```
//--------------------------------------------------------------------------

//
//
//--------------------------------------------------------------------------


//==========================================================================

// Global variable definitions for flute calculations
// inputs
//==========================================================================

//
//  Initialization stuff
//
//==========================================================================


var isound = 13584.0;
var msound = isound * 25.4;

// Create a default whistle.  Note that the form takes the values from this by
// default.  If you want to change the default values, then change them here.
var num_holes = 6;      // number of holes in the instrument
var standard = true;    // metric or inch, default is inch
var vsound = isound;   // inches
var usefracs = true;    // show fractions if true and if standard is true
var base_freq = 587.33; // base frequency
var embouchure = .375;  // diameter of the embouchure
var bore_dia = .5;      // inside diameter of the tube
var wall = .015;        // wall thickness
var xend = 0.0;
var xemb = 0.0;         // location of the embouchure
var selected = 6;
var calcuate_func = FindLocations2;
var title = "My Whistle";

var max_holes = 12;
var hole_dias = new Array(max_holes);
var hole_locs = new Array(max_holes);
var frequencies = new Array(max_holes);
var diffs = new Array(max_holes);
var intervals = new Array(max_holes);
var notes = new Array(max_holes);
var cutoff_freqs = new Array(max_holes);

var saved_whistle_slots = 0;
var saved_whistles = new Array();
var saved_whistle_names = new Array();

// these are constants
var bell_notes = [  "G#", "G", "F#", "F", "E", "D#", "D",
                    "C#", "C", "B", "A#", "A", "G#", "G",
                    "F#", "F", "E", "D#", "D", "C#", "C",
                    "B", "A#", "A"];
var bell_freqs = [830.61, 783.99, 739.99, 698.46, 659.26, 622.25, 587.33,
                  554.37, 523.25, 493.88, 466.16, 440.0,  415.3,  392.0,
                  369.99, 349.23, 329.63, 311.13, 293.66, 277.18, 261.63,
```

```
                246.94, 233.08, 220.0 ];
var note_names = ["A", "A#", "B", "C", "C#", "D", "D#", "E", "F", "F#", "G", "G#
"];
var note_map = [11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, 11, 10, 9, 8, 7, 6, 5, 4,
3, 2, 1, 0];

// system defaults that will be the same on an initial load, but not on a recalc
.
var intervals = [0,2,2,1,2,2,2];
var hole_dias = [0.0, 1/4, (3/8)-(1/16), (1/4)-(1/16), 1/4, 1/4, (1/4)-(1/32)];

// these values should be cleared upon reload
for(var i = 0; i < num_holes+1; i++) {
    diffs[i]= 0;
    hole_locs[i] = 0;
    cutoff_freqs[i] = 0;
}


//============================================================================

//
//  Whistle calculations
//
//============================================================================


// effective wall thickness, i.e. height of air column at open finger holes
// air column extends out past end of hole 3/4 of the hole diameter
function eff_wall(n)
{
    return (1.0 * wall) + (0.75 * hole_dias[n]);
}

// Closed hole for tone hole n.  The length of the vibrating air column is
// effectively increased by each closed tone hole which exists above the
// first open tone hole. Corrections must be added for each such closed tone
// tone hole to C_end, C_s, and C_o.
function closed_correction(n)
{
    return 0.25 * wall * Math.pow(hole_dias[n] / bore_dia, 2);
}

// Calculates the distance from physical open end of flute to effective
// end of vibrating air column.  The vibrating air column ends beyond the
// end of the flute and C_end is always positive. NOTE: Closed hole
// corrections must be added to this value!
function end_correction()
{
    return 0.6133 * bore_dia / 2;
}

// Calculates the effective distance from the first ("single") tone hole to
// the end of the vibrating air column when only that hole is open.
// NOTE: closed hole corrections must be added to this value!
function first_correction()
{
    return  eff_wall(1) /
            (Math.pow(hole_dias[1] / bore_dia, 2) +
             eff_wall(1) / (xend - hole_locs[1]) );
```

```
}

// Calculates the effective distance from the second and subsequent tone holes
// to the end of the vibrating air column when all holes below are open.
// NOTE: closed hole corrections must be added to this value!
// NOTE: the value of this correction is invalid if the frequency of the note
// played is above the cutoff frequency f_c.
function open_correction(n)
{
    return ((hole_locs[n-1] - hole_locs[n]) / 2) *
            (Math.sqrt(1.0 + 4.0 * (eff_wall(n) / (hole_locs[n-1] - hole_locs[n]
)) *
                        Math.pow(bore_dia / hole_dias[n], 2)) - 1);
}

// C_emb = distance from theoretical start of air column to center of embouchure
 hole;
// the air column effectively extends beyond the blow hole center by this distan
ce.
// (the cork face should be about 1 to 1.5 embouchure diameters from emb. center
)
//C_emb := (Bore/Demb)*(Bore/Demb)*(wall+0.75*Demb); // per spreadsheet
//C_emb := (Bore/Demb)*(Bore/Demb)*(Bore/2 + wall + 0.6133*Demb/2); // an altern
ative
//C_emb := (Bore/Demb)*(Bore/Demb)*10.84*wall*Demb/(Bore + 2*wall); // kosel's e
mpirical fit
function emb_correction()
{
    return Math.pow(bore_dia / embouchure, 2) *
                10.84 *
                wall *
                embouchure /
                (1.0 * bore_dia + 2.0 * wall);
}

// Calculates the cutoff frequency above which the open hole correction
// is not valid.  Instrument should be designed so that all second register
// notes are well below this frequency.
function cutoff(n)
{
    var s;
    if(n == 1)
        s = xend - hole_locs[n];
    else
        s = hole_locs[n-1] - hole_locs[n];

    var bacon = vsound / 2.0 / Math.PI * hole_dias[n] / bore_dia / Math.sqrt(eff
_wall(n) * s);
    //alert("cutoff = "+bacon);
    return bacon;
    //return vsound /
    //        (2.0 * Math.PI) * (hole_dias[n] / bore_dia) * 1 /
    //        Math.sqrt(eff_wall(n) * (hole_locs[n-1] - hole_locs[n]));
}

// This procedure finds the locations of end of flute, all finger holes, and
// emb. hole using the Benade equations above in an interative manner.
function FindLocations()
{
    var i, X, oldX, holeNum;
```

```
    // find end location...
    xend = vsound * 0.5 / base_freq;  // uncorrected location
    xend = xend - end_correction();  // subtract end correction
    for(i = 1; i <= num_holes; i++)
        xend = xend - closed_correction(i); // subtract closed hole corrections

    // find first finger hole location
    X = vsound * 0.5 / frequencies[1];
    hole_locs[1] = 0;
    do {
        oldX = hole_locs[1];
        hole_locs[1] = X - first_correction();
        for(i = 2; i <= num_holes; i++)
            hole_locs[1] = hole_locs[1] - closed_correction(i);
    } while(!Math.abs(hole_locs[1] - oldX) < 0.0001);

    // set subsequent finger hole locations
    if(num_holes >= 2)
    for(holeNum = 2; holeNum <= num_holes; holeNum++) {
    X = vsound * 0.5 / frequencies[holeNum];
    hole_locs[holeNum] = 0;
    do {
        oldX = hole_locs[holeNum];
        hole_locs[holeNum] = X - open_correction(holeNum);
        if(holeNum < num_holes)
                    for(i = holeNum + 1; i <= num_holes; i++)
                        hole_locs[holeNum] = hole_locs[holeNum] - closed_correct
ion(i);
    } while(!Math.abs(hole_locs[holeNum] - oldX) < 0.0001);
    }

    // set embouchure hole location
    xemb = emb_correction();
    for(var i = 1; i <= num_holes; i++)
        cutoff_freqs[i] = cutoff(i);
}

// This is a non-iterative procedure equivalent to the above procedure.  It invo
lves use
// of quadratic solutions of the Benade equations obtained by "simple but tediou
s algebraic
// manipulation".
function FindLocations2()
{
    var i;
    var L;
    var holeNum;
    var a,b,c;


    // find end location...
    xend = vsound * 0.5 / base_freq;  // uncorrected location
    xend = xend - end_correction();  // subtract end correction
    for(i = 1; i <= num_holes; i++)
        xend = xend - closed_correction(i);  // subtract closed hole corrections

    //alert("Xend="+Xend)

    // find first finger hole location
```

```
    L = vsound * 0.5 / frequencies[1];
    for(i = 2; i <= num_holes; i++)
        L = L - closed_correction(i);  // subtract closed hole corrections
    a = (hole_dias[1]/bore_dia)*(hole_dias[1]/bore_dia);
    b = -(xend + L)*(hole_dias[1]/bore_dia)*(hole_dias[1]/bore_dia);
    c = xend * L * (hole_dias[1]/bore_dia)*(hole_dias[1]/bore_dia) + eff_wall(1)
*(L-xend);
    //alert("eff_wall(1)="+eff_wall(1))
    //alert("hole_dias[1]="+hole_dias[1]+"frequencies[1]="+frequencies[1]+" a="+
a+" b="+b+" c="+c+" L="+L)
    hole_locs[1] = ( -b - Math.sqrt((b*b) - 4*a*c) ) / (2*a);

    // find subsequent finger hole locations
    if(num_holes >= 2)
    for(holeNum=2;holeNum<=num_holes;holeNum++)
        {
        L = vsound * 0.5 / frequencies[holeNum];
        if (holeNum < num_holes)for(i=holeNum;i<=num_holes;i++) L = L - closed_c
orrection(i);
        a = 2;
        b = - hole_locs[holeNum-1] - 3*L + eff_wall(holeNum)*(bore_dia/hole_dias
[holeNum])*(bore_dia/hole_dias[holeNum]);
        c = hole_locs[holeNum-1]*(L - eff_wall(holeNum)*(bore_dia/hole_dias[hole
Num])*(bore_dia/hole_dias[holeNum])) + (L*L);
        hole_locs[holeNum] = ( -b - Math.sqrt((b*b) - 4*a*c) ) / (2*a);
        }

    // set embouchure hole location
    xemb = emb_correction();
    for(var i = 1; i <= num_holes; i++)
        cutoff_freqs[i] = cutoff(i);
}

//================================================================================

//
//  User interface
//
//================================================================================


//--------------------------------------------------------------------------------

// Convert a number into a fraction that has been reduced to the minimum
// denominator.  For example, if you pass the number 0.5 the return string will
// be 1/2.
//--------------------------------------------------------------------------------

function reduce(num)
{
    var w;
    var f;
    var i;

    w = Math.round(num/(1.0/64.0));
    f = 64;

    for(i = 0; (w % 2) == 0; )
    {
        w = w / 2;
```

```
        f = f / 2;

        if(w <= 0 || f < 0 )
            break;
    }

    return w+"/"+f;
}

//-----------------------------------------------------------------------------

//  Convert a fraction to a double
//-----------------------------------------------------------------------------

function fractod(str)
{
    var lst = str.split("/");
    if(lst.length != 2) {
        alert("Invalid fraction: "+str+"\nPress OK to continue");
        return str;
    }
    return parseFloat(lst[0]) / parseFloat(lst[1]);
}


//-----------------------------------------------------------------------------

// Round off a floating point number to the number of decimal places specified.
// For example, if you want to round 1.12345 off to 3 places, then call it with
// round(num, 3) and the value returned will be 1.123.
//-----------------------------------------------------------------------------

function round(num, places)
{
    var fact = 1;
    for(var i = 0; i < places; i++)
        fact *= 10;
    return Math.round(fact * num) / fact;
}

//-----------------------------------------------------------------------------

// round off a metric number to the nearest 0.5
//-----------------------------------------------------------------------------

function round_metric(num)
{
    var whole = parseInt(num);
    var frac = num - whole;
    //alert("num = "+num+"  whole = "+whole+"  frac = "+frac);
    if(frac < 0.25)
        return whole;
    else if(frac <= 0.5)
        return whole + 0.5;
    else
        return whole + 1;
}


//-----------------------------------------------------------------------------
```

```
// Set up the frequencies
//--------------------------------------------------------------------------

function keyChange(sel)
{
    //alert("key change event called with "+sel);
    base_freq = bell_freqs[sel];
    var ctrl = document.getElementById("freqEnd");
    ctrl.value = base_freq;
    frequencies[0] = base_freq;
    selected = sel;

    var note = note_map[sel];
    var mult = 1.05946309436

    for(var i = 0; i < num_holes; i++) {

        //alert("intervals["+(i+1)+"] = "+intervals[i+1]+"\nnote = "+note);
        note += parseInt(intervals[i+1]);
        if(note > (note_names.length-1)) {
            note -= parseInt(note_names.length);
        }

        var strg = "note"+(i+1);
        ctrl = document.getElementById(strg);
        ctrl.value = note_names[note];

        frequencies[i+1] = frequencies[i] * mult;
        if(intervals[i+1] > 1)
            frequencies[i+1] = frequencies[i+1] * mult;
        if(intervals[i+1] > 2)
            frequencies[i+1] = frequencies[i+1] * mult;
        if(intervals[i+1] > 3)
            frequencies[i+1] = frequencies[i+1] * mult;
        if(intervals[i+1] > 4)
            frequencies[i+1] = frequencies[i+1] * mult;
        frequencies[i+1] = round(frequencies[i+1], 2);

        strg = "freq"+(i+1);
        ctrl = document.getElementById(strg);
        ctrl.value = frequencies[i+1];
    }

    calcuate_func();
    put_form();
}

//--------------------------------------------------------------------------

// This function does the gruntwork of getting input, calling the calculation
// routine, and delivering the results.
//--------------------------------------------------------------------------

function RefreshAll()
{
    //alert("calculate button clicked");
    get_form();
    calcuate_func();
    put_form();
```

```
}

var newWindow;
//---------------------------------------------------------------------------

//---------------------------------------------------------------------------

function doPrint()
{
    //alert("Print whistle");
    if (!newWindow || newWindow.closed) {
        newWindow = window.open("","","status,height=380,width=640")
        if (!newWindow.opener) {
        newWindow.opener = window
        }
        // force small delay for IE to catch up
        setTimeout("show_print_window()", 50)
    } else {
        // window's already open; bring to front
        newWindow.focus()
    }
}

//---------------------------------------------------------------------------

//---------------------------------------------------------------------------

function show_print_window() {
    // assemble content for new window
    var newContent = "<HTML><HEAD><TITLE>Print Whistle</TITLE></HEAD>";
    newContent += "<BODY><H2>"+document.getElementById("title").value+"</H2>";
    newContent += "Bell note: "+bell_notes[selected]+"  "+bell_freqs[s
elected]+"Hz<br/>";
    newContent += "Bore: "+bore_dia+"  Wall: "+wall+"<br/>";
    newContent += "<hr/><br/>";
    newContent += "<table border=\"0\">";
    newContent += "<tr style=\"background-color:#f0f0f0\">";
    newContent += "<th style=\"width:100px;\"> </th>";
    newContent += "<th style=\"width:100px;\" align=\"left\"> <b>Note</b>&n
bsp;</th>";
    newContent += "<th style=\"width:100px;\" align=\"left\"> <b>Hole Dia</
b> </th>";
    newContent += "<th style=\"width:100px;\" align=\"left\"> <b>Location</
b> </th>";
    newContent += "</tr>";

    for(var i = 0; i < num_holes; i++) {
        newContent += "<tr>";
        newContent += "<td> Hole "+(i+1)+"</td>";
        newContent += "<td> "+document.getElementById("note"+(i+1)).value+"
</td>";
        newContent += "<td> "+document.getElementById("diam"+(i+1)).value+"
</td>";
        newContent += "<td> "+document.getElementById("location"+(i+1)).val
ue+"</td>";
        newContent += "</tr>";
    }

    newContent += "</table><hr/>";
    newContent += "<table border=\"0\">";
```

```
    newContent += "<tr><td><input type=\"button\" value=\"Close\" onClick=\"wind
ow.close()\"></td>";
    newContent += "<td><input type=\"button\" value=\"Print\" onClick=\"window.p
rint()\"></td></tr>";
    newContent += "</table><hr/>";
    newContent += "</BODY></HTML>";
    // write HTML to new window document
    newWindow.document.write(newContent);
    newWindow.document.close(); // close layout stream
}

//-----------------------------------------------------------------------------

// Cookies are saved as one long string with a common name.  The index of each
// record in a cookie is an index into the string.  This is similar to a heap
// data strucure.  This string is copied to a local string when the page is
// loaded and manipulate there.
//-----------------------------------------------------------------------------

var cookie_name = "whistle_calc";
//-----------------------------------------------------------------------------

// generic get cookie
//-----------------------------------------------------------------------------

function get_cookie(name)
{
    var c_start, c_end;
    if(document.cookie.length > 0) {
        c_start=document.cookie.indexOf(name + "=");
        if (c_start!=-1) {
            c_start=c_start + name.length+1;
            c_end=document.cookie.indexOf(";",c_start);
            if (c_end==-1)
                c_end=document.cookie.length;
            return unescape(document.cookie.substring(c_start,c_end));
        }
    }
    return "";
}

//-----------------------------------------------------------------------------

// generic save cookie
//-----------------------------------------------------------------------------

function save_cookie(name, value, days)
{
    var exdate=new Date();
    exdate.setDate(exdate.getDate()+days);
    document.cookie = "";
    document.cookie = name+"="+escape(value)+
                ((days==null) ? "" : ";expires="+exdate.toGMTString());
    //alert("cookie \""+name+"\" saved\n"+document.cookie);
}

//-----------------------------------------------------------------------------

// generic delete cookie
//-----------------------------------------------------------------------------
```

```
function del_cookie(name)
{
    var exp = new Date();
    exp.setTime(exp.getTime () - 1);
    var cval = getCookie (name);
    document.cookie = name + "=" + cval + "; expires =" + exp.toGMTString();
    //alert("cookie \""+name+"\" deleted");
}

//---------------------------------------------------------------------------

// load all of the whistles from the cookie and put them into the saved_whistles

// array. called when the page is loaded and when a load, save, or delete event
// takes place.
//---------------------------------------------------------------------------

function load_whistles()
{
    //alert("load_whistles() called");
    var str = get_cookie(cookie_name);
    var len = str.length;
    if(len > 0) {
        var idx = 0;
        var start = 0;
        do {
            var end = str.indexOf("end", start) + 3;
            saved_whistles[idx] = str.substring(start, end);
            saved_whistle_names[idx] = saved_whistles[idx].split(",")[0];
            //alert("idx: "+idx+"  start: "+start+"  end: "+end+"\nname: \""+
            //  saved_whistle_names[idx]+"\"\nloaded whistle:\n"+saved_whistles[
idx]);
            start = end + 1;
            idx++;
        } while(start < len);
    }

    saved_whistle_slots = idx;

    var lctrl = document.getElementById("loadCtrl");
    var dctrl = document.getElementById("delCtrl");

    // update the form
    if(saved_whistles[0] != "") {

        while(lctrl.length > 2) {
            lctrl.remove(2);
        }

        while(dctrl.length > 2) {
            dctrl.remove(2);
        }

        var opt;
        for(var i = 0; i < saved_whistle_slots; i++) {
            if(saved_whistle_names[i] != "") {
                //alert("name: \""+saved_whistle_names[i]+"\"");
                opt = document.createElement("option");
                opt.text = saved_whistle_names[i];
```

```
                lctrl.add(opt, null);
                opt = document.createElement("option");
                opt.text = saved_whistle_names[i];
                dctrl.add(opt, null);
            }
        }
    }
    lctrl.selectedIndex = 0;
    dctrl.selectedIndex = 0;
}

//--------------------------------------------------------------------------

// save all of the saved whistles as a single cookie.  called when a load, save,

// or delete even takes place.  First the cookie is saved and then it is reloade
d.
//--------------------------------------------------------------------------

function save_whistles()
{
    save_cookie(cookie_name, saved_whistles.toString(), 100000);
}


//--------------------------------------------------------------------------

// Read the entire form and place the result into an array, which is then
// converted into a string and placed directly into the cookie.
//--------------------------------------------------------------------------

function make_cookie_string()
{
    // generate the save string
    var idx = 0;
    var arr = new Array();
    arr[idx++] = document.getElementById("title").value;
    arr[idx++] = document.getElementById("numHoles").value;
    arr[idx++] = document.getElementById("calcMethod").selectedIndex;
    arr[idx++] = document.getElementById("bore").value;
    arr[idx++] = document.getElementById("wall").value;
    arr[idx++] = document.getElementById("diamEmb").value;
    arr[idx++] = document.getElementById("keySelect").selectedIndex;
    arr[idx++] = (standard)? "true": "false";
    arr[idx++] = (usefracs)? "true": "false";

    for(var i = 0; i < num_holes; i++) {
        arr[idx++] = document.getElementById("interval"+(i+1)).value;
        if(usefracs == true && standard == true)
            arr[idx++] = fractod(document.getElementById("diam"+(i+1)).value);
        else
            arr[idx++] = document.getElementById("diam"+(i+1)).value;
    }
    arr[idx++] = "end";
    //alert("cookie string:\n"+arr.toString());
    return arr.toString();
}

//--------------------------------------------------------------------------
```

```
// save the form into a cookie with the specified index.  called from the
// onChange event on the save button.
//-----------------------------------------------------------------------------

function doSave()
{
    //alert("doSave() called");
    var cookie = make_cookie_string();
    var name = cookie.split(",")[0];
    // see if this name exists in the list.  If it does, replace it.  Otherwise,

    // add it to the end.
    for(var i = 0; saved_whistles[i] != null; i++) {
        var tmp = saved_whistles[i].split(",")[0];
        if(tmp == name) {
            break;
        }
    }
    saved_whistles[i] = cookie;
    save_whistles();
    load_whistles();
    alert("Whistle \""+name+"\" saved ok");
}


//-----------------------------------------------------------------------------

// load a specific line from the load pull-down into the form. called from the
// onChange event on the load pulldown.
//-----------------------------------------------------------------------------

function doLoad(index)
{
    //alert("doLoad() called");
    if(saved_whistles.length <= 0) {
        alert("No whistle has been saved!");
        load_whistles();
        return;
    }
    index -= 2;
    if(index < 0) {
        load_whistles();
        return;
    }

    if(false == confirm("Load whistle \""+saved_whistle_names[index]+"\"?"))
        return;
    //alert("loading index "+index);
    var cookie_strg = saved_whistles[index];
    var ary = cookie_strg.split(",");

    var idx = 0;
    title = ary[idx++];
    var nh = ary[idx++];
    document.getElementById("calcMethod").selectedIndex = ary[idx++];
    bore_dia = ary[idx++];
    wall = ary[idx++];
    embouchure = ary[idx++];
    selected = parseInt(ary[idx++]);
    standard = (ary[idx++] == "true")? true: false;
```

```
    usefracs = (ary[idx++] == "true")? true: false;

    for(var i = 0; i < nh; i++) { //num_holes; i++) {
        intervals[i+1] = ary[idx++];
        hole_dias[i+1] = ary[idx++];
    }

    if(standard == true)
        vsound = isound;
    else
        vsound = msound;

    load_whistles();
    changeHoles(nh);
    keyChange(selected);
    calcuate_func();
    put_form();
}

//-----------------------------------------------------------------------------

//-----------------------------------------------------------------------------

function showSaved()
{
    var str = "Saved Whistles\n\n";
    for(var i = 0; saved_whistles[i] != null; i++) {
        str += saved_whistles[i]+"\n";
    }
    alert(str);
}

//-----------------------------------------------------------------------------

// delete the specified cookie and reload the controls
//-----------------------------------------------------------------------------

function doDelete(index)
{
    //alert("doDelete() called");
    index -= 2;
    if(index < 0) {
        load_whistles();
        return;
    }

    if(confirm("Delete whistle \""+saved_whistle_names[index]+"\"?") == true)  {

        var arr = new Array();
        saved_whistles[index] = "";
        var j = 0;
        for(var i = 0; saved_whistles[i] != null; i++) {
            if(i != index) {
                arr[j++] = saved_whistles[i];
            }
            saved_whistles[i] = null;
        }

        save_cookie(cookie_name, arr.toString(), 100000);
    }
```

```
    load_whistles();
}

//--------------------------------------------------------------------------------

// change the number of holes in the form.
//--------------------------------------------------------------------------------

function changeHoles(ho)
{
    if(ho > 12) {
        alert("Maximum number of holes is 12");
        ho = 12;
    }
    else if(ho < 1) {
        alert("Minimum number of holes is 1");
        ho = 1;
    }

    //alert("changing number of holes from "+num_holes+" to "+ho);
    // fix up the arrays
    if(parseInt(ho) > parseInt(num_holes)) {
        // create space for the new entries
        //alert("blarg");
        for(var i = num_holes+1; i <= ho; i++) {
            intervals[i] = intervals[num_holes];
            hole_dias[i] = hole_dias[num_holes];
            hole_locs[i] = hole_locs[num_holes];
            diffs[i] = diffs[num_holes];
            cutoff_freqs[i] = cutoff_freqs[num_holes];
            frequencies[i] = 0.0;
            notes[i] = "";
        }
    }

    //alert("here1");
    // do the DHTML now
    var table = document.getElementById("rowTable");
    // delete the old hole lines
    for(var i = 0; i < num_holes; i++) {
        table.deleteRow(1);
    }

    //alert("here2");
    num_holes = ho;
    var row;
    var cell;
    var size = 7;
    // add the hole lines to the form
    for(var i = 1; i <= num_holes; i++) {
        row = table.insertRow(i);

        cell = row.insertCell(0);
        cell.innerHTML = "<strong>Hole "+i+"</strong>";

        cell = row.insertCell(1);
        cell.innerHTML = "<input TYPE=\"TEXT\" ID=\"interval"+
                        i+"\" VALUE=\""+intervals[i]+"\" SIZE=\""+2+"\""+
                        "onChange=\"changeInterval("+i+", this.value)\">";
```

```
        cell = row.insertCell(2);
        cell.innerHTML = "<input TYPE=\"TEXT\" ID=\"note"+
                        i+"\" READONLY VALUE=\""+notes[i]+"\" SIZE=\""+3+"\">";

        cell = row.insertCell(3);
        cell.innerHTML = "<input READONLY TYPE=\"TEXT\" ID=\"freq"+i
                        +"\" VALUE=\""+frequencies[i]+"\" SIZE=\""+size+"\">";

        // do the hole diameters with the increment/decrement buttons
        cell = row.insertCell(4);
        cell.innerHTML = "<table border=\"0\" cellspacing=\"0\"><tr><td>"+
                    "<table border=\"0\" cellspacing=\"0\"><tr><td>"+
                    "<input TYPE=\"TEXT\" ID=\"diam"+i+"\" onChange=\"RefreshAll
()\" VALUE=\""+
                            hole_dias[i]+"\" SIZE=\""+size+"\">"+
                    "</td></tr></table></td><td>"+
                    "<table border=\"0\" cellspacing=\"0\"><tr><td>"+
                    "<input type=\"BUTTON\" value=\"+\" "+
                            "style=\"border:1px solid;height:10px;width:5px;text
-align:top;font-size:7px\" "+
                            "onClick=\"incrementDia("+i+")\"/>"+
                    "</td></tr><tr><td>"+
                    "<input type=\"BUTTON\" value=\"-\" "+
                            "style=\"border:1px solid;height:10px;width:5px;text
-align:top;font-size:8px\" "+
                            "onClick=\"decrementDia("+i+")\"/>"+
                    "</td></tr></table></td></td></tr></table>";


        cell = row.insertCell(5);
        cell.innerHTML = "<input READONLY TYPE=\"TEXT\" ID=\"location"+i
                        +"\" VALUE=\""+hole_locs[i]+"\" SIZE=\""+size+"\">";

        cell = row.insertCell(6);
        cell.innerHTML = "<input READONLY TYPE=\"TEXT\" ID=\"diff"+i
                        +"\" VALUE=\""+diffs[i]+"\" SIZE=\""+size+"\">";

        cell = row.insertCell(7);
        cell.innerHTML = "<input READONLY TYPE=\"TEXT\" ID=\"cutoff"+i
                        +"\" VALUE=\""+cutoff_freqs[i]+"\" SIZE=\""+size+"\">";
    }

    //alert("here3");

    // update the form
    keyChange(selected);
    calcuate_func();
    put_form();
    //alert("finished changing holes");
}

//--------------------------------------------------------------------------

// Update the intervals in the array.
//--------------------------------------------------------------------------

function changeInterval(idx, value)
{
    //alert("index: "+idx+" value: "+value);
    intervals[idx] = value;
```

```
    keyChange(selected);
}

//--------------------------------------------------------------------------------

// Call this funciton when you need to transfer the data to the form.
//--------------------------------------------------------------------------------

function put_form()
{
    //alert("put form called");
    var ctrl = document.getElementById("bore");
    ctrl.value = bore_dia;
    ctrl = document.getElementById("title");
    ctrl.value = title;
    ctrl = document.getElementById("wall");
    ctrl.value = wall;
    ctrl = document.getElementById("numHoles");
    ctrl.value = num_holes;
    ctrl = document.getElementById("keySelect");
    ctrl.selectedIndex = parseInt(selected);
    ctrl = document.getElementById("diamEmb");
    ctrl.value = embouchure;
    ctrl = document.getElementById("freqEnd");
    ctrl.value = base_freq;
    if(standard == true) {
        document.getElementById("units1").checked = true;
        document.getElementById("units0").checked = false;
    }
    else {
        document.getElementById("units1").checked = false;
        document.getElementById("units0").checked = true;
    }

    if(usefracs == true) {
        document.getElementById("fractions0").checked = false;
        document.getElementById("fractions1").checked = true;
    }
    else {
        document.getElementById("fractions0").checked = true;
        document.getElementById("fractions1").checked = false;
    }

    // write the intervals
    var strg;
    for(var i = 1; i <= num_holes; i++) {
        strg = "interval"+i;
        ctrl = document.getElementById(strg);
        ctrl.value = intervals[i];
    }

    // write the frequencies
    for(var i = 1; i <= num_holes; i++) {
        strg = "freq"+i;
        ctrl = document.getElementById(strg);
        ctrl.value = frequencies[i];
    }

    // write the hole diameters
    for(var i = 1; i <= num_holes; i++) {
```

```
        strg = "diam"+i;
        ctrl = document.getElementById(strg);
        if(standard == true && usefracs == true)
            ctrl.value = reduce(hole_dias[i]);
        else
            ctrl.value = hole_dias[i];
    }

    // write the locations
    var num, val;
    for(var i = 1; i <= num_holes; i++) {
        strg = "location"+i;
        ctrl = document.getElementById(strg);
        val = round(xend - hole_locs[i], 4);
        ctrl.value = val;
        if(i > 1) {
            strg = "diff"+i;
            ctrl = document.getElementById(strg);
            ctrl.value = round(val - num, 3);
        }
        else {
            strg = "diff"+i;
            ctrl = document.getElementById(strg);
            ctrl.value = "---";
        }
        num = val;
    }

    // write the cutoff frequencies
    for(var i = 1; i <= num_holes; i++) {
        document.getElementById("cutoff"+i).value = round(cutoff_freqs[i], 2);
    }

    drawGraph();
}

//-----------------------------------------------------------------------------

// Call this function when you want to get the data from the form into the
// arrays of data so that the calculations can take place.
//-----------------------------------------------------------------------------

function get_form()
{
    //alert("get form called");
    var ctrl = document.getElementById("bore");
    bore_dia = parseFloat(ctrl.value);
    ctrl = document.getElementById("title");
    title = ctrl.value;
    ctrl = document.getElementById("wall");
    wall = parseFloat(ctrl.value);
    ctrl = document.getElementById("numHoles");
    if(ctrl.value > 12) {
        alert("Maximum number of holes is 12");
        num_holes = 12;
    }
    else if(ctrl.value < 1) {
        alert("Minimum number of holes is 3");
        num_holes = 1;
    }
```

```
    else
        num_holes = parseInt(ctrl.value);

    ctrl = document.getElementById("keySelect");
    selected = ctrl.selectedIndex;
    ctrl = document.getElementById("diamEmb");
    embouchure = parseFloat(ctrl.value);
    ctrl = document.getElementById("freqEnd");
    base_freq = parseFloat(ctrl.value);

    // read the intervals
    var strg;
    for(var i = 1; i <= num_holes; i++) {
        strg = "interval"+i;
        ctrl = document.getElementById(strg);
        intervals[i] = parseFloat(ctrl.value);
    }

    // read the frequencies
    for(var i = 1; i <= num_holes; i++) {
        strg = "freq"+i;
        ctrl = document.getElementById(strg);
        frequencies[i] = parseFloat(ctrl.value);
    }

    // read the hole diameters
    //alert("here2");
    for(var i = 1; i <= num_holes; i++) {
        strg = "diam"+i;
        ctrl = document.getElementById(strg);
        if(standard == true && usefracs == true)
            hole_dias[i] = fractod(ctrl.value);
        else
            hole_dias[i] = parseFloat(ctrl.value);
    }
}

//-------------------------------------------------------------------------

// Increment the hole number by 1/64", or by .5mm
//-------------------------------------------------------------------------

function incrementDia(num)
{
    //alert("increment the value = "+num);
    get_form();
    var strg = "diam"+num;
    var ctrl = document.getElementById(strg);
    if(standard == true) {
        // inch measurements
        if(usefracs == true)
            hole_dias[num] = fractod(ctrl.value);
        else
            hole_dias[num] = parseFloat(ctrl.value);
        hole_dias[num] = hole_dias[num] + (1.0 / 64.0);
    }
    else {
        // metric measurements
        hole_dias[num] = round_metric(hole_dias[num] + 0.5);
    }
```

```
    ctrl.value = hole_dias[num];
    calcuate_func();
    put_form();
}

//-------------------------------------------------------------------------------

// Decrement the hole number by 1/64", or by .5mm
//-------------------------------------------------------------------------------

function decrementDia(num)
{
    //alert("decrement the value = "+num);
    get_form();
    var strg = "diam"+num;
    var ctrl = document.getElementById(strg);
    if(standard == true) {
        // inch measurements
        if(usefracs == true)
            hole_dias[num] = fractod(ctrl.value);
        else
            hole_dias[num] = parseFloat(ctrl.value);
        hole_dias[num] = hole_dias[num] - (1.0 / 64.0);
    }
    else {
        // metric measurements
        hole_dias[num] = round_metric(hole_dias[num] - 0.5);
    }
    ctrl.value = hole_dias[num];
    calcuate_func();
    put_form();
}


//-------------------------------------------------------------------------------

//  Switch from decimal to metric or visa versa.
//-------------------------------------------------------------------------------

function changeUnits(num)
{
    //alert("changeUnits called = "+num);
    var old_standard = standard;
    get_form();

    // convert the internal parameters
    if(num != old_standard) {
        // convert metric to inches
        if(num == true) {
            for(var i = 1; i <= num_holes; i++) {
                var tmpval = round(parseFloat(hole_dias[i]) *  0.03937007874, 5)
;
                tmpval = reduce(tmpval);    // convert it to a fraction
                hole_dias[i] = parseFloat(fractod(tmpval)); // convert it back t
o a number
            }
            bore_dia = round(parseFloat(bore_dia) * 0.03937007874, 5);
            wall = round(parseFloat(wall) * 0.03937007874, 5);
            vsound = isound; //round(parseFloat(vsound) * 0.03937007874, 5);
            embouchure = round(parseFloat(embouchure) * 0.03937007874, 5);
```

```
        }
        // convert inches to metric
        else {
            for(var i = 1; i <= num_holes; i++) {
                hole_dias[i] = round_metric(parseFloat(hole_dias[i]) *  25.4);
            }
            bore_dia = round_metric(parseFloat(bore_dia) * 25.4);
            wall = round_metric(parseFloat(wall) * 25.4);
            vsound = msound; //round_metric(parseFloat(vsound) * 25.4);
            embouchure = round_metric(parseFloat(embouchure) * 25.4);
        }

        // update the display
        standard = num;
        //alert("standard set to \""+standard"+\"");
        if(num == false) {
            document.getElementById("units0").checked = true;
            document.getElementById("units1").checked = false;
            document.getElementById("fractions0").disabled = true;
            document.getElementById("fractions1").disabled = true;
        }
        else {
            document.getElementById("units1").checked = true;
            document.getElementById("units0").checked = false;
            document.getElementById("fractions0").disabled = false;
            document.getElementById("fractions1").disabled = false;
        }
    }
    calcuate_func();
    put_form();
}

//---------------------------------------------------------------------------

// Cahnge from decimal notation to fractional or visa versa.  This will not be
// called if the metric setting is active.
//---------------------------------------------------------------------------

function changeFracs(num)
{
    get_form();
    usefracs = num;
    if(num == false) {
        document.getElementById("fractions0").checked = true;
        document.getElementById("fractions1").checked = false;
    }
    else {
        document.getElementById("fractions0").checked = false;
        document.getElementById("fractions1").checked = true;
    }
    put_form();
}

//---------------------------------------------------------------------------

//  Change the calculation function
//---------------------------------------------------------------------------

function changeCalc(num)
{
```

```
    get_form();
    switch(num) {
        case 0: calcuate_func = FindLocations2;  break;
        case 1: calcuate_func = FindLocations;  break;
        default:
            alert("Invalid calculation method selected!\nShould never happen!");

    }
    calcuate_func();
    put_form();
    //alert("Calculate function set to "+calcuate_func.toString());
}

//--------------------------------------------------------------------------

//
//  Show the entire form, including all 3 tables.  This is only called when
//  the page is loaded.  Everything else is called in response to an event.
//
//  Events:
//  RefreshAll()    The big work horse
//  changeHoles()   Change the number of holes
//  keyChange()     Change the bell note
//  changeUnits()   Toggle using metric or imperial
//  changeFracs()   Toggle using farctions or decimal on imperial only
//  changeInterval()    Change an interval
//  incrementDia()  Increase the diameter of a hole
//  decrementDia()  Decrease the diameter of a hole
//  doPrint()       Response to the Print button
//  doSave()        Response to the Save button
//  doLoad()        Response to the Load pull-down
//  doDelete()      Response to the Delete pull-down
//  changeCalc()    Change the calculation method
//  showSaved()     Debugging only
//
//--------------------------------------------------------------------------

function show_form()
{
    var size = 7;   // cell size

    document.write("<form method=\"post\" action=\"START\" id=\"_flute_form\" na
me=\"fluteForm\">");

    document.write("");
    // this table has the controls that are common to all of the different
    // calculator scenarios.
    document.write("<table border=\"1\" id=\"fluteTable\">");
    document.write("<tr>");
    document.write("<td>Title</td>");
    document.write("<td colspan=\"3\"><input TYPE=\"TEXT\" ID=\"title\" VALUE=\"
My Whistle\" SIZE=\"35\"></td>");
    document.write("</tr>");
    document.write("<tr>");
    document.write("<td>Inside Diameter</td>");
    document.write("<td><input TYPE=\"TEXT\" onChange=\"RefreshAll()\" ID=\"bore
\" VALUE=\""+bore_dia+"\" SIZE=\""+size+"\"></td>");
    document.write("<td>Wall Thickness</td>");
    document.write("<td><input TYPE=\"TEXT\" onChange=\"RefreshAll()\" ID=\"wall
\" VALUE=\""+wall+"\" SIZE=\""+size+"\"></td>");
```

```
    document.write("</tr>");

    document.write("<tr>");
    document.write("<td>Number of holes</td>");
    document.write("<td><input TYPE=\"TEXT\" onChange=\"changeHoles(this.value)\
" ID=\"numHoles\" VALUE=\""+num_holes+"\" SIZE=\""+size+"\"></td>");
    document.write("<td>Select bell note</td>");
    document.write("<td><select ID=\"keySelect\" onChange=\"keyChange(this.selec
tedIndex)\">");
    for(var i = 0; i < bell_notes.length; i++)
        document.write("<option"+((selected==i)?" SELECTED":"")+"> "+bell_n
otes[i]+"  "+bell_freqs[i]+"Hz</option>");
    document.write("</select></td>");
    document.write("</tr>");

    document.write("<tr>");
    document.write("<td>Embouchure Diameter</td>");
    document.write("<td><input TYPE=\"TEXT\" onChange=\"RefreshAll()\" ID=\"diam
Emb\" VALUE=\""+embouchure+"\" SIZE=\""+size+"\"></td>");
    document.write("<td>Bell Frequency</td>");
    document.write("<td><input READONLY TYPE=\"TEXT\" ID=\"freqEnd\" VALUE=\""+b
ase_freq+"\" SIZE=\""+size+"\"></td>");
    document.write("</tr>");

    document.write("<tr>");
    document.write("<td>Units of measure</td>");
    document.write("<td><input TYPE=\"RADIO\" ID=\"units0\" onClick=\"changeUnit
s(false)\" "+(standard?"":"CHECKED")+">mm ");
    document.write("<input TYPE=\"RADIO\" ID=\"units1\" onClick=\"changeUnits(tr
ue)\" "+(standard?"CHECKED":"")+">inch </td>");
    document.write("<td>Use Fractions</td>");
    document.write("<td><input "+(standard==false? "DISABLED": "")+"TYPE=\"RADIO
\" ID=\"fractions0\" onClick=\"changeFracs(false)\" "+(usefracs?"":"CHECKED")+">
decimals ");
    document.write("<input "+(standard==false? "DISABLED": "")+"TYPE=\"RADIO\" I
D=\"fractions1\" onClick=\"changeFracs(true)\" "+(usefracs?"CHECKED":"")+">fract
ions </td>");
    document.write("</tr>");
    document.write("</table><br/>");

    // This table has the whistle rows.  One for each hole in the instrument.
    document.write("<table border=\"1\" id=\"rowTable\">");
    document.write("<tr><th>        &nbs
p;   </th>");
    document.write("<th> <b>Interval</b> </th>");
    document.write("<th> <b>Note</b> </th>");
    document.write("<th> <b>Frequency</b> </th>");
    document.write("<th> <b>Hole Dia</b> </th>");
    document.write("<th> <b>Location</b> </th>");
    document.write("<th> <b>Diff</b></th>");
    document.write("<th> <b>Cutoff</b></th></tr>");

    for(var i = 0; i < num_holes; i++) {
        document.write("<tr>");
        document.write("<th><b>Hole "+(i+1)+"</b></th>");
        document.write("<td><input TYPE=\"TEXT\" ID=\"interval"+
                       (i+1)+"\" VALUE=\""+intervals[i+1]+"\" SIZE=\""+2+"\""+
                       "onChange=\"changeInterval("+(i+1)+", this.value)\"></td
>");
        document.write("<td><input TYPE=\"TEXT\" ID=\"note"+
```

```
                        (i+1)+"\" READONLY VALUE=\""+notes[i+1]+"\" SIZE=\""+3+"
\"></td>");
        document.write("<td><input READONLY TYPE=\"TEXT\" ID=\"freq"+(i+1)
                        +"\" VALUE=\""+frequencies[i+1]+"\" SIZE=\""+size+"\"></
td>");

        document.write("<td>");
        // this is a nested table, used to get the buttons to show up in the
        // right places.
        document.write("<table border=\"0\" cellspacing=\"0\">");
            document.write("<tr><td>");

            document.write("<table border=\"0\" cellspacing=\"0\">");
                document.write("<tr><td>");
                document.write("<input TYPE=\"TEXT\" onChange=\"RefreshAll()\" I
D=\"diam"+(i+1)
                                +"\" VALUE=\""+hole_dias[i+1]+"\" SIZE=\""+size+
"\">");
                document.write("</td></tr>");
            document.write("</table>");
            document.write("</td><td>");
            document.write("<table border=\"0\" cellspacing=\"0\">");
                document.write("<tr><td>");
                document.write("<input type=\"BUTTON\" value=\"+\" style=\"borde
r:1px solid;height:10px;width:5px;text-align:top;font-size:7px\" onClick=\"incre
mentDia("+(i+1)+")\"/>");
                document.write("</td></tr>");
                document.write("<tr><td>");
                document.write("<input type=\"BUTTON\" value=\"-\" style=\"borde
r:1px solid;height:10px;width:5px;text-align:top;font-size:8px\" onClick=\"decre
mentDia("+(i+1)+")\"/>");
                document.write("</td></tr>");
            document.write("</table></td>");

            document.write("</td></tr>");
        document.write("</table>");
        document.write("</td>");

        document.write("<td><input READONLY TYPE=\"TEXT\" ID=\"location"+(i+1)
                        +"\" VALUE=\""+hole_locs[i+1]+"\" SIZE=\""+size+"\"></td
>");
        document.write("<td><input READONLY TYPE=\"TEXT\" ID=\"diff"+(i+1)
                        +"\" VALUE=\""+diffs[i+1]+"\" SIZE=\""+size+"\"></td>");

        document.write("<td><input READONLY TYPE=\"TEXT\" ID=\"cutoff"+(i+1)
                        +"\" VALUE=\""+cutoff_freqs[i+1]+"\" SIZE=\""+size+"\"><
/td>");
        document.write("</tr>");
    }
    document.write("</table><br/>");

    // This table has the controls, such as "calculate".
    document.write("<table border=\"0\" id=\"ctrlTable\">");
    document.write("<tr>");
    document.write("<td align=\"center\"><input STYLE=\"width:65px\" TYPE=\"BUTT
ON\" VALUE=\"Refresh\" onClick=\"RefreshAll();\"></td>");
    document.write("<td align=\"center\"><input STYLE=\"width:65px\" TYPE=\"BUTT
ON\" VALUE=\"Print\" onClick=\"doPrint();\"></td>");
    document.write("<td align=\"center\"><input STYLE=\"width:65px\" TYPE=\"BUTT
ON\" VALUE=\"Save\" onClick=\"doSave();\"></td>");
```

```
    document.write("<td align=\"center\"><select STYLE=\"max-width:75px\" ID=\"l
oadCtrl\" onChange=\"doLoad(this.selectedIndex);\"><option SELECTED>Load</option
><option>----------</option></select></td>");
    document.write("<td align=\"center\"><select STYLE=\"max-width:75px\" ID=\"d
elCtrl\" onChange=\"doDelete(this.selectedIndex);\"><option SELECTED>Delete</opt
ion><option>----------</option></select></td>");
    document.write("</tr>");
    document.write("<tr>");
    document.write("<td align=\"center\" colspan=\"3\">Calculation Method: <sele
ct ID=\"calcMethod\" onChange=\"changeCalc(this.selectedIndex)\">");
    document.write("<option SELECTED>Non-iterative Benade</option>");
    document.write("<option>Iterative Benade</option>");
    document.write("</select></td>");
    //document.write("<td align=\"center\"><input TYPE=\"BUTTON\" VALUE=\"Show S
aved\" onClick=\"showSaved();\"></td>");
    document.write("</tr>");
    document.write("</table>");
    document.write("</form>");
    document.write("<canvas id=\"cutoff_graph\" width=\"610\" height=\"200\">");

    document.write("some silly schtuff");
    document.write("</canvas>");
    load_whistles();
}


//============================================================================

//
//  Drawing the graph
//
//============================================================================


//----------------------------------------------------------------------------

//  Draw the graph
//----------------------------------------------------------------------------

function drawGraph()
{
    var canvas = document.getElementById("cutoff_graph");
    // drawing area
    var start_x = 50;
    var start_y = 40;
    var end_x = canvas.width;
    var end_y = canvas.height - 20;

    if(canvas.getContext) {
        var ctx = canvas.getContext("2d");

        // Create the graph background
        ctx.clearRect(0, 0, canvas.width, canvas.height);
        ctx.globalAlpha = 1.0;
        ctx.fillStyle = "rgba(200,200,255,0.66)";
        ctx.fillRect(start_x, start_y, end_x, end_y-start_y);
        drawLine(ctx, 0, 0, canvas.width, 0, "#000", 4);
        drawLine(ctx, canvas.width, 0, canvas.width, canvas.height, "#000", 4);
        drawLine(ctx, canvas.width, canvas.height, 0, canvas.height, "#000", 4);
```

```
        drawLine(ctx, 0, canvas.height, 0, 0, "#000", 4);

        //ctx.strokeRect(0, 0, canvas.width, canvas.height);

        drawLine(ctx, start_x, start_y, end_x, start_y, "#505050", 1);
        // make 10 horizontal lines
        var inc = (end_y - start_y) / 11;
        var x = inc + start_y;
        for(var i = 0; i <= 10; i++) {
            drawLine(ctx, start_x, x, end_x, x, "#505050", 1);
            x += inc;
        }

        drawLine(ctx, start_x, start_y, start_x, end_y, "#505050", 1);
        // make a vertical line for each hole
        inc = (end_x - start_x) / (parseInt(num_holes) + 1);
        x = inc + start_x;
        for(var i = 0; i < parseInt(num_holes); i++) {
            drawLine(ctx, x, start_y, x, end_y, "#505050", 1);
            x += inc;
        }

        // Add the text
        // this adds the text functions to the ctx
        CanvasTextFunctions.enable(ctx);
        var font = "sans";
        var fontsize = 10.75;
        var y = ctx.fontAscent(font, fontsize) + 5;
        var tcolor = "#000";
        ctx.drawTextCenter(font, fontsize, tcolor, canvas.width/2, y, "Cutoff Fr
equency Graph");
        y = (canvas.height - ctx.fontAscent(font, fontsize)) + 5;
        ctx.drawText(font, fontsize, tcolor, start_x, y, "octave freqs");
        ctx.drawText(font, fontsize, tcolor, start_x + 150, y, "cutoff freqs");
        y -= 5;
        drawLine(ctx, start_x + 90, y,  start_x + 120, y, "#00ffff", 2);
        drawLine(ctx, start_x + 240, y,  start_x + 270, y, "#0000ff", 2);

        // draw the text for the scaled frequency labels
        var minf = parseInt(((base_freq * 2) - ((base_freq * 2) * 0.01)) / 100)
* 100;
        var maxf = parseInt(((frequencies[num_holes] * 2) + ((frequencies[num_ho
les] * 2) * 0.5)) / 100) * 100;
        var vinc = (maxf - minf) / 10;
        inc = (end_y - start_y) / 11;
        x = inc + start_y + 3.75;
        //alert("minf = "+minf+"  maxf = "+maxf+"  vinc = "+vinc);
        for(var i = 0; i < 10; i++) {
            ctx.drawText(font, 8.5, tcolor, 10, x, ""+(maxf));
            maxf -= vinc;
            x += inc;
        }

        // draw the text for the hole labels
        inc = (end_x - start_x) / (parseInt(num_holes) + 1);
        x = inc + start_x;
        for(var i = 1; i <= num_holes; i++) {
            ctx.drawTextCenter(font, 8.5, tcolor, x, 35, "Hole"+i);
            x += inc;
        }
```

```
        // draw the octave freqs

        // draw the cutoff freqs

    }
    else {
        // canvas is unsupported
        document.write("Cannot display graph because \"canvas\" is not supported
 by your browser. Get a newer/better browser.");
    }
}

function drawLine(ctx, sx, sy, ex, ey, color, width)
{
    ctx.strokeStyle = color;
    ctx.lineWidth = width;
    ctx.beginPath();
    ctx.moveTo(sx, sy);
    ctx.lineTo(ex, ey);
    ctx.stroke();
    ctx.closePath();
}

//
// This code is released to the public domain by Jim Studt, 2007.
// He may keep some sort of up to date copy at http://www.federated.com/~jim/can
vastext/
//
var CanvasTextFunctions = { };

CanvasTextFunctions.letters = {
    ' ': { width: 16, points: [] },
    '!': { width: 10, points: [[5,21],[5,7],[-1,-1],[5,2],[4,1],[5,0],[6,1],[5,2
]] },
    '"': { width: 16, points: [[4,21],[4,14],[-1,-1],[12,21],[12,14]] },
    '#': { width: 21, points: [[11,25],[4,-7],[-1,-1],[17,25],[10,-7],[-1,-1],[4
,12],[18,12],[-1,-1],[3,6],[17,6]] },
    '$': { width: 20, points: [[8,25],[8,-4],[-1,-1],[12,25],[12,-4],[-1,-1],[17
,18],[15,20],[12,21],[8,21],[5,20],[3,18],[3,16],[4,14],[5,13],[7,12],[13,10],[1
5,9],[16,8],[17,6],[17,3],[15,1],[12,0],[8,0],[5,1],[3,3]] },
    '%': { width: 24, points: [[21,21],[3,0],[-1,-1],[8,21],[10,19],[10,17],[9,1
5],[7,14],[5,14],[3,16],[3,18],[4,20],[6,21],[8,21],[10,20],[13,19],[16,19],[19,
20],[21,21],[-1,-1],[17,7],[15,6],[14,4],[14,2],[16,0],[18,0],[20,1],[21,3],[21,
5],[19,7],[17,7]] },
    '&': { width: 26, points: [[23,12],[23,13],[22,14],[21,14],[20,13],[19,11],[
17,6],[15,3],[13,1],[11,0],[7,0],[5,1],[4,2],[3,4],[3,6],[4,8],[5,9],[12,13],[13
,14],[14,16],[14,18],[13,20],[11,21],[9,20],[8,18],[8,16],[9,13],[11,10],[16,3],
[18,1],[20,0],[22,0],[23,1],[23,2]] },
    '\'': { width: 10, points: [[5,19],[4,20],[5,21],[6,20],[6,18],[5,16],[4,15]
] },
    '(': { width: 14, points: [[11,25],[9,23],[7,20],[5,16],[4,11],[4,7],[5,2],[
7,-2],[9,-5],[11,-7]] },
    ')': { width: 14, points: [[3,25],[5,23],[7,20],[9,16],[10,11],[10,7],[9,2],
[7,-2],[5,-5],[3,-7]] },
    '*': { width: 16, points: [[8,21],[8,9],[-1,-1],[3,18],[13,12],[-1,-1],[13,1
8],[3,12]] },
    '+': { width: 26, points: [[13,18],[13,0],[-1,-1],[4,9],[22,9]] },
    ',': { width: 10, points: [[6,1],[5,0],[4,1],[5,2],[6,1],[6,-1],[5,-3],[4,-4
]] },
```

```
    '-': { width: 26, points: [[4,9],[22,9]] },
    '.': { width: 10, points: [[5,2],[4,1],[5,0],[6,1],[5,2]] },
    '/': { width: 22, points: [[20,25],[2,-7]] },
    '0': { width: 20, points: [[9,21],[6,20],[4,17],[3,12],[3,9],[4,4],[6,1],[9,
0],[11,0],[14,1],[16,4],[17,9],[17,12],[16,17],[14,20],[11,21],[9,21]] },
    '1': { width: 20, points: [[6,17],[8,18],[11,21],[11,0]] },
    '2': { width: 20, points: [[4,16],[4,17],[5,19],[6,20],[8,21],[12,21],[14,20
],[15,19],[16,17],[16,15],[15,13],[13,10],[3,0],[17,0]] },
    '3': { width: 20, points: [[5,21],[16,21],[10,13],[13,13],[15,12],[16,11],[1
7,8],[17,6],[16,3],[14,1],[11,0],[8,0],[5,1],[4,2],[3,4]] },
    '4': { width: 20, points: [[13,21],[3,7],[18,7],[-1,-1],[13,21],[13,0]] },
    '5': { width: 20, points: [[15,21],[5,21],[4,12],[5,13],[8,14],[11,14],[14,1
3],[16,11],[17,8],[17,6],[16,3],[14,1],[11,0],[8,0],[5,1],[4,2],[3,4]] },
    '6': { width: 20, points: [[16,18],[15,20],[12,21],[10,21],[7,20],[5,17],[4,
12],[4,7],[5,3],[7,1],[10,0],[11,0],[14,1],[16,3],[17,6],[17,7],[16,10],[14,12],
[11,13],[10,13],[7,12],[5,10],[4,7]] },
    '7': { width: 20, points: [[17,21],[7,0],[-1,-1],[3,21],[17,21]] },
    '8': { width: 20, points: [[8,21],[5,20],[4,18],[4,16],[5,14],[7,13],[11,12]
,[14,11],[16,9],[17,7],[17,4],[16,2],[15,1],[12,0],[8,0],[5,1],[4,2],[3,4],[3,7]
,[4,9],[6,11],[9,12],[13,13],[15,14],[16,16],[16,18],[15,20],[12,21],[8,21]] },
    '9': { width: 20, points: [[16,14],[15,11],[13,9],[10,8],[9,8],[6,9],[4,11],
[3,14],[3,15],[4,18],[6,20],[9,21],[10,21],[13,20],[15,18],[16,14],[16,9],[15,4]
,[13,1],[10,0],[8,0],[5,1],[4,3]] },
    ':': { width: 10, points: [[5,14],[4,13],[5,12],[6,13],[5,14],[-1,-1],[5,2],
[4,1],[5,0],[6,1],[5,2]] },
    ',': { width: 10, points: [[5,14],[4,13],[5,12],[6,13],[5,14],[-1,-1],[6,1],
[5,0],[4,1],[5,2],[6,1],[6,-1],[5,-3],[4,-4]] },
    '<': { width: 24, points: [[20,18],[4,9],[20,0]] },
    '=': { width: 26, points: [[4,12],[22,12],[-1,-1],[4,6],[22,6]] },
    '>': { width: 24, points: [[4,18],[20,9],[4,0]] },
    '?': { width: 18, points: [[3,16],[3,17],[4,19],[5,20],[7,21],[11,21],[13,20
],[14,19],[15,17],[15,15],[14,13],[13,12],[9,10],[9,7],[-1,-1],[9,2],[8,1],[9,0]
,[10,1],[9,2]] },
    '@': { width: 27, points: [[18,13],[17,15],[15,16],[12,16],[10,15],[9,14],[8
,11],[8,8],[9,6],[11,5],[14,5],[16,6],[17,8],[-1,-1],[12,16],[10,14],[9,11],[9,8
],[10,6],[11,5],[-1,-1],[18,16],[17,8],[17,6],[19,5],[21,5],[23,7],[24,10],[24,1
2],[23,15],[22,17],[20,19],[18,20],[15,21],[12,21],[9,20],[7,19],[5,17],[4,15],[
3,12],[3,9],[4,6],[5,4],[7,2],[9,1],[12,0],[15,0],[18,1],[20,2],[21,3],[-1,-1],[
19,16],[18,8],[18,6],[19,5]] },
    'A': { width: 18, points: [[9,21],[1,0],[-1,-1],[9,21],[17,0],[-1,-1],[4,7],
[14,7]] },
    'B': { width: 21, points: [[4,21],[4,0],[-1,-1],[4,21],[13,21],[16,20],[17,1
9],[18,17],[18,15],[17,13],[16,12],[13,11],[-1,-1],[4,11],[13,11],[16,10],[17,9]
,[18,7],[18,4],[17,2],[16,1],[13,0],[4,0]] },
    'C': { width: 21, points: [[18,16],[17,18],[15,20],[13,21],[9,21],[7,20],[5,
18],[4,16],[3,13],[3,8],[4,5],[5,3],[7,1],[9,0],[13,0],[15,1],[17,3],[18,5]] },
    'D': { width: 21, points: [[4,21],[4,0],[-1,-1],[4,21],[11,21],[14,20],[16,1
8],[17,16],[18,13],[18,8],[17,5],[16,3],[14,1],[11,0],[4,0]] },
    'E': { width: 19, points: [[4,21],[4,0],[-1,-1],[4,21],[17,21],[-1,-1],[4,11
],[12,11],[-1,-1],[4,0],[17,0]] },
    'F': { width: 18, points: [[4,21],[4,0],[-1,-1],[4,21],[17,21],[-1,-1],[4,11
],[12,11]] },
    'G': { width: 21, points: [[18,16],[17,18],[15,20],[13,21],[9,21],[7,20],[5,
18],[4,16],[3,13],[3,8],[4,5],[5,3],[7,1],[9,0],[13,0],[15,1],[17,3],[18,5],[18,
8],[-1,-1],[13,8],[18,8]] },
    'H': { width: 22, points: [[4,21],[4,0],[-1,-1],[18,21],[18,0],[-1,-1],[4,11
],[18,11]] },
    'I': { width: 8, points: [[4,21],[4,0]] },
    'J': { width: 16, points: [[12,21],[12,5],[11,2],[10,1],[8,0],[6,0],[4,1],[3
,2],[2,5],[2,7]] },
```

```
    'K': { width: 21, points: [[4,21],[4,0],[-1,-1],[18,21],[4,7],[-1,-1],[9,12]
,[18,0]] },
    'L': { width: 17, points: [[4,21],[4,0],[-1,-1],[4,0],[16,0]] },
    'M': { width: 24, points: [[4,21],[4,0],[-1,-1],[4,21],[12,0],[-1,-1],[20,21
],[12,0],[-1,-1],[20,21],[20,0]] },
    'N': { width: 22, points: [[4,21],[4,0],[-1,-1],[4,21],[18,0],[-1,-1],[18,21
],[18,0]] },
    'O': { width: 22, points: [[9,21],[7,20],[5,18],[4,16],[3,13],[3,8],[4,5],[5
,3],[7,1],[9,0],[13,0],[15,1],[17,3],[18,5],[19,8],[19,13],[18,16],[17,18],[15,2
0],[13,21],[9,21]] },
    'P': { width: 21, points: [[4,21],[4,0],[-1,-1],[4,21],[13,21],[16,20],[17,1
9],[18,17],[18,14],[17,12],[16,11],[13,10],[4,10]] },
    'Q': { width: 22, points: [[9,21],[7,20],[5,18],[4,16],[3,13],[3,8],[4,5],[5
,3],[7,1],[9,0],[13,0],[15,1],[17,3],[18,5],[19,8],[19,13],[18,16],[17,18],[15,2
0],[13,21],[9,21],[-1,-1],[12,4],[18,-2]] },
    'R': { width: 21, points: [[4,21],[4,0],[-1,-1],[4,21],[13,21],[16,20],[17,1
9],[18,17],[18,15],[17,13],[16,12],[13,11],[4,11],[-1,-1],[11,11],[18,0]] },
    'S': { width: 20, points: [[17,18],[15,20],[12,21],[8,21],[5,20],[3,18],[3,1
6],[4,14],[5,13],[7,12],[13,10],[15,9],[16,8],[17,6],[17,3],[15,1],[12,0],[8,0],
[5,1],[3,3]] },
    'T': { width: 16, points: [[8,21],[8,0],[-1,-1],[1,21],[15,21]] },
    'U': { width: 22, points: [[4,21],[4,6],[5,3],[7,1],[10,0],[12,0],[15,1],[17
,3],[18,6],[18,21]] },
    'V': { width: 18, points: [[1,21],[9,0],[-1,-1],[17,21],[9,0]] },
    'W': { width: 24, points: [[2,21],[7,0],[-1,-1],[12,21],[7,0],[-1,-1],[12,21
],[17,0],[-1,-1],[22,21],[17,0]] },
    'X': { width: 20, points: [[3,21],[17,0],[-1,-1],[17,21],[3,0]] },
    'Y': { width: 18, points: [[1,21],[9,11],[9,0],[-1,-1],[17,21],[9,11]] },
    'Z': { width: 20, points: [[17,21],[3,0],[-1,-1],[3,21],[17,21],[-1,-1],[3,0
],[17,0]] },
    '[': { width: 14, points: [[4,25],[4,-7],[-1,-1],[5,25],[5,-7],[-1,-1],[4,25
],[11,25],[-1,-1],[4,-7],[11,-7]] },
    '\\': { width: 14, points: [[0,21],[14,-3]] },
    ']': { width: 14, points: [[9,25],[9,-7],[-1,-1],[10,25],[10,-7],[-1,-1],[3,
25],[10,25],[-1,-1],[3,-7],[10,-7]] },
    '^': { width: 16, points: [[6,15],[8,18],[10,15],[-1,-1],[3,12],[8,17],[13,1
2],[-1,-1],[8,17],[8,0]] },
    '_': { width: 16, points: [[0,-2],[16,-2]] },
    '`': { width: 10, points: [[6,21],[5,20],[4,18],[4,16],[5,15],[6,16],[5,17]]
 },
    'a': { width: 19, points: [[15,14],[15,0],[-1,-1],[15,11],[13,13],[11,14],[8
,14],[6,13],[4,11],[3,8],[3,6],[4,3],[6,1],[8,0],[11,0],[13,1],[15,3]] },
    'b': { width: 19, points: [[4,21],[4,0],[-1,-1],[4,11],[6,13],[8,14],[11,14]
,[13,13],[15,11],[16,8],[16,6],[15,3],[13,1],[11,0],[8,0],[6,1],[4,3]] },
    'c': { width: 18, points: [[15,11],[13,13],[11,14],[8,14],[6,13],[4,11],[3,8
],[3,6],[4,3],[6,1],[8,0],[11,0],[13,1],[15,3]] },
    'd': { width: 19, points: [[15,21],[15,0],[-1,-1],[15,11],[13,13],[11,14],[8
,14],[6,13],[4,11],[3,8],[3,6],[4,3],[6,1],[8,0],[11,0],[13,1],[15,3]] },
    'e': { width: 18, points: [[3,8],[15,8],[15,10],[14,12],[13,13],[11,14],[8,1
4],[6,13],[4,11],[3,8],[3,6],[4,3],[6,1],[8,0],[11,0],[13,1],[15,3]] },
    'f': { width: 12, points: [[10,21],[8,21],[6,20],[5,17],[5,0],[-1,-1],[2,14]
,[9,14]] },
    'g': { width: 19, points: [[15,14],[15,-2],[14,-5],[13,-6],[11,-7],[8,-7],[6
,-6],[-1,-1],[15,11],[13,13],[11,14],[8,14],[6,13],[4,11],[3,8],[3,6],[4,3],[6,1
],[8,0],[11,0],[13,1],[15,3]] },
    'h': { width: 19, points: [[4,21],[4,0],[-1,-1],[4,10],[7,13],[9,14],[12,14]
,[14,13],[15,10],[15,0]] },
    'i': { width: 8, points: [[3,21],[4,20],[5,21],[4,22],[3,21],[-1,-1],[4,14],
[4,0]] },
    'j': { width: 10, points: [[5,21],[6,20],[7,21],[6,22],[5,21],[-1,-1],[6,14]
```

```
,[6,-3],[5,-6],[3,-7],[1,-7]] },
    'k': { width: 17, points: [[4,21],[4,0],[-1,-1],[14,14],[4,4],[-1,-1],[8,8],
[15,0]] },
    'l': { width: 8, points: [[4,21],[4,0]] },
    'm': { width: 30, points: [[4,14],[4,0],[-1,-1],[4,10],[7,13],[9,14],[12,14]
,[14,13],[15,10],[15,0],[-1,-1],[15,10],[18,13],[20,14],[23,14],[25,13],[26,10],
[26,0]] },
    'n': { width: 19, points: [[4,14],[4,0],[-1,-1],[4,10],[7,13],[9,14],[12,14]
,[14,13],[15,10],[15,0]] },
    'o': { width: 19, points: [[8,14],[6,13],[4,11],[3,8],[3,6],[4,3],[6,1],[8,0
],[11,0],[13,1],[15,3],[16,6],[16,8],[15,11],[13,13],[11,14],[8,14]] },
    'p': { width: 19, points: [[4,14],[4,-7],[-1,-1],[4,11],[6,13],[8,14],[11,14
],[13,13],[15,11],[16,8],[16,6],[15,3],[13,1],[11,0],[8,0],[6,1],[4,3]] },
    'q': { width: 19, points: [[15,14],[15,-7],[-1,-1],[15,11],[13,13],[11,14],[
8,14],[6,13],[4,11],[3,8],[3,6],[4,3],[6,1],[8,0],[11,0],[13,1],[15,3]] },
    'r': { width: 13, points: [[4,14],[4,0],[-1,-1],[4,8],[5,11],[7,13],[9,14],[
12,14]] },
    's': { width: 17, points: [[14,11],[13,13],[10,14],[7,14],[4,13],[3,11],[4,9
],[6,8],[11,7],[13,6],[14,4],[14,3],[13,1],[10,0],[7,0],[4,1],[3,3]] },
    't': { width: 12, points: [[5,21],[5,4],[6,1],[8,0],[10,0],[-1,-1],[2,14],[9
,14]] },
    'u': { width: 19, points: [[4,14],[4,4],[5,1],[7,0],[10,0],[12,1],[15,4],[-1
,-1],[15,14],[15,0]] },
    'v': { width: 16, points: [[2,14],[8,0],[-1,-1],[14,14],[8,0]] },
    'w': { width: 22, points: [[3,14],[7,0],[-1,-1],[11,14],[7,0],[-1,-1],[11,14
],[15,0],[-1,-1],[19,14],[15,0]] },
    'x': { width: 17, points: [[3,14],[14,0],[-1,-1],[14,14],[3,0]] },
    'y': { width: 16, points: [[2,14],[8,0],[-1,-1],[14,14],[8,0],[6,-4],[4,-6],
[2,-7],[1,-7]] },
    'z': { width: 17, points: [[14,14],[3,0],[-1,-1],[3,14],[14,14],[-1,-1],[3,0
],[14,0]] },
    '{': { width: 14, points: [[9,25],[7,24],[6,23],[5,21],[5,19],[6,17],[7,16],
[8,14],[8,12],[6,10],[-1,-1],[7,24],[6,22],[6,20],[7,18],[8,17],[9,15],[9,13],[8
,11],[4,9],[8,7],[9,5],[9,3],[8,1],[7,0],[6,-2],[6,-4],[7,-6],[-1,-1],[6,8],[8,6
],[8,4],[7,2],[6,1],[5,-1],[5,-3],[6,-5],[7,-6],[9,-7]] },
    '|': { width: 8, points: [[4,25],[4,-7]] },
    '}': { width: 14, points: [[5,25],[7,24],[8,23],[9,21],[9,19],[8,17],[7,16],
[6,14],[6,12],[8,10],[-1,-1],[7,24],[8,22],[8,20],[7,18],[6,17],[5,15],[5,13],[6
,11],[10,9],[6,7],[5,5],[5,3],[6,1],[7,0],[8,-2],[8,-4],[7,-6],[-1,-1],[8,8],[6,
6],[6,4],[7,2],[8,1],[9,-1],[9,-3],[8,-5],[7,-6],[5,-7]] },
    '~': { width: 24, points: [[3,6],[3,8],[4,11],[6,12],[8,12],[10,11],[14,8],[
16,7],[18,7],[20,8],[21,10],[-1,-1],[3,8],[4,10],[6,11],[8,11],[10,10],[14,7],[1
6,6],[18,6],[20,7],[21,10],[21,12]] }
};

CanvasTextFunctions.letter = function (ch)
{
    return CanvasTextFunctions.letters[ch];
}

CanvasTextFunctions.ascent = function( font, size)
{
    return size;
}

CanvasTextFunctions.descent = function( font, size)
{
    return 7.0*size/25.0;
}
```

```
CanvasTextFunctions.measure = function( font, size, str)
{
    var total = 0;
    var len = str.length;

    for ( i = 0; i < len; i++) {
    var c = CanvasTextFunctions.letter( str.charAt(i));
    if ( c) total += c.width * size / 25.0;
    }
    return total;
}

CanvasTextFunctions.draw = function(ctx,font,size,color,x,y,str)
{
    var total = 0;
    var len = str.length;
    var mag = size / 25.0;

    ctx.save();
    ctx.lineCap = "round";
    ctx.lineWidth = 2.0 * mag;
    ctx.strokeStyle = color;

    for ( i = 0; i < len; i++) {
    var c = CanvasTextFunctions.letter( str.charAt(i));
    if ( !c) continue;

    ctx.beginPath();

    var penUp = 1;
    var needStroke = 0;
    for ( j = 0; j < c.points.length; j++) {
        var a = c.points[j];
        if ( a[0] == -1 && a[1] == -1) {
        penUp = 1;
        continue;
        }
        if ( penUp) {
        ctx.moveTo( x + a[0]*mag, y - a[1]*mag);
        penUp = false;
        } else {
        ctx.lineTo( x + a[0]*mag, y - a[1]*mag);
        }
    }
    ctx.stroke();
    x += c.width*mag;
    }
    ctx.restore();
    return total;
}

CanvasTextFunctions.enable = function( ctx)
{
    ctx.drawText = function(font,size,color,x,y,text) { return CanvasTextFunctio
ns.draw( ctx, font,size,color,x,y,text); };
    ctx.measureText = function(font,size,text) { return CanvasTextFunctions.meas
ure( font,size,text); };
    ctx.fontAscent = function(font,size) { return CanvasTextFunctions.ascent(fon
t,size); }
    ctx.fontDescent = function(font,size) { return CanvasTextFunctions.descent(f
```

```
ont,size); }

    ctx.drawTextRight = function(font,size,color,x,y,text) {
    var w = CanvasTextFunctions.measure(font,size,text);
    return CanvasTextFunctions.draw( ctx, font,size,color,x-w,y,text);
    };
    ctx.drawTextCenter = function(font,size,color,x,y,text) {
    var w = CanvasTextFunctions.measure(font,size,text);
    return CanvasTextFunctions.draw( ctx, font,size,color,x-w/2,y,text);
    };
}
```