# Стоковая цена

непрерывный процесс, дифференцируесый случ процесс

*W*0 = 0 с вероятностью 1.

2. *W={Wt}t≥0* – процесс с независимыми приращениями.

3. *Wt – Ws ~ N*(0, *t – s*), где *s* < *t*, *N*(0, *t – s*) – нормальное распределение с нулевым средним и дисперсией *t – s*.

4. Траектории процесса *Wt* (*ω*) – непрерывные функции времени с веро-ятностью 1

интегралы Ито и связь с бенчмарком

# Аналитическое решение

При ограничениях рынка, построенного с помощью модели Блэка-Шоулза, цена акции через промежуток времени T вычисляется по следующей формуле:

(1),

где – винеровский процесс в момент времени t.

//являются случайными числами, распределёнными по нормальному закону с параметрами (0, Т)

Для моделирования значений винеровского процесса используется функция vsRngGaussian из библиотеки mkl, генерирующая последовательность псевдослучайных числе, распределённых по нормальному закону с необходимыми параметрами распределения.

Реализация функции getStockPrice представляет собой вычисление цены акции по формуле (1).

Аналитическое значение цены акции в момент её продажи через Т лет можно получить путём множественного моделирования цены по формуле (1) и последующего усреднения получаемых значений. Ниже представлен алгоритм.

1. Инициализировать переменную, в которой будет накапливаться общая сумма значений цены
2. С помощью функции vsRngGaussian смоделировать псевдослучайную последовательность чисел, распределённых по N(0, sqrt(T)). Поскольку W(0)=0, W(Time) = dW, W и приращение W совпадают, и можно интерпретировать результат как nPaths значений W(Time)
3. В цикле по числу запусков (i от 0 до nPaths):

{

* + - Вычислить – значение цены акции в момент времени Т;
    - Добавить полученное значение к текущей сумме

}

1. Поделить полученную сумму на число запусков

В итоге будет получена оценка среднего значения цены акции, точность которой очевидно зависит от числа повторений. Ниже Представлен график зависимости и таблицы результатов

# Численное решение

В тех случаях, когда ограничения рынка для модели Блэка-Шоулза не выполняются, для расчёта цены акции спустя время Т применить аналитическую формулу не удастся. Интегрировать диффур сложнее, нужны стохастические чм, блаблабла.

# Сходимость численных методов

Интегралы Ито

Слабая, сильная, порядок, все дела

# Описание 4М

В предположении, что следующие формулы верны:

,

,

,

формула для вычисления шага метода Рунге-Кутты запишется в следующем виде:

=

Или

return S \* (1.0f + R \* h + SIG \* h \* dw + 0.5f \* SIG \* (R \* h + SIG \* sqrtf(h)) \* (dw \* dw \* h - 1.0f) \* sqrtf(h));

# Вычисление стоковой цены

Для вычисления стоковой цены реализована возможность использовать ряд численных методов и двух генераторов псевдо- и квази-случайных чисел.

Стоковая цена вычисляется с помощью функции SimulateStockPrices по следующему алгоритму:

Для корректного вычисления стоковой цены с помощью численных методов необходимо получить множество значений цены, а после усреднить её, и итоговое значение будет являться оценкой математического ожидания.

* В зависимости от входных параметров, инициализируется генератор MCG-59 или Соболя
* Выделяется память для массива случайных чисел и происходит их генерация
* В цикле по числу запусков (i от 0 до npaths) с помощью функции stockPricesIntegrator вычисляется значение стоковой цены и её накопление в переменной StockPrice
* Поделить полученную сумму на число запусков

(int StepIndex, int indexGen, int npaths, int nsteps, float pS0, float pR, float pSig, float time, unsigned int seed)

В функции stockPricesIntegrator происходит вычисление стоковой цены на одной траектории:

* Происходит инициализация переменной начальным значением в момент Т = 0
* В цикле по числе шагов на траектории происходит вычисление цены в момент времени Т + dt
* Итоговое значение ЫStockPrice возвращается из функции

# Доказательство корректности

1. Прежде всего, хотелось бы увидеть в экспериментальных данных подтвер-ждение соотношения \* ( ) ( ) + . Во-первых, будем рассмат-ривать уравнение (6), точное решение которого ( ) нам известно. Во-вторых, рассмотрим **nPaths** траекторий Винеровского процесса *Wt*. На каждой траектории вычислим по аналитической формуле значение ( ), а также численно значение ( ). Просуммируем модули разностей ука-*Численное решение стохастических дифференциальных уравнений* занных величин для каждой из траекторий (разумеется, вычитаются вели-чины, соответствующие одной и той же траектории) и усредним их.

Будем рассматривать величину в качестве оценки математического ожи-дания \* ( ) ( ) +. В соответствии с центральной предельной теоре-мой – асимптотически нормально распределенная случайная величина при больших **nPaths** и сходится к истинному значению средней ошибки при **nPaths**, стремящемся к бесконечности [1].

В вычислительных экспериментах значение **nPaths** всегда конечно, одна-ко, это не мешает поступить следующим образом: разбить все множество траекторий на M групп, для каждой траектории посчитать модуль отклоне-ния от точного значения, вычислить M средних и, рассматривая их как вы-борку, найти ее выборочное среднее и выборочную дисперсию. Используя указанные параметры, можно построить доверительный интервал для зна-чения (см. в [1] обоснование применимости указанной процедуры).

Учитывая тот факт, что оценка средней ошибки не зависит от того, будем ли мы разбивать траектории на группы, ограничимся изучением этого па-раметра. Построение доверительного интервала и изучение вопроса об его уменьшении может быть выполнено в рамках дополнительных заданий к работе.

Итак, будем оценивать среднюю ошибку по формуле (11), организовав численное моделирование траекторий Винеровского процесса *Wt*, а также, для метода порядка 1.5 – траекторий процесса *Zt*, на каждой траектории – вычисление аналитического и численного решений, модуль разности кото-рых входит в формулу (11). В итоге получим значение оценки средней ошибки .

Проведем эти вычисления несколько раз при разных значениях шага инте-грирования *h*. Для уменьшения затрат времени на моделирование будем получать траектории Винеровского процесса с некоторым шагом *d*, впо-следствии выбирая *h* = *d*, *h* = *2d*, *h* = *4d* и т.д. пока шаг не станет слишком большим. Значения шага интегрирования и полученной средней ошибки будем заносить в таблицу.

1. Вернемся к формуле \* ( ) ( )

|  |  |
| --- | --- |
|  |  |

# Справедливая цена

# Аналитическое решение

# Численное решение