Equation Library Reference Manual

Han Duong

March 25, 2017

Contents

1 Installation

1.1 Installation from Equation Library.hpappdir

More recent versions of the `Equation Library` are published in an "hpappdir" format. This is essentially a directory containing various binaries that form the app. To install an app in hpappdir format:

1. Install the connectivity kit and any updates to the connectivity kit as appropriate.
2. Install any firmware updates to the calculator if prompted. Refer to the user manual for more details on updating firmware.
3. Run the connectivity kit program, and ensure that the `Content` pane is visible (in the bottom left corner of the connectivity kit window). If necessary, use the program menu and select ***Window*** and then ***Content***.
4. After uncompressing the zip file, determine which version to install (merged vs. separated).
5. Drag the `Equation Library.hpappdir` folder into the `Content` pane in the connectivity kit. Make sure that this folder is not inadvertently placed inside another folder or existing hpappdir.
6. Run the simulator program (emulator), or connect the calculator via USB.
7. Drag the `Equation Library.hpappdir` folder **inside the** `Content` **pane** and drop it into the calculator or simulator shown above the `Content` pane within the connectivity kit.
8. If you are using the separated version, you will additionally need to drag the `SVD2.hpprgm` file into the calculator or virtual calculator. On the calculator (or simulator), type `restart` in the command line and then open and close the `SVD2` program file.

1.2 Installation on HP Prime simulator from source

Each HP Prime comes with a CD that includes both a simulator and connectivity software. We will assume that both software packages have already been installed and are in working order. To install `Equation Library` on the simulator:

1. Run the simulator program.
2. Press the Apps Info key and highlight the `Solve` app.
3. Click on the `Save` menu option. When prompted for a new app name, click the `Edit` menu option and replace the `Advanced Graphing` text with `Equation Library`. Press Enter twice and the newly created copy of the `Advanced Graphing` app should appear in the apps menu with the name `Equation Library`.
4. Select the `Equation Library` app by clicking on the `Equation Library` icon in the apps menu.
5. Press Shift 1 Program to open the `Program Editor` and highlight the entry listed as `Equation Library (App)`. Click the `Edit` menu option to open the program listing.
6. Clear the existing program listing using Shift Esc Clear.
7. Using a word editor such as `Wordpad`, open the source code file for the `Equation Library` app and copy the entire source listing.
8. On the simulator, (with the `Program Editor` still opened), paste the source listing by clicking on `Edit` and then `Paste` found in the menu at the top of the simulator window. (If the simulator title bar does not appear, right-click on the simulator window and select ***Calculator*** and then ***Show Titlebar***.)

9. If installing the separated version, the SVD2 program must also be installed. In the Program Editor, select the New menu option and use the name SVD2. Copy the contents of the SVD2 program in the svd2.txt file and replace the contents of the SVD2 program within the Program Editor.

10. Press Esc/Clear to exit the Program Editor.

11. To ensure that the simulator saves its memory state, close the simulator and rerun it.

1.3 Installation Onto the HP Prime calculator

The easiest method is to first install Equation Library on the simulator. Once the app is installed on the simulator, follow these steps:

1. Ensure that the app is installed on the simulator, and press the Apps/Info key on the simulator to display the apps menu. Highlight the Equation Library app.

2. Connect the calculator to the computer running the simulator using the USB cable provided with the calculator.

3. In the simulator window, select (from the menu at the top of the window): Calculator, Connect To, <***name of calculator***>. If the calculator has not been named, then its serial number will appear instead.

4. In the simulator (with the apps menu open and Equation Library highlighted), click on the Send menu option at the bottom of the screen.

5. After the app has been sent to the calculator, disconnect the simulator by selecting: Calculator, Connect To, None from the menu at the top of the simulator window.

6. Turn off the calculator to ensure that the app is saved to permanent storage.

1.4 Updating Equation Library.lib

The data for every single system of equations is stored in an app file named Equation Library.lib within the Equation Library app. If there is an update to the library data, the update may be applied in the following manner. (We will assume that the Equation Library is currently installed on the simulator.)

1. Open the text file containing the library data (list of equations, variables, etc.) on the PC or Mac.

2. Select the entire content of the text file and copy the content into the PC or Mac clipboard.

3. On the simulator (virtual calculator), ensure that the current app is the Equation Library. Type into the command line

   ```
   AFiles("Equation Library.lib"):=
   ```

   but do not press the Enter/≈ key. Make sure that the cursor is to the immediate right of the "=" symbol. Then paste the content of the library data that was previously copied into the simulator. Then press Enter/≈ and save the new data.

4. Once this has been updated on the simulator, connect the actual calculator to the PC or Mac.

5. On the simulator, press the Apps/Info key and select the Equation Library app. The menu at the bottom should have a "Send" button. Press this menu button to transfer the updated app.

2 Using the Equation Library

To use the `Equation Library` press the Apps key and select the `Equation Library` app. The app will start with a blank Symb view. At this point, we may either select and existing system of equations stored in the app, or create a new system of equations to solve.

2.1 Using Existing Systems of Equations

To solve a system of equations that is stored in the app, press the Plot key and a menu of categories will appears. Select the appropriate category, and a selection of systems of equations within that category will appear as shown in Figure 1 below. Upon selecting a system, the Symb view will be populated with the

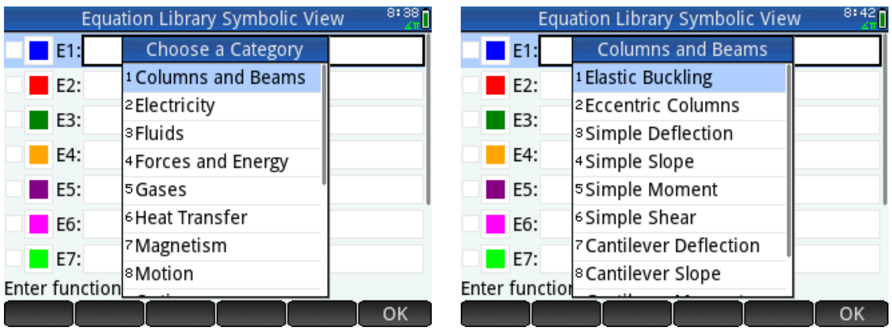


Figure 1: Category and system selection menus

equations from that system as shown in the left screenshot in Figure 2. At this point, we may immediately solve the system of equations, or we may modify the selection of equations into a smaller subset of equations by using the check boxes to the left of each equation. A check mark indicates that that particular equation will be included in the system, whereas a blank box indicates that it will be excluded. Please note that some systems have more than 10 equations. To view additional pages of equations, press the View key and select the `Select Page` option. To view the list of variables of the entire system (for both checked and unchecked equations), press the View key and select `View Variables` option. All variables, and their descriptions, will

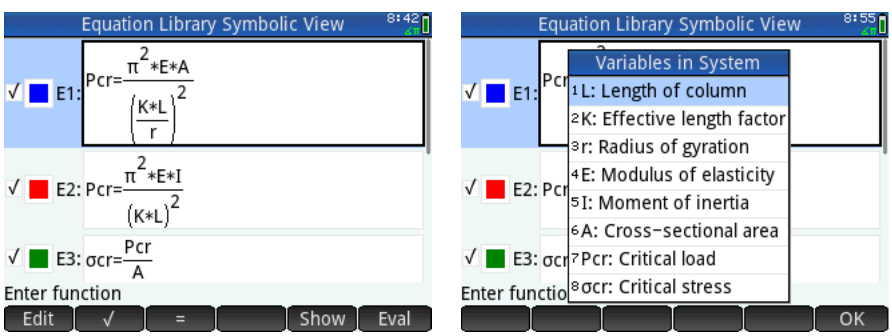


Figure 2: Symb view (left) and list of variables (right)

be listed. Entries surrounded by parentheses are actually constants, though they may be treated as variables in the sense that we may change their values (perhaps due to a difference in preferred units of measure).

2.2 Solving a System

Once we have decided which combination of equations to solve, we then enter in known values and initial guesses using the Num key. Note that there may be several pages of variables and constants. Each input

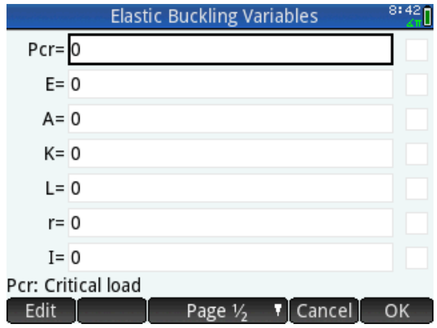


Figure 3: Num view for initial guesses and known values

field has a check box to the right to indicate whether the value entered is to be treated as a fixed value (usually the case for known values or constants). Otherwise the value will be treated as an initial guess for the corresponding variable, and the system will solve for this variable. A description of the variable is displayed at the bottom of the screen, just above the list of menu options. Once all values and initial guesses have been entered, press the `OK` menu option to begin solving. Depending on the options selected, you may

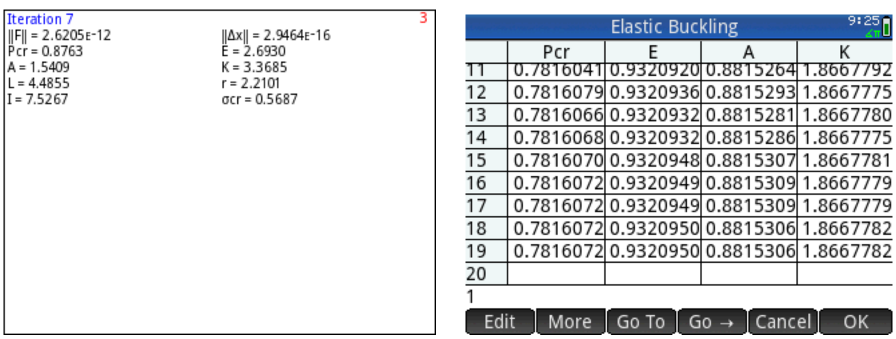


Figure 4: Iteration information (left) and solution matrix (right)

see information about the current state of the solver as shown in Figure 4 on the left. The information shown are the Euclidean norm $\|\mathbf{F}\|$ of the current system and the Euclidean norm $\|\Delta\mathbf{x}\|$ of the change in the vector of variables $\mathbf{x}$. If a solution has been found, a message box indicating a zero was found will be displayed. Also, a matrix of each individual Newton iteration in the solving process is shown, with the last row of values being the solution (unless the index of the last row is the one more than the maximum number of iterations configured).

2.3 Creating a New System

To create a new system to solve, press Plot and select `New System` at the bottom of the list of categories. Before we can enter in our equations in the Symb view, we must initialize the variables that we wish to use in our system. This can be done by pressing View and selecting the `Add/Edit Variables` option. In this input screen, we may also add a description for our variable. The app will continue to prompt for more variable names and descriptions after creating a variable. To stop creating new variables, simply select the `Cancel` option. This process can be automated if we restrict ourselves to pre-existing variables **and** by selecting to have variables be managed automatically. See the ***Settings*** section for more information.

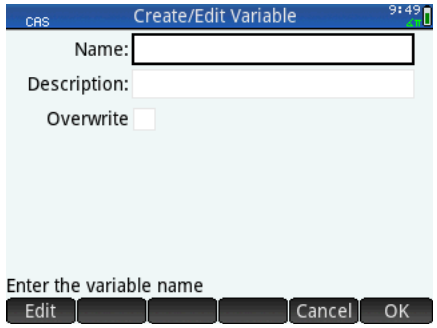


Figure 5: Input form for adding or editing variables

There is no option to delete variables. If we should accidentally create a variable we do not wish to use, we simply avoid using that variable. The system will detect whether a declared variable has been abandoned. After continued use of the solver, the unused variable(s) will be automatically removed. Variables that are marked for eventual removal will be marked with `(*)` in the list of variables (accessible from the View menu).

Once all variables have been declared, we proceed by populating the Symb view with equations. (Note that any syntax error during the creation of an equation is likely due to the use of a variable that either does not currently exist in the system or has not been declared.) After entering the desired equations, press View and select the `Save System` option to add the system to the library data file used by the `Equation Library` app.

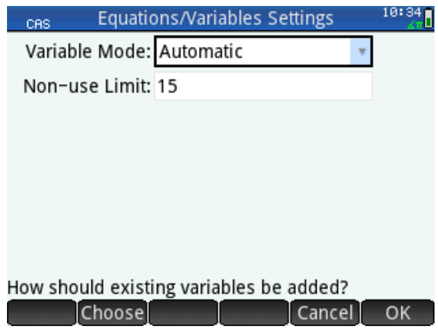
Remark. If using built-in variables such as `A` through `Z`, be aware that these variables only store real values. The solver itself keeps its own record of variables and values and will still solve properly, but we will not be able to reference any solutions through these variables. That is, if we use the variable `A`, and the solution for `A` is a complex number, then typing `A` will recall the initial value and not the solved value. Moreover, subsequent attempts to refine our solution (by pressing Num and solving again) will not work. So if we wish to use complex numbers for our initial values, or if we wish to solve for complex solutions, then we must use `Z0` through `Z9` (if the intention is to only use built-in variables), or we must create a new variable which will always be able to store both real and complex values.

3 Settings

The settings for the `Equation Library` are explained below. **Do not forget to press the `OK` menu button to save your settings!**

3.1 Variables Settings

Shift Symb — Press Shift Symb to access the settings for how variables are managed when creating new systems, or when variables become abandoned.



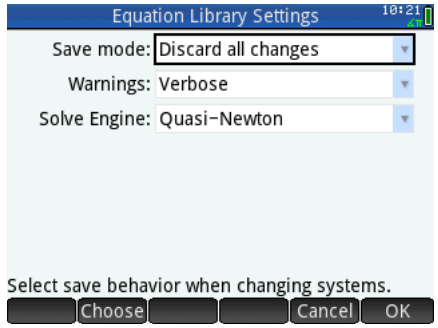
Variable Mode: When creating a new system, variables can either be created manually using View and selecting `Add/Edit Variables`, or they can be managed automatically. When set to `Automatic`, then equations that use pre-existing variables will have those variables added to the system automatically. However, this is only automatic for pre-existing variables. On the other hand, if the equation uses a variable that does not currently exist, then such variable must still be manually added into the system regardless of the `Variable Mode` setting. This has to do with how the HP Prime manages variables. Regardless of the currently active app, equations in any Symb view may only use pre-existing variables; otherwise a syntax error will be returned. This is why the `Equation Library` app provides a customized method for adding variables (with a description).

Non-use Limit: Variables are not deleted manually. Instead, they are simply abandoned by non-use. Every time the solver is activated, it scans the current list of equations and determines which variables are actually used by the system. Those that are no longer used in any equation are marked for deletion. The `Non-use Limit` refers to the number of times the system is solved without using an abandoned variable. Upon exceeding the limit, the abandoned variable is then removed permanently. If a variable that has been marked as abandoned is re-used before the limit is reached, then it is returned to the active state. Variables marked for eventual removal are listed in the Variables list with `(*)`.

3.2 General Settings

Shift **Plot Setup**

Press **Shift** **Plot Setup** to access the settings for systems management and warning levels.



Save Mode: Any time a new system of equations is selected, the `Equation Library` app will need to know what to do with the system of equations that is currently in use. The `Save Mode` determines the default behavior regarding the changes to the current system of equations.

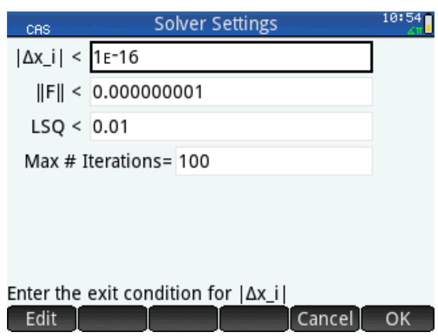
Warning: Set the warning/error message level according to your preference. Note that even with `None` as the setting, critical errors will still appear.

Solve Engine: Select the desired solve engine. The `Quasi-Newton` engine is the default engine; it is essentially the same solver used in the `SolveSys` program for the older HP48 series of calculators. An explanation of the algorithm may be found in the section titled ***Newton's Method***. The `Solve App` engine is engine used by the `Solve App`.

3.3 Quasi-Newton Solver Settings

Shift **Num Setup**

Press **Shift** **Symb Setup** to access the settings for the solver.



|Δx_i|: This value determines the exit condition for the Newton solver. After each iteration, the solver checks the relative change in the each variable x_i by computing

$$\max\left(\frac{|\Delta x_i|}{\max(|x_i|, 1)}\right)$$

If this value is less than that specified by `|Δx_i|` then the exit condition is met and the solver will try to provide an appropriate conclusion about the final iteration based on the Euclidean norm $\|\mathbf{F}\|$. In general, for a value of 1×10^{-p}, the iterations stop when the x_i's differ only in the p-th digit or further.

$\|\mathbf{F}\|$: After an exit condition is met, this value determines whether the current values of the variables x_i are solutions. Here, `F` refers to the vector $\mathbf{F}$ of functions f_i, where each f_i corresponds to the i-th equation $f_i(\mathbf{x}) = 0$ and $\mathbf{x}$ is the vector of variables the n variables $x_1, x_2, \cdots, x_n$ being solved in the system.

LSQ: `LSQ` refers to the least squares tolerance. When a system is inconsistent, the Newton solver will try to return a least-squares solution, i.e. when $f = \|\mathbf{F}\| = \sum(f_i)^2$ is locally minimal. The test is an orthogonality test comparing the gradient of f with the Newton direction. If the vectors are close to orthogonal (test value close to `0`), then the solution is very likely a local minimum. Usually a value of `0.01` is sufficient.

Max # Iterations: This value limits the number of Newton iterations before the solver quits. We may continue solving the system if the results show that the solutions appear to be converging (albeit slowly). The most recent iteration values are stored in their respective variables. Pressing Num Setup will allow us to continue solving from the most recent iteration.

3.4 Variables/Files Used by the `Equation Library`

Below is a description of the variables and/or files used by and created by the `Equation Library`.

- The `Equation Library` uses the Home variable `eqlib.dat` to store all settings. The variable is a list of the settings above.
- All systems of equations, their corresponding variables, and any initial/solved values are saved in the app file named `Equation Library.lib`. The contents of this file can be accessed with the command

  ```
  AFiles("Equation Library.lib")
  ```

  and is a list of lists. The content may be converted into a `Note` using the program `LibToNote` listed in the ***Appendix***.
- Messages during the solve routine are printed into the graphics variable `G1`. To view the most recent messages, use

  ```
  blit_p(G1); freeze; wait(-1);
  ```
- All input forms that are generated at runtime (as opposed to being explicitly coded into the app) are copied into the `Note` named `eqlib_debug.log`.
- Several `CAS` variables are created during the solve routine.
  - `ssF`: this is a vector of the equations being solved
  - `ssFx`: this is a `CAS`-function equivalent of `ssF`
  - `ssJ`: this is the Jacobian of `ssF`

4 The View Menu (View Key)

The View key provides additional interfaces for managing the system of equations. The menu options are described below.

Select Page The Symb view shows only ten equations at any given time. However, a system of equations may have more than 10 equations. In this case, the `Select Page` option will switch between desired pages of equations. If a new page is desired while creating a new system, then select `Create New Page`. A new page will be created. Upon saving the system, all pages will be consolidated. Any equation field left blank will be removed so as few pages as possible are used.

View Variables This option lists all the variables and their descriptions. Variables that have become abandoned through non-use and are marked by deletion will be listed with `(*)` next to their name.

View Picture If the system of equations has a diagram associated with it, then this option will display the diagram. A diagram may be a single image or an animation of several images. All images are saved as Portable Network Graphics (PNG) files, and use the naming convention `imgNN.png` where NN is the "image index."

Add/Edit Variable In order to create a new system of equations, or modify an existing system to use additional variables, new variables must be declared for the system. Note that this is different from creating a variable. The HP Prime is designed to only recognize pre-existing variables (such as the global variables `A` through `Z`) or those created by the user. Thus, when adding to equations to the Symb view, usage of non-existent variables will result in a syntax error.
The `Add/Edit Variable` option will not only declare the variables, it will also create them if and when necessary. This option will also continue to prompt for new variable definitions until the user selects the `Cancel` menu option. This option may also be used to modify an existing variable (mainly to change the description).

Restore Equations This is the equivalent of an "undo" for the Symb view. The last saved set of equations will replace the current equations.

Save System This option provides direct access to the interface for saving systems of equations. Use the `Save as a new system` check box to create a new copy of the current system. This can be used to "export" variables and equations to a new system for future modification.

Delete System This option provides direct access to the interface for deleting systems of equations. In the event that there are more than 70 equations in the current library, the systems will be presented in blocks of 70 (seven per page, with a maximum of ten pages).

5 Equation Reference

The following equation reference comes from the ***HP48G Advanced User's Reference***.

5.1 Columns and Beams

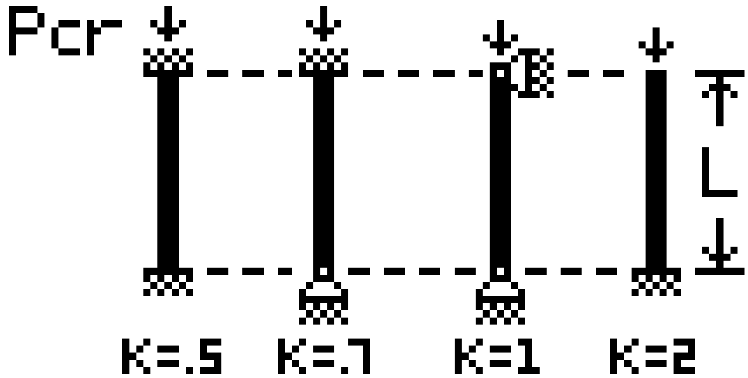
Variable	Description
ϵ	Eccentricity (offset) of load
σcr	Critical stress
σmax	Maximum stress
θ	Slope at x
A	Cross-sectional area
a	Distance to point load
c	Distance to edge fiber (Eccentric Columns), or distance to applied moment (beams)
E	Modulus of elasticity
I	Moment of inertia
K	Effective length factor of column
L	Length of column or beam
M	Applied moment
Mx	Internal bending moment at x
P	Load (Eccentric Columns), or oint load (beams)
Pcr	Critical load
r	Radius of gyration
V	Shear force at x
w	Distributed load
x	Distance along beam
y	Deflection at x

Remark. For simply supported beams and cantilever beams (***Simple Deflection*** through ***Cantilever Shear***), the calculations differ depending upon location of x relative to the loads.

- Applied loads are positive downward.
- The applied moment is positive upward.
- Deflection is positive upward.
- Slope is positive counterclockwise.
- Internal bending moment is positive counterclockwise on the left-hand part.
- Shear force is positive downward on the left-hand part.

5.1.1 Elastic Buckling

These equations apply to a slender column ($\frac{K \cdot L}{r} > 100$) with length factor K.

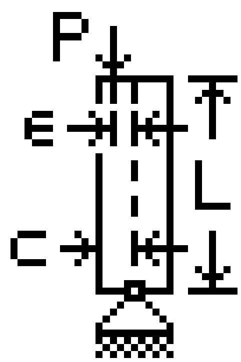


$$Pcr = \frac{\pi^2 \cdot E \cdot A}{\left(\frac{K \cdot L}{r}\right)^2} \qquad \sigma cr = \frac{Pcr}{A}$$

$$Pcr = \frac{\pi^2 \cdot E \cdot I}{(K \cdot L)^2} \qquad r = \sqrt{\frac{I}{A}}$$

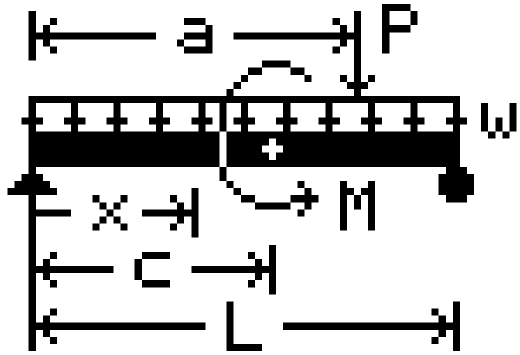
5.1.2 Eccentric Columns

These equations apply to a slender column $\left(\frac{K \cdot L}{r} > 100\right)$ with length factor K.



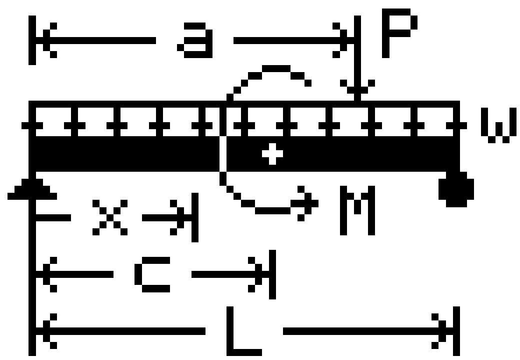
$$\sigma max = \frac{P}{A} \cdot \left[1 + \frac{\epsilon \cdot c}{r^2} \cdot \left(\frac{1}{\cos\left(\frac{K \cdot L}{2 \cdot r} \cdot \sqrt{\frac{P}{E \cdot A}}\right)}\right)\right] \qquad r = \sqrt{\frac{I}{A}}$$

5.1.3 Simple Deflection



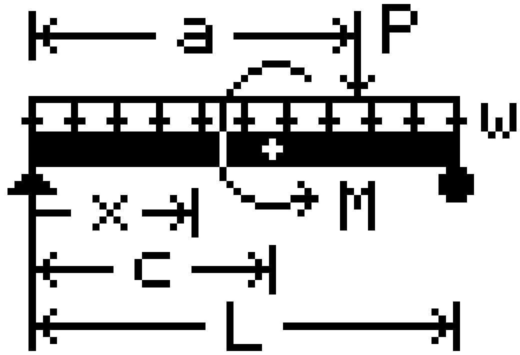
$$y = \frac{P\cdot(L-a)\cdot x}{6\cdot L\cdot E\cdot I}\cdot\begin{cases} x^2+(L-a)^2-L^2, & \text{if } x\le a\\ (L-x)^2+a^2-L^2, & \text{if } x>a\end{cases}$$
$$-\frac{M\cdot x}{E\cdot I}\cdot\begin{cases} c-\frac{x^2}{6\cdot L}-\frac{L}{3}-\frac{c^2}{2\cdot L}, & \text{if } x\le c\\ \frac{c^2+x^2}{2\cdot x}-\frac{x^2}{6\cdot L}-\frac{L}{3}-\frac{c^2}{2\cdot L}, & \text{if } x>c\end{cases}$$
$$-\frac{w\cdot x}{24\cdot E\cdot I}\cdot\left(L^3+x^2\cdot(x-2\cdot L)\right)$$

5.1.4 Simple Slope



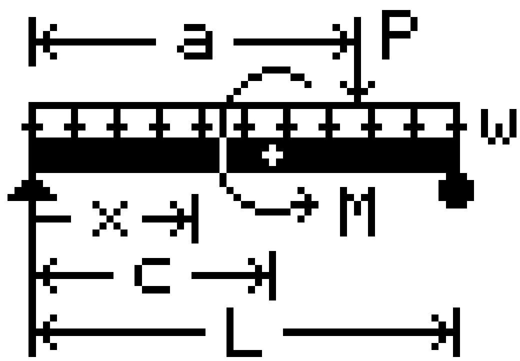
$$\theta = \frac{P}{6\cdot L\cdot E\cdot I}\cdot\begin{cases} (L-a)\cdot\left(3\cdot x^2+(L-a)^2-L^2\right), & \text{if } x\le a\\ -a\cdot\left(3\cdot(L-x)^2+a^2-L^2\right), & \text{if } x>a\end{cases}$$
$$-\frac{M}{E\cdot I}\cdot\begin{cases} c-\frac{x^2}{2\cdot L}-\frac{L}{3}-\frac{c^2}{2\cdot L}, & \text{if } x\le c\\ x-\frac{x^2}{2\cdot L}-\frac{L}{3}-\frac{c^2}{2\cdot L}, & \text{if } x>c\end{cases}$$
$$-\frac{w\cdot x}{24\cdot E\cdot I}\cdot\left(L^3+x^2\cdot(x-2\cdot L)\right)$$

5.1.5 Simple Moment



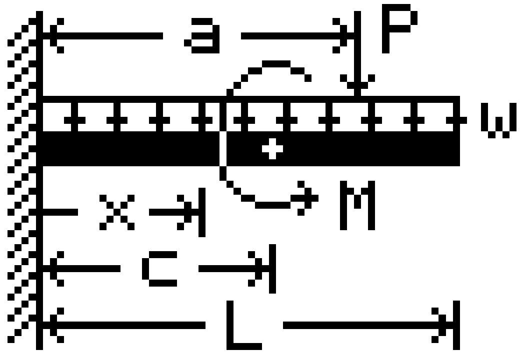
$$Mx = \frac{P}{L}\begin{cases}(L-a)\cdot x, & \text{if } x \leq a\\ a\cdot(L-x), & \text{if } x < a\end{cases} \quad + \quad \frac{M}{L}\cdot\begin{cases}x, & \text{if } x \leq c\\ -(L-x), & \text{if } x > c\end{cases} \quad + \quad \frac{w\cdot x}{2}\cdot(L-x)$$

5.1.6 Simple Shear



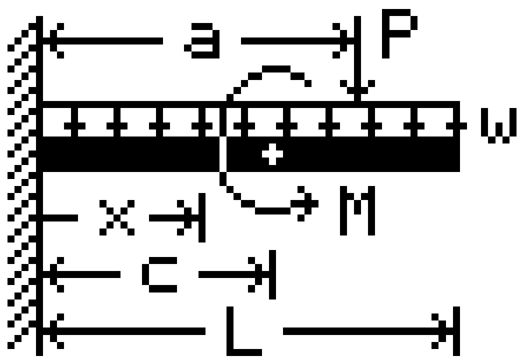
$$V = \frac{P}{L}\cdot\begin{cases}L-a, & \text{if } x \leq a\\ -a, & \text{if } x > a\end{cases} \quad + \quad \frac{M}{L} \quad + \quad \frac{w}{2}(L-2\cdot x)$$

5.1.7 Cantilever Deflection



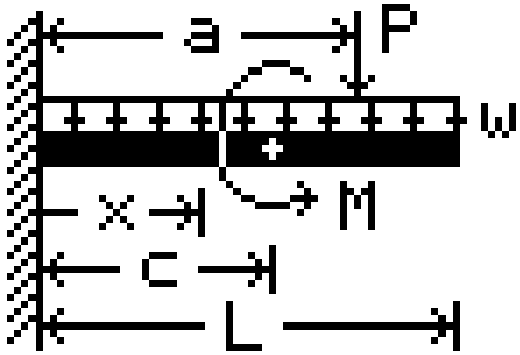
$$y = \frac{P}{6 \cdot E \cdot I} \cdot \begin{cases} x^2 \cdot (x - 3 \cdot a), & \text{if } x \leq a \\ a^2 \cdot (a - 3 \cdot x), & \text{if } x > a \end{cases} \quad + \quad \frac{M}{2 \cdot E \cdot I} \cdot \begin{cases} x^2, & \text{if } x \leq c \\ c \cdot (2x - c), & \text{if } x > c \end{cases}$$
$$- \frac{w \cdot x^2}{24 \cdot E \cdot I} \cdot (6 \cdot L^2 - 4 \cdot L \cdot x + x^2)$$

5.1.8 Cantilever Slope



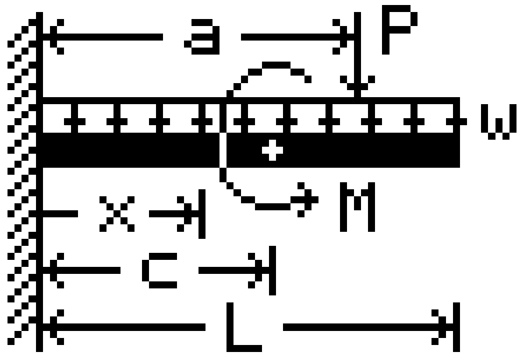
$$\theta = \frac{P}{2 \cdot E \cdot I} \cdot \begin{cases} x \cdot (x - 2 \cdot a), & \text{if } x \leq a \\ -a^2, & \text{if } x > a \end{cases} \quad + \quad \frac{M}{E \cdot I} \cdot \begin{cases} x, & \text{if } x \leq c \\ c, & \text{if } x > c \end{cases}$$
$$- \frac{w \cdot x}{6 \cdot E \cdot I} \cdot (3 \cdot L^2 - 3 \cdot L \cdot x + x^2)$$

5.1.9 Cantilever Moment



$$Mx = \begin{cases} P \cdot (x - a), & \text{if } x \le a \\ 0, & \text{if } x > a \end{cases} \quad + \quad \begin{cases} M, & \text{if } x \le c \\ 0, & \text{if } x > 0 \end{cases} \quad - \quad \frac{w}{2} \cdot (L^2 - 2 \cdot L \cdot x + x^2)$$

5.1.10 Cantilever Shear



$$V = \begin{cases} P, & \text{if } x \le a \\ 0, & \text{if } x > a \end{cases} \quad + \quad w \cdot (L - x)$$

5.2 Electricity

Variable	Description
ϵr	Relative permittivity
μr	Relative permeability
ω	Angular frequency
$\omega 0$	Resonant angular frequency
ϕ	Phase angle
ϕp, ϕs	Parallel and series phase angles
ρ	Resistivity
ΔI	Current change
Δt	Time change
ΔV	Voltage change
A	Wire cross-section area (Wire Resistance), or solenoid cross-section area (Solenoid Inductance), or plate area (Plate Capacitor)
C, $C1$, $C2$	Capacitance
Cp, Cs	Parallel and series capacitance
d	Plate separation
E	Energy
F	Force between charges
f	Frequency
$f0$	Resonant frequency
I	Current, or total current (Current Divider)
$I1$	Current in $r1$
$Imax$	Maximum current
L	Inductance, or length (Wire Resistance, Cylindrical Capacitor)
$l1$, $l2$	Inductance
Lp, Ls	Parallel and series inductances
N	Number of turns
n	Number of turns per unit length
P	Power
q	Charge
$q1$, $q2$	Point charge
Qp, Qs	Parallel and series quality factors
r	Charge distance
R, $r1$, $r2$	Resistance
ri, ro	Inside and outside radii
Rp, Rs	Parallel and series resistances
t	Time
ti, tf	Initial and final times
V	Voltage, or total voltage (Voltage Divider)
$V1$	Voltage across $r1$
Vi, Vf	Initial and final voltages
$Vmax$	Maximum voltage
XC	Reactance of capacitor
XL	Reactance of inductor

5.2.1 Coulumb's Law

This equation describes the electrostatic force between two charged particles.

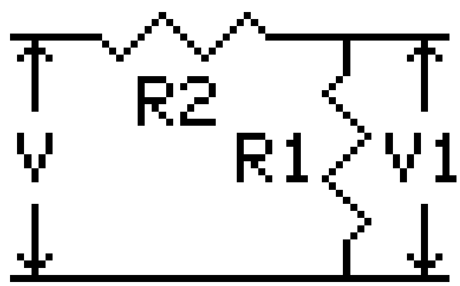
$$F = \frac{1}{4\pi \cdot \epsilon 0 \cdot \epsilon r} \cdot \frac{q1 \cdot q2}{r^2}$$

5.2.2 Ohm's Law and Power

$$V = I \cdot R \qquad\qquad P = V \cdot I$$

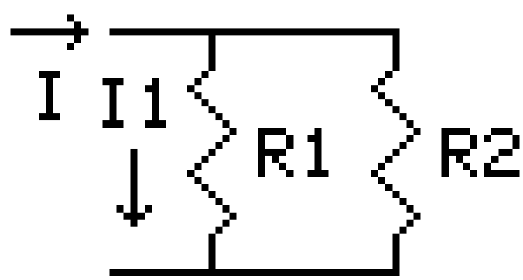
$$P = I^2 \cdot R \qquad\qquad P = \frac{V^2}{R}$$

5.2.3 Voltage Divider



$$V1 = V \cdot \frac{r1}{r1 + r2}$$

5.2.4 Current Divider

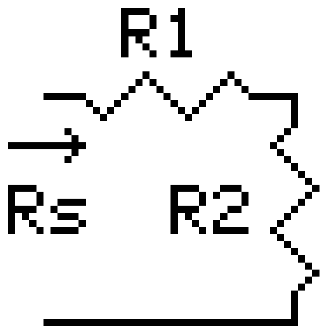


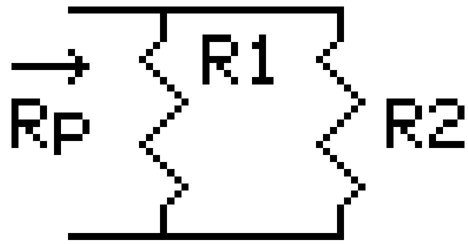
$$I1 = I \cdot \frac{r2}{r1 + r2}$$

5.2.5 Wire Resistance

$$R = \frac{\rho \cdot L}{A}$$

5.2.6 Series/Parallel Resistance

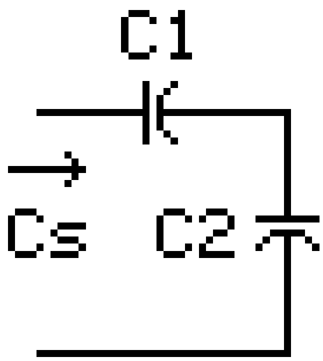


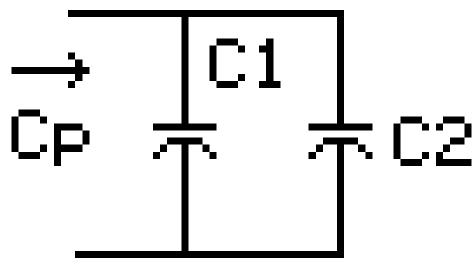


$$Rs = r1 + r2$$

$$\frac{1}{Rp} = \frac{1}{r1} + \frac{1}{r2}$$

5.2.7 Series/Parallel Capacitance

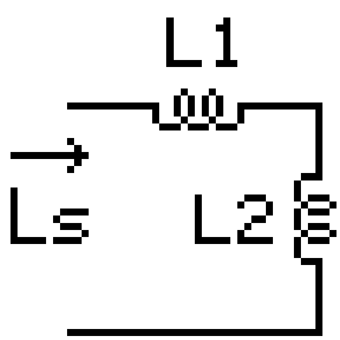


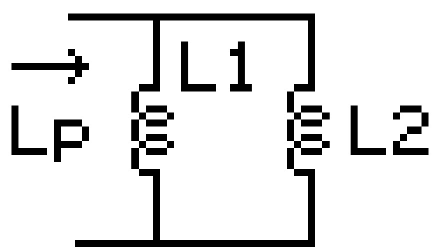


$$\frac{1}{Cs} = \frac{1}{C1} + \frac{1}{C2}$$

$$Cp = C1 + C2$$

5.2.8 Series/Parallel Inductance





$$Ls = l1 + l2$$

$$\frac{1}{Lp} = \frac{1}{l1} + \frac{1}{l2}$$

5.2.9 Capacitive Energy

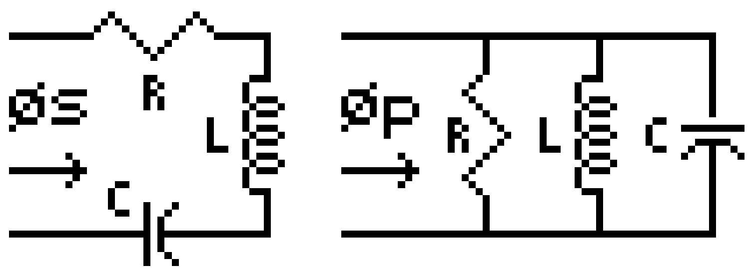
$$E = \frac{C \cdot V^2}{2}$$

5.2.10 Inductive Energy

$$E = \frac{L \cdot I^2}{2}$$

5.2.11 RLC Current Delay

The phase delay (angle) is positive for current lagging voltage.



$$\tan(\phi s) = \frac{XL - XC}{R} \qquad \tan(\phi p) = \frac{\frac{1}{XC} - \frac{1}{XL}}{\frac{1}{R}}$$

$$XC = \frac{1}{\omega \cdot XC} \qquad XL = \omega \cdot L$$

$$\omega = 2\pi \cdot f$$

5.2.12 DC Capacitor Current

These equations approximate the DC current required to change the voltage on a capacitor in a certain time interval.

$$I = C \cdot \frac{\Delta V}{\Delta t} \qquad \Delta V = Vf - Vi \qquad \Delta t = tf - ti$$

5.2.13 Capacitor Charge

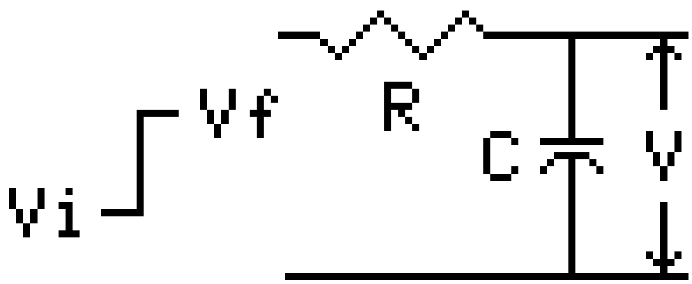
$$q = C \cdot V$$

5.2.14 DC Inductor Voltage

These equations approximate the DC voltage induced in an inductor by a change in current in a certain time interval.

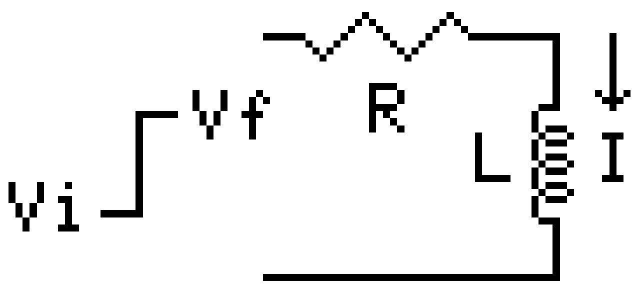
$$V = -L \cdot \frac{\Delta I}{\Delta t} \qquad \Delta I = I1 - I0 \qquad \Delta t = tf - ti$$

5.2.15 RC Transient



$$V = Vf - (Vf - Vi) \cdot e^{-t/(R \cdot C)}$$

5.2.16 RL Transient



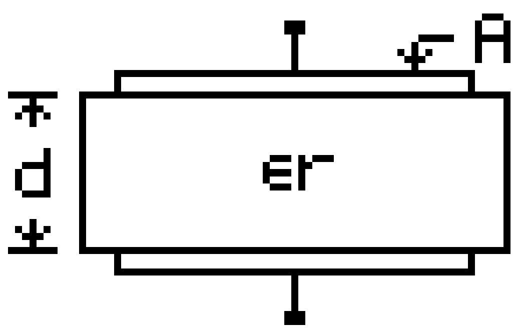
$$I = \frac{1}{R} \cdot \left(Vf - (Vf - Vi) e^{-t \cdot R/L} \right)$$

5.2.17 Resonant Frequency

$$\omega 0 = \frac{1}{\sqrt{L \cdot C}} \qquad Qs = \frac{1}{R} \cdot \sqrt{\frac{L}{C}}$$

$$Qp = R \cdot \sqrt{\frac{C}{L}} \qquad \omega 0 = 2\pi \cdot f0$$

5.2.18 Plate Capacitor

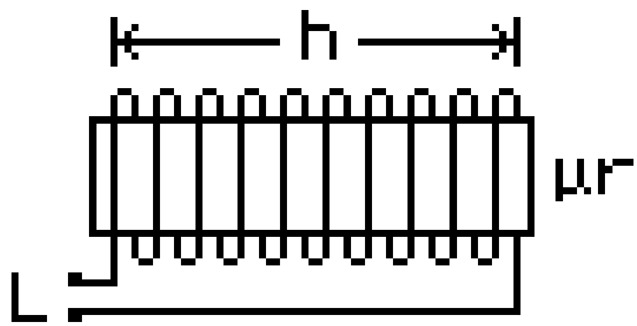


$$C = \frac{\epsilon 0 \cdot \epsilon r \cdot A}{d}$$

5.2.19 Cylindrical Capacitor

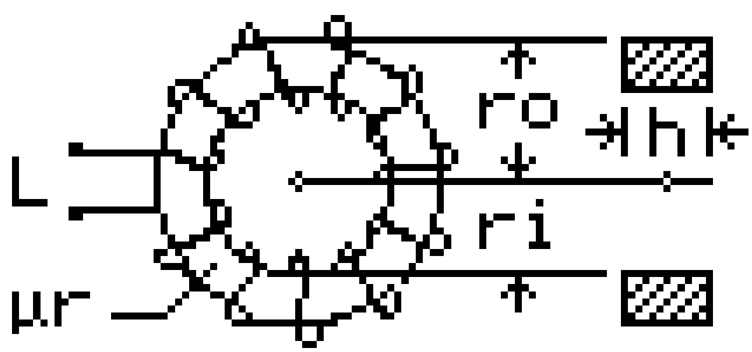
$$C = \frac{2\pi \cdot \epsilon 0 \cdot \epsilon r \cdot L}{\ln\left(\frac{ro}{ri}\right)}$$

5.2.20 Solenoid Inductance



$$L = \mu 0 \cdot \mu r \cdot n^2 \cdot A \cdot h$$

5.2.21 Toroid Inductance



$$L = \frac{\mu 0 \cdot \mu r \cdot N^2 \cdot h}{2\pi} \cdot \ln\left(\frac{ro}{ri}\right)$$

5.2.22 Sinusoidal Voltage

$$V = Vmax \cdot \sin(\omega \cdot t + \phi)$$

5.2.23 Sinusoidal Current

$$I = Imax \cdot \sin(\omega \cdot t + \phi)$$

5.3 Fluids

Variable	Description
ϵ	Roughness
μ	Dynamic viscosity
ρ	Density
ΔP	Pressure change
Δy	Height change
ΣK	Total fitting coefficients
A	Cross-sectional area
$A1$, $A2$	Initial and final cross-sectional areas
D	Diameter
$D1$, $D2$	Initial and final diameters
h	Depth relative $P0$ reference depth
hL	Head loss
L	Length
M	Mass flow rate
n	Kinematic viscosity
P	Pressure at h
$P0$	Reference pressure
$P1$, $P2$	Initial and final pressures
Q	Volume flow rate
RN	Reynolds number
$v1$, $v2$	Initial and final velocities
$vavg$	Average velocity
W	Power input
$y1$, $y2$	Initial and final heights

Remark. The parameter f (see ***Flow in Full Pipes***) is the Fanning friction factor and is based on the parameters $\frac{\epsilon}{D}$ and RN. The Fanning factor is computed as follows.

$$f = \texttt{FANNING}\left(\frac{\epsilon}{D}, RN\right) = \begin{cases} \frac{16}{RN}, & \text{if } RN < 2100 \\ \frac{1}{4} \cdot f_{\text{DW}}, & \text{if } RN \geq 2100 \end{cases}$$

where f_{DW} is the Darcy-Weisbach friction factor as calculated using Serghide's solution. Let

$$A = -2 \cdot \log\left(\frac{\epsilon/D}{3.7} + \frac{12}{RN}\right), \quad B = -2 \cdot \log\left(\frac{\epsilon/D}{3.7} + \frac{2.51 \cdot A}{RN}\right), \quad C = -2 \cdot \log\left(\frac{\epsilon/D}{3.7} + \frac{2.51 \cdot B}{RN}\right).$$

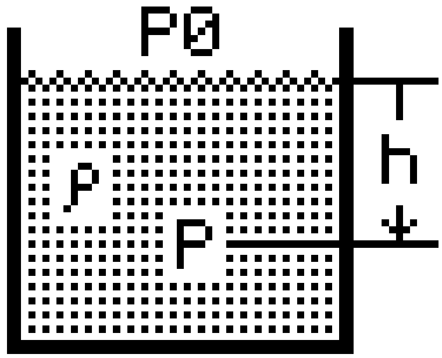
Then

$$\frac{1}{\sqrt{f_{\text{DW}}}} = A - \frac{(B-A)^2}{C - 2 \cdot B + A}.$$

The Fanning factor (if used) should be computed first, and then set as a constant value once computed.

5.3.1 Pressure at Depth

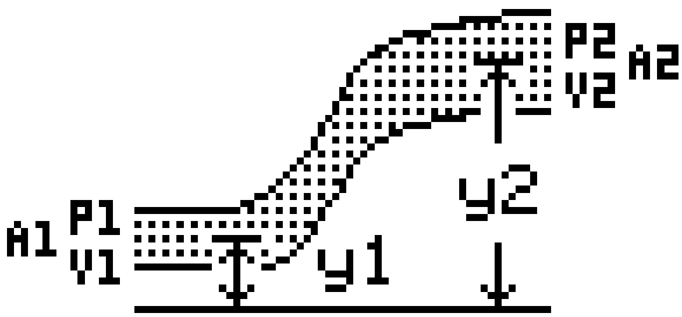
This equation describes hydrostatic pressure for an incompressible fluid. Depth h is positive downward from the reference.



$$P = P0 + \rho \cdot g \cdot h$$

5.3.2 Bernoulli Equation

These equations represent the streamlined flow of incompressible fluid.



$$\Delta P = P2 - P1$$

$$\Delta y = y2 - y1$$

$$M = \rho \cdot Q$$

$$\frac{\Delta P}{\rho} + \frac{v2^2 - v1^2}{2} + g \cdot \Delta y = 0$$

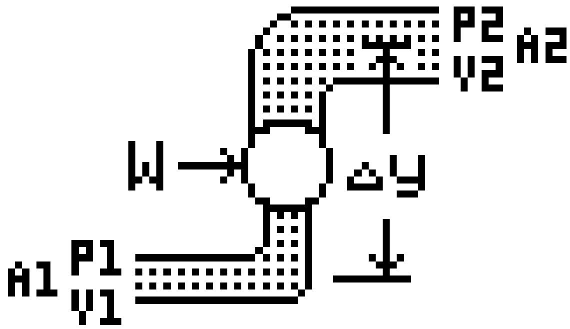
$$\frac{\Delta P}{\rho} + \frac{v2^2 \cdot \left[1 - \left(\frac{A2}{A1}\right)^2\right]}{2} + g \cdot \Delta y = 0$$

$$\frac{\Delta P}{\rho} + \frac{v1^2 \cdot \left[\left(\frac{A2}{A1}\right)^2 - 1\right]}{2} + g \cdot \Delta y = 0$$

$$Q = A2 \cdot v2 \qquad Q = A1 \cdot v1 \qquad A1 = \frac{\pi \cdot D1^2}{4} \qquad A2 = \frac{\pi \cdot D2^2}{4}$$

5.3.3 Flow with Losses

These equations extend Bernoulli's equation to include power input (or output) and head loss.



$$\Delta P = P2 - P1 \qquad M \cdot \left(\frac{\Delta P}{\rho} + \frac{v2^2 - v1^2}{2} + g \cdot \Delta y + hL\right) = W$$

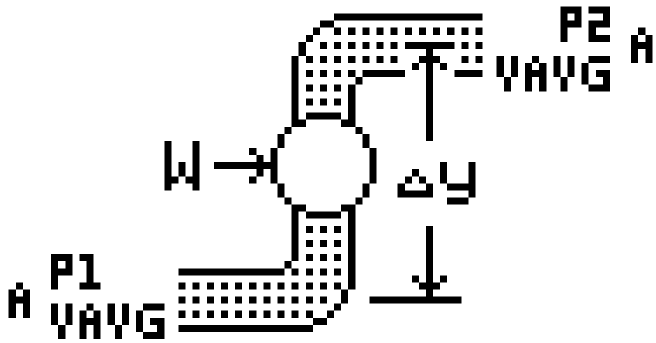
$$\Delta y = y2 - y1 \qquad M \cdot \left(\frac{\Delta P}{\rho} + \frac{v2^2 \cdot \left[1 - \left(\frac{A2}{A1}\right)^2\right]}{2} + g \cdot \Delta y + hL\right) = W$$

$$M = \rho \cdot Q \qquad M \cdot \left(\frac{\Delta P}{\rho} + \frac{v1^2 \cdot \left[\left(\frac{A2}{A1}\right)^2 - 1\right]}{2} + g \cdot \Delta y + hL\right) = W$$

$$Q = A2 \cdot v2 \qquad Q = A1 \cdot v1 \qquad A1 = \frac{\pi \cdot D1^2}{4} \qquad A2 = \frac{\pi \cdot D2^2}{4}$$

5.3.4 Flow in Full Pipes

These equations adapt Bernoulli's equation for flow in a round, full pipe, including power input (or output) and frictional losses.



$$\rho \cdot \left(\frac{\pi \cdot D^2}{4}\right) \cdot vavg \cdot \left[\frac{\Delta P}{\rho} + g \cdot \Delta y + vavg^2 \cdot \left(2 \cdot f \cdot \frac{L}{D} + \frac{\Sigma K}{2}\right)\right] = W$$

$$\Delta P = P2 - P1 \qquad \Delta y = y2 - y1 \qquad M = \rho \cdot Q$$

$$A1 = \frac{\pi \cdot D1^2}{4} \qquad A2 = \frac{\pi \cdot D2^2}{4} \qquad RN = \frac{D \cdot vavg \cdot \rho}{\mu} \qquad n = \frac{\mu}{\rho}$$

5.4 Forces and Energy

Variable	**Description**
α	Angular acceleration
ω	Angular velocity
ωi, ωf	Initial and final angular velocity
ρ	Fluid density
τ	Torque
θ	Angular displacement
a	Acceleration
A	Projected area relative to flow
ar	Centripetal acceleration at r
at	Tangential acceleration at r
Cd	Drag coefficient
E	Energy
F	Force at r or x, or spring force (Hooke's Law), or attractive force (Law of Gravitation), or drag force (Drag Force)
I	Moment of inertia
k	Spring constant
Ki, Kf	Initial and final kinetic energies
m, $m1$, $m2$	Mass
N	Rotational speed
Ni, Nf	Initial and final rotational speeds
P	Instantaneous power
$Pavg$	Average power
r	Radius from rotation axis, or separation distance (Law of Gravitation)
t	Time
v	Velocity
vf, $v1f$, $v2f$	Final velocity
vi, $v1i$	Initial velocity
W	Work
x	Displacement

5.4.1 Linear Mechanics

$$F = m \cdot a \qquad Ki = \frac{1}{2} \cdot m \cdot vi^2 \qquad Kf = \frac{1}{2} \cdot m \cdot vf^2 \qquad W = F \cdot d$$

$$W = Kf - Ki \qquad P = F \cdot v \qquad Pavg = \frac{W}{t} \qquad vf = vi + a \cdot t$$

5.4.2 Angular Mechanics

$$\tau = I \cdot \alpha \qquad Ki = \frac{1}{2} \cdot I \cdot \omega i^2 \qquad Kf = \frac{1}{2} \cdot I \cdot \omega f^2 \qquad W = \tau \cdot \theta$$

$$W = Kf - Ki \qquad P = \tau \cdot \omega \qquad Pavg = \frac{W}{t} \qquad \omega f = \omega i + \alpha \cdot t$$

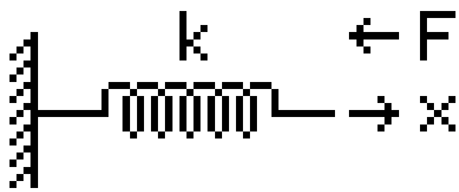
$$at = \alpha \cdot r \qquad \omega = 2\pi \cdot N \qquad \omega i = 2\pi \cdot Ni \qquad \omega f = 2\pi \cdot Nf$$

5.4.3 Centripetal Force

$$F = m \cdot \omega^2 \cdot r \qquad \omega = \frac{v}{r} \qquad ar = \frac{v^2}{r} \qquad \omega = 2\pi \cdot N$$

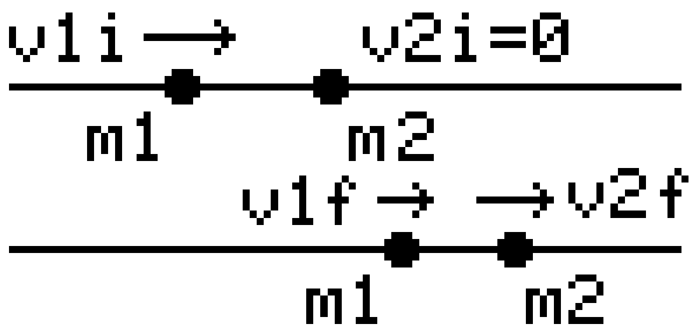
5.4.4 Hooke's Law

The force is that exerted by the spring.



$$F = -k \cdot x \qquad W = -\frac{1}{2} \cdot k \cdot x^2$$

5.4.5 1D Elastic Collisions



$$v1f = \frac{m1 - m2}{m1 + m2} \cdot v1i \qquad v2f = \frac{2 \cdot m1}{m1 + m2} \cdot v1i$$

5.4.6 Drag Force

$$F = Cd \cdot \left(\frac{\rho \cdot v^2}{2}\right) \cdot A$$

5.4.7 Law of Gravitation

$$F = G \cdot \left(\frac{m1 \cdot m2}{r^2}\right)$$

5.4.8 Mass-Energy Relation

$$E = m \cdot c^2$$

5.5 Gases

Variable	Description
λ	Mean free path
ρ	Flow density
$\rho 0$	Stagnation density
A	Flow area
At	Throat area
d	Molecular diameter
k	Specific heat ratio
M	Mach number
m	Mass
MW	Molecular weigth
n	Number of moles, or polytropic constant (Polytropic Processes)
$P,$	Pressure, or flow pressure (Isentropic Flow)
Po, Pf	Initial and final pressure
$P0$	Stagnation pressure
Pc	Pseudocritical pressure
Po, Pf	Initial and final pressure
T	Temperature, or flow temperature (Isentropic Flow)
$T0$	Stagnation temperature
Tc	Pseudocritical temperature
Ti, Tf	Initial and final temperatures
V	Volume
Vi, Vf	Initial and final volumes
$vrms$	Root-mean-square velocity
W	Work

Remark. See ***Real Gas Law*** and ***Real Gas State Change***. The parameter Z is the `ZFACTOR` (compressibility factor) and Z is computed from the Redlich-Kwong equations

$$P = \frac{R \cdot T}{Vm - b} - \frac{a}{Vm \cdot (Vm + b) \cdot \sqrt{T}} \tag{1}$$

$$a = \frac{R^2 \cdot Tc^{5/2}}{9 \cdot (\sqrt[3]{2} - 1) \cdot Pc} \approx \frac{R^2 \cdot Tc^{5/2}}{2.3393 \cdot Pc} \qquad b = \frac{\sqrt[3]{2} - 1}{3} \cdot \frac{R \cdot Tc}{Pc} \approx \frac{R \cdot Tc}{11.5420 \cdot Pc}$$

where $Vm = \frac{V}{m}$ is the molar volume. Set $Vm = \frac{Z \cdot R \cdot T}{P}$ and substitute into equation (1) to obtain the cubic equation

$$Z^3 - Z^2 - q \cdot Z - r = 0 \tag{2}$$

where $r = A \cdot B$, $q = B^2 + B - A$, and

$$A = \frac{Pr}{2.3393 \cdot Tr^{5/2}} \qquad B = \frac{Pr}{11.5420 \cdot Tr} \qquad Tr = \frac{T}{Tc} \qquad Pr = \frac{P}{Pc}$$

Then Z is the largest real-valued solution to equation (2).

5.5.1 Ideal Gas Law

$$P \cdot V = n \cdot R \cdot T \qquad\qquad m = n \cdot MW$$

5.5.2 Ideal Gas State Change

$$\frac{P2 \cdot V2}{T2} = \frac{P1 \cdot V1}{T1}$$

5.5.3 Isothermal Expansion

These equations apply to an ideal gas.

$$W = n \cdot R \cdot T \cdot \ln\left(\frac{Vf}{Vi}\right)$$

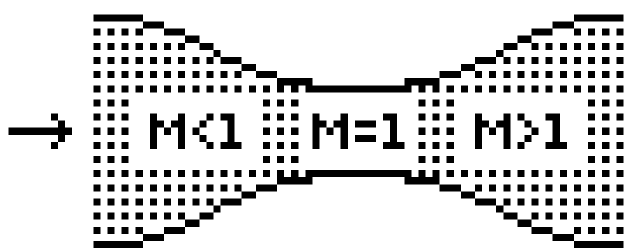
5.5.4 Polytropic Process

These equations describe a reversible pressure-volume change of an ideal gas such that $P \cdot V^n$ is constant. Special cases include isothermal processes ($n = 1$), isentropic processes ($n = k$, the specific heat ratio), and constant-pressure processes ($n = 0$).

$$\frac{Pf}{Po} = \left(\frac{Vf}{Vi}\right)^{-n} \qquad\qquad \frac{Tf}{Ti} = \left(\frac{Pf}{Po}\right)^{(n-1)/n}$$

5.5.5 Isentropic Flow

The calculation differs at velocities below and above Mach 1. The Mach number is based on the speed of sound in the compressible fluid.



$$\frac{T}{T0} = \frac{2}{2 + (k-1) \cdot M^2} \qquad\qquad \frac{\rho}{\rho 0} = \left(\frac{T}{T0}\right)^{1/(k-1)}$$

$$\frac{P}{P0} = \left(\frac{T}{T0}\right)^{k/(k-1)} \qquad\qquad \frac{A}{At} = \frac{1}{M} \cdot \left[\frac{2}{k+1} \cdot \left(1 + \frac{k-1}{2} \cdot M^2\right)\right]^{(k+1)/[2\cdot(k-1)]}$$

5.5.6 Real Gas Law

These equations adapt the ideal gas law to emulate real-gas behavior.

$$P \cdot V = n \cdot Z \cdot R \cdot T \qquad\qquad m = n \cdot MW$$

Remark. In this case,

$$Z = \texttt{ZFACTOR}\left(\frac{T}{Tc}, \frac{P}{Pc}\right).$$

5.5.7 Real Gas State Change

This equation adapt the ideal gas state change equation to emulate real-gas behavior.

$$\frac{Pf \cdot Vf}{Zf \cdot Tf} = \frac{Po \cdot Vi}{Zi \cdot Ti}$$

Remark. In the equation above,

$$Zf = \text{ZFACTOR}\left(\frac{Tf}{Tc}, \frac{Pf}{Pc}\right) \quad \text{and} \quad Zi = \text{ZFACTOR}\left(\frac{Ti}{Tc}, \frac{Po}{Pc}\right).$$

5.5.8 Kinetic Theory

These equations describe properties of an ideal gas.

$$P = \frac{n \cdot MW \cdot vrms^2}{3 \cdot V} \qquad vrms = \sqrt{\frac{3 \cdot R \cdot T}{MW}}$$

$$\lambda = \frac{1}{\sqrt{2} \cdot \pi \cdot \left(\frac{n \cdot NA}{V}\right) \cdot d^2} \qquad m = n \cdot MW$$

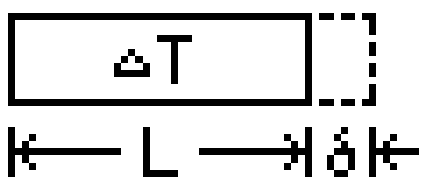
5.6 Heat Transfer

Variable	Description
α	Expansion coefficient
δ	Elongation
$\lambda 1$, $\lambda 2$	Lower and upper wavelength limits
λmax	Wavelength of maximum emissive power
ΔT	Temperature difference
A	Area
c	Specific heat
$eb12$	Emmissive power in range $\lambda 1$ to $\lambda 2$
eb	Total emissive power
f	Fraction of emissive power in the range $\lambda 1$ to $\lambda 2$
h, $h1$, $h3$	Convective heat-transfer coefficient
k, $k1$, $k2$, $k3$	Thermal conductivity
L, $l1$, $l2$, $l3$	Length
m	Mass
Q	Heat capacity
q	Heat transfer rate
T	Temperature
Tc	Cold surface temperature (Conduction), or cold fluid temperature
Th	Hot surface temperature, or hot fluid temperature (Conduction + Convection)
Ti, Tf	Initial and final temperatures
U	Overall heat transfer coefficient

5.6.1 Heat Capacity

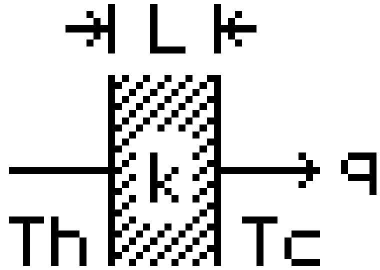
$$Q = m \cdot c \cdot \Delta T \qquad Q = m \cdot c \cdot (Tf - Ti)$$

5.6.2 Thermal Expansion



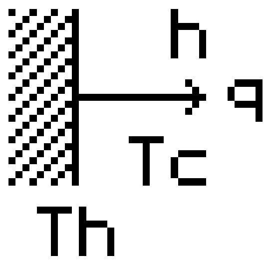
$$\delta = \alpha \cdot L \cdot \Delta T \qquad \delta = \alpha \cdot L \cdot (Tf - Ti)$$

5.6.3 Conduction



$$q = \frac{k \cdot A}{L} \cdot \Delta T \qquad q = \frac{k \cdot A}{L} \cdot (Th - Tc)$$

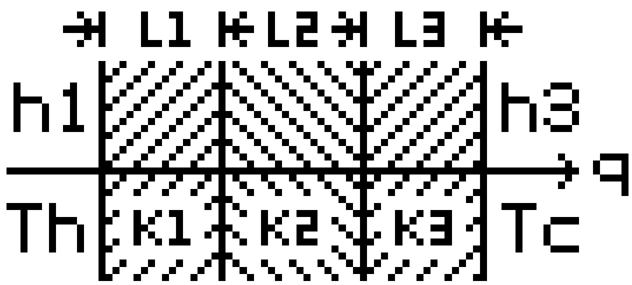
5.6.4 Convection



$$q = h \cdot A \cdot \Delta T \qquad q = h \cdot A \cdot (Th - Tc)$$

5.6.5 Conduction + Convection

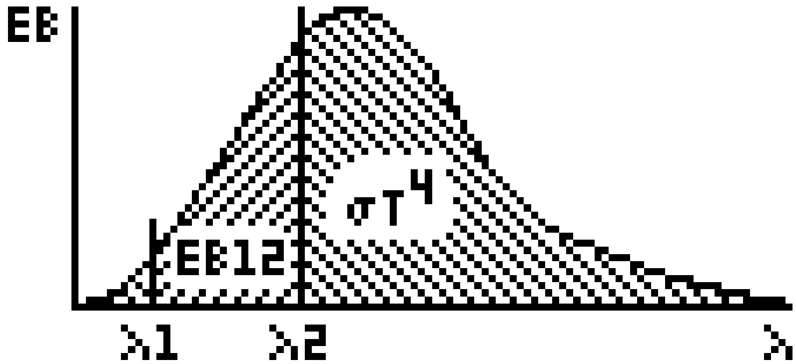
If we have fewer than three layers, we give the extra layers a zero thickness and any nonzero conductivity. The two temperatures are fluid temperatures—if instead we know a ***surface*** temperature, we set the corresponding convective coefficient to 10^{499}.



$$q = \frac{A \cdot \Delta T}{\frac{1}{h1} + \frac{l1}{k1} + \frac{l2}{k2} + \frac{l3}{k3} + \frac{1}{h3}} \qquad U = \frac{q}{A \cdot \Delta T}$$

$$q = \frac{A \cdot (Th - Tc)}{\frac{1}{h1} + \frac{l1}{k1} + \frac{l2}{k2} + \frac{l3}{k3} + \frac{1}{h3}} \qquad U = \frac{q}{A \cdot (Th - Tc)}$$

5.6.6 Black Body Radiation



$$eb = \sigma \cdot T^4 \qquad eb12 = f \cdot eb$$
$$q = eb \cdot A \qquad \lambda max = c3$$

$$f = \mathtt{F0}\lambda(\lambda 2, T) - \mathtt{F0}\lambda(\lambda 1, T)$$

Remark. The $\mathtt{F0}\lambda$ function returns the fraction of total black-body emissive power at temperature T between wavelengths 0 and λ.

$$\mathtt{F0}\lambda = F_{0\to\lambda T} = \frac{2\pi \cdot C_1}{\sigma C_2^4} \int_{\zeta}^{\infty} \frac{\zeta^3}{e^{\zeta} - 1}\, d\zeta$$

where

$$\sigma = \frac{2 \cdot C1 \cdot \pi^5}{15 \cdot C_2^4}, \qquad \zeta = \frac{C_2}{\lambda \cdot T}, \qquad C_1 = h \cdot c_0^2, \qquad C_2 = \frac{h \cdot c_0}{k_B},$$

h is Planck's constant, c_0 is the speed of light, and k_B is the Boltzmann constant. The $\mathtt{F0}\lambda$ function approximates this integral by using the series expansion

$$\frac{1}{e^{\zeta} - 1} = \frac{e^{-\zeta}}{1 - e^{-\zeta}} = e^{-\zeta} + e^{-2\zeta} + e^{-3\zeta} + \cdots$$

and integration by parts. Thus

$$F_{0\to\lambda T} = \frac{15}{\pi^4} \sum_{m=1}^{\infty} \left[\frac{e^{-m\zeta}}{m} \cdot \left(\zeta^3 + \frac{3 \cdot \zeta^2}{m} + \frac{6 \cdot \zeta}{m^2} + \frac{6}{m^3} \right) \right]$$

5.7 Magnetism

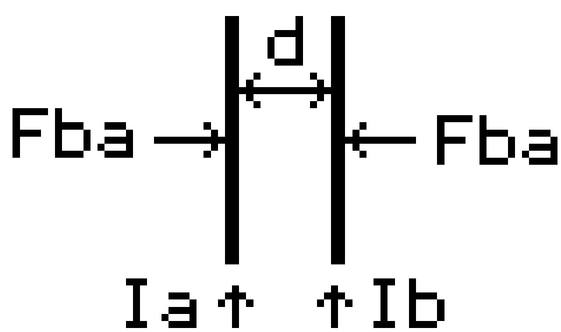
Variable	Description
μr	Relative permeability
B	Magnetic field
d	Separation distance
Fba	Force
I, Ia, Ib	Current
L	Length
N	Total number of turns
n	Number of turns per unit length
r	Distance from center of wire
ri, ro	Inside and outside radii of toroid
rw	Radius of wire

5.7.1 Straight Wire

$$B = \frac{\mu 0 \cdot \mu r \cdot I}{2\pi} \cdot \begin{cases} \frac{r}{rw^2}, & \text{if } r < rw \\ \frac{1}{r}, & \text{if } r \geq rw \end{cases}$$

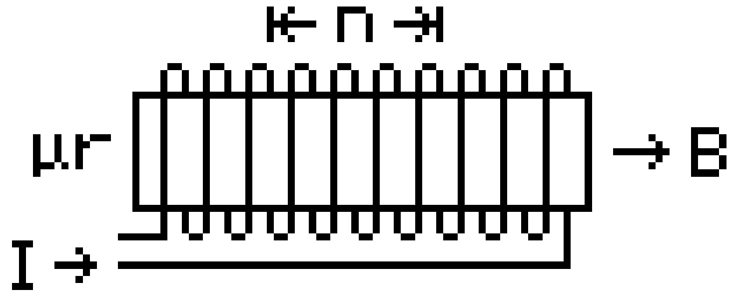
5.7.2 Force Between Wires

The force between wires is positive for an attractive force (for currents having the same sign).



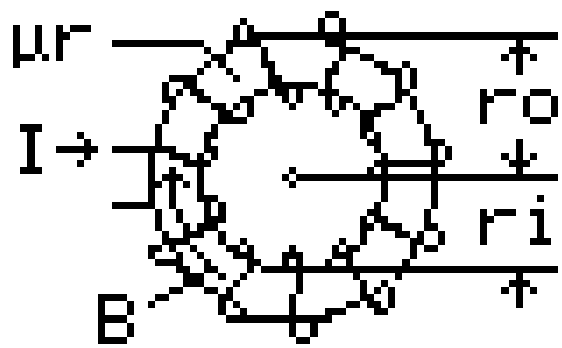
$$Fba = \frac{\mu 0 \cdot \mu r \cdot Ib \cdot Ia}{2\pi \cdot d}$$

5.7.3 Magnetic Field in Solenoid



$$B = \mu 0 \cdot \mu r \cdot I \cdot n$$

5.7.4 Magnetic Field in Toroid



$$B = \frac{\mu 0 \cdot \mu r \cdot I \cdot N}{2\pi} \cdot \left(\frac{2}{ro + ri}\right)$$

5.8 Motion

Variable	Description
α	Angular acceleration
ω	Angular velocity (Circular Motion), or angular velocity at t (Angular Motion)
$\omega 0$	Initial angular velocity
ρ	Fluid density
θ	Angular position at t
$\theta 0$	Initial angular position (Angular Motion), or initial vertical angle (Projectile Motion)
a	Acceleration
A	Projected horizontal area
ar	Centripetal acceleration at r
Cd	Drag coefficient
m	Mass
M	Planet mass
N	Rotational speed
R	Horizontal range (Projectile Motion), or planet radius (Escape Velocity)
r	Radius
t	Time
v	Velocity at t (Linear Motion), or tangential velocity at r (Circular Motion), or terminal velocity (Terminal Velocity), or escape velocity (Escape Velocity)
$v0$	Initial velocity
vx	Horizontal component of velocity at t
vy	Vertical component of velocity at t
x	Horizontal position at t
$x0$	Initial horizontal position
y	Vertical position at t
$y0$	Initial vertical position

5.8.1 Linear Motion

$$x = x0 + v0 \cdot t + \frac{1}{2} \cdot a \cdot t^2 \qquad x = x0 + \frac{1}{2} \cdot (v0 + v) \cdot t$$

$$x = x0 + v \cdot t - \frac{1}{2} \cdot a \cdot t^2 \qquad v = v0 + a \cdot t$$

5.8.2 Object in Free Fall

$$y = y0 + v0 \cdot t - \frac{1}{2} \cdot g \cdot t^2 \qquad v^2 = v0^2 - 2 \cdot g \cdot (y - y0)$$

$$y = y0 + v \cdot t + \frac{1}{2} \cdot g \cdot t^2 \qquad v = v0 - g \cdot g$$

5.8.3 Projectile Motion

$$x = x0 + v0 \cdot \cos(\theta 0) \cdot t \qquad y = y0 + v0 \cdot \sin(\theta 0) \cdot t$$

$$vx = v0 \cdot \cos(\theta 0) \qquad vy = v0 \cdot \sin(\theta 0) - g \cdot t$$

$$R = \frac{v0^2}{g} \cdot \sin(2 \cdot \theta 0)$$

5.8.4 Angular Motion

$$\theta = \theta 0 + \omega 0 \cdot t + \frac{1}{2} \cdot \alpha \cdot t^2 \qquad \theta = \theta 0 + \omega \cdot t - \frac{1}{2} \cdot \alpha \cdot t^2$$

$$\theta = \theta 0 + \frac{1}{2} \cdot (\omega 0 + \omega) \cdot t \qquad \omega = \omega 0 + \alpha \cdot t$$

5.8.5 Circular Motion

$$\omega = \frac{v}{r} \qquad ar = \frac{v^2}{r} \qquad \omega = 2\pi \cdot N$$

5.8.6 Terminal Velocity

$$v = \sqrt{\frac{2 \cdot m \cdot g}{Cd \cdot \rho \cdot A}}$$

5.8.7 Escape Velocity

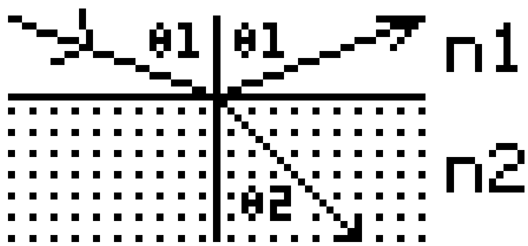
$$v = \sqrt{\frac{2 \cdot G \cdot M}{R}}$$

5.9 Optics

Variable	Description
$\theta 1$	Angle of incidence
$\theta 2$	Angle of refraction
θB	Brewster angle
θc	Critical angle
f	Focal length
m	Magnification
n, $n1$, $n2$	Index of refraction
r, $r1$, $r2$	Radius of curvature
u	Distance to object
v	Distance to image

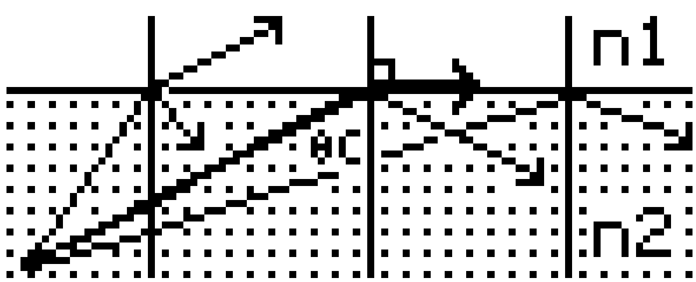
For reflection and refraction problems, the focal length and radius of curvature are positive in the direction of the outgoing light (reflected or refracted). The object distance is positive in front of the surface. The image distance is positive in the direction of the outgoing light (reflected or refracted). The magnification is positive for an upright image.

5.9.1 Law of Refraction



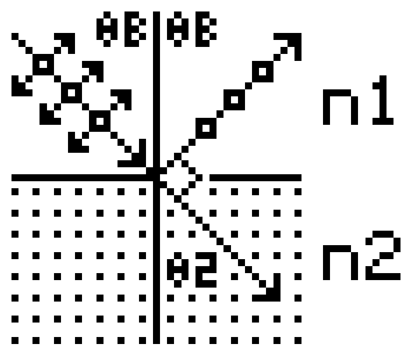
$$n1 \cdot \sin(\theta 1) = n2 \cdot \sin(\theta 2)$$

5.9.2 Critical Angle



$$\sin(\theta c) = \frac{n1}{n2}$$

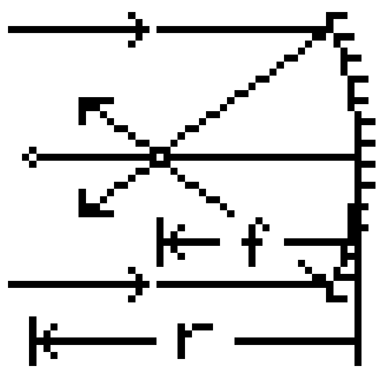
5.9.3 Brewster's Law

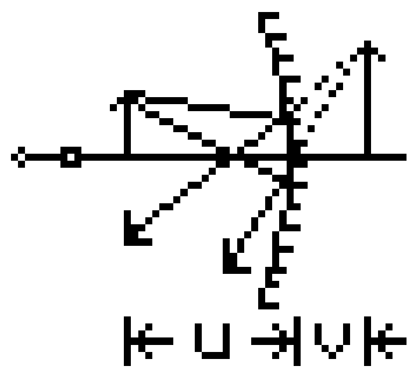


The Brewster angle is the angle of incidence at which the reflected wave is completely polarized.

$$\tan(\theta B) = \frac{n2}{n1} \qquad \theta B + \theta 2 = 90$$

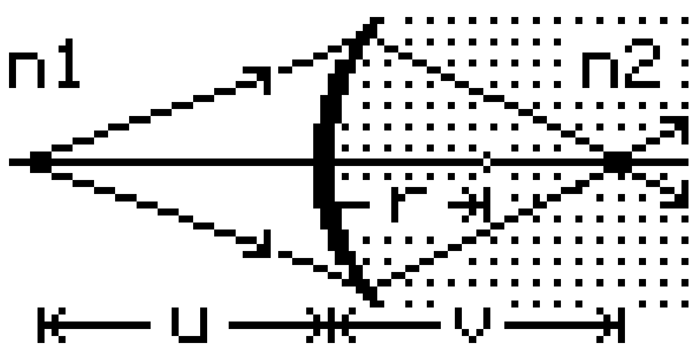
5.9.4 Spherical Reflection





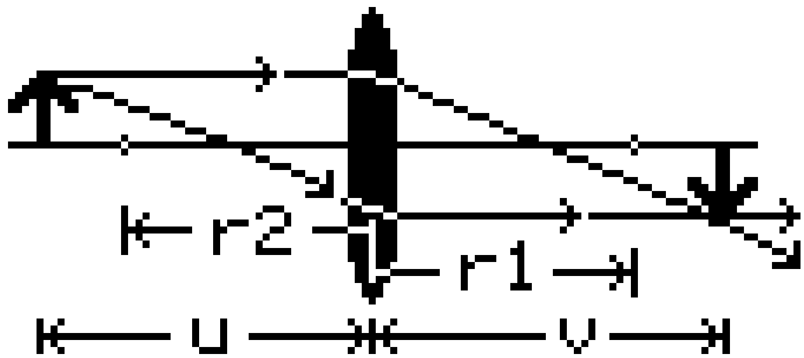
$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f} \qquad f = \frac{1}{2} \cdot r \qquad m = -\frac{v}{u}$$

5.9.5 Spherical Refraction



$$\frac{n1}{u} + \frac{n2}{v} = \frac{n2 - n1}{r}$$

5.9.6 Thin Lens

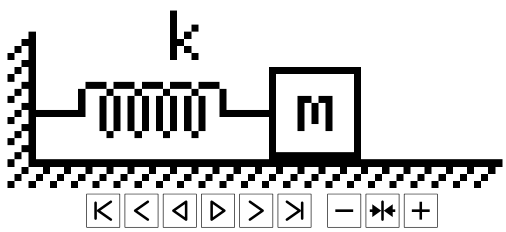


$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f} \qquad f = (n-1) \cdot \left(\frac{1}{r1} - \frac{1}{r2}\right) \qquad m = -\frac{v}{u}$$

5.10 Oscillations

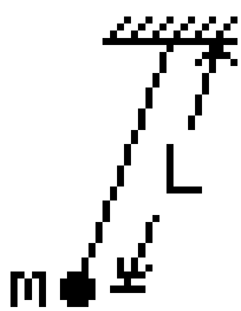
Variable	Description
ω	Angular frequency
ϕ	Phase angle
θ	Cone angle
a	Acceleration at t
f	Frequency
G	Shear modulus of elasticity
h	Cone height
I	Moment of inertia
J	Polar moment of inertia
k	Spring constant
L	Length of pendulum
m	Mass
t	Time
T	Period
v	Velocity at t
x	Displacement at t
xm	Displacement amplitude

5.10.1 Mass-Spring System



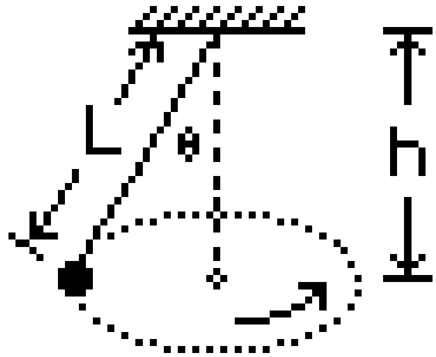
$$\omega = \sqrt{\frac{k}{m}} \qquad T = \frac{2\pi}{\omega} \qquad \omega = 2\pi \cdot f$$

5.10.2 Simple Pendulum



$$\omega = \sqrt{\frac{g}{L}} \qquad T = \frac{2\pi}{\omega} \qquad \omega = 2\pi \cdot f$$

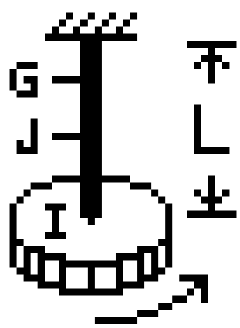
5.10.3 Conical Pendulum



$$\omega = \sqrt{\frac{g}{h}} \qquad T = \frac{2\pi}{\omega} \qquad \omega = 2\pi \cdot f$$

$$h = L \cdot \cos(\theta)$$

5.10.4 Torsional Pendulum



$$\omega = \sqrt{\frac{G \cdot J}{L \cdot I}} \qquad T = \frac{2\pi}{\omega} \qquad \omega = 2\pi \cdot f$$

5.10.5 Simple Harmonic

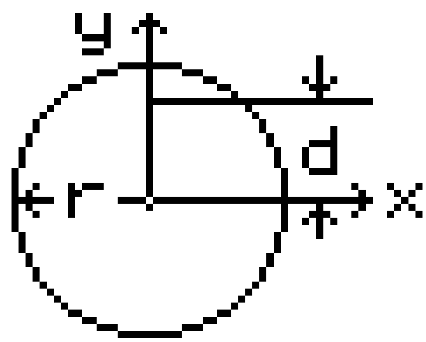
$$x = xm \cdot \cos(\omega \cdot t + \phi) \qquad v = -\omega \cdot xm \cdot \sin(\omega \cdot t + \phi)$$

$$a = -\omega^2 \cdot xm \cdot \cos(\omega \cdot t + \phi) \qquad \omega = 2\pi \cdot f$$

5.11 Plane Geometry

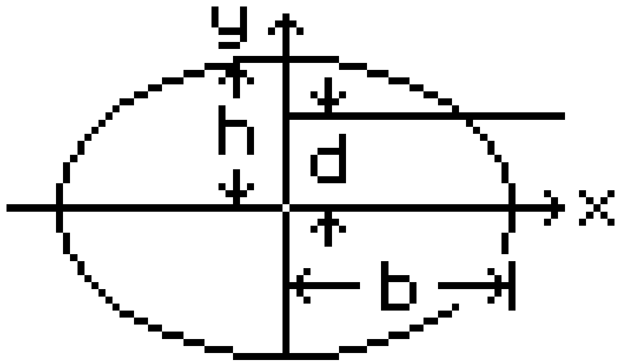
Variable	Description
β	Central angle of polygon
θ	Vertex angle of polygon
A	Area
b	Base length (Rectangle, Triangle), or length of semiaxis in x direction (Ellipse)
C	Circumference
d	Distance to rotation axis in y direction
h	Height (Rectangle, Triangle), or length of semiaxis in y direction (Ellipse)
I, Ix	Moment of inertia about x axis
Id	Moment of inertia about y axis
J	Polar moment of inertia at centroid
L	Side length of polygon
n	Number of sides
P	Perimeter
r	Radius
ri, ro	Inside and outside radii
rs	Distance to side of polygon
rv	Distance to vertex of polygon
v	Horizontal distance to vertex

5.11.1 Circle



$$A = \pi \cdot r^2 \qquad C = 2\pi \cdot r \qquad I = \frac{\pi \cdot r^4}{4} \qquad J = \frac{\pi \cdot r^4}{2} \qquad Id = I + A \cdot d^2$$

5.11.2 Ellipse

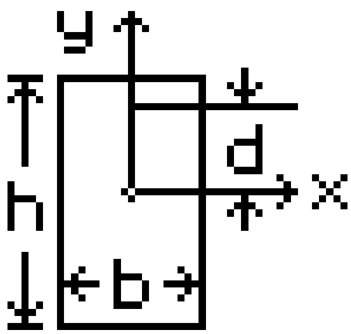


$$A = \pi \cdot b \cdot h \qquad C = 2\pi\sqrt{\frac{b^2 + h^2}{2}}$$

$$I = \frac{\pi \cdot b \cdot h^3}{4} \qquad J = \frac{\pi \cdot b \cdot h}{4} \cdot (b^2 + h^2)$$

$$Id = I + A \cdot d^2$$

5.11.3 Rectangle

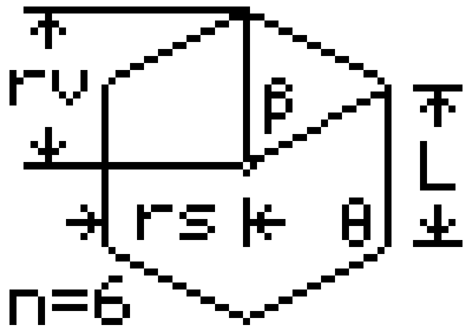


$$A = b \cdot h \qquad P = 2 \cdot b + 2 \cdot h$$

$$I = \frac{b \cdot h^3}{12} \qquad J = \frac{b \cdot h}{12} \cdot (b^2 + h^2)$$

$$Id = I + A \cdot d^2$$

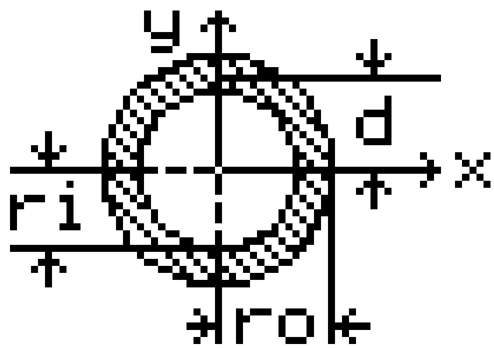
5.11.4 Regular Polygon



$$A = \frac{\frac{1}{4} \cdot n \cdot L^2}{\tan\left(\frac{180}{n}\right)} \qquad P = n \cdot L \qquad rs = \frac{\frac{L}{2}}{\tan\left(\frac{180}{n}\right)}$$

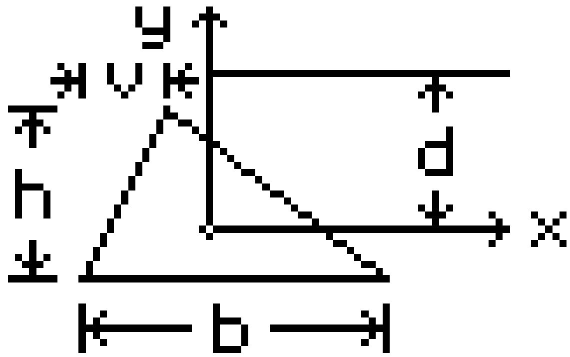
$$rv = \frac{\frac{L}{2}}{\sin\left(\frac{180}{n}\right)} \qquad \theta = \frac{n-2}{n} \cdot 180 \qquad \beta = \frac{360}{n}$$

5.11.5 Circular Ring



$$A = \pi(ro^2 - ri^2) \qquad I = \frac{\pi}{4} \cdot (ro^4 - ri^4) \qquad J = \frac{\pi}{2} \cdot (ro^4 - ri^4) \qquad Id = I + A \cdot d^2$$

5.11.6 Triangle



$$A = \frac{1}{2} \cdot b \cdot h \qquad P = b + \sqrt{v^2 + h^2} + \sqrt{(b-v)^2 + h^2}$$

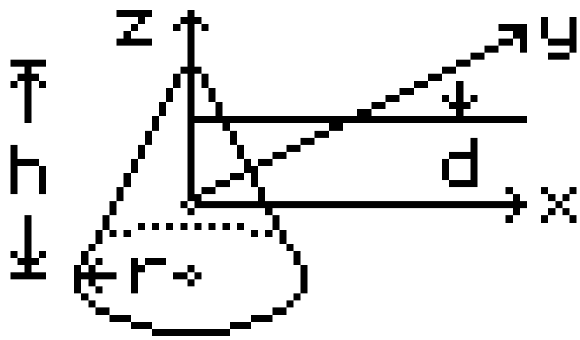
$$Ix = \frac{1}{36} \cdot b \cdot h^3 \qquad Iy = \frac{1}{36} \cdot b \cdot h \cdot (b^2 - b \cdot v + v^2)$$

$$J = \frac{1}{36} \cdot b \cdot h \cdot (h^2 + b^2 - b \cdot v + v^2) \qquad Id = Ix + A \cdot d^2$$

5.12 Solid Geometry

Variable	Description
A	Total surface area
b	Base length
d	Distance to rotation axis in z direction
h	Height in z direction (Cone, Cylinder), or height in y direction (Parallelepiped)
I, Ixx	Moment of inertia about x axis
Id	Moment of inertia in x direction at d
Izz	Moment of inertia about z axis
m	Mass
r	Radius
t	Thickness in z direction
V	Volume

5.12.1 Cone

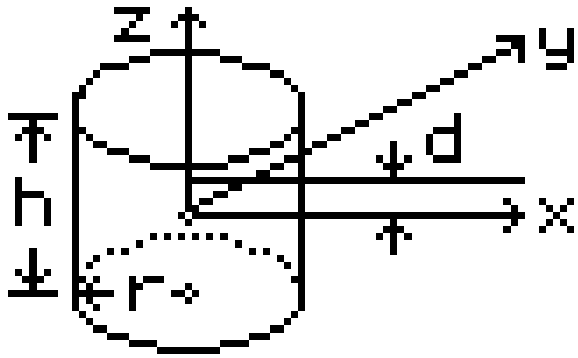


$$V = \frac{\pi}{3} \cdot r^2 \cdot h \qquad A = \pi \cdot r^2 + \pi \cdot r \cdot \sqrt{r^2 + h^2}$$

$$Ixx = \frac{3}{20} \cdot m \cdot r^3 + \frac{3}{80} \cdot m \cdot h^2 \qquad Izz = \frac{3}{10} \cdot m \cdot r^2$$

$$Id = Ixx + m \cdot d^2$$

5.12.2 Cylinder

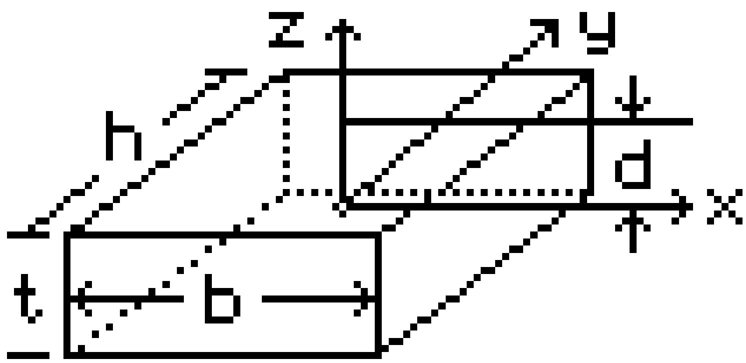


$$V = \pi \cdot r^2 \cdot h \qquad A = 2\pi \cdot r^2 + 2\pi \cdot r \cdot h$$

$$Ixx = \frac{1}{4} \cdot m \cdot r^2 + \frac{1}{12} \cdot m \cdot h^2 \qquad Izz = \frac{1}{2} \cdot m \cdot r^2$$

$$Id = Ixx + m \cdot d^2$$

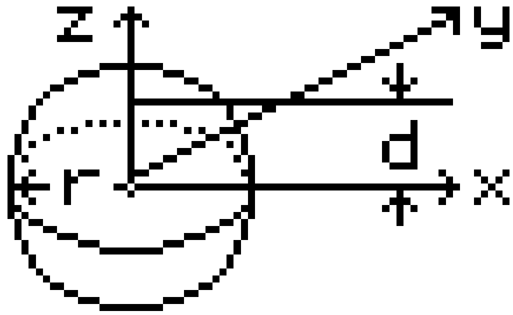
5.12.3 Parallelepiped



$$V = b \cdot h \cdot t \qquad A = 2 \cdot (b \cdot h + b \cdot t + h \cdot t)$$

$$I = \frac{1}{12} \cdot m \cdot (h^2 + t^2) \qquad Id = I + m \cdot d^2$$

5.12.4 Sphere



$$V = \frac{4}{3} \cdot \pi \cdot r^3 \qquad A = 4\pi \cdot r^2 \qquad I = \frac{2}{5} \cdot m \cdot r^2 \qquad Id = I + m \cdot d^2$$

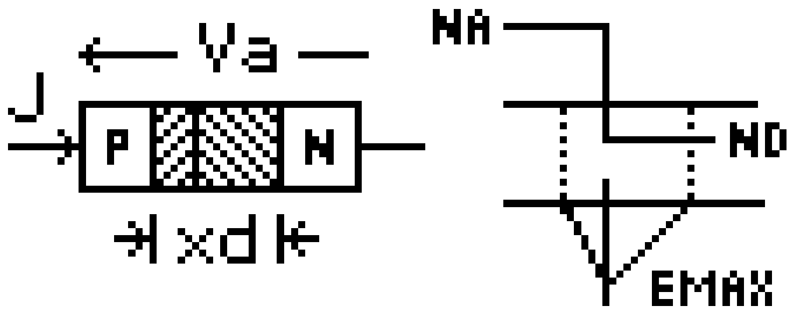
5.13 Solid State Devices

Variable	Description
αF	Forward common-base current gain
αR	Reverse common-base current gain
γ	Body factor
λ	Modulation parameter
μn	Electron mobility
ϕp	Fermi potential
ΔL	Length adjustment (PN Step Junctions), or channel encroachment (NMOS Transistors)
ΔW	Width adjustment (PN Step Junctions), or width contraction (NMOS Transistors)
a	Channel thickness
Aj	Effection junction area
BV	Breakdown voltage
$CG0$	Channel conductance
Cj	Junction capacitance per unit area
Cox	Silicon dioxide capacitance per unit area
$E1$	Breakdown voltage field factor
$Emax$	Maximum electric field
gds	Output conductance
gm	Transconductance
I	Diode current
IB	Total base current
IC	Total collector current
$ICEO$	Collector current (collector-to-base open)
ICO	Collector current (emitter-to-base open)
ICS	Collector-to-base saturation current
ID, IDS	Drain current
IE	Total emitter current
IES	Emitter-to-base saturation current
IS	Transistor saturation current
J	Current density
Js	Saturation current density
L	Drawn mask length (PN Step Junctions), or drawn gate length (NMOS Transistors), or channel length (JFETs)
Le	Effective gate length
NA	P-side doping (PN Step Junctions), or substrate doping (NMOS Transistors)
ND	N-side doping (PN Step Junctions), or N-channel doping (JFETs)
T	Temperature
tox	Gate silicon dioxide thickness
Va	Applied voltage
VBC	Base-to-collector voltage
VBE	Base-to-emitter voltage
Vbi	Built-in voltage
VBS	Substrate voltage
$VCEsat$	Collector-to-emitter saturation voltage
VDS	Applied drain voltage
$VDsat$	Saturation voltage
VGS	Applied gate voltage

Vt	Threshold voltage
$Vt0$	Threshold voltage (at zero substrate voltage)
W	Drawn mask width (PN Step Junctions), or drawn widht (NMOS Transistors), or channel width (JFETs)
We	Effective width
xd	Depletion-region width
$xdmax$	Depletion-layer width
xj	Junction depth

5.13.1 PN Step Junctions

These equations for a silicon PN-junction diode use a "two-sided step-junction" model—the doping density changes abruptly at the junction. The equations assume the current density is determined by minority carriers injected across the depletion region and the PN junction is rectangular in its layout. The temperature should be between 77 and 500 K.



$$Vbi = \frac{k \cdot T}{q} \cdot \ln\left(\frac{NA \cdot ND}{ni^2}\right) \qquad xd = \sqrt{\frac{2\epsilon si \cdot \epsilon 0}{q} \cdot (Vbi - Va) \cdot \left(\frac{1}{NA} + \frac{1}{ND}\right)}$$

$$Cj = \frac{\epsilon si \cdot \epsilon 0}{xd} \qquad Emax = \frac{2 \cdot (Vbi - Va)}{xd}$$

$$BV = \frac{\epsilon si \cdot \epsilon 0 \cdot E1^2}{2 \cdot q} \cdot \left(\frac{1}{NA} + \frac{1}{ND}\right) \qquad J = Js \cdot \left(e^{q \cdot Va/(k \cdot T)} - 1\right)$$

$$I = J \cdot Aj$$

$$Aj = (W + 2 \cdot \Delta W) \cdot (L + 2 \cdot \Delta L) + \pi \cdot (W + L + 2 \cdot \Delta W + 2 \cdot \Delta L) \cdot xj + 2\pi \cdot xj^2$$

Remark. The ni paremeter here refers to the silicon density (intrinsic carrier concentration) at temperature T. This is computed by the `SIDENS` function, which returns

$$ni(T) = \sqrt{N_c \cdot N_v} \cdot e^{-E_g/(2 \cdot k \cdot T)}$$

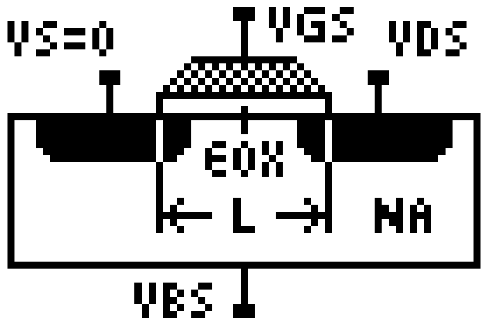
where

$$E_g = 1.17 - 4.73 \cdot 10^{-4} \cdot \frac{T^2}{T + 636}$$

and T is temperature in degrees K.

5.13.2 NMOS Transistors

These equations for a silicon NMOS transistor use a two-port network model. They include linear and non-linear regions in the device characteristics and are based on a gradual-channel approximation (the electric fields in the direction of current flow are small compared to those perpendicular to the flow). The drain current and tansconductance calculations differ depending on wether the transistor is in the linear, saturated, or cutoff region. The equations assume the physical geometry of the device is a rectangle, second-order length-parameter effects are negligible, short-channel, hot-carrier, and velocity-saturation effects are negligible, and subthreshold currents are negligible. See ***PN Step Junctions*** regarding ni.



$$We = W - 2 \cdot \Delta W \qquad Le = L - 2 \cdot \Delta L \qquad Cox = \frac{\epsilon ox \cdot \epsilon 0}{tox}$$

$$IDS = Cox \cdot \mu n \cdot \left(\frac{We}{Le}\right) \cdot (1 + \lambda \cdot VDS) \cdot \begin{cases} 0, & \text{if } VGS < Vt \\ (VGS - Vt) \cdot VDS - \frac{VDS^2}{2}, & \text{if } 0 \le VDS \le VGS - Vt \\ \frac{1}{2} \cdot (VGS - Vt)^2, & \text{if } 0 \le VGS - Vt < VDS \end{cases}$$

$$\gamma = \frac{\sqrt{2 \cdot \epsilon si \cdot \epsilon 0 \cdot q \cdot NA}}{Cox} \qquad Vt = Vt0 + \gamma \cdot (\sqrt{2 \cdot |\phi p| + |VBS|} - \sqrt{2 \cdot |\phi p|})$$

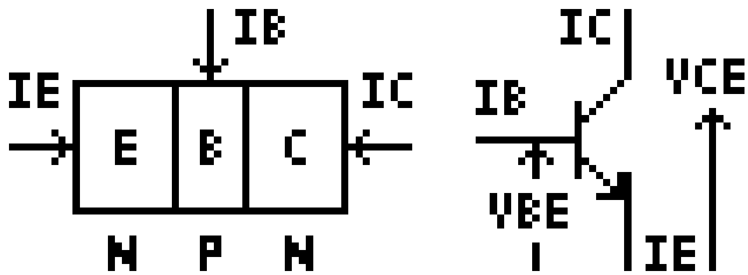
$$\phi p = -\frac{k \cdot T}{q} \cdot \ln\left(\frac{NA}{ni}\right) \qquad gds = IDS \cdot \lambda$$

$$VDsat = VGS - Vt$$

$$gm = \begin{cases} 0, & \text{if } VGS < Vt \\ Cox \cdot \mu n \cdot \left(\frac{We}{Le}\right) \cdot VDS \cdot (1 + \lambda \cdot VDS), & \text{if } 0 \le VDS \le VGS - Vt \\ \sqrt{Cox \cdot \mu n \cdot \left(\frac{We}{Le}\right) \cdot (1 + \lambda \cdot VDS) \cdot 2 \cdot IDS}, & \text{if } 0 \le VGS - Vt < VDS \end{cases}$$

5.13.3 Bipolar Transistors

These equations for an NPN silicon bipolar transistor are based on large-signal models developed by J.J Ebers and J.L. Moll. The offset-voltage calculation differs depending on whether the transistor is saturated or not. The equations also include the special conditions when the emitter-base or collector-base junction is open, which are convenient for measuring transistor parameters.



$$IE = -IES \cdot \left(e^{q \cdot VBE/(k \cdot T)} - 1\right) + \alpha R \cdot ICS \cdot \left(e^{q \cdot VBC/(k \cdot T} - 1\right)$$

$$IC = -ICS \cdot \left(e^{q \cdot VBC/(k \cdot T)} - 1\right) + \alpha F \cdot IES \cdot \left(e^{q \cdot VBE/(k \cdot T} - 1\right)$$

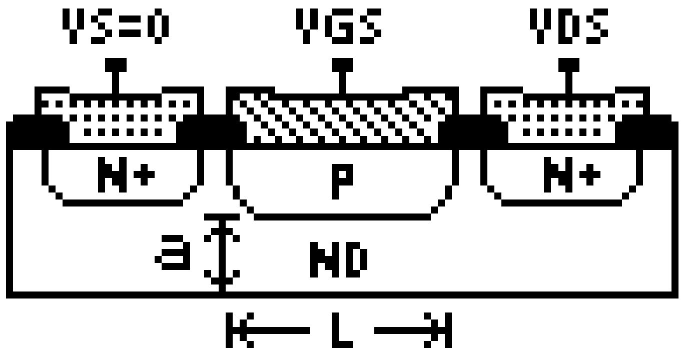
$$VCEsat = \begin{cases} 0, & \text{if } \frac{IC}{IB} \cdot \left(\frac{1-\alpha F}{\alpha F}\right) \geq 1 \\ \dfrac{k \cdot T}{q} \cdot \ln\left[\dfrac{1 + \frac{IC}{IB} \cdot (1 - \alpha R)}{\alpha R \cdot \left(1 - \frac{IC}{IB} \cdot \left(\frac{1-\alpha F}{\alpha F}\right)\right)}\right], & \text{if } \frac{IC}{IB} \cdot \left(\frac{1-\alpha F}{\alpha F}\right) < 1 \end{cases}$$

$$IS = -\alpha F \cdot IES \qquad IS = \alpha R \cdot ICS \qquad IB + IE + IC = 0$$

$$ICO = ICS \cdot (1 - \alpha F \cdot \alpha R) \qquad ICEO = \frac{ICO}{1 - \alpha F}$$

5.13.4 JFETs

These equations for a silicon N-channel junction field-effect transistor (JFET) are based on single-sided step-junction approximation, which assumes the gates are heavily doped compared to the channel doping. The drain-current calculation differs depending on whether the gate-junction depletion-layer thickness is less than or greater than the channel thickness. The equations assume the channel is uniformly doped and end effects (such as contact, drain, and source resistances) are negligible. See ***PN Step Junctions*** regarding ni.



$$Vbi = \frac{k \cdot T}{q} \cdot \ln\left(\frac{ND}{ni}\right) \qquad xdmax = \sqrt{\frac{2 \cdot \epsilon si \epsilon 0}{q \cdot ND} \cdot (Vbi - VGS + VDS)}$$

$$CG0 = q \cdot ND \cdot \mu n \cdot \left(\frac{a \cdot W}{L}\right) \qquad VDsat = \frac{q \cdot ND \cdot a^2}{2 \cdot \epsilon si \cdot \epsilon 0} - (Vbi - VGS)$$

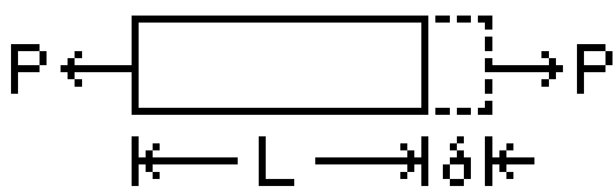
$$Vt = Vbi - \frac{q \cdot ND \cdot a^2}{2 \cdot \epsilon si \cdot \epsilon 0} \qquad gm = CG0 \cdot \left(1 - \sqrt{\frac{2 \cdot \epsilon si \cdot \epsilon 0}{q \cdot ND \cdot a^2} \cdot (Vbi - VGS)}\right)$$

$$ID = CG0 \cdot \begin{cases} VGS - \frac{2}{3} \cdot \sqrt{\frac{2 \cdot \epsilon si \cdot \epsilon 0}{q \cdot ND \cdot a^2}} \left((Vbi - VGS + VDS)^{3/2} - (Vbi - VGS)^{3/2}\right), & \text{if } xdmax < a \\ \frac{q \cdot ND \cdot a^2}{6 \cdot \epsilon si \cdot \epsilon 0} - (Vbi - VGS) \cdot \left(1 - \frac{2}{3} \cdot \sqrt{\frac{2 \epsilon si \cdot \epsilon 0}{q \cdot ND \cdot a^2} \cdot (Vbi - VGS)}\right), & \text{if } xdmax \geq a \end{cases}$$

5.14 Stress Analysis

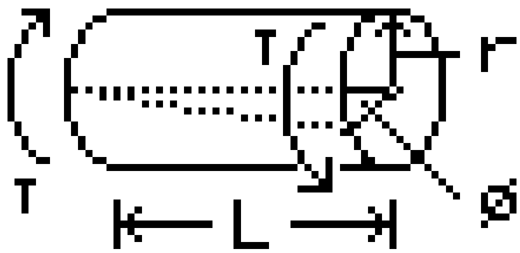
Variable	Description
δ	Elongation
ϵ	Normal strain
γ	Shear strain
ϕ	Angle of twist
σn	Normal stress
$\sigma 1$	Maximum principal normal stress
$\sigma 2$	Minimum principal normal stress
σavg	Normal stress on plane of maximum shear stress
σx	Normal stress in x direction
$\sigma x1$	Normal stress in rotated-x direction
σy	Normal stress in y direction
$\sigma x1$	Normal stress in rotated-y direction
τ	Shear stress
τmax	Maximum shear stress
$\tau x1y1$	Rotated shear stress
τxy	Shear stress
θ	Rotation angle
$\theta p1$	Angle to plane of maximum principal normal stress
$\theta p2$	Angle to plane of minimum principal normal stress
θs	Angle to plane of maximum shear stress
A	Area
E	Modulus of elasticity
G	Shear modulus of elasticity
J	Polar moment of inertia
L	Length
P	Load
r	Radius
T	Torque

5.14.1 Normal Stress



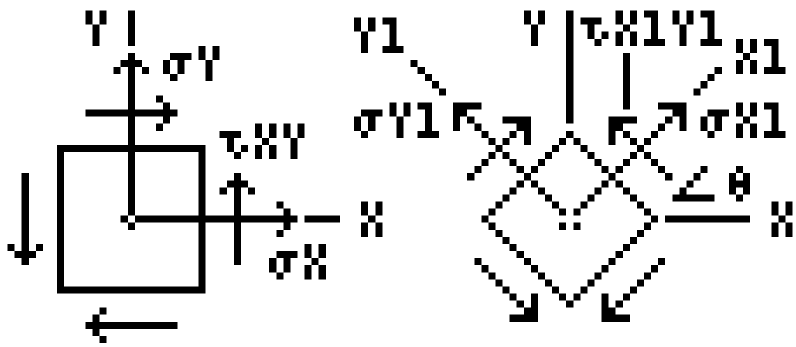
$$\sigma n = E \cdot \epsilon \qquad \epsilon = \frac{\delta}{L} \qquad \sigma n = \frac{P}{A}$$

5.14.2 Shear Stress



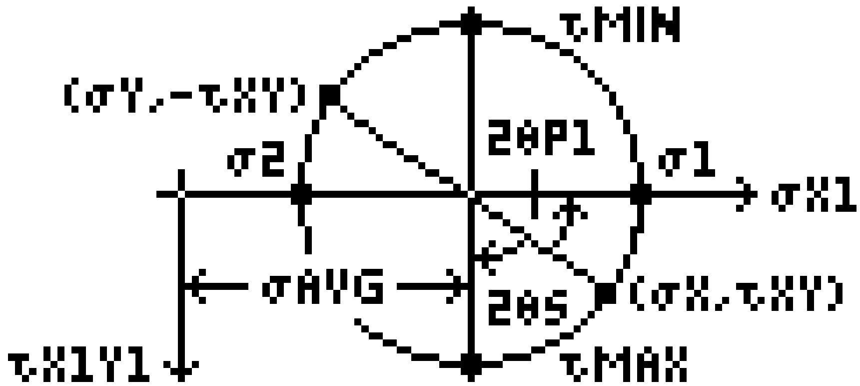
$$\tau = G \cdot \gamma \qquad \gamma = \frac{r \cdot \phi}{L} \qquad \tau = \frac{T \cdot r}{J}$$

5.14.3 Stress on an Element



$$\sigma x1 = \frac{\sigma x + \sigma y}{2} + \frac{\sigma x - \sigma y}{2} \cdot \cos(2 \cdot \theta) + \tau xy \cdot \sin(2 \cdot \theta)$$
$$\tau x1y1 = -\left(\frac{\sigma x - \sigma y}{2}\right) \cdot \sin(2\theta) + \tau xy \cdot \cos(2 \cdot \theta)$$
$$\sigma x1 + \sigma y1 = \sigma x + \sigma y$$

5.14.4 Mohr's Circle



$$\sigma 1 = \frac{\sigma x + \sigma y}{2} + \sqrt{\left(\frac{\sigma x - \sigma y}{2}\right)^2 + \tau xy^2}$$

$$\sigma 1 + \sigma 2 = \sigma x + \sigma y$$

$$\tau max = \frac{\sigma 1 - \sigma 2}{2}$$

$$\sigma avg = \frac{\sigma x + \sigma y}{2}$$

$$\sin(2 \cdot \theta p1) = \frac{\tau xy}{\sqrt{\left(\frac{\sigma x - \sigma y}{2}\right)^2 + \tau xy^2}}$$

$$\theta p2 = \theta p1 + 90$$

$$\theta s = \theta p1 - 45$$

5.15 Waves

Variable	Description
β	Sound level
λ	Wavelength
ω	Angular frequency
ρ	Density of medium
B	Bulk modulus of elasticity
f	Frequency
I	Sound intensity
k	Angular wave number
s	Longitudinal displacement at x and t
sm	Longitudinal amplitude
t	Time
v	Speed of sound in medium (Sound Waves), or wave speed (Transverse Waves, Longitudinal Waves)
x	Position
y	Transverse displacement at x and t
ym	Transverse amplitude

5.15.1 Transverse Waves

$$y = ym \cdot \sin(k \cdot x - \omega \cdot t) \qquad v = \lambda \cdot f \qquad k = \frac{2\pi}{\lambda} \qquad \omega = 2\pi \cdot f$$

5.15.2 Longitudinal Waves

$$s = sm \cdot \cos(k \cdot x - \omega \cdot t) \qquad v = \lambda \cdot f \qquad k = \frac{2\pi}{\lambda} \qquad \omega = 2\pi \cdot f$$

5.15.3 Sound Waves

$$v = \sqrt{\beta}\rho \qquad I = \frac{1}{2} \cdot \rho \cdot v \cdot \omega^2 \cdot sm^2 \qquad \beta = 10 \cdot \log\left(\frac{I}{10}\right) \qquad \omega = 2\pi \cdot f$$

5.16 References

Hewlett-Packard. ***HP 48G Advanced User's Reference Manual, 4th Ed.***, 1994.

6 Newton's Method

Newton's method is essentially an application of tangent lines. The equation of a line tangent to a single-variable function $f(x)$ at $x = x_0$ is given by

$$y = f'(x_0) \cdot (x - x_0) + f(x_0). \tag{3}$$

This comes from the point-slope form of a line: $y - y_0 = m \cdot (x - x_0)$, where $y_0 = f(x_0)$, and $m = f'(x_0)$. Newton's method approximates the zero of $f(x)$ as the horizontal intercept of equation (3).

$$x = x_0 - \frac{f(x_0)}{f'(x_0)} \tag{4}$$

Upon converting equation (4) to iterative form, we have

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

A similar derivation works equally well for multivariable functions. For example, if we have a two-variable function $f(x, y)$, with partial derivatives $f_x(x, y)$ and $f_y(x, y)$, then linear approximation of $f(x, y)$ is essentially the equation of its tangent plane at some specific location $x = x_0$, $y = y_0$:

$$z = f(x_0, y_0) + \underbrace{f_x(x_0, y_0) \cdot (x - x_0) + f_y(x_0, y_0) \cdot (y - y_0)}_{\nabla f \cdot \Delta \mathbf{x}}.$$

Observe that part of the right hand side can be interpreted as the (matrix) product $\nabla f \cdot \Delta\mathbf{x}$, where

$$\nabla f = \begin{bmatrix} f_x(x_0, y_0) & f_y(x_0, y_0) \end{bmatrix} \qquad \text{and} \qquad \Delta\mathbf{x} = \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix}$$

So finding an approximation of a zero of $f(x, y)$, with an initial guess of $x = x_0$ and $y = y_0$, amounts to solving

$$\nabla f \cdot \Delta\mathbf{x} = -f(x_0, y_0) \tag{5}$$

for $\Delta\mathbf{x}$ and then computing

$$\underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} x_0 \\ y_0 \end{bmatrix}}_{\mathbf{x}_0} + \underbrace{\begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix}}_{\Delta\mathbf{x}}$$

Solving a system of m equations in n unknowns is equivalent to finding the zeros of m functions in the n variables $x_1, x_2, \cdots x_n$ as shown below.

$$\begin{aligned} f_1(x_1, x_2 \cdots x_n) &= 0 \\ f_2(x_1, x_2 \cdots x_n) &= 0 \\ &\vdots \\ f_m(x_1, x_2 \cdots x_n) &= 0 \end{aligned}$$

Let $\mathbf{x} = [x_1, x_2, \cdots, x_n]^T$. Using Newton's method with an initial guess of $\mathbf{x} = \mathbf{x}_0$, we would have to solve m equations similar to (5), with each equation corresponding to a linear approximation of f_i at the point $\mathbf{x}_0$.

$$\begin{aligned} \nabla f_1 \cdot \Delta\mathbf{x} &= -f_1(\mathbf{x}_0) \\ \nabla f_2 \cdot \Delta\mathbf{x} &= -f_2(\mathbf{x}_0) \\ &\vdots \\ \nabla f_m \cdot \Delta\mathbf{x} &= -f_m(\mathbf{x}_0) \end{aligned}.$$

The system above is equivalent to the matrix equation

$$\mathbf{J} \cdot \Delta\mathbf{x} = -\mathbf{F}_0$$

where $\mathbf{F}_0 = \mathbf{F}(\mathbf{x}_0)$,

$$\mathbf{J} = \begin{bmatrix} (f_1)_{x_1} & (f_1)_{x_2} & \cdots & (f_1)_{x_n} \\ (f_2)_{x_1} & (f_2)_{x_2} & \cdots & (f_2)_{x_n} \\ \vdots & \vdots & \ddots & \vdots \\ (f_m)_{x_1} & (f_m)_{x_2} & \cdots & (f_m)_{x_n} \end{bmatrix} \quad \text{and} \quad \mathbf{F}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{bmatrix}.$$

The matrix $\mathbf{J}$ is called the Jacobian (and is evaluated at $\mathbf{x_0}$). The (i, j)-th entry of $\mathbf{J}$ is the partial derivative of f_i with respect variable x_j (and the partial derivative is evaluated at $\mathbf{x} = \mathbf{x_0}$). Again, we solve for $\Delta\mathbf{x}$.

$$\Delta\mathbf{x} = -\mathbf{J}^{-1} \cdot \mathbf{F}_0 \tag{6}$$

We then obtain an approximation for the solution to our system of equations using $\mathbf{x} = \mathbf{x}_0 + \Delta\mathbf{x}$, and iterate on formula (6) to refine our solution. While Newton's method can be used to solve a system of equations, whether a solution will be found depends on the initial guess. A poor initial guess may very well lead to a worse estimation of a solution.

6.1 Minimization

If $\mathbf{x}$ is a solution to $\mathbf{F}(\mathbf{x}) = \mathbf{0}$, then $\mathbf{x}$ is also a zero of the function

$$g(\mathbf{x}) = \frac{1}{2}\mathbf{F}(\mathbf{x}).\mathbf{F}(\mathbf{x}) = \frac{1}{2}[(f_1(\mathbf{x}))^2 + (f_2(\mathbf{x}))^2 + \cdots + (f_m(\mathbf{x}))^2].$$

(Here, the operation $\mathbf{u}.\mathbf{v}$ denotes the dot product of $\mathbf{u}$ and $\mathbf{v}$.) Note that g is non-negative, so solutions to $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ are minimums of g. (On the other hand, the minimums of g may not necessarily correspond to solutions of $\mathbf{F}(\mathbf{x}) = \mathbf{0}$.) Another important observation is that $\nabla g = \mathbf{F}^T \cdot \mathbf{J}$, which implies

$$\nabla g \cdot \Delta\mathbf{x} = (\mathbf{F}^T \cdot \mathbf{J}) \cdot (-\mathbf{J}^{-1} \cdot \mathbf{F}) = -\mathbf{F}^T \cdot \mathbf{F} = -\mathbf{F}.\mathbf{F} = -2g < 0$$

So the Newton step $\Delta\mathbf{x}$ is a descent direction toward a minimum of g. If the minimum is also a zero of g, then it is also a solution to $\mathbf{F}(\mathbf{x}) = \mathbf{0}$. In practice, we can simply check whether the norm of $\mathbf{F}$ is sufficiently small. Otherwise, we try again with a different initial guess.

6.2 Line Search

Note that while $\Delta\mathbf{x}$ is a descent direction, the full step $\Delta\mathbf{x}$ may be "too far" and actually result in $g(\mathbf{x}+\Delta\mathbf{x}) > g(\mathbf{x})$. When this happens, we can simply scale back the Newton step by a factor of λ so that $g(\mathbf{x}+\lambda\Delta\mathbf{x}) < g(\mathbf{x})$. Such a λ is guaranteed by the fact that $\Delta\mathbf{x}$ is a descent direction. The scaling factor λ can be determined by a line search.

Suppose that $g(\mathbf{x} + \Delta\mathbf{x}) > g(\mathbf{x}) > 0$ and let $h(\lambda) = g(\mathbf{x} + \lambda\Delta\mathbf{x})$. Then h must have a minimum in the interval $(0, 1)$ since h is non-negative and $\Delta\mathbf{x}$ is a descent direction from the point $\mathbf{x}$. We can approximate $h(\lambda)$ as a quadratic function

$$h(\lambda) \approx [h(1) - h(0) - h'(0)]\lambda^2 + h'(0)\lambda + h(0)$$

where $h(0) = g(\mathbf{x})$, $h(1) = g(\mathbf{x} + \Delta\mathbf{x})$, and

$$h'(0) = \nabla g \cdot \Delta\mathbf{x} = -2g(\mathbf{x}) = -2h(0).$$

The quadratic approximation of h has a minimum when

$$\underbrace{2[h(1) - h(0) - h'(0)]\lambda + h'(0)}_{h'(\lambda)} = 0.$$

The equation above has the solution

$$\lambda_q = \frac{h'(0)}{2[h(1) - h(0) - h'(0)]}. \tag{7}$$

We then test to see if $h(\lambda_q) = g(\mathbf{x} + \lambda_q \mathbf{x})$ is "sufficiently less than" $g(\mathbf{x})$. If not, then we approximate h by a cubic function using this additional evaluation of $g(\mathbf{x} + \lambda_q \Delta\mathbf{x})$.

Remark. In practice we compute (7) using $h'(0) = -2h(0)$ so that

$$\lambda_q = \frac{-2h(0)}{2[h(1) - h(0) + 2h(0)]} = \frac{h(0)}{h(1) + h(0)}$$

Should the quadratic approximation fail to yield a suitable λ, we can proceed with a cubic approximation of h using the formula

$$h(\lambda) = a\lambda^3 + b\lambda^2 + h'(0)\lambda + h(0).$$

At this point, we have already evaluated $h(1)$ and $h(\lambda_q)$, which we use to solve the linear system

$$a + b + h'(0) + h(0) = h(1)$$
$$a\lambda_q^3 + b\lambda_q^2 + h'(0)\lambda_q + h(0) = h(\lambda_q)$$

for a and b. The cubic approximation of h has critical points when $h'(\lambda) = 0$. The solutions to

$$3a\lambda^2 + 2b\lambda + h'(0) = 0$$

are

$$\lambda_c = \frac{-2b \pm \sqrt{(2b)^2 - 4(3a)h'(0)}}{2(3a)} = \frac{-b \pm \sqrt{b^2 + 6h(0)}}{3a}.$$

6.3 Computing $\mathbf{J}^{-1}$

In equation (6), we assumed that $\mathbf{J}$ was invertible. It is quite possible for $\mathbf{J}$ to be non-square (in the case of over determined or under determined systems). And even when $\mathbf{J}$ is a square matrix, it may very well be singular (or nearly singular) at some particular iteration of Newton's method. There are several approaches to solving the equation $\mathbf{J} \cdot \Delta\mathbf{x} = -\mathbf{F}$, though the most robust (and perhaps the slowest) method is to compute the pseudo-inverse $\mathbf{J}^+$ of $\mathbf{J}$ using the singular value decomposition (SVD) of $\mathbf{J}$.

When $\mathbf{J}$ is square and non-singular, the pseudo-inverse $\mathbf{J}^+$ is exactly the inverse of $\mathbf{J}$. That is, $\mathbf{J}^+ = \mathbf{J}^{-1}$, and

$$\Delta\mathbf{x} = -\mathbf{J}^+ \cdot \mathbf{F} = -\mathbf{J}^{-1} \cdot \mathbf{F}.$$

If $\mathbf{J}$ is non-square but still has full rank, then

$$\mathbf{J}^+ = (\mathbf{J}^* \cdot \mathbf{J})^{-1} \cdot \mathbf{J}^*$$

where $\mathbf{J}^*$ is the Hermitian transpose (conjugate transpose). (If $\mathbf{J}$ has only real-valued entries, then $\mathbf{J}^* = \mathbf{J}^T$.) Moreover, in this case, $\mathbf{J}^+$ constitutes a left inverse since

$$\mathbf{J}^+ \cdot \mathbf{J} = ((\mathbf{J}^* \cdot \mathbf{J})^{-1} \cdot \mathbf{J}^*) \cdot \mathbf{J} = (\mathbf{J}^* \cdot \mathbf{J})^{-1}(\mathbf{J}^* \cdot \mathbf{J}) = \mathbf{I}.$$

Solve for $\Delta\mathbf{x}$ in $\mathbf{J} \cdot \Delta\mathbf{x} = -\mathbf{F}$ to obtain $\Delta\mathbf{x} = -\mathbf{J}^+ \cdot \mathbf{F}$. Lastly, if $\mathbf{J}$ is singular, then $\mathbf{J}^+$ is a generalized inverse of $\mathbf{J}$ and has the property $\mathbf{J} \cdot \mathbf{J}^+ \cdot \mathbf{J} = \mathbf{J}$. Notice that

$$\mathbf{J} \cdot \Delta\mathbf{x} = (\mathbf{J} \cdot \mathbf{J}^+ \cdot \mathbf{J}) \cdot \Delta\mathbf{x} = \mathbf{J} \cdot \mathbf{J}^+ \cdot (\mathbf{J} \cdot \Delta\mathbf{x}) = \mathbf{J} \cdot \mathbf{J}^+ \cdot (-\mathbf{F})$$

So even in the singular case, we can obtain $\Delta\mathbf{x}$ since $\mathbf{J} \cdot \Delta\mathbf{x} = \mathbf{J} \cdot \mathbf{J}^+ \cdot (-\mathbf{F})$ suggests that $\Delta\mathbf{x} = -\mathbf{J}^+ \cdot \mathbf{F}$.

6.4 Points of Failure

Despite the robustness of the pseudo-inverse, and the fact that the Newton step $\Delta\mathbf{x}$ is a descent step, the algorithm can fail in several places.

1. Machine precision (or the lack thereof) could produce a λ value (in the line search) so small that its effect would not propagate significantly into the computation of $\mathbf{F}$ and its norm.
2. A combination of precision, bad guess, and singular functions (i.e. the f_i's in $\mathbf{F}$ might be discontinuous) could lead to an evaluation error in $\mathbf{F}$.
3. The Jacobian $\mathbf{J}$ could similarly have points of discontinuity.
4. The system is inconsistent (no solution). Though in this case, Newton's method can still produce a "least squares" solution (i.e. a solution for which the norm of $\mathbf{F}$ is locally minimal).

7 Appendix

7.1 Library File Format

The default library file is saved in `Equation Library.lib`. It consists of a list of lists

```
{
  system1,
  system2,
  ...
  systemK
}
```

where `system1` through `systemK` are themselves lists. Below is a generic example of one such system (of M equations and N variables/constants) and the descriptions of its components.

```
{
  "System Name",
  {
    "equation1",
    "equation2",
    ...
    "equationM"
  },
  { eqchk1,eqchk2,...,eqchkM },
  { "var1", "var2", ..., "varN" },
  { "desc1", "desc2", ..., "descN" },
  { val1,val2,...,valN },
  { const1,const2,...,constN },
  "Category",
  "Image Index"
}
```

`"System Name"` — This is the name or title of the system of equations.

`"equation`i`"` — Each equation is stored as a string. For example, the equation $F = m \cdot a$ for force as a product of mass and acceleration is stored as `"F=m*a"`.

`eqchk`i — While a system of equations may have M equations, the user may choose to use only a subset of these equations. The user may select the subset using the checkboxes in Symb Setup view. Upon saving the system, the selection list is saved into a list of `eqchk`i's. These values are either 0 or 1 and correspond to whether the equation was selected (set to 1) in the most recently saved solving session.

`"var`j`"` — Each variable is listed as a string. For example, the variables F, m, and a in the equation $F = m \cdot a$ are stored as `"F"`, `"m"`, and `"a"` respectively.

`"desc`j`"` — This is the description of the j-th variable. These descriptions are used in two places: the help prompt in the Num Setup view and the list of variables shown by the `View Variables` option in the View Copy menu. Some variables are actually constants. For example, NA is the conventional constant for Avogadro's number. The `"var`j`"` string would be `"NA"`. The description for `"NA"` is `"(Avagadro's Number)"`—note the parentheses. The use of `()`'s provides a visual cue to users that this variable should be treated as a fixed constant.

`val`j — This is the most recently saved value of the corresponding variable `var`j.

`const`j — This value is normally 0 or 1, and determines whether the corresponding variable `var`j should be treated as a constant (set to 1). Actual constants should always have this value set to 1. Variables whose values are considered as "known" during the solving process will also have this set to 1 so that the solve engine does not try to solve for that particular variable. Variables that have been abandoned (no longer used in the system) will have their `const`j value increment up until the `Non-use Limit` value, upon which these variables (and their corresponding `"desc`j`"`, `val`j, and `const`j) are removed from the system definition.

`"Category"` — This string determines the category to which the system belongs.

`"Image Index"` — All diagrams associated with a system of equations are stored as PNG files. These files are app files with the name `imgNN.png` where `NN` denotes the image index. The unmodified equation library uses 63 images stored in the files `img01.png` through `img63.png`. The image index is a string containing the characters between the letters `img` and file extension `.png` in the file name (`"01"` through `"63"`). The desired convention is to retain this file name scheme. Of course, a user's equation library may grow to use more than 100 images (in which case the image index string would be three characters wide). That said, users may use whatever naming indexing scheme they wish (i.e. `NN` may be anything that is allowed in an app file name).

For animations, `"Image Index"` may be replaced by a list whose content are strings of image indices followed by a wait value. For example, the `Mass-Spring System` uses the list

```
{ "40", "41", "42", "41", .1 }
```

instead of a single image index string. The value `.1` refers to the delay between each frame of animation. The frames are looped until a key is pressed or the screen detects a touch event.

7.2 LibToNote

The following program may be used to convert an existing library file (named `Equation Library.lib`) into a `Note` that can be then copied (from the emulator) into a text file for manual editing.

```
EXPORT LibToNote()
BEGIN
  local note:="EqLib.txt";
  local lib:=AFiles("Equation Library.lib");
  local n:=size(lib);
  local sys, m, j, k, r;

  Notes(note):="{\n";
  for j from 1 to n do
      Notes(note):=Notes(note) + "  {\n";
      sys:=lib(j);
      for k from 1 to 9 do
          Notes(note):=Notes(note) + "    ";
          if (k == 2) or (k == 5) then
              Notes(note):=Notes(note) + "{\n";
              for r from 1 to (size(sys(k))-1) do
                  Notes(note):=Notes(note) + "      ";
                  Notes(note):=Notes(note) + string(sys(k,r)) + ",\n";
              end;
              Notes(note):=Notes(note) + "      ";
              Notes(note):=Notes(note) + string(sys(k,r)) + "\n    }";
          else
              Notes(note):=Notes(note) + string(sys(k));
          end;
          if (k < 9) then Notes(note):=Notes(note) + ","; end;
          Notes(note):=Notes(note) + "\n";
      end;
      Notes(note):=Notes(note) + "  }";
      if (j < n) then Notes(note):=Notes(note) + ",\n"; end;
      Notes(note):=Notes(note) + "\n";
  end;
  Notes(note):=Notes(note) + "}";
END;
```

7.3 Creating Diagrams

As of version 0.900, diagrams may be associated with a system of equations. Systems may be saved with related diagrams starting with version 1.000 onward. The `Equation Library` does not provide any mechanism for creating diagrams. Users may either create the diagrams on a computer and use the connectivity kit software to import the images into the app, or use the built-in `Geometry` app to create a diagram. Instructions on how to use the `Geometry` app is beyond the scope of this reference manual. However, a general workflow is outlined below.

1. Create a diagram using the `Geometry` app.

2. While the `Geometry` app is active, take a "snapshot" of the diagram using the following code snippet:

   ```
   DIMGROB_P(G1,320,240); STARTVIEW(1,1); BLIT_P(G1,G0);
   ```

   The code above initializes the graphic object `G1` to a 320×240 image, displays the Plot Setup view, and then copies the current screen `G0` into the graphic object `G1`.

3. Switch over to the `Equation Library` app and save the diagram using:

   ```
   AFiles("img64.png"):=G1;
   ```

   Change the file name from `"img64.png"` to any appropriately indexed name. Make sure to use the format `imgXYZ.png` for the file name, where `XYZ` may be any alpha-numeric sequence of characters that are allowed in an app file name.

4. When saving a system of equations, select `XYZ` as the image index.

Remark. Make sure that all images are no larger than 320 pixels in width and 240 pixels in height. Images whose width and height are simultaneously less than 160 and 120 pixels, respectively, will be scaled by a factor of 2 to fill the screen.