Notes on a non-academic job search

John Kerl

Department of Mathematics, University of Arizona
Graduate student colloquium

April 28, 2010

Overview

- Why
- When
- Who/what/where
- How

What do I (and don't I) know about it?

Everyone's job-search experiences — and results! — differ. I'll tell you about things I learned which I think are worth repeating.

- I'm completing my PhD this spring. My coursework and dissertation research focused on probability, statistics, Monte Carlo methods, and stochastic processes.
- I came back to graduate school after a decade of industrial software engineering. This is not a typical sequence of events for a math grad!
- For personal reasons, I looked solely at D.C., New York, Boston, and Chicago, and the general area including Knoxville, Blacksburg, Raleigh-Durham, Winston-Salem, Greensboro.
- I applied for about 80 jobs total: approximately 30 academic jobs (postdoc, tenure-track, adjunct), two dozen laboratory (public or private) positions, and two dozen industry jobs.
- I received interest (in terms of on-site interviews) from one private lab, two defense contractors, and three finance firms.
- I will be working in computational finance in New York, starting in just over a month.

Why consider a non-academic job?

- If you have solid math and analytical skills and know how to code, it's likely that several organizations will want to talk to you. Your particular area of mathematical research is, for many employers at least, less important than your ability to think analytically and quantitatively. (That is to say, number theorists can make a living doing applied math!)
- If academia is really your dream and industry jobs are a distant fallback, interviewers will probably figure that out ...
- ... but if you're really interested in happily going either way, you can diversify your options in this today's tight labor market.
- You may, officially or unofficially, think of it as an industrial/laboratory postdoc.
- If working in an industry or laboratory environment is your dream, listen on!

Plusses and minuses

- Hours are less flexible in some industry jobs . . . but there's no grading before going to bed.
- Pay is almost certainly higher.
- In academia, you work on your own research. In a non-R&D job, you might do no research at all. In an R&D job, you'll be pursuing the research agenda of an entire organization. This may be a good thing: it takes a team to tackle Big Questions: cancer research, drug design, high-performance computing, climate modeling, alternative energy, next-generation genome sequencing, proteomics, ...
- Non-academic research tends to be more practically oriented.
- R&D groups aren't too different from academic departments. Depending on the job, you may or may not publish as many papers. But, people learn from one another, grow, change jobs . . . and your ideas will spread around. And the total number of people who can understand your stuff may be about the same inside or outside academia.
- There is competition between companies, some secrecy for competitive advantage, and thus some re-inventing of the wheel.
- Once you've worked outside academia for a few years, will it be harder to get back in? Once you've worked in academia for a few years, will it be harder to get out? I don't know how to answer either of those questions.

Skills to develop ahead of time (i.e. now)

- Teach yourself to **code**, **write**, and **present**.
- There are very few jobs available for pure chalkboard thinkers — even then, they're probably just a few very senior people in the company, and you'll be an entry-level researcher.
- They may be willing to teach you their favorite language — but you should be comfortable coding in some language. Matlab at the very least. Preferably C, C++, Java, C#, Fortran, Python, Perl, Ruby, ... Implement mathematical concepts, however naively, when you see them in a course. Do this throughout your grad-school career. After a few years, you will be an experienced programmer.
- Write up (some portion of) your dissertation research for publication, and/or go to a conference which publishes its proceedings. Throughout school, write things down when you figure them out. After a few years, you will be an experienced writer.
- Give grad talks (easy to do since you've got some stuff written down already), give area-seminar talks in the department, present at the Joint Meetings. After a few years, you will be an experienced speaker.
- Interdisciplinary education: Make use of the out-of-department requirement which we have in the PhD program. Take digital design, processor architecture, numerical methods for (department name goes here), etc. etc. In our department: numerical analysis, ODEs, PDEs, SDEs. Take statistics. Minor in statistics.
- Look into summer internships.

Timeline for your final year in grad school

- Academic-job timeline: ads go out in the fall. Application deadlines mostly Dec. 1 and Jan. 1. Joint Meetings in mid-January. Interviews February or so. Offers go out March or so. Some visiting-professor and postdoc positions open up later.
- Some PhD-hiring employers understand this schedule, some don't. (The standard corporate timeline for people who don't hire PhDs, at the other extreme, involves a two-week notice.) A few picky employers have a process which takes a couple months.
- Know yourself. What's your 1st and 2nd choice of academic vs. non-academic employment?
- If academia is your 1st choice: maybe wait, and put in industry applications in March or so.
- If industry is your 1st choice: trickier. Would you want to accept a December job offer? Make it clear in your cover letter when you will graduate and when you will be available. Some employers are happy to hire early in the spring, and have you start in June.
- Don't forget — as a highly educated, mathematically literate person, you are an asset.

Employers and how to find them

Academia: it's easy to find jobs and apply for them. The Joint Meetings, eims.org, and mathjobs.org will show you almost everything. (There's more to it: see Josh Chesler's notes on the academic job search, which I've archived at math.arizona.edu/~kerl.)

Outside academia: There is no central resource. You have to look around. Some places I'm aware of (URLs on the next slide):

- National labs such as Ames, Argonne, Brookhaven, Fermilab, Livermore, Los Alamos, NREL, Oak Ridge, Sandia, ...
- NIST, NIH, NSA, NASA; IBM, GE, AT&T; Novartis, Merck; various engineering firms. And you can take statistics skills all over the place.
- Large defense contractors such as Northrup Grumman, Lockheed Martin, Boeing.
- Small defense contractors such as Metron and Daniel H. Wagner Associates. (Both of these attend the Joint Meetings and recruit actively.)
- Finance jobs, mostly in New York, Chicago, and London. Warning: some finance employers have long hours and poor working conditions; others are employee-friendly. This field has a strongly bimodal distribution.
- Some postdoc positions in computational biology, computational genomics, materials science, etc.: you just need to be good at math and know how to code, and they'll be happy to help you learn their application domain. These are not in math departments and aren't advertised at MathJobs.

Web resources (page 1 of 2)

It's the 21st century and lots of things are on-line. Many employers specifically ask you *not* to send a paper application. Here are some resources:

- There are job boards such as CareerBuilder, JobFox, Monster, Doostang, Juju, ... Use them, but not only them. The applicant/position ratio is high; many listings are stale; not all employers use them. Re-check frequently.
- Job-board postings are either direct (Company X) or indirect (Agent A who recruits for companies X and Y). Avoid clicking through the job board's convoluted multi-screen application. Find the name of the agent and write to them directly. They work on commission filling openings for companies X and Y, not for you, but they do network with their clients and it's in their interest to try to find something you'd fit into at that moment. (If you don't, don't expect to hear from that agent again.)
- Consider LinkedIn. It's a social network for professional contacts, completely separate from Facebook. (I.e. no embarrassing photos.) You can find some job listings there. Also, if you're thinking of applying somewhere, you might find out that you have a friend of a friend who works there. (This may have helped me get a call back from the company that ended up hiring me.)

Web resources (page 2 of 2)

- If you want to do, say, computational genomics, search for that phrase and find out who does it. Search creatively and persistently. Look not only for job ads, but people and companies whether they say they're hiring or not. (I had a couple interviews at places to which I submitted unsolicited resumés.)
- `usajobs.gov` for lots and lots of government jobs. Check back every week or so.
- `mathjobs.org`, `eims.org`, `siam.org`: some non-academic employers post here.
- `newscientistjobs.com`
- `naturejobs.com`
- `sciencemag.org`
- `postdocjobs.com`
- `jobs.phds.org`
- `jobs.stackoverflow.com`
- `thesciencejobs.com`
- `www.nature.com/naturejobs/science`
- Google for more ...

Non-web resources

Each personal contact outweighs several items found by web search. (Your computer is a tool — a very useful one — to help you find people.)

- Talk to your friends, relatives, professors, classmates. Remember six degrees of separation. E.g. my wife's classmate has a friend at NIH; a cousin's friend told me salary expectations for a certain industry; a professor's friend works at Oak Ridge; one of my interviewers went to grad school with my master's advisor.
- Go to conferences and workshops and mention you're on the market. Give a talk.
- Think you don't know much? Go and give a talk anyway. People know you're a graduate student and they don't expect you to know everything. You will probably find more kind advice than you expected. (I did, anyhow.)
- Talk to people on conference breaks, lunch, dinner. Too shy? Say something anyway. You don't have to have the hairspray and the glittering smile and the laminated business cards. But do shy your way up to people, tell them you're graduating, and ask if they can give you any advice.

Resumé vs. CV

The difference: an academic CV contains every talk you ever gave, every paper you ever wrote, etc. If you're a grad student, it's not much different than a resumé; if you're a full professor, your CV can easily run a dozen pages.

Write a resumé for your non-academic jobs, along with your CV for academic jobs. Keep it two pages at most — preferably one page. Contact information, brief personal statement of what you're excited about and why you get out of bed in the morning, education, work experience, skills list.

Keep it quantitative. Not *I was responsible for* X but problem-action-response: *I found situation* P*, I did* A *about it, and I saved the organization a factor of* R *in time/money/student retention/ whatever*. If you've gone straight through school without much job experience, that's fine — still, do what you can to keep it accomplishment-oriented.

Proofread and have someone else (roommate?) look at it too. An employer/recruiter with 200 resumés to look at will probably give each one 5-10 seconds at first. Don't let a typo give them a reason to throw yours out.

Rule number one: **whatever accomplishments or technical skills you claim, you must be able to back them up**. You will be asked specific questions about what you claim to know.

Cover letter

For e-mailed applications, the e-mail *is* the cover letter. Attach the resumé and refer to it in your cover letter. For MathJobs et al., make a nice PDF. One page max.

What you say is up to you. Don't copy and paste anyone else's ideas — it must be unique to be memorable. Don't be monotone — if they don't remember you as an individual, it doesn't help. Take time to customize each letter.

In the first sentence, state what position you're interested in. (They may have several different positions open.)

Tell them what excites you, what you're eager to do (with or without them). Enumerate some technical skills. Focus on something you accomplished (even if it's just a term project). Focus on the specific skill set they're looking for. If you don't know much about an item but are eager to learn more about it, say so. Include what makes you a good fit for them and what makes you unique. Tell them what things you're most proud of. This gives them something to ask you about on a phone interview.

Mention your availability, including your graduation date.

Summarize, and express a desire to speak to them in greater detail. Remember — the sole purpose of the cover letter is to get them to pick up the phone (or e-mail you). Include contact information (phone number and e-mail address).

Portfolio

You have a math-department website — you can (and should) use it to help you find work. Post your

- CV and resumé,
- research statement,
- teaching statement,
- lecture notes for talks,
- term projects,
- preprints of papers, and
- maybe a few tasteful pictures of you doing something besides math (e.g. hiking).

Make yourself searchable: I *just* missed a job (had two good on-site interviews) via a recruiter who googled me and found a good fit for a company he was recruiting for. (So why didn't I get the job? More information below.)

Tips for phone interviews

Prepare! A phone interview is a real interview. Set aside an hour, even though it may end up being shorter. Also, they may *say* 15-20 minutes but end up talking longer if they like you enough. Pick a quiet space with no distractions. Use a land line (not a cell phone) if possible. If not, find a very quiet spot. An airport lobby is *not* a good place. (Ask me how I know.)

Learn about the company, what their business is, who their customers are, where their offices are, about how many employees they have. Prepare for commonly asked questions. There may still be an out-of-the-blue question, but there are recurring patterns:

- *Tell me about yourself*: Don't start with childhood. Tell them what you want out of your life (in the short to medium term), why you went to grad school, what kind(s) of work will fulfill you, what kind of people you want to work with. If this is a phone call worth having, you'll both realize that much of this jives with the organization's needs. Other versions of the same question: *what's your ideal job / ideal role*; *what do you want to be doing in five years*; *where else are you applying.*
- *Describe something you're most proud of. What are your greatest strength and weakness?*
- You may get programming or puzzle questions on the phone. More info about this, and about *tell me about your research*, on the on-site slides below.

Tips for phone interviews

Have questions for them. Not only does this help you interview them, but it shows your individuality and preparedness, and it helps them evaluate you. A few interviewers I spoke with didn't ask me any questions at all — they wanted me to do the asking.

- What have other employees found to be the toughest challenge in this position? What concerns need to immediately addressed in this position?
- How would you describe the corporate culture? Describe a typical workday for one of the people I'd be working alongside.
- What do you, the interviewer, personally most and least look forward to each week?
- Based on what you know about me at present, what do you see as my greatest strength and greatest weakness with respect to this position?
- After three months, and after a year, how would you decide if I had been successful?
- How many employees did you have two years ago? How many today and how many do you expect to have two years from now? How will you know if you have enough people?
- What's your turnover rate?
- What mathematical problems are you trying to solve?
- What are some of the most/least useful mathematical/software skills for people working in your organization?

Tips for on-site interviews

Prepare!! Everything mentioned above for phone interviews, in spades. If they've flown you out, someone at the company likes you a lot. You're probably in their top five — even at the pickiest employers, you're in the top 20. Invest time in this interview.

Go ahead and tell them you'll wear a suit unless they suggest otherwise. At all but one place I interviewed at, they told me to leave the suit at home and be comfortable. Still: wear nice shoes, slacks/skirt, and something with a collar.

Prepare to go all day. If it looks like it isn't going to be a fit, they may do you and themselves a favor by letting you go at lunch. If you meet with several future co-workers, a future supervisor, an executive in the company, and an HR employee who talks you through their benefit plan at the end of the day, you probably have good chances.

Joke around a little bit (within reason) — be willing to smile and laugh. Be an interesting future co-worker. They're less likely to remember a quiet, passive applicant in a grey suit, and more likely to remember the applicant who was technically competent, thought on their feet, and shared their love of hiking.

Don't be afraid to let them know about other interviews and/or offers — this makes you look in-demand. If you're making a few long-distance trips to different employers, don't be afraid to schedule them on back-to-back days. Places which knew they had competition, replied to me more quickly and eagerly. Coincidence?

Tips for on-site interviews: technical material

Companies hiring PhD mathematicians may ask you to prepare a half-hour or one-hour talk on your research. Ask them about the intended audience and technical level. Maybe send them a draft of your slides, or slides from a previous talk, and ask what they think. Standard advice for any talk: 1/3 tutorial, 1/3 for people in the general area, 1/3 for the experts in your subject.

Only one place I interviewed asked for a formal job talk. But at every one of eight employers I interviewed with (on the phone or at the Joint Meetings or on-site), at least one person asked me to explain my research in 5 minutes or so. Prepare for this. Don't ad-lib it because it's only 5 minutes. Make some notes for yourself; practice; maybe prepare some slide-like material printed out on paper which you can show them in lieu of doodling on a notepad. If they only ask for 5 minutes, they do *not* want to understand your entire dissertation — they want to see how you communicate technical material.

Expect programming questions, e.g. write a C routine which reverses a string in-place. They'll ask you about a language you claim to know. The question may or may not be tricky — they want to see not only if you can do it, but how comfortable you are.

Web-search for "programming interview questions". Particular questions vary highly, but certain patterns re-arise: e.g. linked lists, binary trees, sorting, searching. Once you've solved many of these on your own time, oddball questions won't seem so oddball.

Tips for on-site interviews: puzzle questions

Expect puzzle questions. Example: You have two dozen coins, one of which is hollow, and a two-pan scale which tips left, right, or centered. Devise a strategy to find the hollow coin, minimizing the number of weighings.

These may seem maddening: either you see or don't see the magic trick. But not so. In my experience, these almost always involve decision trees. Write down what you know. At each step there are only so many possibilities of what to put where. Whatever the choices you enumerate, walk through them. Don't be afraid to mention a non-optimal solution first (e.g. twelve weighings of one coin in each pan) and then try to improve on it.

Think out loud. Don't try to silently dream up the perfect solution. They're hiring a thinker and they want to see you think, and see what it's like to work with you. They might not even expect a complete solution — they might just want to see how far you get. Don't give up — let them ask you something else if they decide to.

Have fun! Tell them what ideas you have, what might not work out, which fork(s) in the road you're exploring. They will (in my experience) quite likely give you little hints or nudges. Joke around a little bit; be a memorable applicant. The first puzzle question I got scared the bejeebers out of me. By my 5th interview or so, I was disappointed one day when I *didn't* get any puzzle questions.

Tips for on-site interviews: afterward

Send a thank-you note (I believe e-mail is fine) at least to your HR contact — ideally, to everyone you met with. (This is a good reason to get business cards of people who interview you.) Do this after phone interviews as well.

When you get a job offer, ask for some time to think it over — whether you have a competing offer or not. If they want you, they'll wait a week or two. Ask people you know (aunts/uncles, parents, siblings, professors, etc.) for advice. They'll be proud their math-grad relative/student has a job offer; they'll be sure to give you some ideas on how (and how much) to haggle.

If you don't get any offers — don't give up! Keep those applications going. And don't be afraid to write people and ask for a few minutes of feedback. One employer who didn't hire me said that my technical skills were unquestioned, and my experience was relevant and extensive — but ultimately, there was too much overlap between my skill-set and the skill-sets of people already on the team. Good to know! I used this to re-write my cover letters, which before then had been monotone and impersonal; I started emphasizing my uniqueness in phone interviews. After that, good things started happening.

Keep asking — ask people what they least liked about your resumé; ask them what made them pick up the phone and call you. Ask your friends and family for tips. Most importantly, be willing to update your plans as you go along!

Thank you for attending!