

ESIREM

4A INFOTRONIQUE MODULE ITIS43

BRICE NOFIELE

Mini-Projet Salesforce : Sujet 2

Vincent CANDAPPANE

Orann WEBER

30 mars 2018



Table des matières

Table des figures	1
1 Introduction	2
2 Travail demandé	2
3 Création de l'objet personnalisé : "BlackScholes"	3
4 Création de l'application (App) : "Price Action"	4
5 Vérification	4
6 Le formulaire par défaut de Salesforce	5
7 Le formulaire Visualforce	6
8 La Classe Apex : "BlackScholesUpdate"	8
9 Le Trigger Apex : "BlackScholesTrigger"	9
10 Conversion de la date	9
11 Piste d'avancement et choix de méthode	10
12 Fonction de répartition de la loi normale	10
13 Conclusion	12
Sitographie	13

Table des figures

1	Création de l'objet "BlackScholes"	3
2	Fonctionnalité à cocher	3
3	Gestionnaire des applications	4
4	Générateur d'application Lightning	4
5	Formulaire de base généré par Salesforce	5
6	Visualisations des enregistrements	5
7	Fonctionnement de l'infrastructure Visualforce	6
8	Connexion de la page Visualforce vers l'application	6
9	Mini-onglet Visualforce	7
10	Formulaire via l'interface VisualForce "PriceActionForm"	7
11	Conversion date	9
12	Affichage final	11

1 Introduction

Dans le cadre de nos travaux pratiques de Cloud Computing au sein de l'ESIREM, nous allons appliquer et résumer les connaissances acquises au cours du module ITIS43. Ce rapport retrace notre démarche et l'utilisation d'un nouvel environnement Salesforce illustrant les enjeux et les bénéfices du Cloud Computing dans le système d'information.

Salesforce est leader du CRM et le premier cloud d'entreprise. En 1999, Salesforce est né d'une vision : celle de réinventer la gestion de la relation client(CRM). La Customer Success Platform aide les clients à être plus efficaces dans tous leurs services(ventes, service client, marketing...). Tous leurs produits sont 100% cloud, tout est en ligne : pas de matériel, pas de logiciel. Ainsi, comparé à une infrastructure Legacy on-premises, il n'y a aucun coût de configuration, aucune maintenance et les clients peuvent travailler depuis tous les terminaux avec une simple connexion Internet.

Salesforce possède plusieurs plateformes client dont :

- Sales Cloud : Outil de gestion de la CRM et automatisation
- Service Cloud : Service client et gestion de requêtes
- Marketing Cloud : Création de campagnes personnalisées et multicanaux
- Analytics Cloud : Logiciel de veille stratégique pour exploration de données
- Community Cloud : Communautés en ligne et en libre-service
- App Cloud : Communautés de développement d'application

Afin de simuler et nous familiariser à cet environnement, nous allons modéliser un actif financier de type « Option sur Action » sous Salesforce.

2 Travail demandé

Nous souhaitons modéliser un actif financier de type « Option sur Action ». Pour cela on souhaite réaliser une application qui sera appelée « Price Action »

La formule de Black-Scholes peut être utilisée pour évaluer une option européenne ne payant pas de dividendes. Pour une option d'achat (call) et une option de vente (put), les formules sont les suivantes, avec :

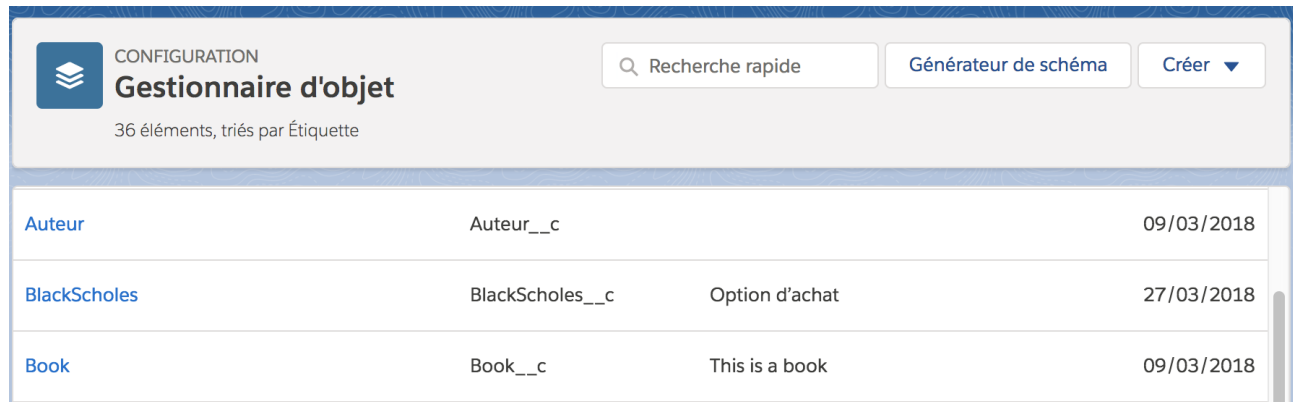
- S = Prix de l'action
- K = Strike de l'option ou « Prix d'exercice »
- r = taux sans risque
- T = Maturité de l'option (en année)
- σ = volatilité implicite du sous-jacent
- $N(x)$ = Fonction de répartition de la loi normale

À partir d'un formulaire Visualforce, l'utilisateur entre les paramètres de l'option. Après validation par l'utilisateur le résultat en sortie est le calcul du Call et du Put.

Afin de modéliser cela sous Salesforce, nous allons procéder par étapes. Nous allons tout d'abord créer un objet personnalisé nommé "BlackScholes". Ensuite nous créerons notre application "Price Action". Enfin nous intégrerons une classe Apex "BlackScholesUpdate" et un trigger "BlackScholesTrigger " directement manipulable via un formulaire Visualforce..

3 Création de l'objet personnalisé : "BlackScholes"

L'élément BlackScholes s'agit d'un objet (custom object) créé à partir de l'onglet "Gestionnaire d'objet", jusqu'à présent vide.



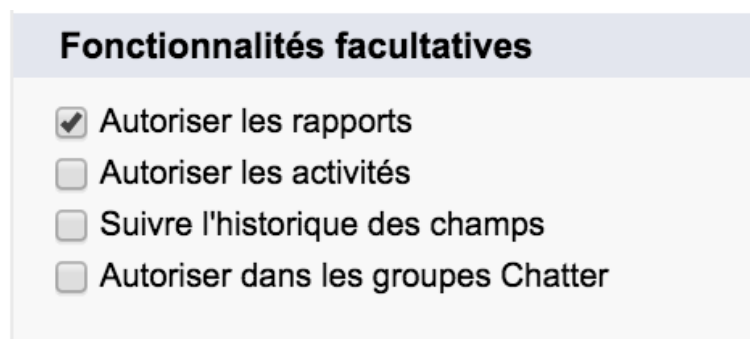
Auteur	Auteur__c		09/03/2018
BlackScholes	BlackScholes__c	Option d'achat	27/03/2018
Book	Book__c	This is a book	09/03/2018

FIGURE 1 – Création de l'objet "BlackScholes"

Afin de l'alimenter en données, nous allons compte tenu de la mission demandée, créer 13 champs (field) associés à l'objet "BlackScholes" avec d'un côté les entrées et de l'autre les sortie :

- *Entree* : « Nom de l'actif financier », « S », « K », « r », « T »
- *Sortie* : « T (en années) », « σ », « d1 », « d2 », « Call », « Put »

En dernière étape, après finalisation de création de l'objet, il est important d'activer la fonctionnalité « Autoriser les rapports » afin de le rendre accessible lors de la création d'une application.



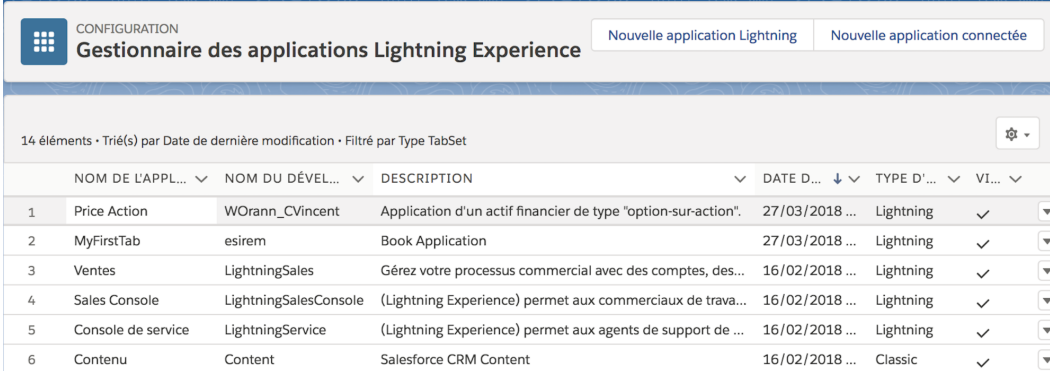
Fonctionnalités facultatives

- ☒ Autoriser les rapports
- ☐ Autoriser les activités
- ☐ Suivre l'historique des champs
- ☐ Autoriser dans les groupes Chatter

FIGURE 2 – Fonctionnalité à cocher

4 Création de l'application (App) : "Price Action"

Se positionner dans la hiérarchie « Configuration », et enfin rechercher « Gestionnaire des applications », en cliquant sur nouveau l'on pourra créer notre application de type Lightning avec une page visualforce intégrée où l'utilisateur pourra entrer les paramètres de l'option.



	NOM DE L'APPL...	NOM DU DÉVEL...	DESCRIPTION	DATE D...	TYPE D'...	VI...
1	Price Action	WOrann_CVincent	Application d'un actif financier de type "option-sur-action".	27/03/2018 ...	Lightning	✓
2	MyFirstTab	esirem	Book Application	27/03/2018 ...	Lightning	✓
3	Ventes	LightningSales	Gérez votre processus commercial avec des comptes, des...	16/02/2018 ...	Lightning	✓
4	Sales Console	LightningSalesConsole	(Lightning Experience) permet aux commerciaux de trava...	16/02/2018 ...	Lightning	✓
5	Console de service	LightningService	(Lightning Experience) permet aux agents de support de ...	16/02/2018 ...	Lightning	✓
6	Contenu	Content	Salesforce CRM Content	16/02/2018 ...	Classic	✓

FIGURE 3 – Gestionnaire des applications

Nous associons à l'App, l'objet "BlackScholes" qui a été créé précédemment.

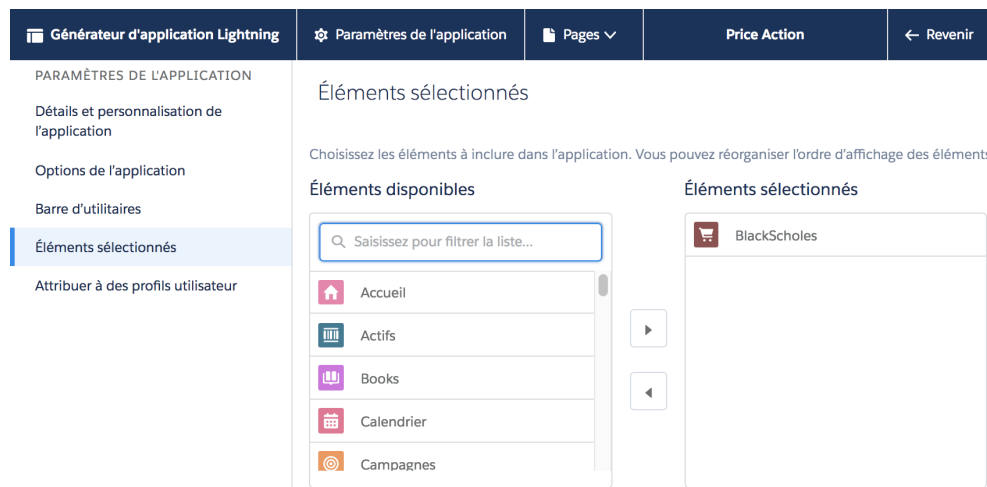


FIGURE 4 – Générateur d'application Lightning

5 Vérification

Pour le bon déroulement du développement de notre application, il est nécessaire de réaliser les vérifications adéquates. Pour cela, il faut se diriger vers la liste des Apps et sélectionner « Price Action ». Nous créons ensuite un ou deux enregistrements afin de détecter les éventuelles erreurs et la cohérence entre les différents typages utilisés pour chaque variable.

6 Le formulaire par défaut de Salesforce

À cette étape l'objet BlackScholes est intégré à notre application et les typages des différents champs ont été mis à jour. Certains champs seront suivis d'une étoile « * » indiquant que les paramètres doivent être rentrés obligatoirement afin de générer le résultat attendu, voici donc le formulaire généré de visualforce :

The screenshot shows a Salesforce form titled 'BlackScholes Action financier n°1'. The form is divided into two main sections: 'ASSOCIÉ' and 'ACTIVITÉ'. The 'ASSOCIÉ' section contains fields for 'Nom de l'actif financier' (Action financier n°1), 'S' (80,00), 'K' (90,00), 'r' (0,05), 'T' (28/06/2018), 'T (en année)' (0,35), 'sigma' (0,35), 'd1', 'd2', 'N(d1)', 'N(d2)', 'call', and 'put'. The 'ACTIVITÉ' section includes buttons for 'Nouvel é...', 'Nouvelle ...', 'Consigne...', and 'Plus', a 'Configurer un événement...' button, and a 'Chronologie des activi...' section. The form also displays 'Créé par Cdpn Vince, 28/03/2018 02:41' and 'Dernière modification par Cdpn Vince, 31/03/2018 19:45'. At the bottom, there are 'Annuler' and 'Enregistrer' buttons.

FIGURE 5 – Formulaire de base généré par Salesforce

Chaque enregistrement est stocké et trié par ordre de création :

The screenshot shows a Salesforce table titled 'BlackScholes Visualisations récentes'. The table has 7 elements and is sorted by 'Nom de l'actif financier'. The table columns are 'NOM DE L'ACTIF FINANCIER' and 'Action financier n°1'. The table contains the following data:

	NOM DE L'ACTIF FINANCIER	Action financier n°1
1	Action financier n°1	
2	Action financier n°2	
3	Action financier n°3	
4	Action financier n°4	
5	Action financier n°5	
6	Action financier n°6	
7	Action financier n°7	

FIGURE 6 – Visualisations des enregistrements

7 Le formulaire Visualforce

Visualforce est une infrastructure de développement Web. Elle permet aux développeurs d'élaborer des interfaces utilisateur sophistiquées et personnalisées pour des applications mobiles et de bureau, qui peuvent être hébergées sur la plate-forme Lightning Platform suivi d'une interface entièrement personnalisable.

Dans Visualforce, le développement d'applications est familier pour toutes les personnes qui ont élaboré des applications Web. Les développeurs créent des pages Visualforce en assemblant des composants, des balises HTML et des éléments de style facultatifs. Visualforce peut intégrer n'importe quelle technologie Web standard ou infrastructure JavaScript pour créer une interface utilisateur plus animée et plus riche. Chaque page est accessible via une URL unique. En voici une représentation rapide de son fonctionnement :

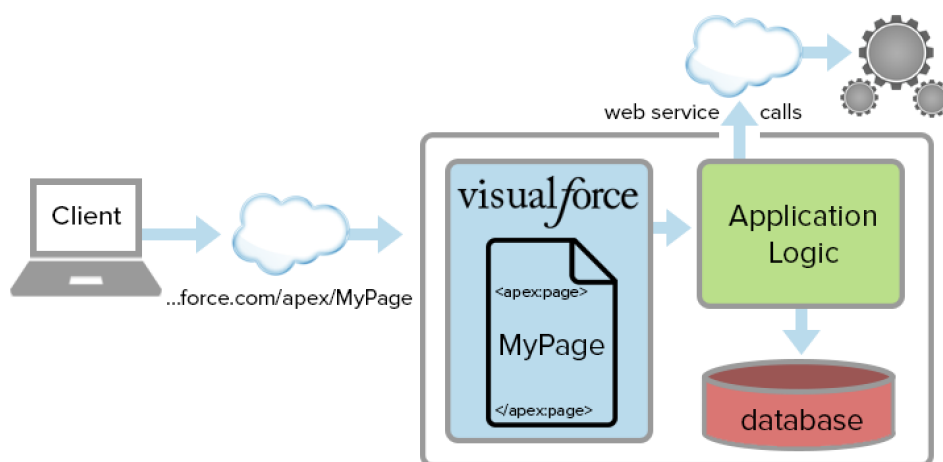


FIGURE 7 – Fonctionnement de l'infrastructure Visualforce

Afin d'affilier la page Visualforce créée au préalable à l'application adéquate, il a fallu passer par le générateur d'application Lightning. Ici, nous réaliserons la connexion entre la page Visualforce « PriceActionForm » et l'application « Price Action » :

CONFIGURATION
Pages Visualforce

Pages Visualforce [Aide sur cette page](#)

Les pages Visualforce fournissent un mécanisme fiable et simple d'utilisation pour créer de nouvelles expériences utilisateur attrayantes pour votre application ou pour améliorer les applications existantes et optimiser la productivité des utilisateurs.

Afficher :

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Autres

Action	Étiquette	Nom	Préfixe d'espace de noms	Version API	Description	Créé par alias	Date de création	Dernière modification par alias	Date de dernière modification
Modifier Suppr. Sécurité 🔗	PriceActionForm	PriceActionForm		42.0	Formulaire pour un actif financier de type « Option sur Action »	CVinc	31/03/2018 19:26	CVinc	31/03/2018 23:17
Modifier Suppr. Sécurité 🔗	ContactForm	ContactForm		42.0	Formulaire pour gestion de « Contact »	CVinc	27/03/2018 15:01	CVinc	31/03/2018 23:07

FIGURE 8 – Connexion de la page Visualforce vers l'application

Maintenant que la carcasse du formulaire a été créée, il ne reste plus qu'à l'alimenter :

```
<apex:page standardController="BlackScholes__c" >
<apex:form >
<apex:pageBlock title="Edit Price Action">
<apex:pageBlockSection columns="1">
<apex:inputField value="{!BlackScholes__c.Name}"/>
<apex:inputField value="{!BlackScholes__c.S__c}"/>
<apex:inputField value="{!BlackScholes__c.K__c}"/>
<apex:inputField value="{!BlackScholes__c.T__c}"/>
<apex:inputField value="{!BlackScholes__c.sigma__c}"/>
<apex:outputField value="{!BlackScholes__c.call__c}"/>
<apex:outputField value="{!BlackScholes__c.put__c}"/>
</apex:pageBlockSection>
<apex:pageBlockButtons >
<apex:commandButton action="{!save}" value="Save"/>
</apex:pageBlockButtons>
</apex:pageBlock>
</apex:form>
</apex:page>
```

En bonus, il est aussi possible de configurer le formulaire sous forme de mini-onglet dans la section « Barre d'utilitaires » de l'application de la manière suivante :

Configuration de l'onglet Visualforce à l'application « Price Action »	Enregistrement non-autorisé du formulaire	Enregistrement autorisé du formulaire

FIGURE 9 – Mini-onglet Visualforce

FIGURE 10 – Formulaire via l'interface VisualForce "PriceActionForm"

8 La Classe Apex : "BlackScholesUpdate"

Afin de pouvoir manipuler les variables, nous allons accéder au « Developper console ». Nous créons et ajoutons à la classe Apex : « BlackScholesUpdate », la méthode « calculer(BlackScholes__c blackscholes) » qui réalisera le calcul de d1, d2, N(d1), N(d2), N(-d1), N(-d1), T, call et put.

```
public class BlackScholesUpdate {
    public void calculer(BlackScholes__c blackscholes){
        //Variable
        Decimal sigma = blackscholes.sigma__c;
        Double S = blackscholes.S__c;
        Double K = blackscholes.K__c;
        Double r = blackscholes.r__c;
        Double d1,d2,nd1,nd2,nd1_minus,nd2_minus,call,put;

        //Conversion date(T) en années(entre 0 et 1)
        Date firstDate = Date.today();
        Date secondDate = blackscholes.T__c;
        Double monthsBetween = firstDate.monthsBetween(secondDate);
        Double T = monthsBetween/12;
        blackscholes.T_by_year__c = T;

        //Calcul de d1
        d1 = ((math.log(S/K))+((r+((sigma.pow(2))/2))*T)/(sigma*math.sqrt(T)));
        blackscholes.d1__c = d1;

        //Calcul de d2
        d2 = d1 - (sigma*math.sqrt(T));
        blackscholes.d2__c = d2;

        //Calcul de N(d1)
        nd1 = (1/(math.sqrt(2*math.PI))) * ((math.sqrt(math.PI/2)) * (erf2(d1/math.sqrt(2))+1));
        blackscholes.N_d1__c = nd1;

        //Calcul de N(d2)
        nd2 = (1/(math.sqrt(2*math.PI))) * ((math.sqrt(math.PI/2)) * (erf2(d2/math.sqrt(2))+1));
        blackscholes.N_d2__c = nd2;

        //Calcul de N(-d1)
        nd1_minus=(1/(math.sqrt(2*math.PI))) * ((math.sqrt(math.PI/2)) * (erf2((-d1)/math.sqrt(2))+1));

        //Calcul de N(-d2)
        nd2_minus=(1/(math.sqrt(2*math.PI))) * ((math.sqrt(math.PI/2)) * (erf2((-d2)/math.sqrt(2))+1));

        //Calcul du CALL
        call = (S*nd1)-(K*math.exp(-r*T)*nd2);
        blackscholes.call__c = call;

        //Calcul du PUT
        put = (K*math.exp(-r*T)*(nd2_minus))-(S*(nd1_minus));
        blackscholes.put__c = put;
    }
}
```

9 Le Trigger Apex : "BlackScholesTrigger"

Un trigger est utilisé afin d'appliquer la fonction BlackScholesUpdate à chaque insertion et à chaque modification (mise à jour) d'un nouvel enregistrement.

```
trigger BlackScholesTrigger on BlackScholes__c (before insert,before update) {  
    BlackScholesUpdate BS = new BlackScholesUpdate();  
    for (BlackScholes__c b : Trigger.new) {  
        BS.calculer(b);  
    }  
}
```

10 Conversion de la date

Dans le formulaire l'utilisateur doit pouvoir saisir une date pour la maturité. Pour cela il suffit de récupérer la date que l'utilisateur sélectionne dans le calendrier, puis de compter le nombre de mois entre la date sélectionnée et la date actuelle. Une fois cela calculer, l'on divise le tout par 12, nous obtenons alors la date en année (compris entre 0 et 1).

The image shows a date picker interface. At the top, a date field displays "28/06/2018". Below it is a calendar for the month of "JUIN" in the year "2018". The date "28" is selected and highlighted with a blue circle. Below the calendar, there is a text input field labeled "T (en année)" which contains the value "0,25". A blue curved arrow points from the date "28" in the calendar to the input field, indicating the calculation process.

FIGURE 11 – Conversion date

11 Piste d'avancement et choix de méthode

Il s'agit ici de l'étape la plus fastidieuse de la mission, car il a fallu traduire un calcul purement mathématique en code informatique, or ceci n'est pas toujours facilement implémentable due au problème de limitations de ressources informatiques. Par exemple, pour calculer une intégrale d'une fonction quelconque, il fallait passer par des méthodes personnelles. En effet, il n'existe pas encore une bibliothèque de fonction ni de documentation Apex pour Salesforce pour ce type de calcul. Les cas particuliers pour calculer une intégrale étant nombreux, il était judicieux de construire nos propres solutions. Il a donc fallu trouver une solution de substitution, nous avons trouvé plusieurs techniques :

- Intégration numérique par la méthode de Simpson
- Intégration numérique par la méthode des trapèzes
- Intégration numérique par la méthode de Gauss-Legendre

Cependant, ses techniques de calcul d'intervalle étant limitées à des intervalles d'entier, il n'était pas possible de calculer une limite convergente vers l'infini. Cela nous posant problème, nous avons changé de solution.

12 Fonction de répartition de la loi normale

Nous avons donc raisonné d'une manière différente, étant donné que l'intégrale à calculer est toujours la même, nous avons tout simplement intégré manuellement la fonction de répartition pour des raisons de facilité, mais aussi pour des raisons d'optimisation du temps de calcul au niveau de la machine. Comme vu précédemment, la loi normale est la loi de probabilité dont la fonction de répartition est donnée par la fonction :

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{1}{2}t^2} dt, \text{ pour tout } x \in \mathbb{R}.$$

Il n'existe pas d'expression analytique de la fonction de répartition, c'est-à-dire qu'elle ne s'exprime pas à partir de fonctions usuelles, mais devient elle-même une fonction usuelle. Nous avons donc procédé par étape d'intégrations successives :

- Étape 1, l'intégrale s'exprime en fonction de la fonction d'erreur (erf) :

$$\int_{-\infty}^x e^{-\frac{t^2}{2}} dt = \sqrt{\frac{\pi}{2}} \left(\operatorname{erf}\left(\frac{x}{\sqrt{2}}\right) + 1 \right)$$

- Étape 2, corrélation avec la fonction de répartition :

$$\Phi(x) = \left(\frac{1}{\sqrt{2\pi}} \right) \sqrt{\frac{\pi}{2}} \left(\operatorname{erf}\left(\frac{x}{\sqrt{2}}\right) + 1 \right)$$

- Étape 3, Afin de reproduire le comportement de la formule, il a fallu créer ou chercher une fonction qui calcule la fonction d'erreur erf. Nous utiliserons pour cela les algorithmes de "Abramowitz and Stegun" :


```

// fractional error less than x.xx * 10 ^ -4.
// Algorithm 26.2.17 in Abramowitz and Stegun, Handbook of Mathematical.
public static double erf2(double z) {
    double t = 1.0 / (1.0 + 0.47047 * Math.abs(z));
    double poly = t * (0.3480242 + t * (-0.0958798 + t * (0.7478556)));
    double ans = 1.0 - poly * Math.exp(-z*z);
    if (z >= 0) return ans;
    else return -ans;
}

```


Nous avons maintenant toutes les entrées pré-calculés nécessaires, nous allons donc pouvoir calculer le résultat souhaité : $N(d1)$, $N(d2)$, call, put.


En voici une démonstration similaire à l'exemple présenté dans le TP : soit un call sur Arcelor, une entreprise spécialisée dans le commerce de l'acier, dont les résultats futurs semblent prometteurs. L'action vaut aujourd'hui 80 euros. Soit un call de strike 90 et de maturité un trimestre (3 mois). Les taux d'intérêt sans risque pour cette période sont équivalents à 5%. La volatilité implicite est estimée à 35%. Donc, $S = 80$, $K = 90$, $T = 0,25$, $r = 0,05$ et $\sigma = 0,35$.


Action financier n°1

ASSOCIÉ

DÉTAILS

Nom de l'actif financier	Propriétaire
Action financier n°1	 Cdpm Vince
S	
80,00	
K	
90,00	
r	
0,05	
T	
28/06/2018	
T (en année)	
0,25	
sigma	
0,35	
d1	
-0,51	
d2	
-0,69	
N(d1)	
0,30	
N(d2)	
0,25	
call	
2,48	
put	
11,36	

Créé par
 Cdpm Vince, 28/03/2018 02:41


Dernière modification par
 Cdpm Vince, 31/03/2018 19:45

FIGURE 12 – Affichage final

Après validation par l'utilisateur le résultat en sortie est le calcul du Call et du Put.

13 Conclusion

Ce mini-projet nous a permis d'avoir une approche plus concrète et plus approfondie des notions pratiques abordées en cours. En effet, cela nous a permis de visualiser les enjeux et les bénéfices du Cloud Computing dans le système d'information. La manipulation et la mise en pratique pour une situation concrète de l'architecture applicative Salesforce nous ont formés sur la réalisation d'applications dans le Cloud et à l'administration d'une plate-forme. Enfin, nous avons appris que l'automatisation dans un projet est un atout crucial à considérer.

Sitographie

- Documentation de la classe Math Apex :
https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_methods_system_math.htm
- Documentation de l'infrastructure Visualforce :
https://trailhead.salesforce.com/fr/modules/visualforce_fundamentals/units/visualforce_intro
- Documentation de la fonction d'erreur (Error function erf) :
https://fr.wikipedia.org/wiki/Fonction_d'erreur#Calcul_num%C3%A9rique
<http://dictionnaire.sensagent.leparisien.fr/Fonction%20d'erreur/fr-fr/>
<https://www.ilemaths.net/sujet-erf-10-arf-364726.html>
<https://introcs.cs.princeton.edu/java/21function/ErrorFunction.java.html>
- Documentation sur les calculs d'intégrale :
<https://www.dcode.fr/integrale-intervalle>
- Documentation de la fonction random() sous Apex :
<https://developer.salesforce.com/forums/?id=9060G000000Bd0YQA0>
- Loi Normale & fonction de répartition :
https://fr.wikipedia.org/wiki/Loi_normale
<https://www.developpez.net/forums/d245215/logiciels/microsoft-office/access/calcul-loi-normale-standard-formule-fonction/>
- Normal-distribution en C++ :
http://www.cplusplus.com/reference/random/normal_distribution/
- Documentation sur la méthode Simpson :
<https://goo.gl/NZy2vM>