Keith On . . .

Engineering Design

K.E. Schubert

Founder
Renaissance Research Labs

Associate Professor
Department of Computer Science and Engineering
California State University, San Bernardino

Contents

Part I

Employment

Chapter 1

Job Hunt

I suggest starting your job hunt 4-6 months before you want the job. This allows you time to figure out the right job and location, then research the companies that fit. You can then look for openings or make one by contacting them directly.

When looking at job descriptions consider your skills, experiences, and goals. Interpret your skills in the broadest sense, by which I mean, you need to see how what you know and have done could fit their needs. No one has done exactly what they advertise. The advertisement is an ideal, you just have to show enough to get past the human resources gauntlet so you can get an interview with the technical people. Usually the biggest barrier will be experience. Some sources of experience that you should count (or if it is early enough that you should do to prepare yourself):

- Summer jobs
- Help desk/ department job (Ken has lots to do)
- Summer research with faculty
- Summer research at another university (UCSB summer research looks great on a resume)
- CSCI 695/595 Independent study/research project
- CSCI 475 Internship
- CSCI 455 Software Engineering (this is a huge project and worthy of listing, I would note that it has been written up in several articles)
- CSCI 303 Computer Engineering design (lots of experience for the CE's)
- Volunteer work (church, charity, or other technical volunteering can look great)
- Working with faculty on a grant

- IT for friends and family
- Any IT actually

Once you have found the jobs that fit it is time to send them a resume and cover letter.

1.1 Resume

Everyone has an opinion about what a resume should look like. People fuss with fonts, styles, pretty paper, and all sorts of silly things to get their resume noticed. I did this too when I started out and it didn't work. What really matters is a clean, easy to read document that highlights what is great about you. Look at your life, skill, experiences, accomplishments, and awards. Lead with that. Work your way down the list, hitting the things that make you stand out. Do you want your paper being noticed or you? When you stand out, you can worry about all the frills to try to make your paper get picked up.

What should be on your resume:

- Education degrees, certificates, special courses
- Work experience (particularly hi-lighting those that fit with what you are seeking)
- Awards, recognitions, and professional licensing
- Professional societies and service organizations
- Skills, languages, etc.

1.2 Curriculum Vitea (CV)

So you want to be an academic? It is a different life with a great many rewards. I worked in industry for several years, and had a great job with great people. I have no complaints, but I would not trade my job as a professor for even that great job no matter what the pay raise would be. A Curriculum Vitea, or CV as we all call them, is an academic resume, though it is occasionally used for very experienced workers or technical experts, whose experience and accomplishments do not fit in the standard resume format.

1.3 Cover Letter

I feel the cover letter is the most important part of your application. It is your chance to pitch yourself. A cover letter should stand out, communicating not only your ability and interest, but also who you are. A cover letter should make the reader feel you know them and what they need, and make them feel for you. This is a tall order, so don't sweat it if you don't do it perfect, do your best and let your best shine through.

Here is how I suggest you write a cover letter. Get the job description and hi-light key words and phrases that describe the job you are applying for. Get to know what they really want. Go to their website and see what they have been doing. What were their most recent products and projects? If you work with them this is what you will probably be doing, and the more you know the better.

Once you have done your homework and selected the key phrases and terms, which quantify what they want, look over your resume and see what you have done that fits their needs. This explanation of how you fit their job is what you are primarily doing in the cover letter. Don't beat around the bush, a quick statement of your enthusiasm is nice but then start showing how everything you have been doing makes you their perfect candidate. Don't be shy or self effacing, there are times for this, but a job application is not one of them. In the job application you are letting them know you are their perfect candidate.

Don't make the letter too long. A good letter catches attention, showcases your perfect fit with the job, encourages action, and is done. It is a letter, not an endurance test. If you are too long and wordy, that lets them know you are going to be so after being hired. Do you want to work with someone who never shuts up? Neither do they.

When writing you should just get something down then polish, polish, polish. The just write it advice is to avoid writers block. If you can get going it tends to flow, even if it needs cleanup. If you try to get perfection on your first try, you will probably just stare at a blank page and become frustrated. When reviewing your letter before sending, be brutally honest with yourself. Do you sound desperate or excited? Do you drone on? Is your first sentence (the hook) exciting or boresville? Put yourself in their shoes. Would you want to read this? Would it convince you that this person can do the job and will be nice to work with? Hiring people is scary, just like applying. Make it easy on them and they will make it easy on you.

1.4 Other Correspondence

You want to stay in their notice but not be annoying. Don't over-do it or you will drive them off. After sending in your application you should get an acknowledgement letter. When you do a brief thank you note with an offer to supply additional information if needed is acceptable but not required. Should you not receive an acknowledgement after a couple weeks, I would send a very polite note asking if they are lacking anything in your file or application. This wakes them up without being obnoxious, and lets you know that you got it. If you haven't heard anything in a reasonable period of time[1], then you can send a polite inquiry letter. Other than that don't make yourself a pill. If you do your job right in the material you send then they will be dying to talk to you.

[1] Sorry I can't give a hard time here. This varies by the level of job, type of work, and company doing the hiring. An entry level job in a technical profession should have the decision made in 4-6 weeks tops in most cases. The more important or difficult the job the longer the time, but I would find it hard to believe more than a few months, which is what it takes in academic environments.

1.5 Phone Interviews

The resume and cover letter are to get you the phone interview. Your job now is to wow them and make them want to do the follow up interview[2]. Be enthusiastic, lively, and professional. Don't let the conversation hang, make them feel comfortable. They want to know if you will be good to work with. Let them see the great person you are.

You should be expecting a call, so have the information on each job readily available, with a quick notes scheme. You want to continue highlighting how you fit their exact needs. A folder with a quick summary[3] of key points for the job. You should have the rest of the application material in the same folder, one folder per application, should you need to quickly reference something. Don't let your material distract you, they are only for reference if you freeze. I would advise looking them over occasionally so you won't even need to look at them when the call comes. It is reassuring to have them if you need them though. Think of what they might want to know and how you should answer them. Think of some questions for them, particularly interest in the company and what they are doing. The web is a great place for this. It lets them know you have done your homework on them and that you really care. Both are good things.

1.6 Site Interviews

This is the deal closer. I would read up on any key skill they are interested in. Write a program in the languages they care about. Do a small example of whatever they are going to do. Look over notes from relevant classes. In short be ready. Then the day before you go, don't do anything related to your job search, just have fun and go to bed early. Get there a few minutes early. Dress nicely, a good rule being to dress for your prospective boss's job. Some of this is corporate culture, which you will have a feel for if you have done your homework. Most places this means formal business wear (read business suit or equivalent). You should look clean, crisp, professional, and with a fun sense of style. Sounds impossible? Most of it is how you act and feel, a warm smile and confident style is the most noticeable thing you wear.

You are not only selling your skills you are selling you as a colleague so be someone you would want to work with. Be proactive in the interview. Don't just sit there saying short answers. Use your answers as opportunities to discuss relevant information and ideas. Particularly for engineers, but it helps others too, be ready to have a design discussion. Talk with them about their previous work and how they did it. Be ready with ideas and insights, while showing you can listen. You want them to imagine you are already fitting in the team. Make each person you speak with feel respected and valued. Any enemy can kill your job chances, conversely a bunch of friends will ensure your offer.

[2] For some entry positions you might be trying to get the hire offer off this. I have heard of some companies that do only one interview, but I am always suspicious of them. This implies to me they either have few applicants and are desperate, or it could imply they aren't that concerned about who is filling the spot. Either way, I have to ask why. Doesn't mean they are bad, it is just a warning flag to me.

[3] I mean quick, a long one will make you read and the conversation will hang. Bulleted points with words or short phrases works best. Keep the bullets short too.

1.7 Negotiating

To be honest this is my worst area, and I don't want to improve it. My philosophy goes like this. If you really want to work with me you will be fair. I should let you know what I need, and we can come rapidly to a fair arrangement. I try not to give too much info out early in this process, so I don't undersell or overprice myself. This is a feel it out maneuver. It is rare to get a major correction in pay or benefits once you are hired, so if you really need something then you have to say it, just keep it real friendly and nice. Look for win-win arrangements. Everyone should be happy when this is over.

Chapter 2

Ladder Climbing

2.1 The Ladder

Before you start climbing the ladder of success, you had better determine if you have the right ladder. There are many ways to be successful. Do you want to go into management? Do you want to do research? Do you want to do design? Do you like customer interaction? Do you want to work for yourself or a company? Do you like teaching? You must figure out what you enjoy.

The typical business progression is to go from technical positions into management. For many people this is natural and even desirable. For me it would be death. The last thing I want to do is give up my research and teaching to push papers. I am very happy there are those who wish to do this. I am just not one of them. This was a big motivator to me in deciding to become a professor.

Another route I strongly considered was the technical specialist route. I was working at Northrop-Grumman when I decided to go back to get my Ph.D., and like many large companies, they had a route for those who wanted to get promotions and pay raises without becoming a manager. They called it the technical specialist. After you progressed through the usual assistant engineer, associate engineer, engineer path, your progression could either go to group leader, or for those rare few who were techies to the core they could follow the tech specialist. This route was very appealing to me, in that you become the tech guru, assisting on a variety of projects. You would get to do a bunch of different things, and see them go through to completion. Not just managing people, working out the big ideas and problems, while letting others carry them out. In the end I didn't choose to do this, not because of a problem with this route, but rather due to the even cooler (to me of course) route of becoming a professor.

My point in all this is that there are different routes. You might not like the academic route. Perhaps management is in your future. Or maybe you would like to be a tech specialist. Perhaps you would rather stay in the rank and file implementing things (most do stay for one reason or another). No matter what you do, you should do it deliberately.

2.2 Worthwhile Work

Few things in life are as satisfying as doing a job well that really matters and you like. Work is important to your well-being. This does not mean you should be only concerned about your work, rather your work should be something that you are proud of, and you should work so that you can be proud of what you do.

2.3 Peers

A big part of a satisfying job is working with people you like. I would not stay long term at a job I didn't enjoy the people I worked with. I would find a new job and then leave. Notice I did not mention pay, because pay is irrelevant if you are unhappy. No amount of money makes up for a lousy life. You have to have a certain amount of money to live, hence why I could find a new job then leave[1].

Even if the job is unsatisfying, if you are working with a great group of people, then it is usually still good. You might still long for a more fulfilling job, but while you are looking for it, you will be happy with your friends.

2.4 Godfathers

One of the most important skills of successful people is the ability to get a godfather (or godmother) at their place of work. When you first start out, you won't know the ropes. Nobody does. What you need is a well positioned and knowledgeable mentor and protector. When I worked at Grumman Aerospace (before it became Northrop-Grumman), such a person was called a godfather (Grumman was a New York company, and the joking mob reference was deliberate).

Godfathers can open doors of opportunity, introduce you to the movers in the company, show you the ropes, and help you when you screw up. In short, they are invaluable. While some companies help you find them, most who have mentoring pair you with another fairly new person. This person may or may not have the ability to do all that a godfather should. They are certainly helpful, and should be rightly appreciated and listened to, but don't fail to look for a godfather. No mentor should be upset if their mente has other supporters.

2.5 Bosses

A good boss is a rare find. I have had some great ones and some really bad ones. The two best I had were at Northrop-Grumman: Steve Gavin and Ken Crane. They had a few things in common.

---

[1] While it is glamorous to say "take this job and shove it" and walk out without a net, this is impractical. Swallowing the annoyance for a short time allows you to get a job, then you can leave. I wouldn't even say "shove it" then as you might want a job reference in the future. It is better not to burn those bridges.

- They knew me. They knew what I could do, and what I was like. They knew those who worked hard and those who slaked. They knew those who were clever and those who were traditional. They could then use this in assigning tasks and overseeing the work.

- They gave me the room I needed to get the job done. They expected regular updates, but weren't watching me every second. They knew I could be trusted to get the job done.

- They kept us informed. They had brief, regular meetings (once a week) in which they let us know what work was going on and what was coming up. Each person would give a statement on their progress. We knew what was happening and what we could do to help.

- They were there when I needed help. If I had a question, they had the answer or more often they knew someone who did. The second is really important, because if you are wise, you will learn the people who are the experts from a good boss.

- They utilized my skills. At first they assigned me projects that were in keeping with what they knew my skills were. As I continued to prove myself, they gave me bigger tasks. Eventually when I showed that I was a go-getter and could find work in the company, they let me locate my own tasks. After that they only assigned me work when they needed my expertise, or if they felt I would really enjoy the job. I worked really hard for them, and still think the world of them.

- They shielded us from above. The took whatever junk was coming down from above, and while they would let us know if we messed up, they would not let us catch it from above. We caught it from them alone. This may not sound great until you are there. You need to find out when you are coming up short, you just need to hear it from someone close but who is in charge. Reprimands from a couple levels up would not be tolerable. A well handled reprimand from your immediate boss can be motivating.

- They gave us credit for the good things we did. The funny thing many bosses don't get is that they look best to everyone when they are giving credit to their people. Far from lessening their prestige it build it, and builds loyalty of their people.

Part II

Engineering Design

Chapter 3

What is Engineering?

3.1 Fun Quotes

Thanks to Harry T. Roman of East Orange, N.J., USA, who compiled the following 21 definitions of Engineering.

> The application of science to the common purpose of life.
>
> Count Rumford (1799)

> Engineering is the art of directing the great sources of power in nature for the use and convenience of man.
>
> Thomas Tredgold (1828)

> It would be well if engineering were less generally thought of, and even defined, as the art of constructing. In a certain sense it is rather the art of not constructing; or, to define it rudely but not inaptly, it is the art of doing that well with one dollar which any bungler can be with two after a fashion.
>
> A. M. Wellington (1887)

> Engineering is the art of organizing and directing men and controlling the forces and materials of nature for the benefit of the human race.
>
> Henry G. Stott (1907)

> Engineering is the science of economy, of conserving the energy, kinetic and potential, provided and stored up by nature for the use of man. It is the business of engineering to utilize this energy to the best advantage, so that there may be the least possible waste.

Willard A. Smith (1908)

Engineering is the conscious application of science to the problems of economic production.

H. P. Gillette (1910)

Engineering is the art or science of utilizing, directing or instructing others in the utilization of the principles, forces, properties and substance of nature in the production, manufacture, construction, operation and use of things ... or of means, methods, machines, devices and structures ...

Alfred W. Kiddle (1920)

Engineering is the practice of safe and economic application of the scientific laws governing the forces and materials of nature by means of organization, design and construction, for the general benefit of mankind.

S. E. Lindsay (1920)

Engineering is an activity other than purely manual and physical work which brings about the utilization of the materials and laws of nature for the good of humanity.

R. E. Hellmund (1929)

Engineering is the science and art of efficient dealing with materials and forces ... it involves the most economic design and execution ... assuring, when properly performed, the most advantageous combination of accuracy, safety, durability, speed, simplicity, efficiency, and economy possible for the conditions of design and service.

J. A. L. Waddell, Frank W. Skinner, and H. E. Wessman (1933)

Engineering is the professional and systematic application of science to the efficient utilization of natural resources to produce wealth.

T. J. Hoover and J. C. L. Fish (1941)

The activity characteristic of professional engineering is the design of structures, machines, circuits, or processes, or of combinations of these elements into systems or plants and the analysis and prediction of their performance and costs under specified working conditions.

M. P. O'Brien (1954)

The ideal engineer is a composite ... He is not a scientist, he is not a mathematician, he is not a sociologist or a writer; but he may use the knowledge and techniques of any or all of these disciplines in solving engineering problems.

N. W. Dougherty (1955)

Engineers participate in the activities which make the resources of nature available in a form beneficial to man and provide systems which will perform optimally and economically.

L. M. K. Boelter (1957)

The engineer is the key figure in the material progress of the world. It is his engineering that makes a reality of the potential value of science by translating scientific knowledge into tools, resources, energy and labor to bring them into the service of man ... To make contributions of this kind the engineer requires the imagination to visualize the needs of society and to appreciate what is possible as well as the technological and broad social age understanding to bring his vision to reality.

Sir Eric Ashby (1958)

The engineer has been, and is, a maker of history.

James Kip Finch (1960)

Engineering is the profession in which a knowledge of the mathematical and natural sciences gained by study, experience, and practice is applied with judgment to develop ways to utilize, economically, the materials and forces of nature for the benefit of mankind.

Engineers Council for Professional Development (1961/1979)

Engineering is the professional art of applying science to the optimum conversion of natural resources to the benefit of man.

Ralph J. Smith (1962)

> Engineering is not merely knowing and being knowledgeable, like a walking encyclopedia; engineering is not merely analysis; engineering is not merely the possession of the capacity to get elegant solutions to non-existent engineering problems; engineering is practicing the art of the organized forcing of technological change ... Engineers operate at the interface between science and society ...
>
> Dean Gordon Brown; Massachusetts Institute of Technology (1962)

> The story of civilization is, in a sense, the story of engineering - that long and arduous struggle to make the forces of nature work for man's good.
>
> L. Sprague DeCamp (1963)

> Engineering is the art or science of making practical.
>
> Samuel C. Florman (1976)

3.1.1 Class Quotes

> Engineers use vision and resources to create a design that is cost-efficient and maximizes performance.
>
> Jonathan Carranza, Jason Fredrick, Charles Korman, Sarah Wade (2008)

> Engineering is a creative and altruistic way to benefit society using broad knowledge and tools efficiently to design technology.
>
> Maged Assad, Diane Bernal, Erin Powers, and Amber Thomas (2008)

> If you are going to reinvent the wheel, do it as optimally, efficiently as possible... and damnit make a better wheel! And remember when designing embrace the laziness.
>
> Tim Castelli, Steven Parker, and Sachithra Udunuwarage (2008)

3.2 Fields

The "Big Three":

Civil The name comes from civilian rather than military engineering, but is used of large construction projects such as buildings and roads.

Electrical Design of electrical or electronic components. Includes such areas as analog and digital circuits, electro-magnetics, signals, power, controls, estimation. This has many sub-fields such as ours - computer engineering.

Mechanical Design of machines and devices. Typically includes such areas as statics, dynamics, materials, fluids, and thermodynamics. This has many sub-fields such as industrial and aerospace.

3.3 Profession

Engineering is a true profession, as opposed to many areas of study often so called because engineering has a licensing process. To become licensed, you must pass the EIT/FE[1] exam, then work a number of years under a professional engineer (PE) or in certain fields (such a aerospace) or get education credit (ABET accredited schools only)[2], then pass a second exam (PE exam)[3]. Typically continuing education courses are required for renewal.

As a profession, engineering is overseen by state laws and professional societies. Ethics in engineering is thus not an arbitrary concept, it is a professional standard and requirement on our actions.

[1] Engineer-In-Training/Fundamentals of Engineering exam is a one day, 8 hour exam with around a 20% pass rate when I took it.

[2] The number of years required varies by the area you are trying to get a license in.

[3] A 2 day exam of about 8 hours a day.

Chapter 4

Communication

4.1 Weekly Status Report

4.1.1 Team Leaders

Weekly status reports have four basic sections: scheduled tasks, accomplished tasks, meetings, project status, goals/assignments for next period. Much of this revolves around the schedule/Gantt chart. The entire report should be 1-2 pages in length. Shorter is better if it covers the points.

In the first section, you should give a brief statement of what was the goals for the week, and how they align with or derive from the schedule/Gantt chart. The person or people assigned should be included. Brief means enough detail is given to understand what was to happen and how it fits in the larger program. Do not give so much that it takes to much to read (a paragraph or two is normal), or so little that the reader can't judge what was done and how it will impact things. Note that the schedule can get modifications and tasks can get added, deleted, or amended, so it is important to cite the date or revision of the schedule/Gantt chart.

In the next section you should cover what tasks were actually accomplished, and if this met the goals. Be sure to cite key players, as this is how people get recognition and thus raises and promotions. It is the goal of a manager to help your employees improve and get rewards. These should also be brief (read: get to the point), as whoever is reading this will probably be reading several of these. People need to know precisely what has been accomplished, not a narrative on the accomplishment. You can give a short statement on why something was difficult if it is only occasionally done.

The meetings section should state the attendance, reason (main agenda items) and results or conclusions.

Next, the weekly status report should show the progress incrementally and cumulatively, i.e. what progress did you make this week as a percent/fraction of the weeks goals, and what is your total progress as a percent/fraction of the total project to date. This should sum up in a numerical way what has been accomplished this week and where the entire project stands. Including an updated Gantt chart is optional, but can be very illuminating.

Finally, the status report should have the assignments or goals for each member of the team by name, giving the task, numerical estimate of completion, and task title.

4.1.2 Team Members

Each team member needs to send their status reports so that the team leader can send a team status report to their superiors, such as project engineers in industry, or professors and company liaisons in college design programs. Team members have three things they have to report

1. Goals for the week. These are usually assigned by the team leader, or at least with the team leader's cognizance, but you are trying to make their life easy and show them that you understood your tasks.

2. Accomplishments for the week. This could be a percentage of work done for a task that is larger or the whole task if you finished it. Let them know if it is ahead or behind schedule and any positive notes. Be brief (not rambling) but don't cut out important facts. This is a bit of an art, but basically you are trying to let them know the great things you did, and what is going on, but not take so long they don't want to read it. As a good guide try two sentences per task, the first stating what amount is done, the second stating descriptively what was accomplished.

3. Assignments for the next week.

I have included a sample of a good status report in figure 4.1. Note the problem/alteration/roadblock was put in italics to emphasize it.

4.2 Log Book

Log books are a necessary legal documentation of work that are crucial for a variety of cases including patents. To be useful in court the pages must not be removed or inserted (thus no binders). Blank pages cannot be left, and all pages must be used in sequence. You cannot leave any large blank space, as this would allow for future tampering. Items cannot be "whited out" or scribbled over. Any corrections must be done by a single, thin line through the error.

Each entry should have a descriptive heading with the date, and should be legible and understandable. The log book must contain all work done, including: meeting notes, sketches, ideas, concepts, calculations, phone call notes, research notes, test data, etc. Logbooks are primarily a daily record of what you thought and did, what your goals were, and how you were progressing. It ought to be more than a diary of general activity. Make sure you label sketches/drawings. Include relevant dimensions or scale. Include assembly diagram for context. Note block diagrams are particularly useful.

Think of future users, and evaluate your logbook from this perspective. Make sure you add enough detail to the notes of design specifications and estimates to make them

Figure 4.1: Sample Team Member Status Report.

Keith Evan Schubert
Associate Engineer
Widget Project

Goals:
All tasks are from Widget Project Gantt chart, 4-17-08.

1. Finish task 3, model system dynamics widget driven by oscillator circuit.

2. Do 20% of task 4, analyze stability of widget.

3. Do task 8, select widget antenna.

Accomplishments:
All tasks are from Widget Project Gantt chart, 4-17-08.

1. Task 3, model system dynamics widget driven by oscillator circuit is 100% complete. The modeling took one day longer than anticipated because nonlinearities in the oscillator proved to be too large to ignore, necessitating a more complicated model. The non-linearity enters only as an input, but can be handled by augmenting the states. The results are in technical report TR-00743-23: Widget Dynamics.

2. Task 4, analyze stability of widget is 10% done, as the non-linear states, which handle the inputs, have been analyzed. The non-linear states form a stable, passive manifold that is bounded-input, bounded-output stable. This will still create complexity in calculating the closed loop control law, which might require an extra week to examine robust control methods if the non-linear control proves infeasible.

3. Task 8, select widget antenna is 100% complete. The Antenna Industries, 9" whip antenna model # 91-471128-678 rev B.

Assignments:
All tasks are from Widget Project Gantt chart, 4-17-08.

1. Continue task 4, analyze stability of widget. Achievement of 70% completion was the original goal, but *given the alterations in the problem complexity an adjustment to 50% completion is suggested*, which would entail completion up to nonlinear analysis, but not a generation of a candidate control law.

reconstructible later. If you use an outside formula or result, cite the source so others can find it. Make sure you include enough detail to facilitate future users reconstruction of your efforts. Note even your dead ends and why you abandoned them, as this information is of particular assistance. Relevant loose materials can be attached to a logbook if small enough, or they can be cited in a clear distinguishable way, and stored in a different location (such as a notebook or filing cabinet). Identify contacts and communications by name, company, location/number, and date for future reference.

Chapter 5

Quality

In the 80's and 90's it was called Total Quality Management, in the 2000's it is called Six Sigma design, what it really means is to do the right thing the right way. This involves a lot of work and is not as easy as I make it sound. In this chapter we will go over many of the pillars of this process.

Six goals of quality programs:

1. Customer focus (internal and external, this is the most popular emphasis)
2. Leadership and individual responsibility (let them do their job)
3. Integrity (quality as a way of life)
4. Enhance communication (avoid silly mistakes)
5. Recognize and reward employees (read find the best and keep them)
6. Streamline processes (save money, the real goal)

The goals are achieved by:

1. Customer focus (Kind of circular don't you think, you could also think of this a statement of the obvious, which sadly most people don't do.)
2. Involvement (The entire organization must support the goals, as a spoiler is deadly.)
3. Metrics (You have to have measurements to improve something)
4. Support (Budget, planning, etc.)
5. Continuous improvement (None of us is perfect, so try to get better at everything in a deliberate fashion.)

Quality programs are all about doing the right things, the right ways. You will incur certain necessary costs to do things right, but you will avoid many other costs and keep happy customers. Most big companies have some sort of quality program, but varying degrees of implementation. Most small companies have no formal system but are more naturally connected to customer needs. In the end, you need to get the customer what is required for the best price and profit.

The moral to me is

Doing it right costs, doing it wrong costs more.

Before we examine the five ways we achieve our goals, we need to scope our discussion to the processes we will be working on. We can break down all we do, including our engineering designs (and the subsequent manufacturing) into a series of processes. Roughly, a process is a series of related tasks that produce a product. The product need not be physical or the final item. It is often useful to note that processes act on an input to produce a desired output according to a specification. Processes have:

boundaries Identify who are the actors and what they must do (responsibilities).

suppliers Produce inputs to according to the process's requirements.

customers Receive outputs and specify requirements.

The best processes are:

Effective Does it work? That is does it produce an output that meets the requirements.

Efficient Does it produce the output at the lowest cost?

Controlled The process is documented, and has metrics that are used to update and improve the process.

Adaptable Does it have built-in mechanisms that allow it to meet new requirements?

Processes must be analyzed and made to be effective, efficient, controlled, and adaptable. Common symptoms that indicate a problem are: customer complaints/dissatisfaction, returns, redos, unresolved issues, excessive workload/overtime, missed deadlines, budget difficulties, bad morale, turnover, productivity drops, constantly changing requirements. Note that most processes aren't going to meet the ideal. You need to find the most problematic and fix it, then repeat.

5.1 Customer Focus

Know your customer, and how your customer uses your product. An ideal customer-supplier relationship is aligned, that is, the supplier's abilities match the customer's requirements. To ensure alignment you need to ask what the customer needs, and what will they use it for. Particularly ask if there are any gaps between what is received and what is needed, often this opens up new opportunities. Make sure you keep written notes of conversations, and have people ok your summaries and requirement specifications. Be inclusive in the meetings, by which I mean you should actively invite those who have an interest in the product/service to be supplied. Make sure you have though things through before you meet so you don't look uninformed or ill-prepared. Think about what they will say and plan questions to elicit their inputs and concerns.

One way of thinking about customer focus and establishing requirements is the acronym PRIDE:

Product/service Does it meet the customer's requirements, needs, and wants? Note they are different. Did you meet them in a pleasing way?

Relationship Is their trust? How about a casual friendship?

Integrity Can you meet the requirements? What will you do if there is a problem?

Delivery Does it arrive on time and budget? Is it usable?

Expense Is the customer happy with the value? Is the price competitive?

5.2 Continuous Improvement

Have you ever heard this saying?

If it ain't broke, don't fix it.

There are certainly times when this is the right course of action[1], but often it is not true. If man followed that saying in all areas, we would be hunting for dinner with a pointed stick[2]. Improvement is risky but the rewards are immeasurable. Improvement is what engineering is all about.

A popular rule is the 1-10-100 rule. Basically it says if you catch a thing in design it costs 1 buck, if in manufacturing it costs 10, and if your customer finds it it will cost you a 100. The point of this rule is to underscore the need for improvement. Four techniques that help improve are active listening, visualization, the why technique, and contingency diagrams.

1. Active listening means paying close attention to what everyone is saying. This involves having your mind and ears working together. Often great ideas or major dangers are brought up in discussions, but don't get noticed or are ignored. If we work hard at listening to others and thinking of their point and perspective then we will all benefit.

[1] Most quality people would shoot me for that, but it is true sometimes. Knowing when to leave it alone and when to improve it is an art you will develop over your life. You can't always trust little sayings, or always reject them. You must learn to think and assess on your own the best ways to handle things. Sometimes that means making and following rules. Sometimes you will have to break the rules. Great engineers and engineering managers know how and when to do that.

[2] Come to think of it, this sounds too much like a fun vacation or a cable survival show. Hopefully you get the point anyway.

2. Visualization is the process where we imagine what we are making, how it will be used and misused, and how it benefits others. Think over the product. Put yourself in the place of needing it and using it. Many people do not consider what they are working on and thus make junk, don't join their ranks.

3. The why technique helps find causes by repeatedly asking why something happens. It is simple, but surprisingly effective, which is a good combination. It is particularly useful in tracking down a failure, but can be used proactively, particularly in trying to figure out how to make something work or to come up with new features.

4. Contingency diagrams[3] are a simple drawing in which a problem is listed in a bubble and the contributing factors listed on arrows pointing to the bubble. For each arrow, one or more solutions are developed. Like the why technique, it is particularly useful in tracking down a failure, but can also be used proactively, in trying to figure out how to make something work or to come up with new features.

Continuous improvement also applies to you. Your skills will degenerate and your mind grow weaker if you don't work at improving them. Do mind challenges like crosswords, sodoku, or brain teasers. Read popularizations of math, physics, computer science, engineering, philosophy, or history. I have a list of suggested books on my r2labs.org website if you need suggestions. Join and be active in professional societies like IEEE, SIAM, and ACM. Read the latest advancements in science and technology. Take certification courses. In short, keep getting better.

5.3 Manager's Side

Managers are responsible for the guiding and empowering of their employees to achieve success for the company. Many managers sadly see their job as keeping costs down, and thus they avoid improving things. One improvement frequently dumped for cost reasons is training. I don't know who first said it, but it is very true,

> *Training is a non-recurring overhead expense;*
> *Ignorance is a recurring direct charge.*

As a manager there are several things you can do to help your team improve.

- Give your group the big picture. To do a truly great job, and constantly improve, they need to know what is really going on. Many improvements happen when people realize how others use their work.

- Actively solicit new ideas. Most people are afraid to say things and contribute. You must make them feel comfortable with you and then seek their input and ideas. Avoid defensiveness and finger pointing (two common actions) and replace them with a desire to learn from mistakes and an open sharing of knowledge.

[3] Personally I dislike the name, and have never thought it was very descriptive, but I am stuck with it.

- Encourage your team to communicate directly with their customer. Make each person responsible for finding out how their work is used and empower them to do so. Bad managers try to control information flow. If you have an employee who is not able to communicate with customers then you have three choices: train them, pair them with a more skillful person, or if all else fails fire them.
- Have the team flowchart their work process and look for ways to improve it. Note that this is really helpful, but if you do it when a major task is due, you are crazy. Quality takes time and effort, no matter what people may tell you.
- Model high standards in your work and reward your team for their work. Encourage and reward when they succeed, you never lose by doing this as it reflects well on you.
- Seek and grow the best people on your team. Make the environment supportive, open, and friendly. People want to work in these type of places, and will often work harder for less if they are working for and with friends. This is not a license to take advantage of people, rather it means that if you all help each other, then you will all succeed together and be happy.

One thing you should keep to a minimum is meetings. Meetings are necessary to keep people informed, but they block real work and create frustration in employees (particularly the good ones). Make sure meetings are planed and stay on track.

Often you will need to guide your team to solve a problem. Here is a four step method I learned at Grumman Aerospace (now Northrop-Grumman).

1. Focus
    (a) List of problems (brainstorm)
    (b) Select one problem (selection grid)
    (c) Verify/Define problem (impact analysis)

    Output: problem statement
2. Analyze
    (a) Identify needed information (checklist)
    (b) Collect data (sampling, surveys, literature)
    (c) Find major factors (statistics, flowchart, fishbone, why)

    Output: baseline data, and list of contributors
3. Develop
    (a) List possible solutions (brainstorming, literature)
    (b) Select best solution (cost-benefit analysis)
    (c) Develop plan (task list, gantt chart, procedure)

Output: plan

4. Execute

   (a) Buy-in (presentations, reports/papers, individual discussions)
   (b) Implementation
   (c) Monitoring and tuning

   Output: metrics

5.4 How to Brainstorm

This sounds easy right. Get a bunch of people together and think up solutions. Sadly, these often fail because of poor implementation. I strongly suggest you follow all of the following:

1. Have one person write the ideas on the board, and another document for later use in a notebook.

2. No criticism or analysis can be done. Not even a comment of "that is the same thing" as this will disturb the process and scare people from contributing. Things that come up repeatedly might be more important, and variants sometimes have important improvements.

3. Each person gets one contribution per turn, and turns go in sequence. Give people paper to jot down ideas so they don't forget. If you don't do this you run the risk of cutting some people out. People can say pass, but should be encouraged to say what they think, even if it is a repeat.

Chapter 6

Engineering Economics

Engineers have to consider the cost of the items designed, thus we must know some economics. This is an introduction to engineering economics. Those regularly dealing with costing are encouraged to study other references for a deeper coverage. This should get you started though. Some standard variables I will use are:

P Present sum of money, sometimes called principle though this does not always make sense.

F Future sum of money, at n periods in the future.

A Annuity, or regular payment of the same amount.

G Geometric payment series.

i Interest rate per compounding period, usually a year.

n Number of interest periods.

r Nominal interest rate per year, does not consider effect of subperiod compounding.

m Number of compounding subperiods.

6.1 Simple Interest

We want to study finance, and to do so we must look at interest. The most basic kind of interest is simple interest. While not used in the US, simple interest is used in some countries in Africa. Let's get a few terms down first.

Principle (P) the amount borrowed or initial balance.

Future worth (F) worth of the investment at that interest rate.

Interest rate (i) the percent of the principle charged for the loan.

The interest charged each year is the same in simple interest, and can be found by multiplying the interest rate times the amount owed, Pi. Simple interest is an example of arithmetic growth. The series that shows how much must be paid for a simple interest loan of P at rate i for n years is

$$\begin{aligned} F &= P + Pi + Pi + \cdots + Pi \\ &= P + nPi \\ &= P(1 + ni) \end{aligned} \tag{6.1}$$

6.2 Compound Interest

In the US we do not use simple interest, but rather compound interest. Before we get into the formal ideas, let's see why it is called compound interest.

Say you put P dollars in a savings account at an interest rate of i. We will assume the interest is calculated and added to your account only once a year. The interest earned at the end of the year is Pi, just as in simple interest. After 1 year you have $P(1 + i)$ dollars. If you leave the money in the account then the next year you will have a new initial amount of $P(1 + i)$, so the interest paid will be $(P(1 + i))i$. The total money you have will be $P(1+i)(1+i) = P(1+i)^2$. As this continues, the money you have at the end of n years is

$$F = P(1 + i)^n. \tag{6.2}$$

The interest builds upon itself to help you, or compound your gain.

We do not have to compound (or add) the interest just once a year, you can compound a number of times a year. Say we compounded interest m times per year, at the end of a year we would have $P(1 + i)^m$ dollars, right?. Interestingly (pardon the pun) no bank uses the same i in yearly interest compounding as in more frequent compounding. The interest used when interest is compounded more than once a year is $r = \frac{i}{m}$. If a bank lists that an account pays a nominal (APR) of $i = rm$ that is compounded m times per year, the rate used in the compounding equation is r. The formula reads $P(1 + \frac{i}{m})^m = P(1 + r)^m$. If we leave our money in for two years we would have $P(1 + \frac{i}{m})^{2m} = P(1 + r)^{2m}$. Sometimes instead of listing the nominal interest (APR), a bank will list the effective rate (APY). This comes from noticing that $(1 + r)^m > 1$, so we can write it as $(1 + r)^m = 1 + r_e$. We then solve for r_e to find that $r_e = (1 + r)^m - 1$. You would have this actual interest rate if the account were compounded annually. After n years, the balance will have grown to

$$F = P(1 + \frac{i}{m})^{nm} \tag{6.3}$$

$$= P(1 + r)^{nm} \tag{6.4}$$

$$= P(1 + r_e)^n. \tag{6.5}$$

This series is a geometric series. This is a commonly calculated equation, so we will do a little bit of work to get it into an even more useful form.

$$\begin{aligned} F &= P(1+r)^k \\ \frac{F}{P} &= (1+r)^k \qquad (6.6) \\ &= (F/P,\ r,\ k) \qquad (6.7) \end{aligned}$$

The final equation (eq. 6.7) is called the **single payment compound amount factor**, and is tabulated in many places for different interest rates. This allows for easy calculation. It is also easy to code the formula for programs.

6.3 Easy Estimation

One challenge with compounded interest is estimating what something is worth in 20 years at some interest rate. There is an easy way to estimate this value.

> **Rule of 69** A sum invested at an interest rate, r, will double every k years where, $k = \frac{69}{100r}$.

For example, at 3% interest, an investment doubles every $\frac{69}{3} = 23$ years. Try it $(1.03)^{23} \approx 1.97$, a very good approximation. This can be used to rapidly estimate values.

How does it work? Magic of course. Just kidding. Here is the proof

$$\begin{aligned} 2 &= (1+r)^t \qquad (6.8) \\ \ln(2) &= t\ln(1+r) \qquad (6.9) \\ t &= \frac{\ln(2)}{\ln(1+r)} \qquad (6.10) \\ t &\approx \frac{0.69}{r} \qquad (6.11) \end{aligned}$$

6.4 Continuous Compounding

You might have noticed something when we looked at multiple compoundings per year. You might have noticed that the more frequent the compoundings, the effective interest rates (APY) increased but not as much each time. Let's look at this trend more closely, see Figure 6.1.

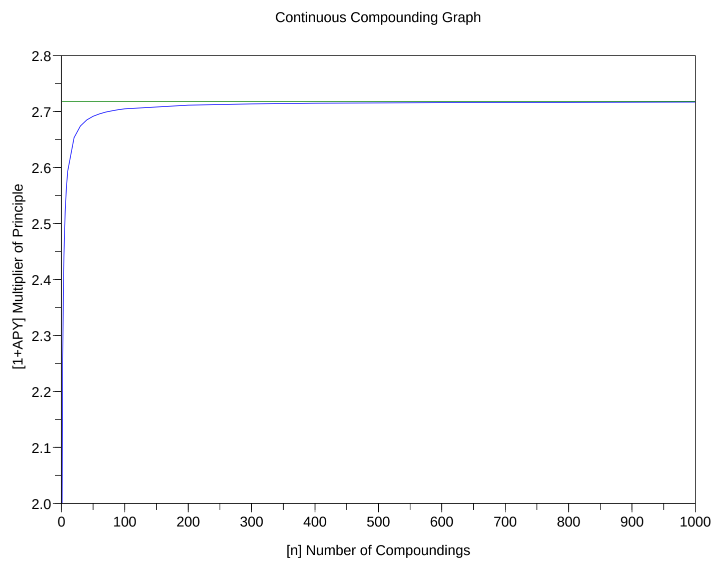
The final term is a special number in mathematics called e. There are many similarities between the constants e and π. Here are a few.

1. Both arise naturally in calculations. Circles, curves, and angles give rise to π, while compounding and calculus operations on exponents give rise to e.

2. Both numbers are not rational, which means they cannot be expressed as the ratio of integers.

Figure 6.1: Continuous Compounding (digits rounded to 8 for display purposes)

n	$1 + APY$
1.	2.
2.	2.25
3.	2.3703704
4.	2.4414062
5.	2.48832
10.	2.5937425
20.	2.6532977
30.	2.6743188
40.	2.6850638
50.	2.691588
100.	2.7048138
200.	2.7115171
300.	2.7137652
400.	2.7148917
500.	2.7155685
1000.	2.7169239
Inf	2.7182818



3. Both numbers are not algebraic and thus they are transcendental. An algebraic number is one which can be expressed as the root of a polynomial with integer coefficients.

4. Both have no repeating patterns in their decimal expansion. This is a result of being irrational.

6.5 Fun with Numbers

I thought I would take this time to give you a little taste of Number Theory. This is the type of thing which mathematicians find fun. Lets look a few of them.

First, consider a repeating decimal like 0.345345345 We would like to know how to express this as a fraction. We know 345/1000 gives us 0.345, which is a little smaller than we want, so we make the denominator a little smaller. We try 345/999 and we find that indeed we get the desired result. Try the following

7/9	75/99	752/999	7528/9999
5/9	50/99	504/999	5046/9999
1/9	12/99	123/999	1234/9999

Why does this work? Consider the following.

Let some number sequence of $n+1$ digits be given by $d_0 d_1 \cdots d_n$. To make it easy to follow note that $n+1$ digits of all 9's is specified by $9_0 9_1 \cdots 9_n$. Finally we will look at the

Table 6.1: Digit recursion.

x	explanation
$0.d_0d_1\cdots d_n$	The first $n+1$ digits must be from the fixed number since the shift is large enough so there is no overlap.
$0.d_0d_1\cdots d_n\ d_0d_1\cdots d_n$	Since we know the first $n+1$ digits, we know from the shift term that the first $n+1$ digits are copied and shifted to the second $n+1$ digits.
$0.d_0d_1\cdots d_n\ d_0d_1\cdots d_n\ d_0d_1\cdots d_n$	Since we know the second $n+1$ digits, we know from the shift term that the second $n+1$ digits are copied and shifted to the third $n+1$ digits.
$\vdots$	The process repeats, generating the pattern.

ratio,

$$\begin{aligned}
x &= \frac{d_0d_1\cdots d_n}{9_09_1\cdots 9_n} \\
&= \frac{d_0d_1\cdots d_n}{10^{n+1}-1} \\
x(10^{n+1}-1) &= d_0d_1\cdots d_n \\
x10^{n+1} &= d_0d_1\cdots d_n + x \\
x &= \frac{d_0d_1\cdots d_n}{10^{n+1}} + x\frac{1}{10^{n+1}} \\
&= 0.d_0d_1\cdots d_n + x\frac{1}{10^{n+1}}
\end{aligned}$$

Think about what this equation means. The ratio, x, is equal to a fixed length ($n+1$ digits) decimal number and a shifted (by $n+1$) copy of itself. Since the shift is equal to the size of the fixed number, they don't overlap. We can then see that this can be handled recursively, see Table 6.1.

Note the term in square brackets is repeat of the term on the on line two so we have for 345/999, that it is 0.345(1.001001...) or 0.345345345...We have seen that this works in general. Try 9/9. We should see 0.99999 from what we have shown, but we see 1! Why? Well, 1=0.99999...! Here is an easy way to see this.

$$\begin{aligned}
y &= 0.999\cdots \\
10y &= 9.999\cdots \\
10y - y &= 9.999\cdots - 0.999\cdots \\
9y &= 9 \\
y &= 1
\end{aligned}$$

An amazing result! Anyway, I hope this little diversion sparks an interest in number theory.

6.6 Amortization and Savings

Up till now we have looked at a single starting value and what it will be worth (or cost). Now we will consider amortization and regular payments. Say we are going to make regular payments of A for m periods per year and n years. Further, say you will get $r = \frac{i}{m}$ as an interest rate. We can treat this as a bunch of compounded equations that are added together.

$$\begin{aligned} F &= A(1+r)^{nm-1} + A(1+r)^{nm-2} + \\ &\quad \ldots + A(1+r) + A \\ &= \sum_{k=0}^{nm-1} A(1+r)^k \\ &= A\sum_{k=0}^{nm-1} (1+r)^k \end{aligned} \tag{6.12}$$

Note that this is nice looking series, so we might suspect there is an easier way of summing. Consider the series

$$s_m(x) = \sum_{k=0}^{m-1} x^k, \tag{6.13}$$

then multiply by $\frac{(x-1)}{(x-1)}$ to obtain

$$\begin{aligned} s_m(x) &= \frac{x s_m(x) - s_m(x)}{x-1} \\ &= \frac{x\left(x^m + s_{m-1}(x)\right)}{x-1} \\ &\quad - \frac{(x s_{m-1}(x) + 1)}{x-1} \\ &= \frac{x^m + x s_{m-1}(x)}{x-1} \\ &\quad - \frac{x s_{m-1}(x) - 1}{x-1} \\ &= \frac{x^m - 1}{x-1}. \end{aligned} \tag{6.14}$$

We note that for us $x = 1 + r$ so we have

$$
\begin{aligned}
F &= A \sum_{k=0}^{nm-1} (1+r)^k \\
&= A \frac{(1+r)^{nm} - 1}{(1+r) - 1} \\
&= A \frac{(1+r)^{nm} - 1}{r} \\
\frac{F}{A} &= \frac{(1+r)^{nm} - 1}{r} \qquad (6.15) \\
&= \left(\frac{F}{A},\ r,\ nm \right). \qquad (6.16)
\end{aligned}
$$

This is called the amortization or savings formula.

When paying off a mortgage or loan, the future value of the amortization (your payment) and the future value of the initial amount of the loan are set equal. Assuming multiple compoundings we have

$$
\begin{aligned}
P(1+r)^{nm} &= A \frac{(1+r)^{nm} - 1}{r} \\
A &= \frac{P(1+r)^{nm} r}{(1+r)^{nm} - 1} \\
\frac{A}{P} &= \frac{r(1+r)^{nm}}{(1+r)^{nm} - 1} \\
&= \left(\frac{A}{P},\ r,\ nm \right). \qquad (6.17)
\end{aligned}
$$

The final equation (eq. 6.17) is called the **Capital Recovery Factor**, and it is tabulated for various values of interest and compounding periods. Notice that this is the formula used for loans (loan value is P, monthly payment is A, and the number of monthly payments is nm). The reciprocal,

$$
\begin{aligned}
\frac{P}{A} &= \frac{(1+r)^{nm} - 1}{r(1+r)^{nm}} \qquad (6.18) \\
&= \left(\frac{P}{A},\ r,\ nm \right), \qquad (6.19)
\end{aligned}
$$

is also an important equation, and is called the **Present Worth Factor, Uniform Series**.

6.7 Arithmetic Gradients

6.7.1 Compound amount, arithmetic gradient

$$\frac{F}{G} = \left(\frac{(1+r)^{nm} - (1+rn)}{r^2}\right) \tag{6.20}$$

$$= \left(\frac{F}{G},\ r,\ nm\right) \tag{6.21}$$

6.7.2 Present worth factor, arithmetic gradient

$$\frac{P}{G} = \left(\frac{(1+r)^{nm} - (1+rn)}{r^2(1+r)^{nm}}\right) \tag{6.22}$$

$$= \left(\frac{P}{G},\ r,\ nm\right) \tag{6.23}$$

6.8 Returns and Costs

6.8.1 Equivalent Uniform Annual Cost(EUAC)

$$EUAC = \left(\frac{A}{P},\ r,\ k\right)P - \left(\frac{A}{F},\ r,\ k\right)S \tag{6.24}$$

where P is the cost of the item, and S is the salvage value (what can you sell it for when you are done).

6.8.2 Equivalent Uniform Annual Benefit(EUAB)

$$EUAB = A - \left(\frac{A}{P},\ r,\ k\right)P \tag{6.25}$$

where P is the cost of the investment, and A is the periodic return.

6.8.3 Net Present Worth(NPW)

Net present worth (NPW) is defined by

$$NPW = \left(\frac{P}{A},\ r,\ k\right)A - P \tag{6.26}$$

where P is the cost of the investment, and A is the periodic return.

6.8.4 Rate of Return(ROR)

The rate of return is the interest rate, which makes the net present worth equal zero. The rate of return is thus:

$$\begin{aligned} ROR &= r \\ & \quad s.t.\ NPW = 0, \end{aligned} \tag{6.27}$$

where $s.t.$ means "such that"[1]. ROR is usually evaluated with respect to a minimum attractive rate of return (MARR), which is selected to reflect current options and relative risk of the investment. MARR can be selected to be a safe investment's ROR (such as a savings account or secure bonds).

6.9 Depreciation

We have seen how things grow. We also need to see how things lose value. Major assets (homes, cars, computers, etc.) do not retain their value. They get used and thus worth less to others, some more than others. For instance, cars tend to lose 25% of their value a year until they are in the \$1000 range where they tend to hover (unless they are a classic). Houses on the other hand tend not to lose a lot of value and actually start gaining value after about 5-10 years of age. Depreciation works just like interest, except the rate is negative.

Consider a new car that costs \$20,000 and loses 25% of its value each year. How much is it worth in 1, 2, 5, 10 years?

Years	Formula	Value
0	$20000(1-0.25)^0$	\$20,000
1	$20000(1-0.25)^1$	\$15,000
2	$20000(1-0.25)^2$	\$11,250
5	$20000(1-0.25)^5$	\$4,746
10	$20000(1-0.25)^{10}$	\$1,126

People often find it difficult to calculate this, so other methods are often used. Some of the most common are below.

6.9.1 Straight Line Depreciation

$$D_k = \frac{C-S}{n} \tag{6.28}$$

where D_k is the depreciation in year k, C is the cost of the item, S is the salvage value (what you get for selling it at the end, and n is the number of years till you salvage it.

[1]Such that means that there is a constraint that must be fulfilled. It is frequently used in constrained minimization, but can be used - as it is here - to specify a one parameter system from which any solution will do.

6.9.2 Accelerated Cost Recovery System (ACRS)

$$D_k = W_j(C - S) \tag{6.29}$$

where D_k is the depreciation in year k, C is the cost of the item, S is the salvage value (what you get for selling it at the end, and W_j is from Table 6.2.

Table 6.2: ACRS Weights

	Recovery Period			
Year	3	5	7	10
1	.333	.200	.143	.100
2	.445	.320	.245	.180
3	.148	.192	.175	.144
4	.074	.115	.125	.115
5		.115	.089	.092
6		.058	.089	.074
7			.089	.066
8			.045	.066
9				.065
10				.065
11				.033

6.10 Non-Renewable Resources

If you have a sum of money, a quantity of oil, etc., that you cannot replace then it is a non-renewable resource. There are two ways you can use it (in practical situation, theoretically you can use it in infinitely many ways). Our primary interest is when the resource will be used up.

The first is static consumption. In it you use the same amount, A, every period. The resource will last

$$t_s = \frac{P}{A} \tag{6.30}$$

The second type is exponential consumption. In it you start by using A the first period, and then you use A(1+r) the next period, and so on. The equation ends up looking like the savings formula (though we are withdrawing not depositing). In any case the formula is

$$\begin{aligned} P &= A + A(1+r) + \cdots + A(1+r)^{t_e - 1} \\ &= A\frac{(1+r)^{t_e} - 1}{r} \end{aligned}$$

We can then solve for t_e, the time when it is all gone.

$$\frac{P}{A}r + 1 = (1+r)^{t_e} \tag{6.31}$$

$$\ln\left(\frac{P}{A}r + 1\right) = t_e \ln(1+r) \tag{6.32}$$

$$t_e = \frac{\ln\left(\frac{P}{A}r + 1\right)}{\ln(1+r)} \tag{6.33}$$

6.11 Renewable Resources

If you have a sum of money, a group of animals, an amount of plants, etc., that can grow, you have a renewable resource. In this case it makes sense to live off the surplus amount above that which is needed to maintain the quantity at current levels. For money you live off the interest.

If we ignore inflation the amount we get in one year is given by

$$F = P(1+r) \tag{6.34}$$

$$= P + rP. \tag{6.35}$$

We want to keep an amount P left over (for the following year) so the net gain (what we can live off) is

$$A = F - P \tag{6.36}$$

$$= P + rP - P \tag{6.37}$$

$$= rP. \tag{6.38}$$

We might want to consider inflation into this equation, so the money we have to spend each year remains of equal value. In this case the amount we need to keep from spending is $P(1+i)$, which gives us an amount to live off of

$$A = F - P(1+i) \tag{6.39}$$

$$= P + rP - P - iP \tag{6.40}$$

$$= (r-i)P. \tag{6.41}$$

Since interest rates are determined by inflation (usually r=i+1% to 3%). For instance it is reasonable to assume that if you live off $\frac{1}{50}$ of an amount you can do so forever.

Chapter 7

Engineering Ethics

Engineering ethics are really more professional codes of conduct than true ethical systems. The main difference is that codes of conduct specify conduct rather than specifying the way to evaluate a situation. Amongst professional codes of conduct, there are two general methods, specific and general. Specific codes of conduct are typified by the Association of Computer Machinist's (ACM) code of conduct, which is very detailed in what is to be done in different situations. On the other extreme is the Institute of Electrical and Electronic Engineer's (IEEE) code of conduct, which has only some general principles.

7.1 IEEE Code of Ethics

As per IEEE Bylaw I-104.14, membership in IEEE in any grade shall carry the obligation to abide by the IEEE Code of Ethics (IEEE Policy 7.8) as stated below.

We, the members of the IEEE, in recognition of the importance of our technologies in affecting the quality of life throughout the world, and in accepting a personal obligation to our profession, its members and the communities we serve, do hereby commit ourselves to the highest ethical and professional conduct and agree:

- to accept responsibility in making decisions consistent with the safety, health and welfare of the public, and to disclose promptly factors that might endanger the public or the environment;
- to avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist;
- to be honest and realistic in stating claims or estimates based on available data;
- to reject bribery in all its forms;
- to improve the understanding of technology, its appropriate application, and potential consequences;

- to maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;
- to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;
- to treat fairly all persons regardless of such factors as race, religion, gender, disability, age, or national origin;
- to avoid injuring others, their property, reputation, or employment by false or malicious action;
- to assist colleagues and co-workers in their professional development and to support them in following this code of ethics.

Approved by the IEEE Board of Directors
February 2006

7.2 ACM Code of Ethics and Professional Conduct

Adopted by ACM Council 10/16/92.

Preamble

Commitment to ethical professional conduct is expected of every member (voting members, associate members, and student members) of the Association for Computing Machinery (ACM).

This code, consisting of 24 imperatives formulated as statements of personal responsibility, identifies the elements of such a commitment. It contains many, but not all, issues professionals are likely to face. Section 1 outlines fundamental ethical considerations, while Section 2 addresses additional, more specific considerations of professional conduct. Statements in Section 3 pertain more specifically to individuals who have a leadership role, whether in the workplace or in a volunteer capacity such as with organizations like ACM. Principles involving compliance with this Code are given in Section 4.

The Code shall be supplemented by a set of Guidelines, which provide explanation to assist members in dealing with the various issues contained in the Code. It is expected that the Guidelines will be changed more frequently than the Code.

The Code and its supplemented Guidelines are intended to serve as a basis for ethical decision making in the conduct of professional work. Secondarily, they may serve as a basis for judging the merit of a formal complaint pertaining to violation of professional ethical standards.

It should be noted that although computing is not mentioned in the imperatives of Section 1, the Code is concerned with how these fundamental imperatives apply to one's conduct as a computing professional. These imperatives are expressed in a general form to emphasize that ethical principles which apply to computer ethics are derived from more general ethical principles.

It is understood that some words and phrases in a code of ethics are subject to varying interpretations, and that any ethical principle may conflict with other ethical principles in specific situations. Questions related to ethical conflicts can best be answered by thoughtful consideration of fundamental principles, rather than reliance on detailed regulations.

1. GENERAL MORAL IMPERATIVES.

As an ACM member I will ....

1.1 Contribute to society and human well-being.

This principle concerning the quality of life of all people affirms an obligation to protect fundamental human rights and to respect the diversity of all cultures. An essential aim of computing professionals is to minimize negative consequences of computing systems, including threats to health and safety. When designing or implementing systems, computing professionals must attempt to ensure that the products of their efforts will be used in socially responsible ways, will meet social needs, and will avoid harmful effects to health and welfare.

In addition to a safe social environment, human well-being includes a safe natural environment. Therefore, computing professionals who design and develop systems must be alert to, and make others aware of, any potential damage to the local or global environment.

1.2 Avoid harm to others.

"Harm" means injury or negative consequences, such as undesirable loss of information, loss of property, property damage, or unwanted environmental impacts. This principle prohibits use of computing technology in ways that result in harm to any of the following: users, the general public, employees, employers. Harmful actions include intentional destruction or modification of files and programs leading to serious loss of resources or unnecessary expenditure of human resources such as the time and effort required to purge systems of "computer viruses."

Well-intended actions, including those that accomplish assigned duties, may lead to harm unexpectedly. In such an event the responsible person or persons are obligated to undo or mitigate the negative consequences as much as possible. One way to avoid unintentional harm is to carefully consider potential impacts on all those affected by decisions made during design and implementation.

To minimize the possibility of indirectly harming others, computing professionals must minimize malfunctions by following generally accepted standards for system design and testing. Furthermore, it is often necessary to assess the social consequences of systems to project the likelihood of any serious harm to others. If system features are misrepresented to users, coworkers, or supervisors, the individual computing professional is responsible for any resulting injury.

In the work environment the computing professional has the additional obligation to report any signs of system dangers that might result in serious personal or social damage. If one's superiors do not act to curtail or mitigate such dangers, it may be necessary to "blow the whistle" to help correct the problem or reduce the risk. However, capricious or misguided reporting of violations can, itself, be harmful. Before reporting violations, all relevant aspects of the incident must be thoroughly assessed. In particular, the assessment of risk and responsibility must be credible. It is suggested that advice be sought from other computing professionals. See principle 2.5 regarding thorough evaluations.

1.3 Be honest and trustworthy.

Honesty is an essential component of trust. Without trust an organization cannot function effectively. The honest computing professional will not make deliberately false or deceptive claims about a system or system design, but will instead provide full disclosure of all pertinent system limitations and problems.

A computer professional has a duty to be honest about his or her own qualifications, and about any circumstances that might lead to conflicts of interest.

Membership in volunteer organizations such as ACM may at times place individuals in situations where their statements or actions could be interpreted as carrying the "weight" of a larger group of professionals. An ACM member will exercise care to not misrepresent ACM or positions and policies of ACM or any ACM units.

1.4 Be fair and take action not to discriminate.

The values of equality, tolerance, respect for others, and the principles of equal justice govern this imperative. Discrimination on the basis of race, sex, religion, age, disability, national origin, or other such factors is an explicit violation of ACM policy and will not be tolerated.

Inequities between different groups of people may result from the use or misuse of information and technology. In a fair society,all individuals would have equal opportunity to participate in, or benefit from, the use of computer resources regardless of race, sex, religion, age, disability, national origin or other such similar factors. However, these ideals do not justify unauthorized use of computer resources nor do they provide an adequate basis for violation of any other ethical imperatives of this code.

1.5 Honor property rights including copyrights and patent.

Violation of copyrights, patents, trade secrets and the terms of license agreements is prohibited by law in most circumstances. Even when software is not so protected, such violations are contrary to professional behavior. Copies of software should be made only with proper authorization. Unauthorized duplication of materials must not be condoned.

1.6 Give proper credit for intellectual property.

Computing professionals are obligated to protect the integrity of intellectual property. Specifically, one must not take credit for other's ideas or work, even in cases where the work has not been explicitly protected by copyright, patent, etc.

1.7 Respect the privacy of others.

Computing and communication technology enables the collection and exchange of personal information on a scale unprecedented in the history of civilization. Thus there is increased potential for violating the privacy of individuals and groups. It is the responsibility of professionals to maintain the privacy and integrity of data describing individuals. This includes taking precautions to ensure the accuracy of data, as well as protecting it from unauthorized access or accidental disclosure to inappropriate individuals. Furthermore, procedures must be established to allow individuals to review their records and correct inaccuracies.

This imperative implies that only the necessary amount of personal information be collected in a system, that retention and disposal periods for that information be clearly defined and enforced, and that personal information gathered for a specific purpose not be used for other purposes without consent of the individual(s). These principles apply to electronic communications, including electronic mail, and prohibit procedures that capture or monitor electronic user data, including messages,without the permission of users or bona fide authorization related to system operation and maintenance. User data observed during the normal duties of system operation and maintenance must be treated with strictest confidentiality, except in cases where it is evidence for the violation of law, organizational regulations, or this Code. In these cases, the nature or contents of that information must be disclosed only to proper authorities.

1.8 Honor confidentiality.

The principle of honesty extends to issues of confidentiality of information whenever one has made an explicit promise to honor confidentiality or, implicitly, when private information not directly related to the performance of one's duties becomes available. The ethical concern is to respect all obligations of confidentiality to employers, clients, and users unless discharged from such obligations by requirements of the law or other principles of this Code.

2. MORE SPECIFIC PROFESSIONAL RESPONSIBILITIES.

As an ACM computing professional I will ....

2.1 Strive to achieve the highest quality, effectiveness and dignity in both the process and products of professional work.

Excellence is perhaps the most important obligation of a professional. The computing professional must strive to achieve quality and to be cognizant of the serious negative consequences that may result from poor quality in a system.

2.2 Acquire and maintain professional competence.

Excellence depends on individuals who take responsibility for acquiring and maintaining professional competence. A professional must participate in setting standards for appropriate levels of competence, and strive to achieve those standards. Upgrading technical knowledge and competence can be achieved in several ways:doing independent study; attending seminars, conferences, or courses; and being involved in professional organizations.

2.3 Know and respect existing laws pertaining to professional work.

ACM members must obey existing local, state,province, national, and international laws unless there is a compelling ethical basis not to do so. Policies and procedures of the organizations in which one participates must also be obeyed. But compliance must be balanced with the recognition that sometimes existing laws and rules may be immoral or inappropriate and, therefore, must be challenged. Violation of a law or regulation may be ethical when that law or rule has inadequate moral basis or when it conflicts with another law judged to be more important. If one decides to violate a law or rule because it is viewed as unethical, or for any other reason, one must fully accept responsibility for one's actions and for the consequences.

2.4 Accept and provide appropriate professional review.

Quality professional work, especially in the computing profession, depends on professional reviewing and critiquing. Whenever appropriate, individual members should seek and utilize peer review as well as provide critical review of the work of others.

2.5 Give comprehensive and thorough evaluations of computer systems and their impacts, including analysis of possible risks.

Computer professionals must strive to be perceptive, thorough, and objective when evaluating, recommending, and presenting system descriptions and alternatives. Computer professionals are in a position of special trust, and therefore have a special responsibility to provide objective, credible evaluations to employers, clients, users, and the public. When providing evaluations the professional must also identify any relevant conflicts of interest, as stated in imperative 1.3.

As noted in the discussion of principle 1.2 on avoiding harm, any signs of danger from systems must be reported to those who have opportunity and/or responsibility to resolve them. See the guidelines for imperative 1.2 for more details concerning harm,including the reporting of professional violations.

2.6 Honor contracts, agreements, and assigned responsibilities.

Honoring one's commitments is a matter of integrity and honesty. For the computer professional this includes ensuring that system elements perform as intended. Also, when one contracts for work with another party, one has an obligation to keep that party properly informed about progress toward completing that work.

A computing professional has a responsibility to request a change in any assignment that he or she feels cannot be completed as defined. Only after serious consideration and with full disclosure of risks and concerns to the employer or client, should one accept the assignment. The major underlying principle here is the obligation to accept personal accountability for professional work. On some occasions other ethical principles may take greater priority.

A judgment that a specific assignment should not be performed may not be accepted. Having clearly identified one's concerns and reasons for that judgment, but failing to procure a change in that assignment, one may yet be obligated, by contract or by law, to proceed as directed. The computing professional's ethical judgment should be the final guide in deciding whether or not to proceed. Regardless of the decision, one must accept the responsibility for the consequences.

However, performing assignments "against one's own judgment" does not relieve the professional of responsibility for any negative consequences.

2.7 Improve public understanding of computing and its consequences.

Computing professionals have a responsibility to share technical knowledge with the public by encouraging understanding of computing, including the impacts of computer systems and their limitations. This imperative implies an obligation to counter any false views related to computing.

2.8 Access computing and communication resources only when authorized to do so.

Theft or destruction of tangible and electronic property is prohibited by imperative 1.2 - "Avoid harm to others." Trespassing and unauthorized use of a computer or communication system is addressed by this imperative. Trespassing includes accessing communication networks and computer systems, or accounts and/or files associated with those systems, without explicit authorization to do so. Individuals and organizations have the right to restrict access to their systems so long as they do not violate the discrimination principle (see 1.4). No one should enter or use another's computer system, software, or data files without permission. One must always have appropriate approval before using system resources, including communication ports, file space, other system peripherals, and computer time.

3. ORGANIZATIONAL LEADERSHIP IMPERATIVES.

As an ACM member and an organizational leader, I will ....

> BACKGROUND NOTE: *This section draws extensively from the draft IFIP Code of Ethics,especially its sections on organizational ethics and international concerns. The ethical obligations of organizations tend to be neglected in most codes of professional conduct, perhaps because these codes are written from the perspective of the individual member. This dilemma is addressed by stating these imperatives from the perspective of the organizational leader. In this context "leader" is viewed as any organizational member who has leadership or educational responsibilities. These imperatives generally may apply to organizations as well as their leaders. In this context "organizations" are corporations, government agencies,and other "employers," as well as volunteer professional organizations.*

3.1 Articulate social responsibilities of members of an organizational unit and encourage full acceptance of those responsibilities.

Because organizations of all kinds have impacts on the public, they must accept responsibilities to society. Organizational procedures and attitudes oriented toward quality and the welfare of society will reduce harm to members of the public, thereby serving public interest and fulfilling social responsibility. Therefore,organizational leaders must encourage full participation in meeting social responsibilities as well as quality performance.

3.2 Manage personnel and resources to design and build information systems

that enhance the quality of working life.

Organizational leaders are responsible for ensuring that computer systems enhance, not degrade, the quality of working life. When implementing a computer system, organizations must consider the personal and professional development, physical safety, and human dignity of all workers. Appropriate human-computer ergonomic standards should be considered in system design and in the workplace.

3.3 Acknowledge and support proper and authorized uses of an organization's computing and communication resources.

Because computer systems can become tools to harm as well as to benefit an organization, the leadership has the responsibility to clearly define appropriate and inappropriate uses of organizational computing resources. While the number and scope of such rules should be minimal, they should be fully enforced when established.

3.4 Ensure that users and those who will be affected by a system have their needs clearly articulated during the assessment and design of requirements; later the system must be validated to meet requirements.

Current system users, potential users and other persons whose lives may be affected by a system must have their needs assessed and incorporated in the statement of requirements. System validation should ensure compliance with those requirements.

3.5 Articulate and support policies that protect the dignity of users and others affected by a computing system.

Designing or implementing systems that deliberately or inadvertently demean individuals or groups is ethically unacceptable. Computer professionals who are in decision making positions should verify that systems are designed and implemented to protect personal privacy and enhance personal dignity.

3.6 Create opportunities for members of the organization to learn the principles and limitations of computer systems.

This complements the imperative on public understanding (2.7). Educational opportunities are essential to facilitate optimal participation of all organizational members. Opportunities must be available to all members to help them improve their knowledge and skills in computing, including courses that familiarize them with the consequences and limitations of particular types of systems.In particular, professionals must be made aware of the dangers of building systems around oversimplified models, the improbability of anticipating and designing for every possible operating condition, and other issues related to the complexity of this profession.

4. COMPLIANCE WITH THE CODE.

As an ACM member I will ....

4.1 Uphold and promote the principles of this Code.

The future of the computing profession depends on both technical and ethical excellence. Not only is it important for ACM computing professionals to adhere to the principles expressed in this Code, each member should encourage and support adherence by other members.

4.2 Treat violations of this code as inconsistent with membership in the ACM. Adherence of professionals to a code of ethics is largely a voluntary matter. However, if a member does not follow this code by engaging in gross misconduct, membership in ACM may be terminated.

This Code and the supplemental Guidelines were developed by the Task Force for the Revision of the ACM Code of Ethics and Professional Conduct: Ronald E. Anderson, Chair, Gerald Engel, Donald Gotterbarn, Grace C. Hertlein, Alex Hoffman, Bruce Jawer, Deborah G. Johnson, Doris K. Lidtke, Joyce Currie Little, Dianne Martin, Donn B. Parker, Judith A. Perrolle, and Richard S. Rosenberg. The Task Force was organized by ACM/SIGCAS and funding was provided by the ACM SIG Discretionary Fund. This Code and the supplemental Guidelines were adopted by the ACM Council on October 16, 1992.

7.3 Case Studies

Appendix A

Modeling

A.1 ASM Charts

Large state diagrams get ugly, due to the need to label each input and output, even if they are irrelevant to that state. Algorithmic State Machine (ASM) Charts take care of this problem, and thus are much easier to read. Only relevant inputs and asserted signals are listed.

ASM Charts are much like flow charts, and consist of states and decisions connected by directed lines. There are three types of symbols used:

rectangle State block. The name goes on the upper left (either above or in the box). The state code goes on the upper right, and the asserted output goes on the lower inside of the box. Each box is a state, and the system remains in it through a whole clock cycle.

diamond Decision block. The condition goes inside, and the potential answers (usually T/F) goes on the lines exiting the diamond.

Rounded rectangle Conditional outputs. If a Mealy style output is used, then these conditional outputs may be asserted in the conditional output block without changing state.

Basic Rules:

- ASMs are deterministic state machines, but they can include parallel conditions, so long as no state is non-deterministic. Both serial and parallel conditions are performed simultaneously.
- Lines must point to a symbol not another line.
- At each moment, the system may only be in a state block. The downstream decision blocks and conditional blocks are calculated during the cycle, so the transition can take place at the clock pulse.

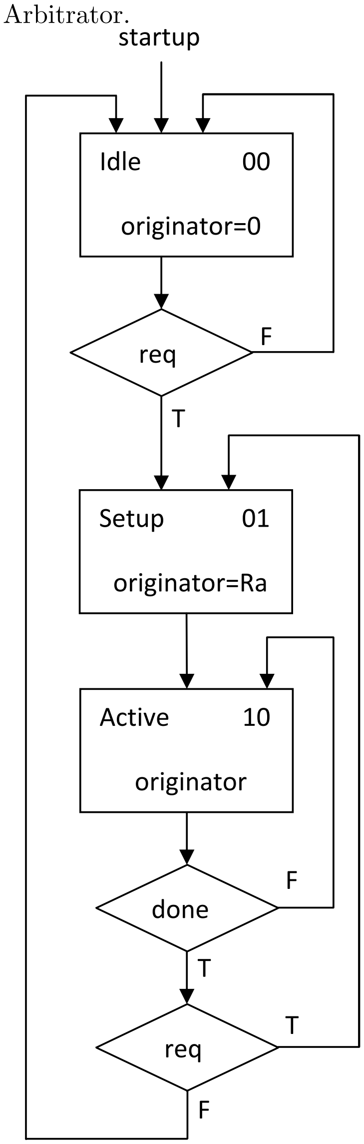
- Only state blocks can have multiple inputs.

Example: Bus Arbitrator

Imagine we are to design a bus arbitrator. The basic scheme is that there is to be a request line, *req*, that any device can assert as long as it is not currently asserted. When the request line is asserted, the requestor also puts their identifier on the requestor's address line , *Ra*. If the system is idle (no one transferring) then the system changes first to a setup state, which sets the originator to the address of the requestor. When the requestor sees they have the bus, the requestor de-asserts the request line and request address. The requestor is now the transmitter and we go to the active state. Since the request lines are free, another device can que a request. Note only one can que a request as no one else can request once the line is asserted. When the transmitter is done, the done line is asserted and the request line is checked. If there is a waiting request this system becomes the transmitter, else the system goes idle.

This is shown in the ASM chart, figure A.1. Let's consider this chart. We start in the idle state, where the originator is set to zero. This makes the assumption there is no device with address 0, which is a common assumption. The alternative is to add a busy line and assert it[1]. If no request comes then we stay in idle. If a request comes we go to setup and copy Ra to originator. The remote machine must drop the request so we do not list this. In the remote devices logic when their address gets copied to address, they would switch states and do this. In any case, the next cycle the system goes to the active state and continues to assert the originator. When the remote system signals done, the request line is checked.

Figure A.1: ASM Chart of Bus Arbitrator.



[1]This is inefficient because one extra line is another bit, which could double the number of addresses, then minus one to account for the unused code

A.2 Block Diagrams

A.3 Control Flow Graphs

Appendix B

Using SciLab

B.1 Basics

When you first start SciLab you will see something like

```
                  ==========
                   scilab-2.7.2
        Copyright (C) 1989-2003 INRIA/ENPC
                   ==========



Startup execution:
  loading initial environment

-->
```

The arrow "$-->$" is the command prompt. SciLab, like MatLab, is a command line interface to a mathematics programming environment. To get started lets do a calculation.

```
3+(2+5*4)/11
```

SciLab performs the calculation and displays the answer.

```
 ans  =

    5.
```

Now lets define a simple variable.

```
a=2
```

SciLab responds with

```
 a  =

    2.
```

Notice anything similar? The response is almost the same but "ans" has been replaced by the variable name "a". In fact it is even more similar than that. When no assignment ("name=") is given, SciLab automatically assigns the result to the variable "ans". Try using it.

```
a*ans
```

SciLab will tell you that "ans" is now 10. Lets move on and define a matrix. Type the following

```
  A=[1,2;3,4;5,6]
```

and press enter. Commas are used to separate elements and semicolons are used to separate rows. Note that you could also have entered "A" using the alternate notation

```
  A=[[1 2];[3 4];[5 6]]
```

or even (command prompt shown so you won't think something is wrong when it automatically appears, also you do not need to space over like I do to enter the numbers, I just find it easier to read)

```
-->  A=[[1 2]
-->     [3 4]
-->     [5 6]]
```

Thus spaces work like commas and returns work like semicolons. In any case, SciLab should respond by showing you that it has created the matrix variable as follows

```
 A  =

!   1.    2. !
!   3.    4. !
!   5.    6. !
```

The variable "A" is now defined and can be used. For instance we might want to define "B" to be "A+A". Do this by typing

```
  B=A+A
```

SciLab will add the matrices and define "B" to be the result, showing you the answer.

```
 B  =

!   2.      4.  !
!   6.      8.  !
!   10.     12. !
```

This mode is useful for doing simple calculations and testing output. We will refer to it as the interactive mode. Since SciLab has an interactive mode that is command driven, it is reasonable to assume it would have a programming interface (we will refer to it as the programming mode). I will show the use of programming mode later.

B.2 Programming

Scilab is a Matlab look alike. We will use Scilab as it is free and open source. If you know one, you can figure the other out easily.

There are two basic ways to interact with Scilab: command line execution, and script files[1]. Yes there are others such as compiled programs (MEX-files in Matlab), a GUI - Scicos (Simulink in Matlab), and several interfacing programs, but they are not relevant to us at the moment. We will primarily be concerned with the use of script files, because they are the most helpful. Command line execution is really just for quick operations and checking of segments of code. Scilab syntax is a high level programming language that interacts with a series of numerical libraries (most notably LinPack, EisPack, and BLAS). Like most programming languages we have two types of programs that can be written. A regular program, which is written as you would type commands on the command line, is the most basic type and is often the way you will start homework problems and other projects. I am not going to try to be exhaustive, as the help files are quite nice (type "help" for a searchable help file or "help command" for help on some command (obviously you must replace command with the actual command you want help on). Functions, which are sub-programs called by another program (even by other functions), are probably the most useful, as they allow you to extend the language by defining new operations. One of the side goals of this book is for you to walk away with a basic familiarity of the tools that you can use to do a variety of tasks. So how do you specify which you want? You will get a regular program unless you start the script file with the command function. The syntax is

```
function [a,b,...,c]=name(x,y,..., z)
```

Note you can return multiple values. You also have several command structures: for, while, and if-elseif-else. To see how these work let's make up a program.

Listing B.1: Quadratic Function

```
function [y]=quadratic(x,a,b,c)
  y=a*x^2+b*x+c;
endfunction
```

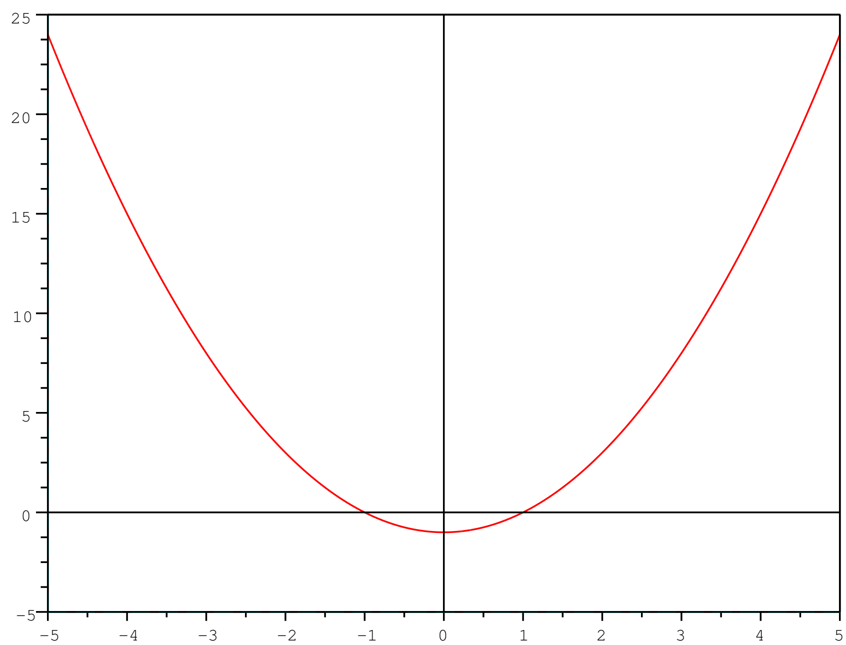
Now we want to run it. Create the following script file in Scipad (Scilab's editor) then save and execute.

[1]Scilab distinguishes between directly executed files (.sce) and function libraries (.sci) which must be called by something else. Matlab just uses M-files, a .m extension.

Listing B.2: Graphing a Quadratic

```
exec("quadratic.sci");
x=-5:.01:5;
y=quadratic(x,1,0,-1);
plot(x,y,'r-')
plot([min(x);max(x)],[0;0],'k-')
plot([0;0],[min(y)-4;max(y)+1],'k-')
```

The resulting graph is



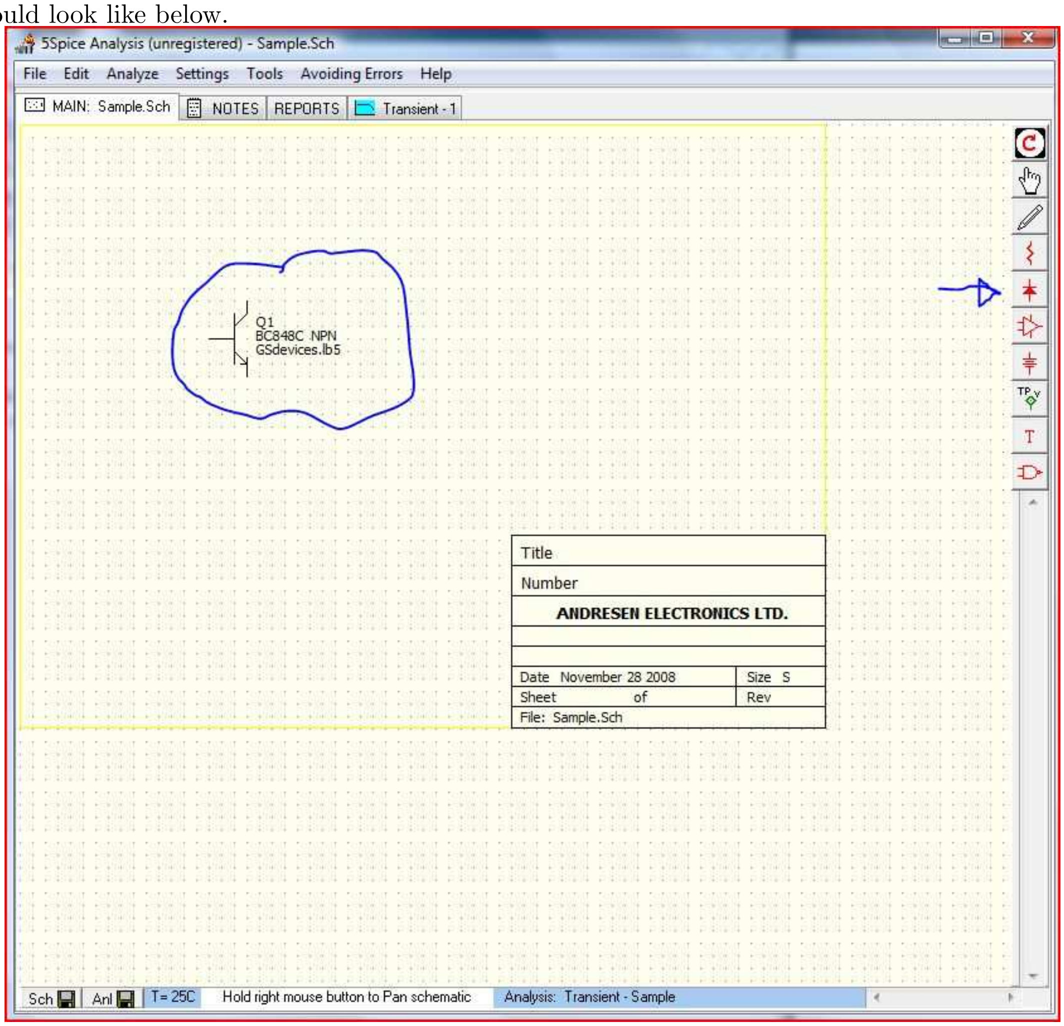
Appendix C

SPICE

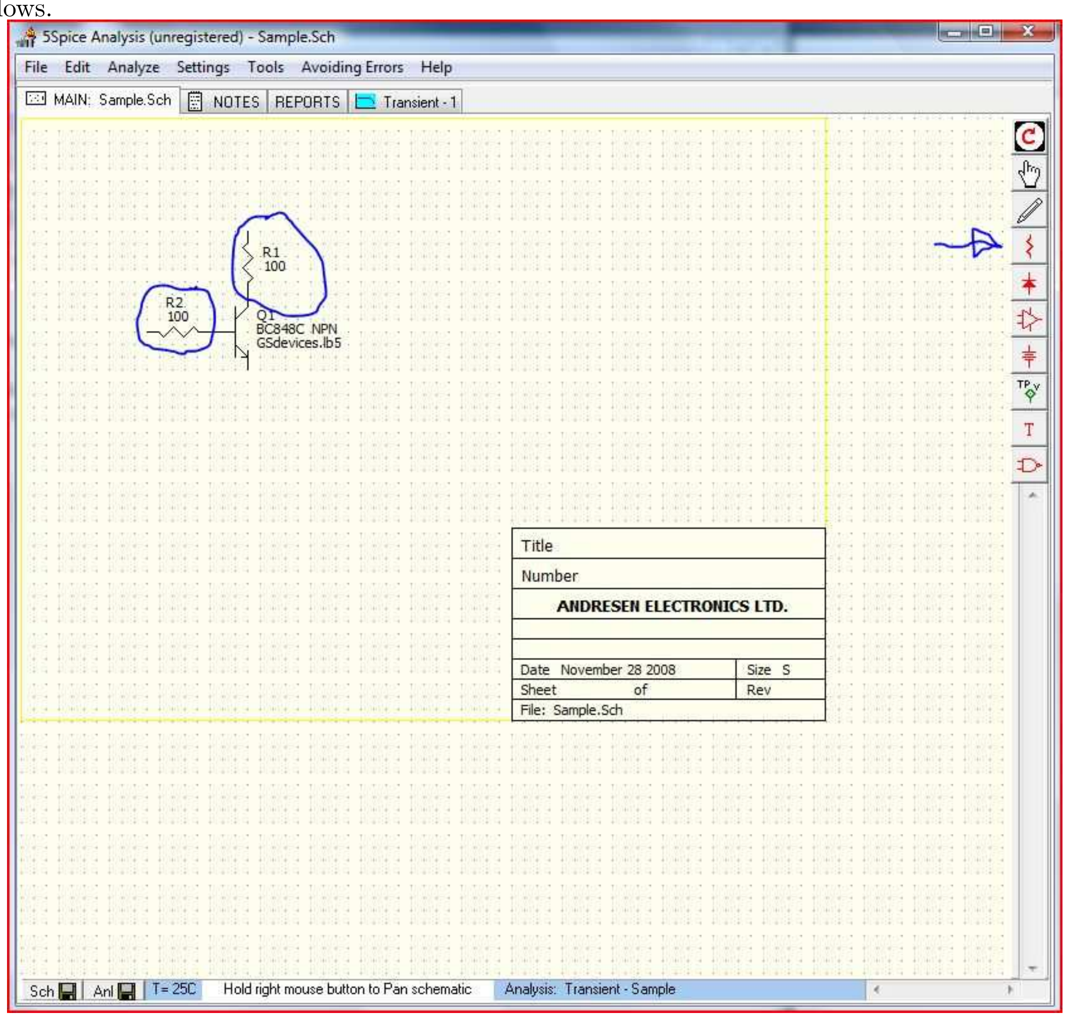
SPICE is an acronym (come on this is engineering of course it is an cutsie acronym to try to make it memorable) that stands for Simulation Program with Integrated Circuit Emphasis. SPICE is often used as a simulator for circuit layout, and is thus usually incorporated into EDA (Electronic Design Analysis tools) like OrCAD or Mentor Graphics. I will cover a separate SPICE tool called 5SPICE, which was designed for prototyping, and is free for educational use.

SPICE allows us to simulate a circuit without having to build it. You can select actual component families, operating temperatures, and other operating conditions. This allows a more accurate simulation, and thus less prototyping is needed. This saves a lot of time and money, while improving the quality of the design, by allowing design changes in the simulation.

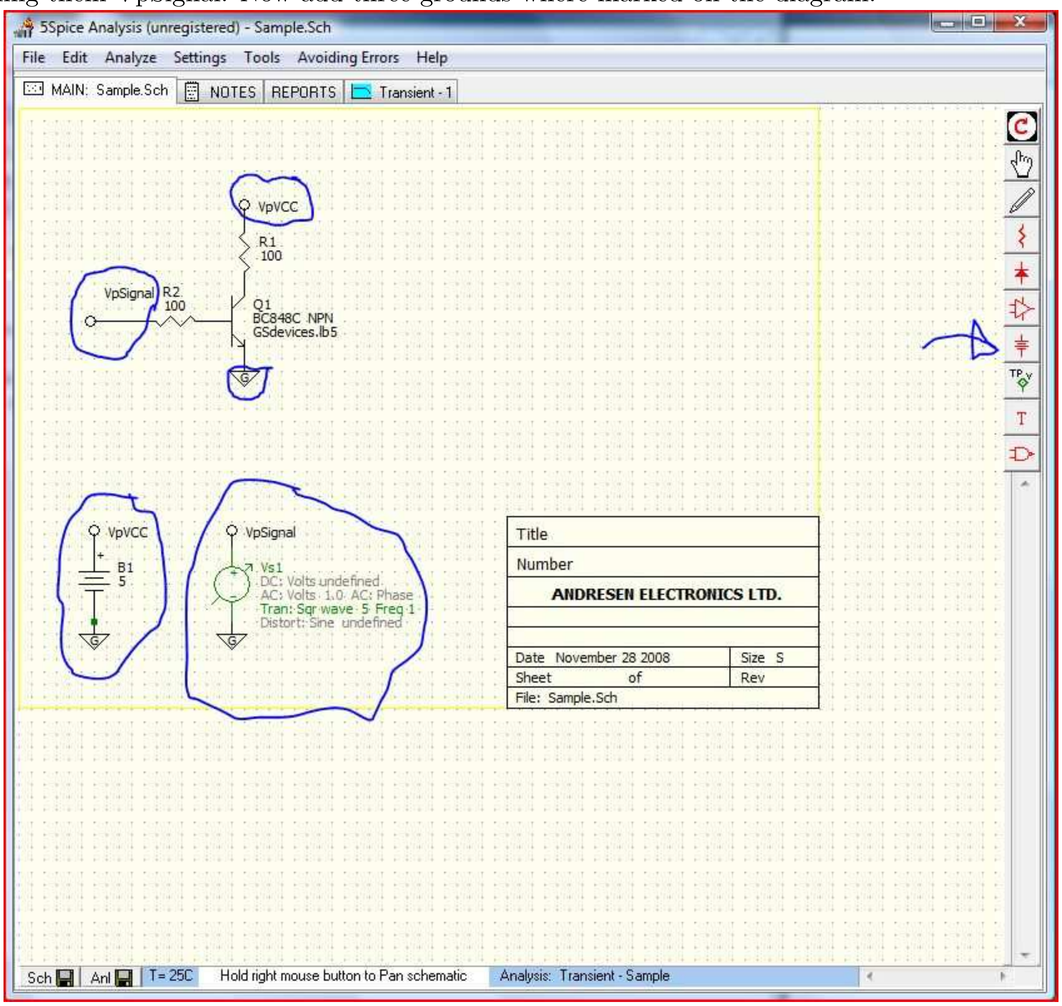
We will create an inverter gate using a transistor and some resistors. First we need to put a transistor in our blank schematic. We do this by selecting the NPN transistor symbol (if you don't know it, the name will appear in the tool tip when you hover) off the diode button on the right button bar. When we have placed it by left clicking, we right click twice (once to end adding transistors and the second to edit the transistor we place). A dialog will appear, from which you should select "BC848C" then click "ok" to accept. Your screen should look like below.



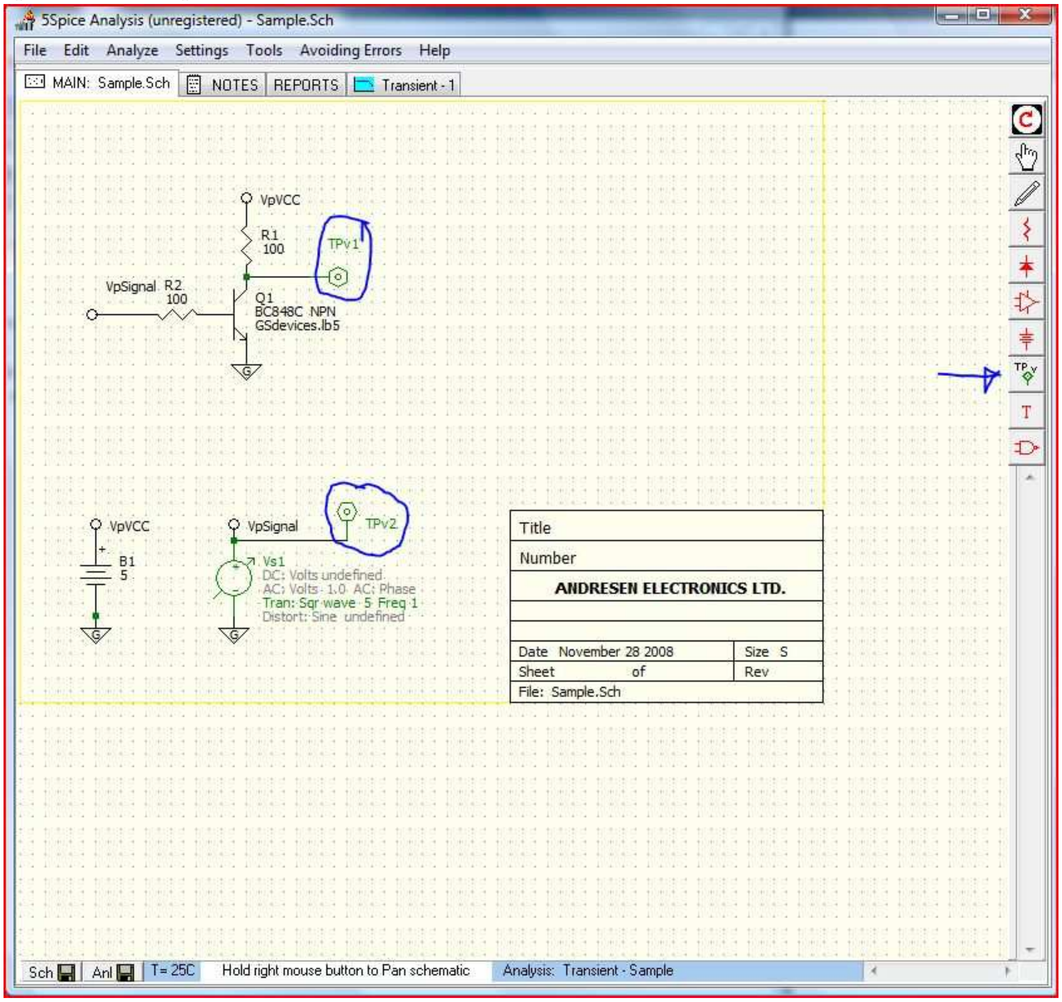
We now need to add two resistors to keep the current flows within reason. Resistors are selected from the resistor button on the right button bar. Place two resistors on the sheet. Right click to end adding and then right click one to edit. Change the resistance to 100 (ohms is implicit) then click ok. Move this one to the collector, to keep the current from collector to emitter to a reasonable value. Now edit the other so it is also 100 ohms and click ok. Right click it again and rotate. Drag to the gate. You schematic should look as follows.



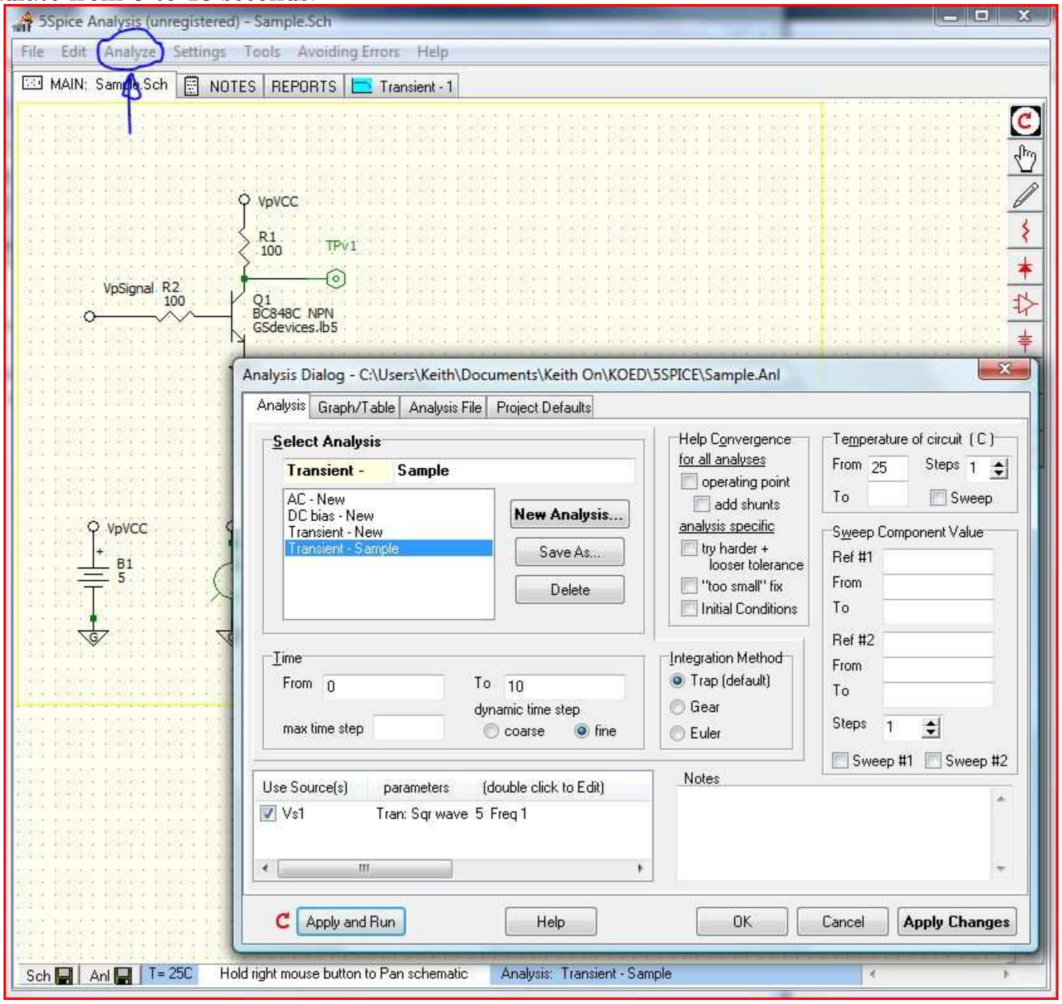
We know need to add several items from the power source button. Start with a DC source to supply power to the transistor. We will need to edit it so it has 5 volts. Since we don't want to drag wires all over the schematic (we could, I just think it looks ugly) we will use two voltage points to connect the DC source to the resistor on the source. Let's edit them to name it VpVCC. Now add an AC voltage source. When you edit its properties there will be several tabs so select the one called transient. Select square wave, 5 volts peak to peak, 2.5 volts bias (so it goes 0-5 not -2.5 to 2.5), a frequency of 1, and a rise time of .1. I also set the delay to 1 so it stays at 0 volts for a second before transitioning. We will again add two voltage points to keep things need, this time putting them on the input signal and calling them VpSignal. Now add three grounds where marked on the diagram.



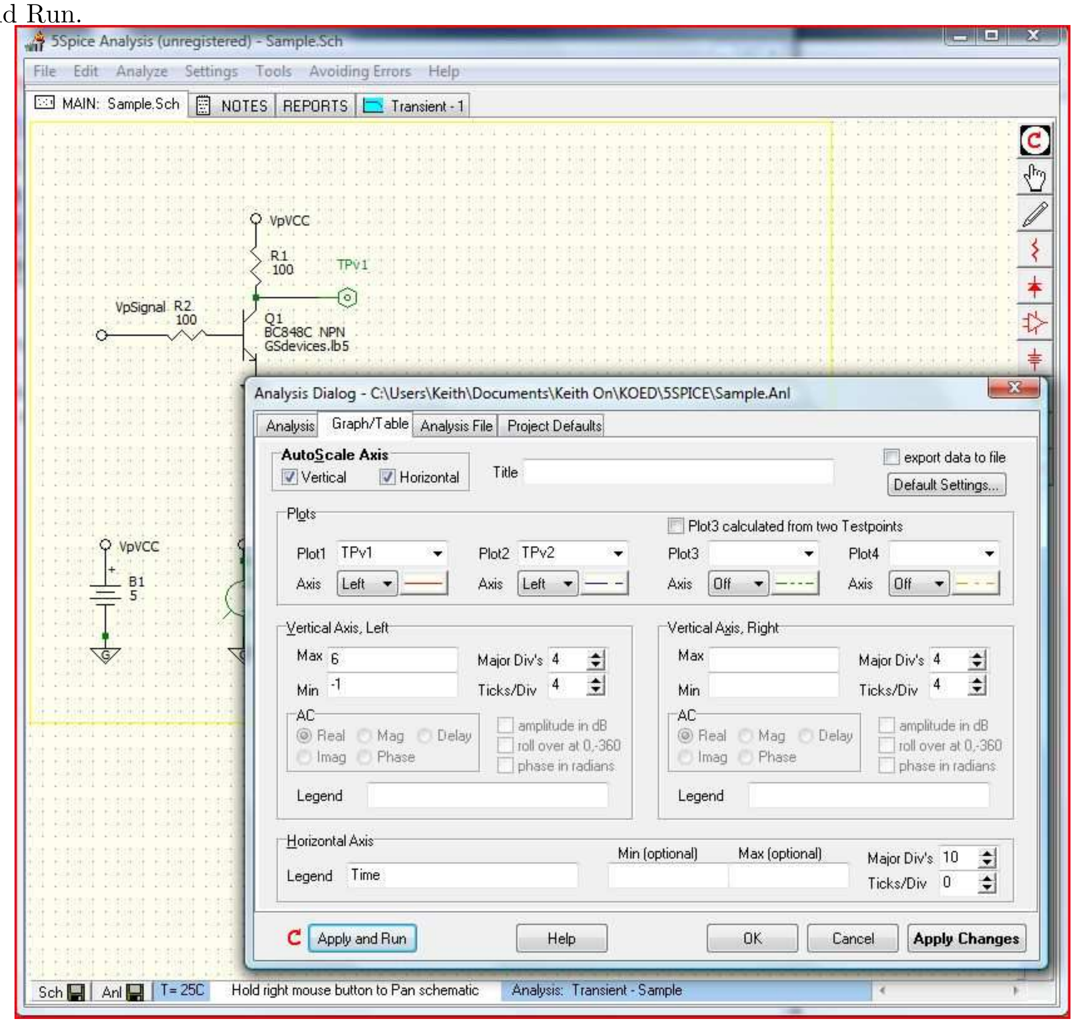
Now we need to add two test points, so we graph something. You can only plot from test points. To keep it neat I used wires to connect them.



We now need to set the defaults for the simulator. From the Analyze menu, select "Project Defaults..." then select the Analysis tab. Pick transient analysis using Vs1 and simulate from 0 to 10 seconds.



Now select the Graph/Table tab and under Plots, select TPv1 with Left axis and TPv2 with Left axis. Note if you don't select the axis it is off by default. Then set the vertical axis to go from -1 to 6 so there is some room around the graph (again for aesthetics). Save and Run.



You will get a graph like below after a few seconds. Note the graph shows we have an inverter. You will also note the inverted signal does not go to zero exactly due to the resistors. If you play with the resistances you will get some interesting results. Try and see. You should also see the cyan circle with the vertical line at the top of the graph. It is the measurement device, and it is the selector for the x,y data on the top right. Try moving it and reading the voltages.