What's new in version 3.0 of viewkel?

The short answer is: all kinds of stuff, I'm not even sure I remember all of it (I haven't used version 2.x since the day it came out).

That's not a very satisfactory answer, so here's a somewhat more systematic attempt to list everything new, in no particular order.

- Ability to export VRML files as well as rayshade files. The VRML exported is either VRML-2.0 or VRML-97. I can't keep them straight, but the VRML that comes out works just fine in CosmoViewer 2.0 from SGI.
- Many of the nasty core leaks that caused crashes and/or excessive growth of the program (particularly when dealing with MOs) have been located and fixed. *Note for lazy programmers:* I've included the functions I wrote to do this with the program, look in `mem_debug.c`.
- Numerous bug fixes.
- Numerous new bugs. Realistically this has to be the case... the thing is fairly well tested, but I'm sure there are things which didn't get tweaked.
- Atoms can be printed using a shading pattern which simulates lighting. They can also be printed in color, or in color with shading, or in the plain, boring old style. If you have a color display, they will even be displayed on screen in color or with the shading pattern shown. This stuff is all controlled in the "Printing Options" window. Makes sense, neh?
- Atoms and bonds can be selected and their drawing characteristics can be altered (by typing `a` with either one or two atoms selected). So you can change the sizes, colors, shading patterns, whatever of atoms and bonds. By selecting two atoms that don't have a bond between them and typing `a` it's even now possible to *add* a new bond to the drawing.
- Atoms can be selected in groups by clicking and dragging in the main window. So you can drag out a box and get all the atoms in the box. This is slower than it ought to be, but it works just fine.
- viewkel can now use the GNU readline library when interacting with the user. This allows the use of command histories and file name completion when entering file names. *hint:* try hitting the up arrow the second time you are prompted for a file name in viewkel, or hitting the Tab key while typing a file name.
- It's possible to hide atoms by selecting them and then hitting 'h' (could you do this in version 2? I don't remember, probably it was, but I forgot to document it.)
- There are no longer 400 different "Read" buttons in the Main Options window. These have been replaced by a single "Read Data" button. Viewkel now tries to figure out

what kind of file it is given automagically. If it cannot do so, then it will ask you what type of file it should read. **Note:** the automatic recognition of file types does not work so well for files from older versions of bind and the fitting utilities.

- Hitting the question mark ('?') in the viewkel window will now show you a list of what the individual keys do. At least what most of them do. This is a convenient way to determine which features I forgot to document in both the manual and here.
- The functions which evaluate contour plots (and contour surfaces) now use symmetry planes when they can. This can result in enormous time savings when looking at MOs.
- The hidden line removal code for psi88 style plots has been cleaned up a little bit. I caught at least one strange case in there, so those annoying little bleed-through problems are now lessened (oh, rest assured that they are still there... the boundary conditions kill me).
- Coordination environments... hide all atoms farther than some specified distance from a particular atom.
- Coordination polyhedra. With shading. Woo Hoo!
- The "fancy lines" option (where bonds are stopped part of the way into atoms) works much better now... provided the user doesn't alter the z scaling too much.
- The white lines which make tube bonds and breaking lines break other lines are now halted at the edge of the atom. This makes plots look a lot better.
- The "Energy (eV)" legend for property graphs (and everything else I remembered to update) is now rotated, so it looks better. (Yes, I keep saying "looks better"... it's a visualization program... "looks better" is important!).
- The MO routines use less memory now. (There, that wasn't about looks).
- You can do contour plots along arbitrary planes, just select three atoms which define the plane before hitting "contour it".
- The program can display (after a fashion) animations now. This is useful for visualizing vibrational modes and molecular dynamics trajectories. More on this below.
- The file formats for the various input files viewkel uses are now documented. At least some of them are. They *should* be.
- The FMO plotting stuff can now plot simple energy level diagrams, and the filling of electrons can be non-*aufbau*. This is useful for ADF energy level diagrams.
- There are a pile of new utility programs for dealing with ADF and LMTO output. More on this below.
- MOs with f-orbitals can now be correctly displayed, i.e. f orbitals are now supported in viewkel, just like they are in bind.

- Ability to display fatbands plots. Currently this is not supported by bind, but it works just fine with output from LMTO programs.
- Ability to deal with spin-polarized DOS data (i.e. the traditional representation where the α and β spins are drawn as positive and negative values).
- Ability to individually toggle the shading of curves in properties (DOS and COOP) plots (shift-right-click in the button).
- viewkel can now dump files containing the grids evaluated when doing a contour plot of an MO. These can be worked with in either gnuplot or in viewkel itself.
- The stuff for dealing with contour plots (i.e. reading in planes of data) now works a whole heck of a lot better and is more general. Specifically, it can now read in the grid files which come out of viewkel or anything else which fits this (hopefully documented) file format.
- Hitting the `Delete` key will now delete the selected object. Due to the author hating such things, it **does not ask you first**, so be careful.
- Changes were made to the "Save All" routine to make it work better and support a couple more object types.
- Labels added for bond lengths/angles/etc. can now be translated around on screen to put them where you want them. The lines connecting them to atoms can also be toggled on and off (by hitting `a` with an label selected). The lines connecting them to atoms also now stop before they cut the label. This makes stuff look better.
- One can add text labels to go with graphs by hitting `Tab` in the main window. These labels can use all the PS tricks that graph labels and titles can (superscripts, subscripts, etc.).

Known Problems

Here we go... This is the list of stuff I know of that is either wrong or that behaves in "unexpected" ways. I've included work-arounds when I know of them.

- If you customize bonds and/or atoms in a crystal, then grow the crystal, there is no guarantee that everything will work the way you expect. With atoms everything *should* (famous last words) be okay, but bonds get all fouled up.
  Work-around: don't customize bonds until you have a crystal slice you are happy with.

Utility Programs

parse_it

parse_it, the program for pulling output from ADF TAPE21 files and producing data that viewkel can read in order to show MO plots has been updated. It now supports unrestricted as well as restricted mode calculations.

In addition, parse_it can be compiled to produce a program called cohp_it. This program, which is not part of the binary distributions for reasons which will become obvious, carries out a crystal orbital Hamilton population (COHP) analysis on the results of the ADF calculation (really it should be called MOHP, but that just sounds stupid). **BUT...** this is not fully tested and there is good reason to believe that it may not be correct. If you want to give this a spin, please contact me so that I can let you know where the pitfalls are. Just to make things as clear as possible: if parse_it and cohp_it didn't share the same source file, **I would not be distributing** cohp_it. You have been warned.

Scripts

I've written a bunch of new utility programs for parsing output from ADF and TB-LMTO-ASA. There are even one or two for Gaussian. These utilities add significantly to the functionality of viewkel. There is a small "problem" with them though. Most of these programs are written in awk, not C. This means that they won't work on the Macintosh, unless you can find a version of awk which works on the Mac. I have so far been unsuccessful in finding such a thing, but I haven't looked very hard. The other possibility is to convert the awk into perl (perl works just find on the Mac). I've done this, so there's a perl (.pl) version of each of the awk scripts, but I haven't really tested these. Assuming you know perl (I don't really), it's probably easy to fix these. If someone does so, please let me know.

Anyway, here's a list of the awk scripts and what they do. The general way these are used is: `awk -f awkfile outfile` where `awkfile` is the script you want to use and `outfile` is the name of the output file you want to extract information from.

- LT_to_moov.awk: Reads the output of an ADF linear transit run and constructs a `.MOOV` file for viewkel, which can then display the linear transit as an animation. Produces file `outfile.moov`.
- LT_to_walsh.awk: Reads the output of an ADF linear transit run and constructs a `.WALSH` file for viewkel, which can then display the evolution of the MO energies along the linear transit. Output is to stdout, so you should do: `awk -f LT_to_walsh.awk outfile > outfile.WALSH` or something like that.
- ICSD_to_b.awk: reads a `.CIF` file from the ICSD and produces a bind input file. An optional second argument can be used to specify a particular database code in order to pull a single entry from a `.CIF` file containing many.
- add_MOs.awk: takes two arguments. These should be grid files generated by viewkel. The two are added to each other and a third grid file is produced containing all three curves.

- sub_MOs.awk: takes two arguments. These should be grid files generated by viewkel. The two are subtracted from each other and a third grid file is produced containing all three curves.

- construct_fmo.awk: produces a `.FMO` file from an ADF output file which has the energy levels of the fragments and the final molecule. Lines are not drawn connecting the FMOs to the MOs.

- extract_freqs.awk: produces a `.vibn` file which can be read by viewkel for the old style vibration plots. This is a simpler replacement for the old C program which did the same thing.

- extract_geom.awk: constructs bind input files from the output of ADF geometry optimization or linear transit calculations.

- extract_orb_E.awk: constructs a `.LEV` file from an ADF output file (single point or geometry optimization). This can be read in by viewkel in order to display the energy level diagram for the molecule. This works with unrestricted and restricted calculations.

- freq_to_moov.awk: reads in a `.vibn` file and generates a file called `the.moov` which can be read into viewkel to display the results of an ADF frequency calculation as an animation.

- g94_to_v.awk: reads the final geometry from a g94 geometry optimization and produces a file viewkel can read to show the geometry.

- opt_to_moov.awk: reads geometries out of an ADF logfile and produces a .moov file which viewkel can read. This is useful for keeping an eye on ADF as it optimizes the geometry. This might, possibly, work for a lineary transit run as well.

- prop_shift.awk: shifts the energies in a .DOS or .COOP curve so that $\varepsilon_F \equiv 0.0$. This is useful when comparing eHT results to LMTO results.

LMTO utilities

I have written some tools which make the LMTO package produce output which can be directly read by viewkel. However, because they are based upon modifications to the LMTO source code (instead of being written from scratch), I am not distributing them. If you want to use viewkel with LMTO, feel free to contact me and I will send you patches so that you can make things work.

When I have time, I will contact the folks down in Stuttgart and find out about either getting permission to distribute these myself or having them integrated into the LMTO distribution.

A View into the Author's Mind

One of the "user transparent" changes in the new versions of both bind and viewkel is the inclusion of a revision history in each file so that we can keep track of who has been mucking around with what. As part of putting together this release, I went through the revision history of each file in viewkel. Along the way, I found some amusing entries which provide a fairly good portrayal of what was going through my mind as this version evolved. A selection of highlights is presented here, for your reading pleasure.

```
---
interface.c
   04.09.98 gL:
     added "support" for click and drag selection.
     This is implemented in an unpleasing fashion, but it
     works and I need to do my research proposals.

   17.02.99 gL:
     added proportionate scaling of contour plots.

   18.02.99 gL:
     fixed said functionality so that it's actually possible to
     get into scale mode with a contour plot (duh).

   06.04.99 gL:
     fixed stupid core dumping bug when the window was clicked in
     without an active object.  Ahem!

---
many files
  26.09.98 gL:
     various modifications to remove warning when compiled with -Wall
     under gcc (yeah yeah... it's anal)
---
graphics.c
   18.10.98 gL:
     modifications to filled_polygon and open_polygon routines to allow
     them to deal with things with more than 3 sides.  This was added as
     part of the fatbands stuff, but it should generalize them significantly.
     NOTE: the Mac version of the filled polygon thing still only deals with
     three sides... this needs to be updated.  The tektronix stuff has not
     been touched at all... not that that matters.  Secretly, g_open_polygon
     has been broken in X mode for as long as it has existed... hahah.
     luckily it has never been used in a situation where this would be
     noticeable.
```

```
---
band_graphs.c
  18.10.98 gL:
     support for fatbands added.  draw_band_graph still needs to be modified
     a little bit to be able to deal with multiple fatbands plots in one
     file.  Uh, not that there is currently any software which generates such
     a data file... there will be though, yes there will... you'll see.
---
fileio.c
   28.01.99 gL:
     added general_read, which allows only a single file read button
     to appear in the main options window.  yay.  (secretly, this is
     all part of the plan to make the Mac port work better and to
     facilitate the appearance of a Windoze port)
---
graphs.c
  11.05.99 gL:
     added sarcastic comment to find_tic_sep
---
molecule.c
   22.02.98 gL:
     modifications to determine_connections:
      cleaned up memory allocation, extra bond insertion at beginning
      added persistence of custom bonds
   30.03.98 gL:
     modifications to determine_connections:
      patched up at least some of the horrible bugs the aforementioned
      changes wrought with animations.  what *was* I thinking?
   06.04.98 gL:
     more mods to the bug farm that is determine_connections:
      sooprise sooprise sooprise, that modification was still completely
      screwed up.  This time it worked for animations, but not for crystals.
      things seem to be okay now.   It still gets broken sometimes
      when crystals are grown with custom bonds, but I haven't tracked
      that little taste of evil down yet... at least it doesn't crash.
      I suspect it has something to do with atoms being renumbered.
      This probably needs to be changed.
   19.05.98 gL:
     YA boundary condition fixed in determine_connections.
   30.05.98 gL:
     determine_connections is still screwed up when custom lines
     are turned on.  Work around is simple: don't customize lines until
     after you're done finding them.
     Will the bugs in this function never end?  A smart
```

```
    boy would just give up and rewrite the damn thing entirely, then
    mabye it would be easier to write the code to allow individual
    lines to be added.
  21.05.1999 gL:
    some of the problems in determine_connectors with respect to
     custom bonds seem to have been due to a stupid oversight on
     my part.  This appears to have been remedied.  In theory
     that stuff should all work now.  Also, in theory, I've now
     grokked enough of the code (again) to be able to add bonds.
     We'll see if that actually works.

    Frighteningly, that actually seemed to work.  So, now, at last,
     the oft-requested add-a-bond(TM) feature seems to function.
     Like I won't find a crashing bug two days after the release,
     but for now I get to WOO HOO!

    Also: the revision history is now Y2K compliant.
---
```

The Future of viewkel

Aside from bug fixes, this is very, very likely to be the last iteration of this particular program that I release. There will still be something called viewkel, and it will still do the same things, but it will be written in a different language and will run on Unix, the Mac, and Windoze (I don't like it, but everybody is using it).

The reason I'm doing this is quite simple: viewkel has gotten too big, too old (the basic parts of the program, those that deal with 3D objects, date back to a class programming project I did in 1990), too crufty, and the user interface is a piece of junk (to put it mildly). I have too many times gone to add a new feature that I want/need and discovered that it would be too difficult to do so because either: (a) I can't figure the code out anymore, (b) the code is a nightmare of cruft (see the entries dealing with `determine_connections` in the revision history above), or (c) there's no reasonable way to add a user-interface element that allows access to the new feature. I worked on this for so long because it was fun and I could always add stuff that I wanted. That has ceased to be the case.

So... I'm starting over. The new version is coming along nicely. It should be ready for beta testing by the Fall of 1999... send me mail if you want to give it a try.