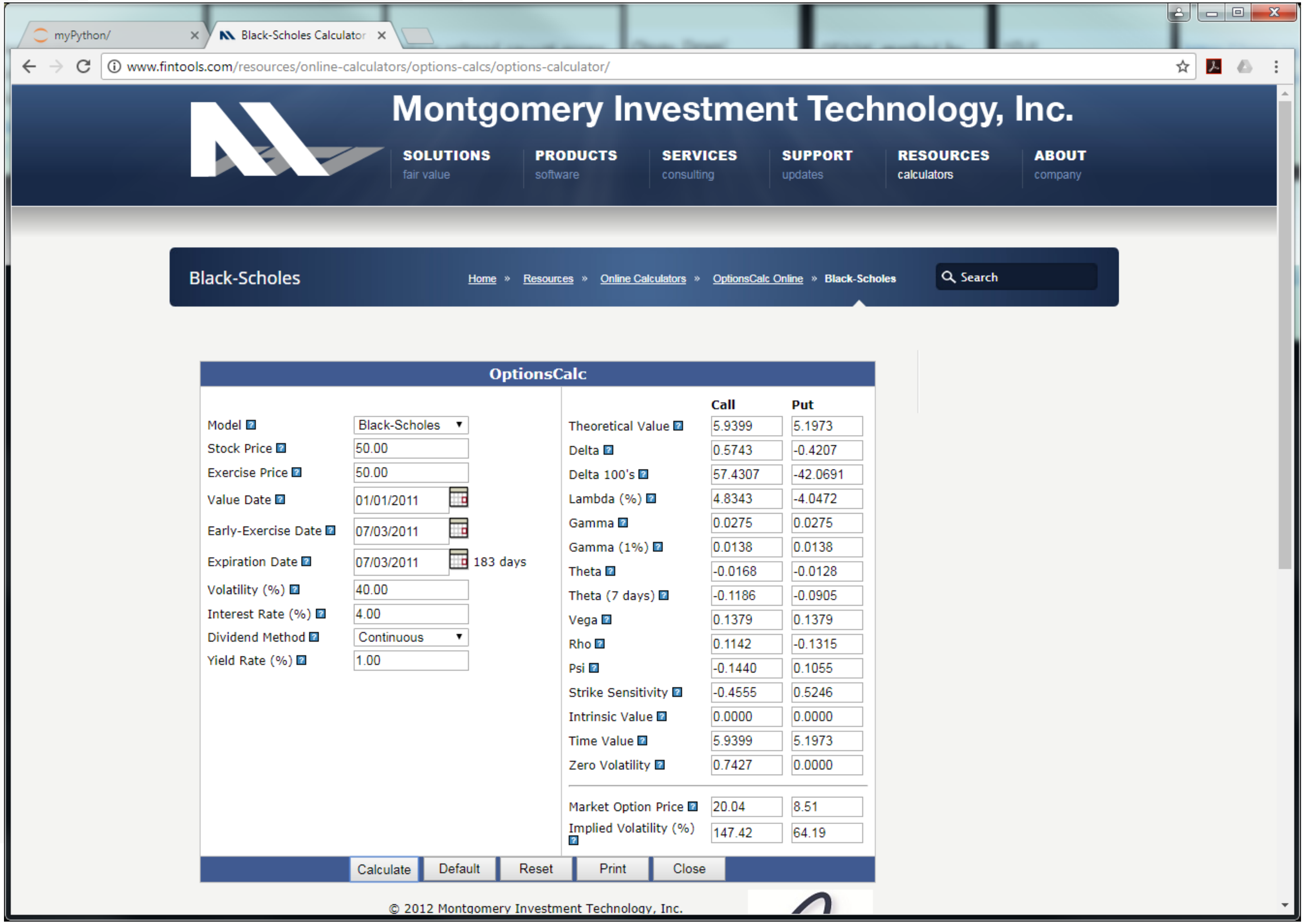
$\tau = T - t$

Read more: Options Pricing:

The Greeks http://www.investopedia.com/university/options-pricing/greeks.asp#ixzz4mVxbx200

Application 4: Option Price (and Greeks) Calculator

http://www.fintools.com/resources/online-calculators/options-calcs/options-calculator/



Gamma (1%)

The rate of change in an option's delta with respect to 1% change in the price of the underlying assets or instrument.

[Delta(1.01S)-Delta(0.99S)]/2

Theta

The rate of change in an option's price as time (1 day) passes with all else remaining the same.

[Price(τ-1 days)-Price(τ days)]

Theta (7 days)

The rate of change in an option's price as time (7 day) passes with all else remaining the same.

[Price(τ-1 days)-Price(τ days)]

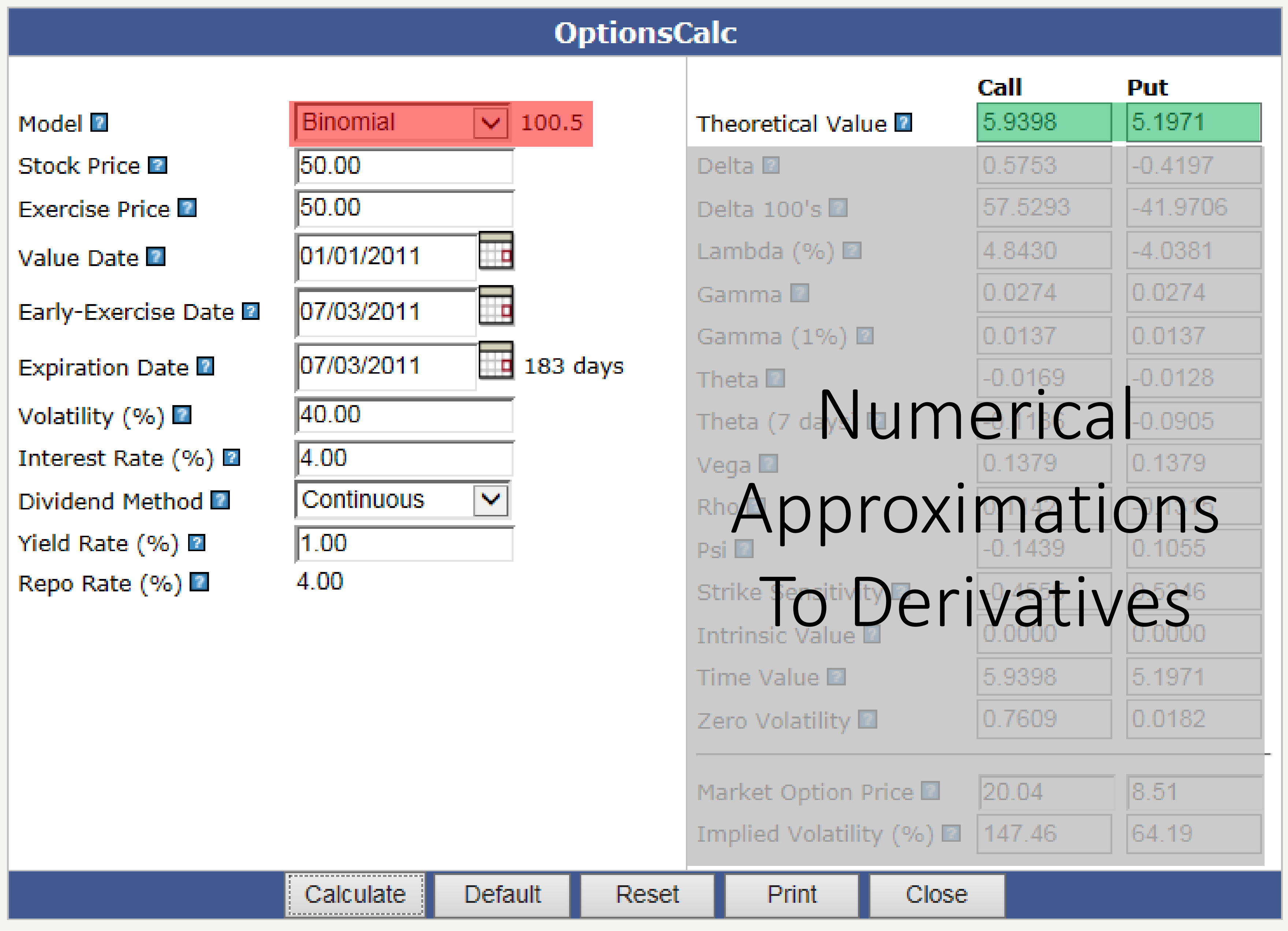
OptionsCalc

Model	Black-Scholes
Stock Price	50.00
Exercise Price	50.00
Value Date	01/01/2011
Early-Exercise Date	07/03/2011
Expiration Date	07/03/2011 183 days
Volatility (%)	40.00
Interest Rate (%)	4.00
Dividend Method	Continuous
Yield Rate (%)	1.00

	Call	Put
Theoretical Value	5.9399	5.1973
Delta	0.5743	-0.4207
Delta 100's	57.4307	-42.0691
Lambda (%)	4.8343	-4.0472
Gamma	0.0275	0.0275
Gamma (1%)	0.0138	0.0138
Theta	-0.0168	-0.0128
Theta (7 days)	-0.1186	-0.0905
Vega	0.1379	0.1379
Rho	0.1142	-0.1315
Psi	-0.1440	0.1055
Strike Sensitivity	-0.4555	0.5246
Intrinsic Value	0.0000	0.0000
Time Value	5.9399	5.1973
Zero Volatility	0.7427	0.0000
Market Option Price	20.04	8.51
Implied Volatility (%)	147.42	64.19

Calculate | Default | Reset | Print | Close

Numerical Approximations To Derivatives



Delta: $\Delta = \frac{\partial V}{\partial S} \approx \frac{V(S+\Delta S)-V(S)}{\Delta S}$ or $\frac{V(S+\Delta S)-V(S-\Delta S)}{2\Delta S}$

Delta 100's: $\Delta_{100} = 100\Delta$

Lambda: $\lambda = \Delta \times \frac{S}{V}$

Gamma: $\Gamma = \frac{\partial^2 V}{\partial S^2} \approx \frac{V(S+\Delta S)-2V(S)+V(S-\Delta S)}{(\Delta S)^2}$

Gamma 1%: $= \frac{\Delta(1.01S)-\Delta(0.99S)}{2}$

Theta 1 day: $\theta_1 = V(\tau - 1/365) - V(\tau)$

Theta 7 days: $\theta_7 = V\left(\tau - \frac{7}{365}\right) - V(t)$

Vega: $\nu = \frac{\partial V}{\partial \sigma} \approx \frac{V(\sigma+\Delta\sigma)-V(\sigma)}{\Delta\sigma}$ or $\frac{V(\sigma+\Delta\sigma)-V(\sigma-\Delta\sigma)}{2\Delta\sigma}$

Rho: $\rho = \frac{\partial V}{\partial r} \approx \frac{V(r+\Delta r)-V(r)}{\Delta r}$ or $\frac{V(r+\Delta r)-V(r-\Delta r)}{2\Delta r}$

Psi: $\Psi = \frac{\partial V}{\partial q} \approx \frac{V(q+\Delta q)-V(q)}{\Delta q}$ or $\frac{V(q+\Delta q)-V(q-\Delta q)}{2\Delta q}$

Strike Sensitivity (SS): $\frac{V-S\Delta}{K}$

Intrinsic Value (IV):

$$\max(S-K,0),$$
$$\max(K-S,0)$$

Time Value: $V - IV$

Zero Volatility:

$$e^{-r\tau}\max\left(Se^{(r-q)\tau}-K,0\right)$$
$$e^{-r\tau}\max\left(K-Se^{(r-q)\tau},0\right)$$

option_BS - DataFrame

Index	Attributes	Call	Put
S	Stock Price	50.0000	50.0000
K	Exercise Price	50.0000	50.0000
t	Time to Maturity	0.5014	0.5014
sigma	Volatility (%)	40.0000	40.0000
r	Interest Rate (%)	4.0000	4.0000
q	Yiled Rate (%)	1.0000	1.0000
V	Theoretical Value	5.9399	5.1973
delta	Delta	0.5743	-0.4207
delta100	Delta 100's	57.4308	-42.0691
lambda	Lambda (%)	4.8343	-4.0472
gamma	Gamma	0.0275	0.0275
gamma1pct	Gamma (1%)	0.0138	0.0138
theta	Theta	-0.0168	-0.0128
theta7day	Theta (7 days)	-0.1186	-0.0905
vega	Vega	0.1379	0.1379
rho	Rho	0.1142	-0.1315
psi	Psi	-0.1440	0.1055
ss	Strike Sensitivity	-0.4555	0.5246
iv	Intrinsic Value	0.0000	0.0000
tv	Time Value	5.9399	5.1973
zv	Zero Volatility	0.7427	0.0000

Format | Resize | Background co | Column min/ma | OK | Cancel

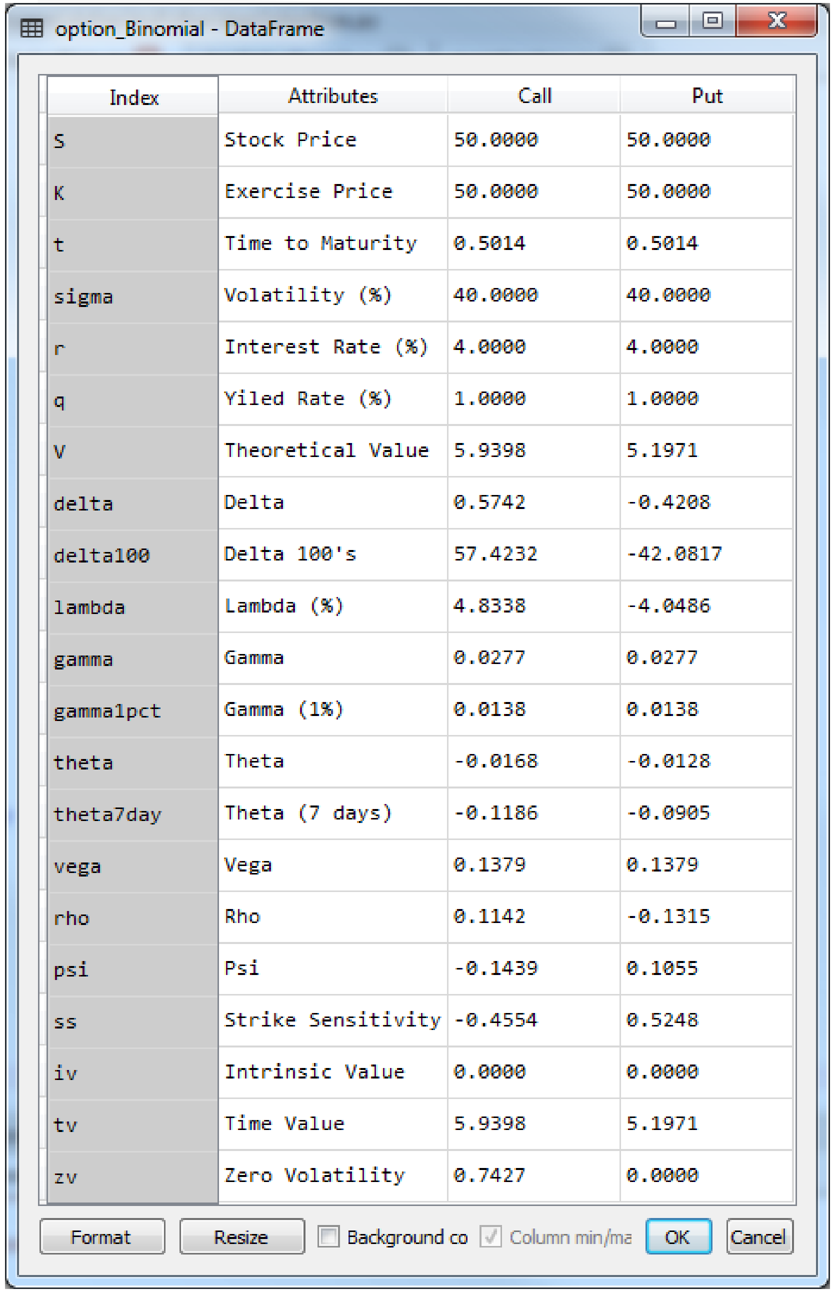
MainWindow

Option Price (and Greeks) Calculator

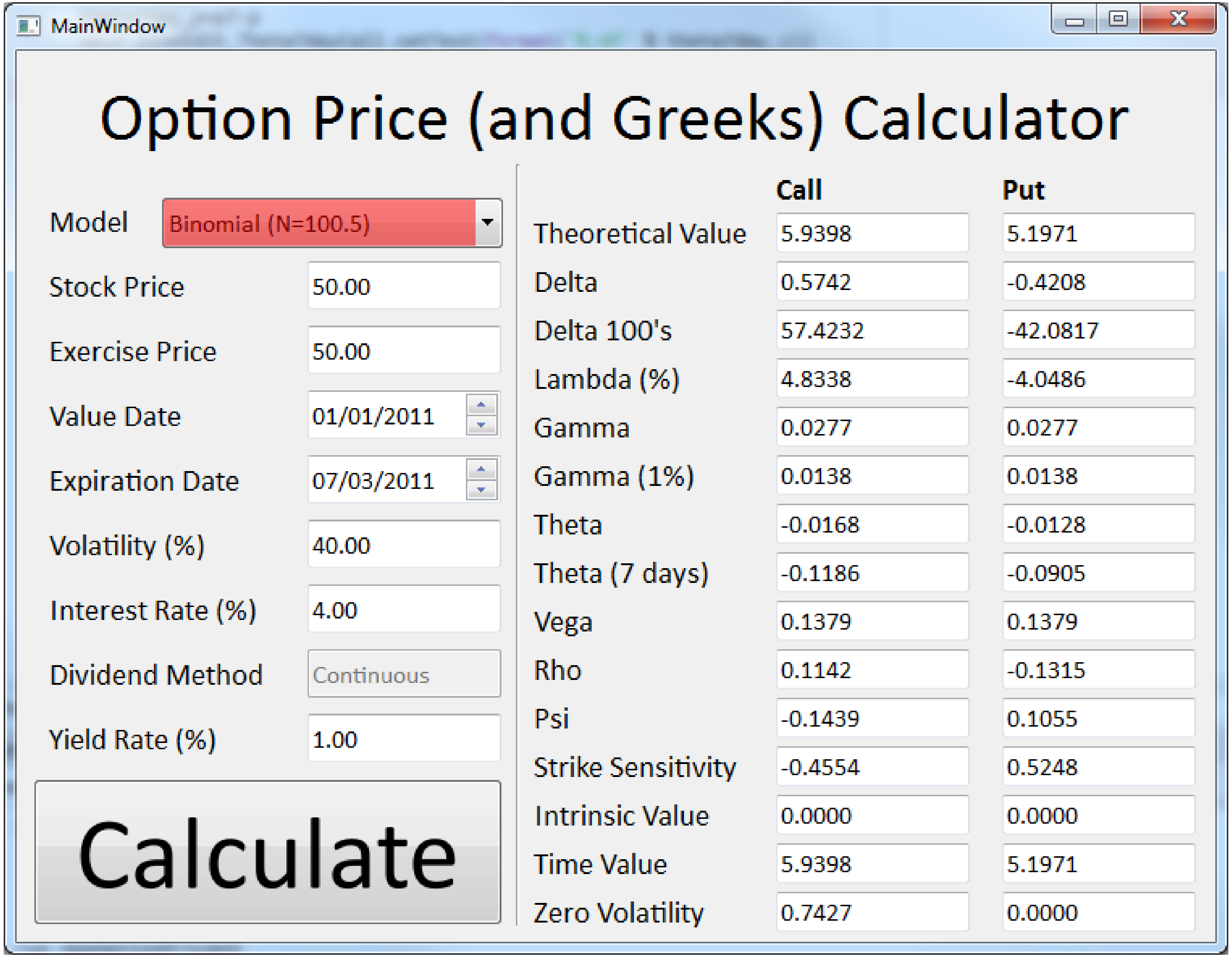
Model	Black-Scholes
Stock Price	50.00
Exercise Price	50.00
Value Date	01/01/2011
Expiration Date	07/03/2011
Volatility (%)	40.00
Interest Rate (%)	4.00
Dividend Method	Continuous
Yield Rate (%)	1.00

Calculate

	Call	Put
Theoretical Value	5.9399	5.1973
Delta	0.5743	-0.4207
Delta 100's	57.4308	-42.0691
Lambda (%)	4.8343	-4.0472
Gamma	0.0275	0.0275
Gamma (1%)	0.0138	0.0138
Theta	-0.0168	-0.0128
Theta (7 days)	-0.1186	-0.0905
Vega	0.1379	0.1379
Rho	0.1142	-0.1315
Psi	-0.1440	0.1055
Strike Sensitivity	-0.4555	0.5246
Intrinsic Value	0.0000	0.0000
Time Value	5.9399	5.1973
Zero Volatility	0.7427	0.0000



Index	Attributes	Call	Put
S	Stock Price	50.0000	50.0000
K	Exercise Price	50.0000	50.0000
t	Time to Maturity	0.5014	0.5014
sigma	Volatility (%)	40.0000	40.0000
r	Interest Rate (%)	4.0000	4.0000
q	Yiled Rate (%)	1.0000	1.0000
V	Theoretical Value	5.9398	5.1971
delta	Delta	0.5742	-0.4208
delta100	Delta 100's	57.4232	-42.0817
lambda	Lambda (%)	4.8338	-4.0486
gamma	Gamma	0.0277	0.0277
gamma1pct	Gamma (1%)	0.0138	0.0138
theta	Theta	-0.0168	-0.0128
theta7day	Theta (7 days)	-0.1186	-0.0905
vega	Vega	0.1379	0.1379
rho	Rho	0.1142	-0.1315
psi	Psi	-0.1439	0.1055
ss	Strike Sensitivity	-0.4554	0.5248
iv	Intrinsic Value	0.0000	0.0000
tv	Time Value	5.9398	5.1971
zv	Zero Volatility	0.7427	0.0000



N=10,000,000+10,000,000 (antithetic), seed(0)

option_MonteCarlo - DataFrame

Index	Attributes	Call	Put
S	Stock Price	50.0000	50.0000
K	Exercise Price	50.0000	50.0000
t	Time to Maturity	0.5014	0.5014
sigma	Volatility (%)	40.0000	40.0000
r	Interest Rate (%)	4.0000	4.0000
q	Yiled Rate (%)	1.0000	1.0000
V	Theoretical Value	5.9383	5.1966
delta	Delta	0.5743	-0.4207
delta100	Delta 100's	57.4298	-42.0682
lambda	Lambda (%)	4.8355	-4.0477
gamma	Gamma	0.0277	0.0277
gamma1pct	Gamma (1%)	0.2871	-0.2104
theta	Theta	-0.0168	-0.0128
theta7day	Theta (7 days)	-0.1185	-0.0905
vega	Vega	0.1378	0.1379
rho	Rho	0.1142	-0.1315
psi	Psi	-0.1440	0.1055
ss	Strike Sensitivity	-0.4555	0.5246
iv	Intrinsic Value	0.0000	0.0000
tv	Time Value	5.9383	5.1966
zv	Zero Volatility	0.7427	0.0000

Format | Resize | Background co | Column min/ma | OK | Cancel

MainWindow

Option Price (and Greeks) Calculator

Model	Monte Carlo Simulation
Stock Price	50.00
Exercise Price	50.00
Value Date	01/01/2011
Expiration Date	07/03/2011
Volatility (%)	40.00
Interest Rate (%)	4.00
Dividend Method	Continuous
Yield Rate (%)	1.00

Calculate

	Call	Put
Theoretical Value	5.9383	5.1966
Delta	0.5743	-0.4207
Delta 100's	57.4298	-42.0682
Lambda (%)	4.8355	-4.0477
Gamma	0.0277	0.0277
Gamma (1%)	0.2871	-0.2104
Theta	-0.0168	-0.0128
Theta (7 days)	-0.1185	-0.0905
Vega	0.1378	0.1379
Rho	0.1142	-0.1315
Psi	-0.1440	0.1055
Strike Sensitivity	-0.4555	0.5246
Intrinsic Value	0.0000	0.0000
Time Value	5.9383	5.1966
Zero Volatility	0.7427	0.0000

Algorithm (Black-Scholes):

Given S, K, r, q, t, T and σ, (1 day=1/365 years)

- Compute Option Prices/Values c and p using

$$c = S \cdot e^{-q(T-t)} \cdot \Phi(d_1) - K \cdot e^{-r(T-t)} \cdot \Phi(d_2)$$

and

$$p = K \cdot e^{-r(T-t)} \cdot \Phi(-d_2) - S \cdot e^{-q(T-t)} \cdot \Phi(-d_1)$$

where

$$d_1 = \frac{\ln\left(\frac{S}{K}\right) + \left(r - q + \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}}, d_2 = d_1 - \sigma\sqrt{T-t}$$

$$c = V_c(S, K, r, q, t, T, \sigma), p = V_p(S, K, r, q, t, T, \sigma)$$

- Compute Greeks using :
  - Delta

$$\Delta_c = e^{-q(T-t)} \cdot \Phi(d_1), \Delta_p = -e^{-q(T-t)} \cdot \Phi(-d_1)$$

$$\Delta_c = \Delta_c(S, K, r, q, t, T, \sigma), \Delta_p = \Delta_p(S, K, r, q, t, T, \sigma)$$

  - Delta 100's

$$100\Delta_c, 100\Delta_p$$

  - Lambda

$$\lambda_c = \Delta_c \times \frac{S}{c}, \lambda_p = \Delta_p \times \frac{S}{p}$$

- Compute Greeks (continued)
  - Gamma

$$\Gamma_c = \Gamma_p = e^{-q(T-t)} \frac{\phi(d_1)}{S\sigma\sqrt{T-t}}$$

  - Gamma (1%)

$$\frac{\Delta_c(1.01S, K, r, q, t, T, \sigma) - \Delta_c(0.99S, K, r, q, t, T, \sigma)}{2},$$

$$\frac{\Delta_p(1.01S, K, r, q, t, T, \sigma) - \Delta_p(0.99S, K, r, q, t, T, \sigma)}{2}$$

  - Theta (1 day)

$$V_c\left(S, K, r, q, t - \frac{1}{365}, T, \sigma\right) - V_c(S, K, r, q, t, T, \sigma),$$

$$V_p\left(S, K, r, q, t - \frac{1}{365}, T, \sigma\right) - V_p(S, K, r, q, t, T, \sigma)$$

  - Theta (7 days)

$$V_c\left(S, K, r, q, t - \frac{7}{365}, T, \sigma\right) - V_c(S, K, r, q, t, T, \sigma),$$

$$V_p\left(S, K, r, q, t - \frac{7}{365}, T, \sigma\right) - V_p(S, K, r, q, t, T, \sigma)$$

  - Vega

$$\nu_c = \nu_p = Se^{-q(T-t)}\phi(d_1)\sqrt{T-t} \times \frac{1}{100}$$

$\Phi(x)$ and $\phi(x)$ are, respectively, the cumulative distribution function and probability density function of the standard normal distribution.

Algorithm (continued):

- Compute Greeks (continued)
  - Rho

$$\rho_c = K(T-t)e^{-r(T-t)}\Phi(d_2) \times \frac{1}{100}, \qquad \rho_c = -K(T-t)e^{-r(T-t)}\Phi(-d_2) \times \frac{1}{100}$$

  - Psi

$$\Psi_c = \left[-S(T-t)e^{-q(T-t)}\Phi(d_1) - \frac{S\sqrt{T-t}}{\sigma}e^{-q(T-t)}\phi(d_1) + \frac{K\sqrt{T-t}}{\sigma}e^{-r(T-t)}\phi(d_2)\right] \times \frac{1}{100},$$

$$\Psi_p = \left[S(T-t)e^{-q(T-t)}\Phi(-d_1) - \frac{S\sqrt{T-t}}{\sigma}e^{-q(T-t)}\phi(-d_1) + \frac{K\sqrt{T-t}}{\sigma}e^{-r(T-t)}\phi(-d_2)\right] \times \frac{1}{100}$$

  - Strike Sensitivity

$$SS_c = \frac{c - S\Delta_c}{K}, SS_p = \frac{p - S\Delta_p}{K}$$

  - Intrinsic Value

$$IV_c = \max(S-K, 0), IV_p = \max(K-S, 0)$$

  - Time Value

$$TV_c = c - IV_c, TV_p = p - IV_p$$

  - Zero Volatility

$$ZV_c = \max\left(Se^{(r-q)(T-t)} - K, 0\right)e^{-r(T-t)}, ZV_p = \max\left(K - Se^{(r-q)(T-t)}, 0\right)e^{-r(T-t)}$$

Appendix:

- Theta (instantaneous rate of change)

$$\theta_c = -e^{-q(T-t)}\frac{S\phi(d_1)\sigma}{2\sqrt{T-t}} - rKe^{-r(T-t)}\Phi(d_2) + qSe^{-q(T-t)}\Phi(d_1)$$

$$\theta_p = -e^{-q(T-t)}\frac{S\phi(d_1)\sigma}{2\sqrt{T-t}} + rKe^{-r(T-t)}\Phi(-d_2) - qSe^{-q(T-t)}\Phi(-d_1)$$

$\tau = T - t$

Algorithm (Binomial Tree):

Given $\boldsymbol{S}, \boldsymbol{K}, \boldsymbol{r}, \boldsymbol{q}, \boldsymbol{t}, \boldsymbol{T}, \boldsymbol{\sigma}$ and $\boldsymbol{N}$ (1 day=1/365 years)

1. $\Delta t = \frac{T-t}{N}, u = e^{\sigma\sqrt{\Delta t}}, d = \frac{1}{u}, p = \frac{e^{(r-q)\Delta t} - d}{u-d}$
2. $f_{N,j} = \max\left(0, Su^j d^{N-j} - K\right)$ (Call), $f_{N,j} = \max\left(0, K - Su^j d^{N-j}\right)$ (Put), for $j = 0, 1, \ldots, N$
3. $f_{i,j} = e^{-r\Delta t}\left[p \cdot f_{i+1,j+1} + (1-p) \cdot f_{i+1,j}\right]$ for $i = N-1, N-2, \ldots, 1, 0; j = 0, 1, \ldots, i$
4. Option (call/put) price: $V = f_{0,0}$
5. Option Greeks (*Delta and *Gamma):

$$\Delta = \frac{f_{1,1} - f_{1,0}}{S \cdot u - S \cdot d}, \Gamma = \frac{d\left(f_{2,2} - f_{2,1}\right) - u\left(f_{2,1} - f_{2,0}\right)}{S^2(u-d)^2}$$

~~Delta: $\Delta = \frac{\partial V}{\partial S} \approx \frac{V(S+\Delta S) - V(S)}{\Delta S}$ or $\frac{V(S+\Delta S) - V(S-\Delta S)}{2\Delta S}$~~

Delta 100's: $\Delta_{100} = 100\Delta$

Lambda: $\lambda = \Delta \times \frac{S}{V}$

~~Gamma: $\Gamma = \frac{\partial^2 V}{\partial S^2} \approx \frac{V(S+\Delta S) - 2V(S) + V(S-\Delta S)}{(\Delta S)^2}$~~

Gamma 1%: $= \frac{\Delta(1.01S) - \Delta(0.99S)}{2}$

Theta 1 day: $\theta_1 = V\left(\tau - \frac{1}{365}\right) - V(\tau)$

Theta 7 days: $\theta_7 = V\left(\tau - \frac{7}{365}\right) - V(\tau)$

Vega: $\nu = \frac{\partial V}{\partial \sigma} \approx \frac{V(\sigma+\Delta\sigma) - V(\sigma)}{\Delta\sigma}$ or $\frac{V(\sigma+\Delta\sigma) - V(\sigma-\Delta\sigma)}{2\Delta\sigma}$

Rho: $\rho = \frac{\partial V}{\partial r} \approx \frac{V(r+\Delta r) - V(r)}{\Delta r}$ or $\frac{V(r+\Delta r) - V(r-\Delta r)}{2\Delta r}$

Psi: $\Psi = \frac{\partial V}{\partial q} \approx \frac{V(q+\Delta q) - V(q)}{\Delta q}$ or $\frac{V(q+\Delta q) - V(q-\Delta q)}{2\Delta q}$

Strike Sensitivity (SS): $\frac{V - S\Delta}{K}$

Intrinsic Value (IV):
$\max(S-K, 0)$,
$\max(K-S, 0)$

Time Value: $V - IV$

Zero Volatility:
$e^{-r\tau}\max\left(Se^{(r-q)\tau} - K, 0\right)$
$e^{-r\tau}\max\left(K - Se^{(r-q)\tau}, 0\right)$

$\tau = T - t$

Algorithm (Monte Carlo Simulation):

Given $\boldsymbol{S}, \boldsymbol{K}, \boldsymbol{r}, \boldsymbol{q}, \boldsymbol{t}, \boldsymbol{T}, \boldsymbol{\sigma}$ and $\boldsymbol{N}$ (1 day=1/365 years)

$$S_T = Se^{\left(r-q-\frac{1}{2}\sigma^2\right)(T-t)+\sigma\sqrt{(T-t)}z}$$

where z is a random variable following the standard normal distribution.

1. Draw N random numbers $z(i), i = 1, 2, \dots, N$, from the standard normal distribution.
2. Calculate $S_T(i)$ using the above equation for each $z = z(i)$.
3. Calculate the option's value at maturity as

   Call option: $h_T(i) = \max(S_T(i) - K, 0)$

   or Put option: $h_T(i) = \max(K - S_T(i), 0)$
4. Estimate the option's <u>price/value</u> using:

$$V = e^{-r(T-t)} \frac{1}{N} \sum_{i=1}^{N} h_T(i)$$

$\tau = T - t$

Delta: $\Delta = \frac{\partial V}{\partial S} \approx \frac{V(S+\Delta S)-V(S)}{\Delta S}$ or $\frac{V(S+\Delta S)-V(S-\Delta S)}{2\Delta S}$

Delta 100's: $\Delta_{100} = 100\Delta$

Lambda: $\lambda = \Delta \times \frac{S}{V}$

Gamma: $\Gamma = \frac{\partial^2 V}{\partial S^2} \approx \frac{V(S+\Delta S)-2V(S)+V(S-\Delta S)}{(\Delta S)^2}$

Gamma 1%: $= \frac{\Delta(1.01S)-\Delta(0.99S)}{2}$

Theta 1 day: $\theta_1 = V\left(\tau - \frac{1}{365}\right) - V(\tau)$

Theta 7 days: $\theta_7 = V\left(\tau - \frac{7}{365}\right) - V(\tau)$

Vega: $\nu = \frac{\partial V}{\partial \sigma} \approx \frac{V(\sigma+\Delta\sigma)-V(\sigma)}{\Delta\sigma}$ or $\frac{V(\sigma+\Delta\sigma)-V(\sigma-\Delta\sigma)}{2\Delta\sigma}$

Rho: $\rho = \frac{\partial V}{\partial r} \approx \frac{V(r+\Delta r)-V(r)}{\Delta r}$ or $\frac{V(r+\Delta r)-V(r-\Delta r)}{2\Delta r}$

Psi: $\Psi = \frac{\partial V}{\partial q} \approx \frac{V(q+\Delta q)-V(q)}{\Delta q}$ or $\frac{V(q+\Delta q)-V(q-\Delta q)}{2\Delta q}$

Strike Sensitivity (SS): $\frac{V-S\Delta}{K}$

Intrinsic Value (IV):

$\max(S-K, 0)$,

$\max(K-S, 0)$

Time Value: $V - IV$

Zero Volatility:

$e^{-r\tau}\max\left(Se^{(r-q)\tau} - K, 0\right)$

$e^{-r\tau}\max\left(K - Se^{(r-q)\tau}, 0\right)$

Application 4a: Option Price (and Greeks) Calculator (★★☆☆☆)

(Using Black-Scholes Formula)

Hint: About 15+60 lines

1. ☐ Literals (Numbers and Strings)
2. ☐ Arithmetic Operators
3. ☐ Variables
4. ☐ Assignment Statements
5. ☐ Function Definitions
6. ☐ `return` Statement
7. ☐ `scipy.stats.norm` Functions `cdf`, `pdf`
8. ☐ `numpy` Functions `sqrt`, `exp`, `log`

- ☐ List or Tuple Indexing and Slicing
- ☐ (optional) `pandas.DataFrame`
- ☐ (optional) Built-in Function `print`

Application 4b: Option Price (and Greeks) Calculator (★★★☆☆)

(Using Binomial Tree)

Hint: About 40+70 lines

1. ☐ Literals (numbers and strings)
2. ☐ Arithmetic Operators
3. ☐ Variables
4. ☐ Assignment Statements
5. ☐ (Nested) List Comprehension
6. ☐ List or Tuple Indexing and Slicing
7. ☐ (Nested) `for` Statements
8. ☐ (Recursive) Functions
9. ☐ Built-in Functions `isinstance`, `range`, `int`, `max`
10. ☐ `return` Statement
11. ☐ `if` Statements
12. ☐ `numpy` Functions `sqrt`, `exp`

- ☐ (optional) `pandas.DataFrame`

Application 4c: Option Price (and Greeks) Calculator (★★☆☆☆)

(Using Monte Carlo Simulation)

Hint: About 10+70 lines

1. ☐ Literals (Numbers and Strings)
2. ☐ Arithmetic Operators
3. ☐ Variables
4. ☐ Assignment Statements
5. ☐ Functions
6. ☐ `if` Statements
7. ☐ `return` Statement
8. ☐ `numpy` Functions `sqrt`, `exp`, `maximum`, `mean`, `random.seed`, `random.standard_normal`, `append`

❖ ☐ (optional) `pandas.DataFrame`

Application 4d: Option Price (and Greeks) Calculator GUI (★★☆☆☆)

✓ ☐ Qt Designer + PyQt5 Template (Label, Push Button, Line Edit, Date Edit, Combo Box)

a) Functions

b) Modules

c) Classes

✓ ☐ `datetime` Attribute `date.days`

✓ ☐ Built-in Function `float`, `format`

✓ ☐ `pandas.DataFrame` Indexing

Hint: About 150 lines

Application 4e: Option Price (and Greeks) Calculator (★★★★☆)

(Using 3 Numerical Solutions to the Black-Scholes PDE with Continuous Dividend)

$$V_t + \frac{1}{2}\sigma^2 S^2 V_{SS} + (r - q)SV_S - rV = 0$$

Explicit Method:

Given $S,\ K,\ r,\ q,\ t=0, T,\ \sigma, M$ and N. (Assume $S<2K$.)

1. Compute $\Delta t=\frac{T}{N}$, $\Delta S=\frac{S_{\max}}{M}$, where $S_{\max}=2K$.
2. Compute (Call option) $f_{N,j}=\max(j(\Delta S)-K,0)$, (Put option) $f_{N,j}=\max(K-j(\Delta S),0)$, for $j=0,1,\dots,M$.
3. For $i=N-1,\ N-2,\dots,1,0$, repeat 3.1 and 3.2.

 3.1. Compute vector $\widehat{\boldsymbol{F}}_{\boldsymbol{i}}=\boldsymbol{A}\cdot\boldsymbol{F}_{\boldsymbol{i+1}}$, where

$$\widehat{F}_i=\begin{pmatrix}\hat{f}_{i,0}\\ \hat{f}_{i,1}\\ \vdots\\ \hat{f}_{i,M-1}\\ \hat{f}_{i,M}\end{pmatrix},\ \boldsymbol{F}_{\boldsymbol{i+1}}=\begin{pmatrix}f_{i+1,0}\\ f_{i+1,1}\\ \vdots\\ f_{i+1,M-1}\\ f_{i+1,M}\end{pmatrix},\ \boldsymbol{A}=\begin{pmatrix}1 & 0 & 0 & 0 & & \cdots & & \\ a_1^* & b_1^* & c_1^* & 0 & & & & \\ 0 & a_2^* & b_2^* & c_2^* & & & & \\ \vdots & & \ddots & \ddots & \ddots & & & \vdots\\ \vdots & & & \ddots & \ddots & \ddots & & \vdots\\ & & & & a_{M-2}^* & b_{M-2}^* & c_{M-2}^* & 0\\ & & & & 0 & a_{M-1}^* & b_{M-1}^* & c_{M-1}^*\\ & & \cdots & & 0 & 0 & 0 & 1\end{pmatrix}\ \text{with}\ \begin{cases}a_j^*=\frac{1}{2}\cdot(\Delta t)\cdot[\sigma^2\cdot j^2-(r-q)\cdot j]\\ b_j^*=1-(\Delta t)\cdot[\sigma^2\cdot j^2+r]\\ c_j^*=\frac{1}{2}\cdot(\Delta t)\cdot[\sigma^2\cdot j^2+(r-q)\cdot j]\end{cases},\text{for } j=1,\dots,M-1.$$

 3.2. Compute vector $\boldsymbol{F}_{\boldsymbol{i}}$,

$$\boldsymbol{F}_{\boldsymbol{i}}=\begin{pmatrix}f_{i,0}\\ f_{i,1}\\ \vdots\\ f_{i,M-1}\\ f_{i,M}\end{pmatrix}\ \text{with}\ f_{i,j}=\begin{cases}0 & ,\text{ for } j=0\\ \hat{f}_{i,j} & ,\text{ for } j=1,2,\dots,M-1\\ S_{\max}-Ke^{-r(N-i)(\Delta t)} & ,\text{ for } j=M\end{cases}\ (\text{Call});\ f_{i,j}=\begin{cases}Ke^{-r(N-i)(\Delta t)} & ,\text{ for } j=0\\ \hat{f}_{i,j} & ,\text{ for } j=1,2,\dots,M-1\\ 0 & ,\text{ for } j=M\end{cases}\ (\text{Put}).$$

4. Find k, such that $k(\Delta S)\le S<(k+1)(\Delta S)$, i.e. $k=\left\lfloor\frac{S}{\Delta S}\right\rfloor$. ($\lfloor\cdot\rfloor$ represents the floor of $\cdot$.)
5. Option price: $V=f_{0,k}+\frac{f_{0,k+1}-f_{0,k}}{\Delta S}[S-k(\Delta S)]$.

Implicit Method:

Given $S,\ K,\ r,\ q,\ t=0, T,\ \sigma, M$ and N. (Assume $S < 2K$.)

1. Compute $\Delta t = \frac{T}{N}$, $\Delta S = \frac{S_{\max}}{M}$, where $S_{\max} = 2K$.
2. Compute (Call option) $f_{N,j} = \max(j(\Delta S) - K, 0)$, (Put option) $f_{N,j} = \max(K - j(\Delta S), 0)$, for $j = 0,1,\ldots,M$.
3. For $i = N-1,\ N-2,\ldots,1,0$, repeat 3.1 and 3.2.

   3.1. Compute vector $\widehat{\boldsymbol{F}}_{i+1}$,

$$\widehat{\boldsymbol{F}}_{i+1} = \begin{pmatrix} \hat{f}_{i+1,0} \\ \hat{f}_{i+1,1} \\ \vdots \\ \hat{f}_{i+1,M-1} \\ \hat{f}_{i+1,M} \end{pmatrix} \text{ with } \hat{f}_{i+1,j} = \begin{cases} 0 & ,\text{ for } j=0 \\ f_{i+1,j} & ,\text{ for } j = 1,2,\ldots,M-1 \\ S_{\max} - Ke^{-r(N-i)(\Delta t)} & ,\text{ for } j = M \end{cases} \text{ (Call); } \hat{f}_{i+1,j} = \begin{cases} Ke^{-r(N-i)(\Delta t)} & ,\text{ for } j=0 \\ f_{i+1,j} & ,\text{ for } j = 1,2,\ldots,M-1 \\ 0 & ,\text{ for } j = M \end{cases} \text{ (Put).}$$

   3.2. Compute vector $\boldsymbol{F}_i = \boldsymbol{A}^{-1} \cdot \widehat{\boldsymbol{F}}_{i+1}$, where

$$\boldsymbol{F}_i = \begin{pmatrix} f_{i,0} \\ f_{i,1} \\ \vdots \\ f_{i,M-1} \\ f_{i,M} \end{pmatrix}, \boldsymbol{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & & \cdots & & & \\ a_1 & b_1 & c_1 & 0 & & & & & \\ 0 & a_2 & b_2 & c_2 & & & & & \\ \vdots & & \ddots & \ddots & \ddots & & & & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & & & \vdots \\ & & & & a_{M-2} & b_{M-2} & c_{M-2} & 0 \\ & & & & 0 & a_{M-1} & b_{M-1} & c_{M-1} \\ & & \cdots & & 0 & 0 & 0 & 1 \end{pmatrix} \text{ with } \begin{cases} a_j = \frac{1}{2} \cdot (\Delta t) \cdot [(r-q) \cdot j - \sigma^2 \cdot j^2] \\ b_j = 1 + (\Delta t) \cdot (\sigma^2 \cdot j^2 + r) \\ c_j = -\frac{1}{2} \cdot (\Delta t) \cdot [\sigma^2 \cdot j^2 + (r-q) \cdot j] \end{cases} ,\text{for } j = 1,\ldots,M-1.$$

4. Find k, such that $k(\Delta S) \leq S < (k+1)(\Delta S)$, i.e. $k = \left\lfloor \frac{S}{\Delta S} \right\rfloor$. ($\lfloor \cdot \rfloor$ represents the floor of $\cdot$.)
5. Option price: $V = f_{0,k} + \frac{f_{0,k+1} - f_{0,k}}{\Delta S}[S - k(\Delta S)]$.

Crank-Nicolson Method:

Given S, K, r, q, $t=0$, T, σ, M and N. (Assume $S<2K$.)

1. Compute $\Delta t=\frac{T}{N}$, $\Delta S=\frac{S_{\max}}{M}$, where $S_{\max}=2K$.
2. Compute (Call option) $f_{N,j}=\max(j(\Delta S)-K,0)$, (Put option) $f_{N,j}=\max(K-j(\Delta S),0)$, for $j=0,1,\dots,M$.
3. For $i=N-1,\ N-2,\dots,1,0$, repeat 3.1, 3.2 and 3.3.

3.1. Compute vector $\widehat{\boldsymbol{b}}=\boldsymbol{M_2}\cdot\boldsymbol{F_{i+1}}$, where

$$\widehat{\boldsymbol{b}}=\begin{pmatrix}\hat{b}_0\\ \hat{b}_1\\ \vdots\\ \hat{b}_{M-1}\\ \hat{b}_M\end{pmatrix},\ \boldsymbol{F_{i+1}}=\begin{pmatrix}f_{i+1,0}\\ f_{i+1,1}\\ \vdots\\ f_{i+1,M-1}\\ f_{i+1,M}\end{pmatrix},\ \boldsymbol{M_2}=\begin{pmatrix}1 & 0 & 0 & 0 & & \cdots & & \\ \alpha_1 & 1+\beta_1 & \gamma_1 & 0 & & & & \\ 0 & \alpha_2 & 1+\beta_2 & \gamma_2 & & & & \\ \vdots & & \ddots & \ddots & \ddots & & & \vdots\\ \vdots & & & \ddots & \ddots & \ddots & & \vdots\\ & & & & \alpha_{M-2} & 1+\beta_{M-2} & \gamma_{M-2} & 0\\ & & & & 0 & \alpha_{M-1} & 1+\beta_{M-1} & \gamma_{M-1}\\ & & \cdots & & 0 & 0 & 0 & 1\end{pmatrix}\ \text{with}\ \begin{aligned}\alpha_j&=\frac{1}{4}\cdot(\Delta t)\cdot[\sigma^2\cdot j^2-(r-q)\cdot j]\\ \beta_j&=-\frac{1}{2}(\Delta t)\cdot(\sigma^2\cdot j^2+r)\\ \gamma_j&=\frac{1}{4}\cdot(\Delta t)\cdot[\sigma^2\cdot j^2+(r-q)\cdot j]\end{aligned}\quad,\text{for } j=1,\dots,M-1.$$

3.2. Compute vector $\boldsymbol{b}$,

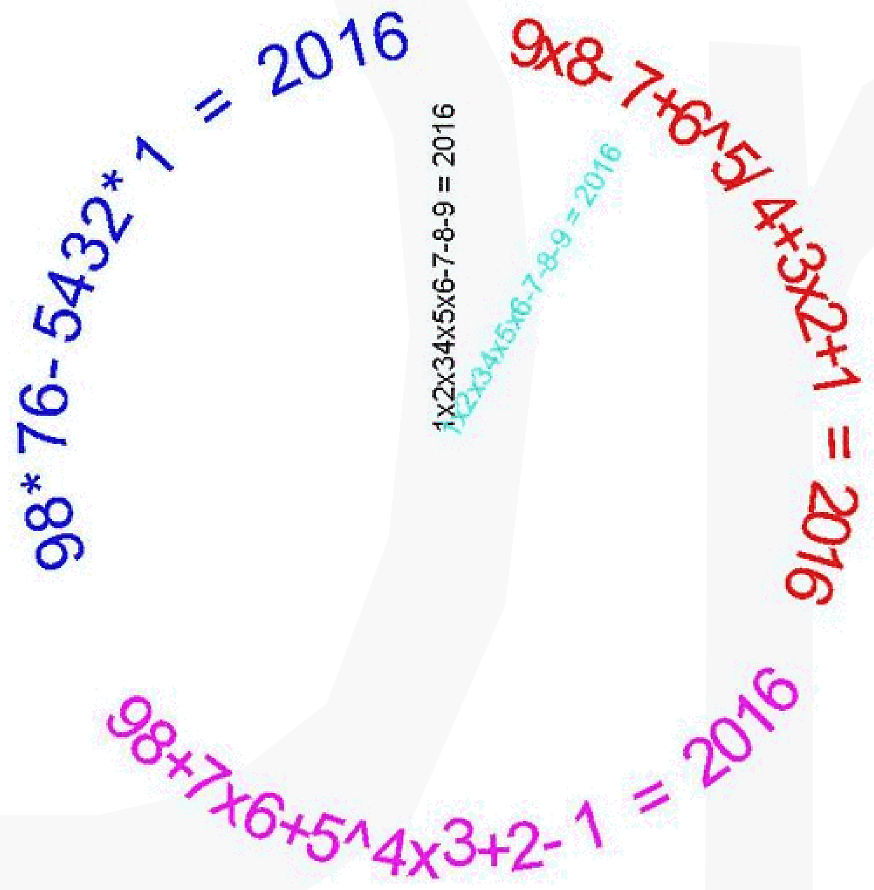
$$\boldsymbol{b}=\begin{pmatrix}b_0\\ b_1\\ \vdots\\ b_{M-1}\\ b_M\end{pmatrix}\ \text{with}\ b_j=\begin{cases}0 & ,\text{ for } j=0\\ \hat{b}_j & ,\text{ for } j=1,2,\dots,M-1\\ S_{\max}-Ke^{-r(N-i)(\Delta t)} & ,\text{ for } j=M\end{cases}\text{(Call)};\ b_j=\begin{cases}Ke^{-r(N-i)(\Delta t)} & ,\text{ for } j=0\\ \hat{b}_j & ,\text{ for } j=1,2,\dots,M-1\\ 0 & ,\text{ for } j=M\end{cases}\text{(Put)}.$$

3.3. Compute vector $\boldsymbol{F_i}=(\boldsymbol{M_1})^{-1}\cdot\boldsymbol{b}$, where

$$\boldsymbol{F_i}=\begin{pmatrix}f_{i,0}\\ f_{i,1}\\ \vdots\\ f_{i,M-1}\\ f_{i,M}\end{pmatrix},\ \boldsymbol{M_1}=\begin{pmatrix}1 & 0 & 0 & 0 & & \cdots & & \\ -\alpha_1 & 1-\beta_1 & -\gamma_1 & 0 & & & & \\ 0 & -\alpha_2 & 1-\beta_2 & -\gamma_2 & & & & \\ \vdots & & \ddots & \ddots & \ddots & & & \vdots\\ \vdots & & & \ddots & \ddots & \ddots & & \vdots\\ & & & & -\alpha_{M-2} & 1-\beta_{M-2} & -\gamma_{M-2} & 0\\ & & & & 0 & -\alpha_{M-1} & 1-\beta_{M-1} & -\gamma_{M-1}\\ & & \cdots & & 0 & 0 & 0 & 1\end{pmatrix}.$$

4. Find k, such that $k(\Delta S)\le S<(k+1)(\Delta S)$, i.e. $k=\left\lfloor\frac{S}{\Delta S}\right\rfloor$.
5. Option price: $f_{0,k}+\frac{f_{0,k+1}-f_{0,k}}{\Delta S}[S-k(\Delta S)]$.

Fun Application 1a: Pandigital Formula For New Year – 1



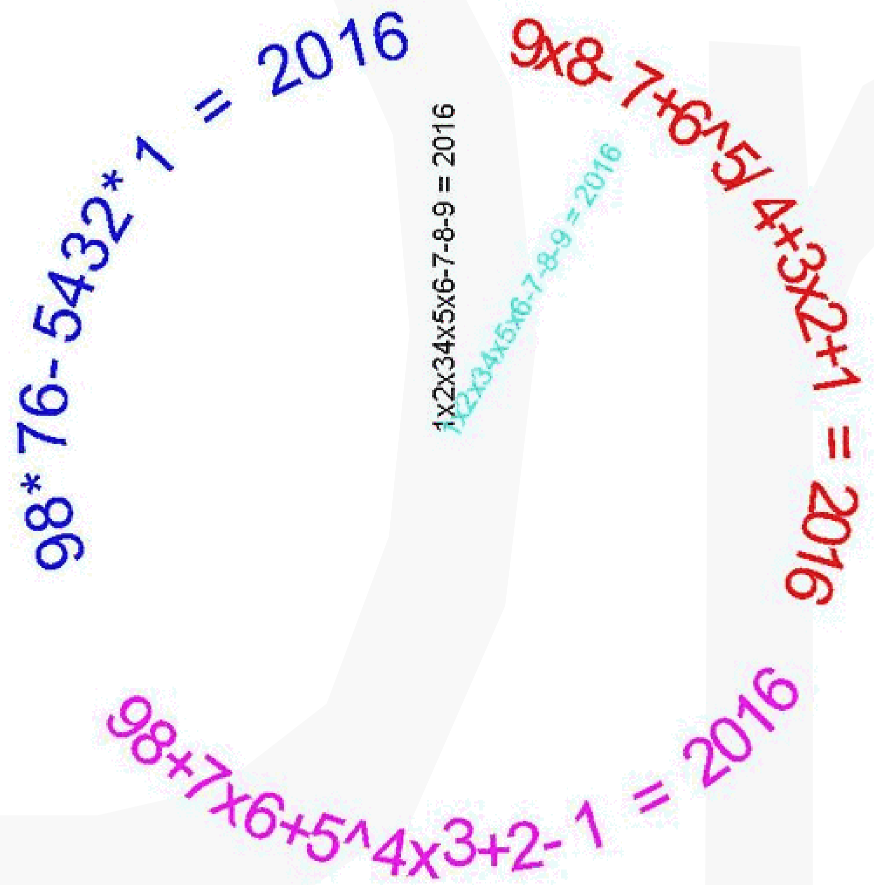
Insert +, -, * or / in “123456789” to obtain 2016.

Fun Application 1a: Pandigital Formula For New Year – 1 (★⯪☆☆☆)

1. ☐ Numeric Literals
2. ☐ Lists
3. ☐ Arithmetic Operators
4. ☐ Comparison (Relational) Operators
5. ☐ Variables
6. ☐ String Concatenation
7. ☐ Assignment Statements
8. ☐ `if` Statements
9. ☐ `for` Statements
10. ☐ Built-in Functions `eval`, `print`
11. ☐ (optional) Error and Exceptions

(brute force solution)
Hint: About 30 lines

Fun Application 1b: Pandigital Formula For New Year – 2



Insert +, -, *, / or ^ in “123456789” to obtain 2016. (Note that Python does not have ^ operator.)

Additional Requirement:

2^3^4: $(2^3)^4$

Fun Application 1b: Pandigital Formula For New Year – 2 (★★★☆☆)

1. ☐ Numeric Literals
2. ☐ Lists
3. ☐ Arithmetic Operators
4. ☐ Comparison (Relational) Operators
5. ☐ Variables
6. ☐ String Concatenation
7. ☐ String Function `rfind`
8. ☐ Assignment Statements
9. ☐ `if` Statements
10. ☐ `for` Statements
11. ☐ Built-in Functions `power`, `eval`, `print`
12. ☐ Recursive Functions
13. ☐ (optional) Error and Exceptions

(brute force solution)

Hint: About 50 lines

Fun Application 2: Sudoku Solving

Fun Application 2: Sudoku Solving (★★★☆☆)

1. ☐ Variables
2. ☐ Assignment Statements
3. ☐ Numeric and String Literals
4. ☐ Arithmetic Operators
5. ☐ Comparison (Relational) Operators
6. ☐ Logical Operators
7. ☐ Strings (Indexing and Slicing, Function `find`)
8. ☐ Recursive Function
9. ☐ Set Comprehension (or `for` loop)
10. ☐ `for` Statements
11. ☐ `if` Statements
12. ☐ Built-in Functions `range`, `print`
13. ☐ Errors and Exceptions
14. ☐ Class
15. ☐ `pass` statement

(brute force solution)
Hint: About 25 lines