Michael Page
CITY

INTERVIEW PREPARATION FOR JUNIOR QUANTS

1) Basic Maths:

As mentioned by a number of my clients the failing of many junior candidates they interview is their lack of ability to solve problems using the fundamentals. For example, whilst 90%+ can apply Ito's lemma in the context of Black Scholes to calculate the price of an option where a stochastic process is involved, over half of these people are not able to solve it as an ODE once the stochastic component has been removed. This suggests two things;

i. They have merely learned the solution by memory; and/or
ii. They are only aware of standard methodologies;

without understanding the fundamental mechanics of the model. What clients are looking for at the entry to 2 year level is strong competency on the basics and strong (almost intuitive) mathematical and logical problem solving skills that would suggest that in the future you will be capable of learning the difficult and complex processes applied to harder problems. If you slip up on the basics it's difficult to judge your potential of coping with the more challenging products or indeed in a longer term your potential to bring innovative ideas to the forum. I would therefore suggest that prior to interview you spend at least a 40-50% (the other 30-40% on programming (C++) and the rest learning about the products and their structures) of your time consolidating your understanding of basic math applicable to quantitative finance. I deem this to include the following areas:

Calculus:

Functions of a single variable:
- o Ordinary calculus
- o Ordinary differential equations
- o Solution methods
- o Basic numerical integration
- o Simple integral equations

Functions of two or more variables:
- o Partial differential calculus
- o Partial differential equations
- o Classification
- o The diffusion equation
- o Solution methods
- o Basic numerical methods

Matrices:
- o Matrix manipulation
- o Eigenvalues and eigenvectors
- o Exponentiation

Probability:

Elementary probability theory:
- o Distributions, discrete and continuous
- o First and second moments (mean and variance)
- o Higher moments (skew and kurtosis)
- o Important distributions
- o Several variables
- o Correlation
- o Central Limit Theorem

Elementary statistics:
- o Data representation
- o Regression
- o Confidence intervals
- o Hypothesis testing

Random walks:
- o Trinomial
- o Transition probability density functions
- o Deterministic equations from random behaviour

2) Basic Financial Maths:

I would then make sure you have mastered the basics math behind of option pricing theory, e.g.: You understand Black Scholes and are able to not only solve the equation using at least two different methods but are able to discuss the application in a real world context (you can apply it to problems and discuss the limitations of the model and perhaps have an insight into other methods that overcome these basic limitations). A complaint from clients is that some candidates are not "pragmatic" mathematicians – i.e. if they do have a good understanding of how an equation works they can often be too zealous about the equation itself rather than its application and practicality.

Example: When asking someone to price an option the "purists" dive straight into finding the solution with the most precision, however before attempting any question like this it is important to know the precision limits required as this may fundamentally change the approach one may take (speed vs precision optimization).

In terms of specific math look at: Markov Processes, Ito Processes, Ito's Lemma, Wiener, Brownian Motion, PDE, Monte Carlo (setting reasonable boundary conditions….). It goes without saying that you should understand the terminology and simple behavior of vanilla derivatives e.g the relationship between option prices and their underlyings, what is a put, what is a call etc.

Finally as a basic rule – ensure you are able to talk in depth about what you have put on your CV, even if it is something you have not done for a few years make sure you are able to give a thorough overview of the projects you've done and a fairly good summary of the considerations behind technical decisions you made in any of those projects. Be prepared to be questioned on these and make sure you have re-familiarized yourself with the subject areas involved.

3) Complex

The complexities are all about how to tailor numerical methods to the problem – usually within an interview for an entry/junior level position if you are presented with a highly complex problem the interviewer will often talk you through the approach to solving it giving you pointers where you get stuck, all you need to ensure is you have mastered the basics and that your showing that you understand the approach being taught to you by the interviewer.

With regard to the more complex problems interviewers are looking for creativity in finding solutions. It is difficult to be creative with mathematical rules unless a) you know them well and therefore can apply them and b) You have been actively applying them to a multitude of problems and therefore understand the different approaches to finding solutions.

Sample Questions on the basics

- Calculate $E[e^X]$, where X follows a Normal distribution $N(u, s)$ of mean u and standard deviation s.

- If

  $$dX_t = a(X_\infty - X_t).dt + s.dW_t$$

  and

  f is a function of X and t;

  calculate df, where a, X_∞ and s are constant.

  What if the Brownian motion term in the above is 0 (i.e. $s.dW_t = 0$)?

- Write a probability density function for normal and log-normal distributions.

4) OO and C++

Software development in C++ is one of the key technologies employed by global financial institutions, particularly due to its support of object-oriented programming. This has resulted in a minimum requirement for all quantitative professionals to have a solid core background in C++ modelling. Often entry level or junior candidates will have used

C++ sporadically in relation to one or two isolated problems (e.g.; implement a PDE solver), but do not have a deeper understanding of the core programming principles (particularly OO) and language whereby they are able to apply it easily to a wider range of problems. You should aim to learn C++ as a skill/subject in isolation so you are able to apply it with a high degree of fluency to a general range of problems/circumstances.

Areas of theory to cover:

Variables, types and Expressions:
- o Identifiers
- o Data Types
- o Declarations
- o Constants and Enumerations
- o Assignment and Expressions

Branch and loop statements:
- o Boolean Values
- o Expressions and Functions
- o 'For', 'While' and 'Do....While' Loops
- o Multiple selection and Switch statements
- o Blocks and Scoping

Functions and Procedural abstraction:
- o User-defined functions
- o Value and Reference parameters
- o Polymorphism and Overloading
- o Procedural abstraction and good programming style
- o Splitting programs into different files

Files and streams:
- o Input and Output using files and streams
- o Streams as arguments to functions
- o Input and Output using '<' and '>'

Arrays and Strings:
- o Declaring arrays and strings
- o Arrays as parameters
- o Sorting arrays
- o Two-dimensional arrays
- o String manipulation

Pointers:
- o Declaring pointers
- o The '*','&','new' and 'delete' operators
- o Pointer arithmetic
- o Automatic and dynamic variables

Recursion:
- o Recursion and iteration
- o Mechanics of a recursive call
- o Recursive data structures
- o Quick sort

Classes:
- o The object-oriented paradigm
- o Encapsulation and inheritance in C++
- o Constructors, friends and overloaded operators
- o Static members

Numerical Methods:
- o Approximating a PDF/CDF
- o Solutions of linear systems
- o Direct methods of solution and iterative techniques
- o Numerical integration
- o Power method
- o Explicit and implicit finite difference methods for parabolic PDEs
- o Monte Carlo method

Sample questions:

Basic:
- What is the difference between a pointer and a reference?
- When would you use a pointer/reference?
- What does it mean to declare a function or variable as *static?*
- What is a class?
- What is the difference between a struct and a class in C++?
- What is the purpose of a constructor/destructor?

What is a constructor, destructor, default constructor, copy constructor?
What does it mean to declare a member function as *virtual/static?*
What is virtual inheritance?
What is polymorphism?
What is the most difficult program you have had to write?

Intermediate: What happens when you have a non-virtual method in a base class and a method of the same name in a derived class?

What about "overriding" a virtual method in a base class with one in a derived class? Why doesn't this work the same?

Can you call a virtual function in a base class when you have overridden it?

Other: How could you determine if a linked list contains a cycle in it?
How would you reverse a doubly linked list?
Write a function to sum 1 to *n* numbers?
How would you traverse a binary tree?

Useful online resource: http://www.parashift.com/c++-faq-lite

These are merely a list of general questions and should be easily answerable in detail if a comprehensive general study of C++ has been undertaken. It is also likely that you will also have questions where you are given a sample of code and are asked what is wrong with it, you may also be given a function and be asked to determine what the function will output. Also in addition to learning the theory it is advisable that you put theory into practice by doing as much implementation as possible. You may find it a useful exercise to implement a framework in which to price a variety of options using the Black-Scholes formula and any appropriate extensions.

5) Problem Solving Questions/Brainteasers.

It is likely that during a course of interview you will be asked some "brainteaser" type questions designed to test your intuition for problem solving. The solutions are often mathematical but can also require simple logic or lateral thinking. Often there can be several solutions, some more optimized than others. Interviewers are looking to see your thought process in solving the problem and will usually require you to prove your answers.

Sample questions:

- What is the sum of all the numbers between 1 and 1000?
- How would you sum a series of 1 to *n* numbers? Demonstrate proof for this.
- You are given a set of balance scales which you are to use to measure eight balls. Seven of these balls have the same weight: the eighth ball is heavier than the rest. What is the minimum number of weighs you could perform to find the heaviest of the eight balls?
- Same as above but with 12 balls?
- To qualify for a race, you need to average 60 mph driving two laps around a 1 mile long track. You have some sort of engine difficulty the first lap so that you only average 30 mph during that lap; how fast do you have to drive the second lap to average 60 for both of them?
- A river is flowing downstream at 15 mph relative to the shore. A rowing team is practicing rowing and at first they row upstream (against the current). They can only go 1.5 mph relative to the shore at this rate. The guy at the back

end of the boat is wearing a hat when they begin, but after a while his hat falls into the water (and floats) and it is 15 minutes before they notice it. They then instantaneously reverse direction and row back to catch up with the hat, rowing with the same strength or power they were rowing with before. How long will it take them to catch up with the hat as it is pushed downstream by the current?

- There are 10 open boxes containing 100 coins each. In 9 of these boxes the coins are made of gold, and in the other the coins are made of copper. You are given a large digital balance which can be used once only. Can you identify the box containing copper coins knowing the weight of both gold and copper coins?
- A bag contains a total of *N* balls with either blue or red colour. If five balls are randomly chosen from the bag, the probability is precisely 1/2 that all five balls are blue. What's the smallest value of *N* for which this is possible? (Hint: Use different number of blue/red balls to get to the answer?)
- You are given 5 bags containing 100 coins each. The bags can contain coins of 3 different types that look identical. The first type weighs 9 grams, the second type 10 and the third type 11 grams. Each bag contains coins of equal weight but you do not know how many of the 5 bags are of the different types. (i.e. all 5 bags might well contain 9 gram coins as far as you are concerned). You are given a huge digital balance. How many times do you need to use the balance to clearly determine the type of coin contained in each bag?
- You are playing Russian roulette with a six chamber revolver, you load 2 bullets into the revolver in adjacent chambers. You spin the barrel place the gun to your head and pull the trigger, you don't shoot yourself. You now have the option of either spinning the barrel or pulling the trigger again, which do you take?
- You are in a boat on a lake, in the boat there is a suitcase, you throw this suitcase over the side of the boat. What happens to the level of the water in the lake? Does it rise, fall or stay the same?
- How many manhole covers are there in London?
- How many petrol stations are there in the UK?
- You are gambling on the roll of a fair six sided dice, in this game if you role a 1 you get $1, if you role a 2 you get $2 if you role a 3 you get $3 and so on. What is the expected return after 100 roles of the dice.
  (Note – there are a large number of variations on this game, you should spend some time looking at various dice games and probabilities).

If you wish to discuss any of this material further, please contact

Doug Ward
Consultant
Quantitative Analysis
Michael Page City
50 Cannon Street
London
ENGLAND
EC4N 6JJ

Tel: +44 (0) 20 7269 1981
Mob: +44 (0) 7876 565 803
Fax: +44 (0) 20 7329 2986
Web: www.michaelpage.co.uk

Dr Tony Ofori
Consultant
Quantitative Analysis
Michael Page City
50 Cannon Street
London
ENGLAND
EC4N 6JJ

Tel: +44 (0) 20 7269 1979
Mob: +44 (0) 781 517 4459
Fax: +44 (0) 20 7329 2986
Web: www.michaelpage.co.uk