

Steering Chemistry in the Right Direction!

R. A. Davies[†]

School of Chemistry, University of Wales, Bangor, Bangor, Gwynedd, LL57 2UW, UK

Abstract

The development of novel molecular visualization and computational steering applications is described in this paper. Improvements to the Molecular Visualizer application, jointly developed between the schools of Chemistry and Informatics are firstly described. The effectiveness of an alternative approach utilising Perl scripting capabilities within the recent commercial release of Materials Studio 4.1 (Accelrys Plc.) will be demonstrated using specific examples. The development of simple, novel, proof-of-concept computational steering applications are also described.

Categories and Subject Descriptors (according to ACM CCS): J.2 [Physical Sciences and Engineering]: Chemistry

1. Introduction

Chemists rely heavily on visualization as a key to understanding chemical processes. Geometrical structures are described in terms of their equilibrium structures (e.g. Lewis), and reactions are visually represented as a scheme. This pictorial approach, incorporating simple perceptions of atom connectivity, bonding (e.g. ionic, covalent, dative, and multi-centre "banana" bonds), bond order (e.g. single, aromatic, double, triple etc.), aromaticity (Kekulé vs resonant forms) and chirality, has allowed the chemist to gain great insight into complicated chemical processes that would otherwise impede scientific progress.

Dreiding and Corey-Pauling-Koltun (CPK) mechanical models have been routinely used in the form of wooden, metallic or plastic model kits, to conceptualize 3D molecular structures for several decades, e.g. [WC53] and myoglobin [KBD*58]. Plastic kits offer cheapness and portability, immediate tactile and visual messaging, and easy user manipulation and interactivity (e.g. conformational analysis, substitution / addition / elimination reactions and pseudorotation).

Molecular graphics has enhanced the accessibility of molecular modelling techniques, and assisted in the analysis and interpretation of the underlying theoretical calculations, e.g. measurements of simple geometric properties, or the vi-

ualization of orbital or electrostatic potential isosurfaces. Molecular graphics are commonly represented using 'stick' or 'space-filling' representations, analogous to the Dreiding and CPK mechanical models, together with other subtle variations on these two basis types such as 'tube' and 'balls and stick'. Inclusion of shading and lighting effects, coupled with an element-specific colour scheme, provide improved visual perception. For large enzymes and proteins, the secondary structure is best represented by a 'ribbon' representation [CLZL06], where the backbone is depicted as a set of cylinders (α -helices), flat arrows (β -sheets), or tubes (no regular structure).

Improved perception of a virtual molecule can be obtained via cooperative deconvolution and interpretation of multiple cues – synaesthesia [Dur05]. Optical, ultrasonic or magnetic tracking allows not only an improved sense of immersion, but improved spatial object manipulation, whilst tactile or force-feedback cues provide additional sensory information, strengthening the links between the quantum and macroscopic worlds, e.g. for molecular docking [OyPH*88].

Rapid-prototyping can generate accurate, physical molecular models from either virtual data (e.g. VRML / X3D or .STL formats) or from atomic coordinates (e.g. experimental or simulated .pdb files). Powder-based models allow atom-specific colours, whereas polymer models are generally more robust and expensive. Printed models can be tracked and spatially registered within augmented reality (AR) applications [SBO*95] [GSS006]. Real-world user

[†] e-mail: chs001@bangor.ac.uk

and physical model presences are combined with stereoscopic rendering of models and data, generating an illusion of tangibility by leveraging the natural semantics of physical object manipulation, e.g. via force-feedback haptics.

2. Molecular Visualizer (MV)

A prototype Molecular Visualiser (MV) has been developed as part of a multidisciplinary collaboration between the Schools of Chemistry and Informatics [DJMH05]. Although a wide range of commercial and academic molecular visualization packages exist (e.g. RasMol, PyMol, JMol etc.), MV and the other solutions described in this paper are cost effective and offer support for both immersive (e.g. via a Powerwall) and non-immersive (e.g. via a standard web-based interface) visualization (VRML 97 / X3D) and potential haptic interaction (Reachin API / Open Haptics Toolkit / H3D) via Web 3D technology. The educational benefits including knowledge-building experiences and the ability to analyze phenomena from different points of view will help students to efficiently learn and understand key concepts, e.g. symmetry, stereochemistry, vibrations or molecular dynamics (MD).

Chemical information can be described by an astonishing array of widely differing file formats, ranging from simple ASCII descriptions, e.g. SMILES, .xyz, .mol and .pdb etc. to XML-markups, e.g. CML (Chemical Markup Language). The expressiveness and extensibility of CML allows molecular data to be encoded and efficiently converted to X3D using an appropriate XML / XSLT style sheet [Pol03], for display within Information-Rich Virtual Environments.



Figure 1: Molecular Visualiser. A stereoscopic projection of the molecule is co-located with the haptic device.

MV parses simple molecular descriptions (e.g. from a .xyz file) within a Cartesian framework, calculates bonding, and generates a VRML 97 scene graph for visualization. For haptic interaction, a Reachin Display can be used which co-locates stereoscopic renderings with a PHANTom Desktop haptic device as shown in Figure 1. MV assigns appropriate haptic properties to the molecule, e.g. a *SimpleSurface* node

allowing the user to feel the spherical CPK atoms. An *Inertia* node describing the mass distribution, in conjunction with a *Dynamic* node, which transforms the local coordinate space, allows molecular rotation and translation via application of forces, momenta and torques.

2.1. Keyframe animation

An .arc file describes a sequence of structures or frames within a molecular dynamics trajectory or vibrational normal mode. The structure of the .arc file varies slightly for helical, periodic and helical systems. For simplicity, only non-periodic, non-helical systems were considered at this stage. The .arc file contains the following information (**lines** and **columns**):

1st : header

2nd : periodic boundary conditions (PBC) choice

3rd : title(1–64), energy (65–80)

4th : !DATE day month date time year

5th : optional PBC information

6th–Nth : (repeated for each molecule) Atom name(1–5), x coordinate (7–20), y coordinate (22–35), z coordinate (37–50), type of residue (52–55), residue sequence (57–63), potential type (64–70), element symbol (72–73), partial charge (75–80).

Final line for a given molecule: end

Final line for entire molecular system: end

Tokenization of the atomic data allows parsing of the Cartesian coordinates and unique IUPAC element symbol. The trajectory is described as a series of nested Vector objects, allowing efficient data storage:

Trajectory → **Frames** → **Molecules** → **Atoms** → **Data**
(e.g. mass, colour, coordinates etc.)

A *TimeSensor* node drives the keyframe animation, in which unique *Transform* and *PositionInterpolator* nodes are assigned to each atom. The **key_value** field of the interpolator is an array of coordinates representing the trajectory of each atom, whilst the **key** field represents an array of fractional times from 0 to 1, whose step size depends on the number of frames within the trajectory, e.g. 0.01 for 100 frames. An *SFFloat fraction_changed* eventOut is routed from the *TimeSensor* into the *SFFloat set_fraction* eventIn of each *PositionInterpolator*. Linear interpolation yields a new set of coordinates which are outputted as *SFVec3f value_changed* eventOuts and routed to specific *SFVec3f translation* eventOuts. The trajectory within the .arc file always begins at frame 1, necessitating the additional parsing of an .xsd file containing the initial (frame 0) molecular conformation.

2.2. Vibrational normal modes

The following example is routinely employed within the Molecular Modelling and 3D Visualization course run by the School of Chemistry. The course augments practical molecular modelling experience gained using the commercial Materials Studio package, with the creation of VRML molecular scene graphs for visualization. The VRML scene graphs are aesthetically compared with those generated using Materials Studio (OpenGL).

For an N atom system, $3N$ harmonic vibrational frequencies may be obtained by diagonalization of the mass-weighted Hessian matrix (\mathbf{F}), whose elements are mass-weighted second derivatives of the total energy, E . Of these, $3N - 6$ ($3N - 5$ if linear, $3N - 3$ if periodic) correspond to the normal vibrational modes, and the remainder being translations and rotations of the entire system.

A water molecule was built using the sketching tools within Materials Studio, and minimised using Restricted Hartree-Fock (RHF) semi-empirical quantum mechanical-based methods. Figure 2 illustrates the normal modes of water, which were parsed from the NormalMode attribute of the Properties element associated of each atom:

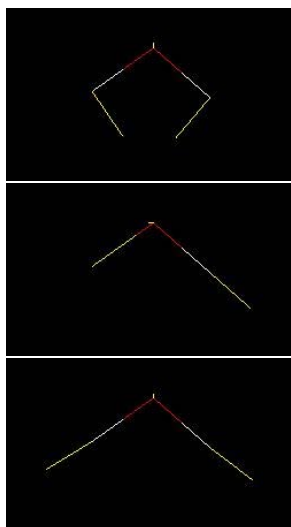


Figure 2: AM1 Vibrational modes in water: bending (top, 1885 cm^{-1}), asymmetric stretch (middle, 3505 cm^{-1}), symmetric stretch (bottom, 3585 cm^{-1}).

2.3. Transition state structures

Under the harmonic approximation, vibrational frequencies are given by the square roots of the eigenvalues of the mass-weighted Hessian matrix, \mathbf{F} , whilst the normal modes are eigenvectors of \mathbf{F} . Negative eigenvalues correspond to modes in which the energy is not a local minimum. For a

first-order transition state or saddle point, only one negative eigenvalue is obtained. Moving along the transition or reaction coordinate, the normal mode corresponding to the negative eigenvalue, in the two opposite senses leads either towards the reactants or products.

The [3,3]-sigmatropic Claisen rearrangement of allyl vinyl ether, represents a powerful tool for the synthesis of γ,δ -unsaturated aldehydes. The AM1-based reaction coordinate for the rearrangement of allyl vinyl ether to pent-4-enal is illustrated in Figure 3. For clarity, the radii of the atoms have been halved. The formation of the C-O bond (coupled with the breaking of the opposite C-C bond) can be clearly seen by the decreasing separation between the red oxygen and the grey carbon atom in the top of each model.

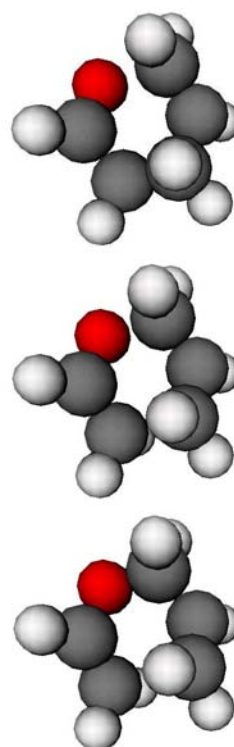


Figure 3: The Claisen rearrangement: Allyl vinyl ether (top), transition state (middle), pent-4-enal (bottom).

2.4. Periodic systems

Molecular descriptions typically use one of three main coordinate frameworks – **Cartesian**, **internal** (based upon bond lengths, bond angles, torsional angles and atom connectivities) and **fractional**.

Any solid or surface, whether crystalline or amorphous, ionic or molecular, perfect or defective, e.g. glasses, diamond, zeolites and catalysts, can be described by either 2-D

or 3-D periodically repeating unit cells, which when translated provides the full structural symmetry of the system. The symmetry properties are tabulated as 230 distinct crystallographic space groups (and their associated sub-groups) described by the Hermann-Mauguin (International) System notation [Hah05]. 2-D surfaces are represented by 17 plane groups, whilst non-periodic 0D-systems are represented by either 32 crystallographic or an infinite number of non-crystallographic general point groups.

Within crystals (3D), atoms are defined relative to the lattice vectors \vec{a} , \vec{b} , \vec{c} , or more frequently by the unit cell lengths a , b , c , and angles α , β , γ . For 2D-surfaces, atoms are defined relative to \vec{u} and \vec{v} and a third vector \vec{w} , a unit vector normal to both \vec{u} and \vec{v} . An atom lying at (0.5, 0.5, 0.5) will therefore be directly in the middle of a 3D unit cell, or can be located 0.5 Å above the middle of a 2D unit cell.

Atoms are placed upon Wyckoff sites, which may be denoted as ‘special’ or ‘general’; the former indicating that one or more symmetry element is associated with it. Application of symmetry operators specific to the Wyckoff sites described by the asymmetric unit, the largest subset of atoms which can be chosen such that no two atoms are related to each other by a symmetry operator or by periodic translation, produces the full unit cell. Conversion from fractional (x_f , y_f , z_f) to Cartesian (x_c , y_c , z_c) coordinates is accomplished using a matrix similar to that shown below for a triclinic unit cell ($a \neq b \neq c$; $\alpha \neq \beta \neq \gamma$) [BAB*95]:

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} a & b \cos \gamma & c \cos \beta \\ 0 & b \sin \gamma & \frac{c(\cos \alpha - \cos \beta \cos \gamma)}{\sin \gamma} \\ 0 & 0 & V/(ab \sin \gamma) \end{pmatrix} \begin{pmatrix} x_f \\ y_f \\ z_f \end{pmatrix}$$

where:

$$V = abc(1 - \cos^2 \alpha - \cos^2 \beta - \cos^2 \gamma + 2 \cos \alpha \cos \beta \cos \gamma)^{1/2}$$

The contents of the periodically repeated cell are expressed within given intervals, the most common being [0, 1), although other intervals based upon translation, e.g. [-0.5, 0.5, 0.5) are equally as valid. The following methodology was employed:

- a) Convert **all** coordinate intervals to [0,1).
- b) For **each** atom / ion within the asymmetric unit:
 - i) Identify the Wyckoff site upon which it lies
 - ii) Calculate **all** Wyckoff positions within the unit cell [0, 1) via application of symmetry operations
- c) Apply repeats, n_a , n_b , n_c (integers ≥ 1) along the cell vectors, to obtain new intervals [0, n_a) etc.
- d) Extend atoms to **all** external edges, faces and corners, i.e. to [0, n_a) etc. using symmetry.
- e) Perform fractional \rightarrow Cartesian conversion

If step d) is performed prior to c), unnecessary duplication occurs at all edges, corners and faces.

Table 1: Wyckoff sites for cubic $Fm\bar{3}m$ space group.

Multiplicity	Wyckoff Site	Site Symmetry	Coordinates
192	l	1	(x, y, z), ...
96	k	$..m$	(x, x, z), ...
96	j	$m..$	($0, y, z$), ...
48	i	$m.m2$	($\frac{1}{2}, y, y$), ...
48	h	$m.m2$	($0, y, y$), ...
48	g	$2.mm$	($x, \frac{1}{4}, \frac{1}{4}$), ...
32	f	$.3m$	(x, x, x), ...
24	e	$4m.m$	($x, 0, 0$), ...
24	d	$.mm$	($0, \frac{1}{4}, \frac{1}{4}$), ...
8	c	$\bar{4}3m$	($\frac{1}{4}, \frac{1}{4}, \frac{1}{4}$) ($\frac{1}{4}, \frac{1}{4}, \frac{3}{4}$) ($\frac{3}{4}, \frac{1}{4}, \frac{1}{4}$)
4	b	$m\bar{3}m$	($\frac{1}{2}, \frac{1}{2}, \frac{1}{2}$)
4	a	$m\bar{3}m$	($0, 0, 0$)

Sodium chloride (Figure 4) belongs to the cubic space group $Fm\bar{3}m$, whose Wyckoff sites are summarised in Table 1. For each space group and its subgroups, the fractional coordinate for any ion is tested against all of its possible symmetry generated positions, starting with the highest symmetry (and lowest multiplicity) special Wyckoff site, e.g. an interstitial ion placed at ($\frac{3}{4}, \frac{1}{4}, \frac{1}{4}$) does not lie on an $4a$ or $4b$ Wyckoff site, but lies on $8c$. For the $Fm\bar{3}m$ space group, we can generate the fractional coordinates of **all** atoms in the unit cell using face centering generators together with the coordinates for each Wyckoff site. Therefore, a Na^+ cation placed at $4a$ (0, 0, 0) produces additional symmetry equivalents at ($\frac{1}{2}, \frac{1}{2}, 0$), ($\frac{1}{2}, 0, \frac{1}{2}$) and ($0, \frac{1}{2}, \frac{1}{2}$); whilst the Cl^- anion at $4b$ ($\frac{1}{2}, \frac{1}{2}, \frac{1}{2}$) generates additional symmetry equivalents at ($0, 0, \frac{1}{2}$), ($0, \frac{1}{2}, 0$), ($\frac{1}{2}, 0, 0$).

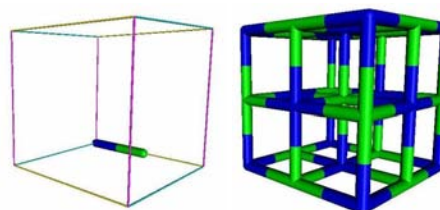


Figure 4: Sodium chloride: Asymmetric unit (left, NaCl), (b) $1 \times 1 \times 1$ conventional unit cell (right, Na_4Cl_4)

Although only selected space groups have been encoded due to the considerable effort involved, scene graphs generated for sodium chloride and other crystalline materials validate the above approach.

3. Perl Scripting within Materials Studio

Version 4.1 of Materials Studio, released in late 2006, introduced simple Perl scripting capabilities via the MaterialsScript API. The API currently focuses on a subset of the Visualizer functionality – the main chemistry model and its associated objects, e.g. the ability to create new atoms, bonds, geometry monitors, clean etc. Properties such as atomic coordinates, unit cell parameters, space group symmetry information, bonding data etc. can be accessed from the Properties Explorer, allowing efficient creation of molecular scene graphs. The hard-coding of specific algorithms into the MV parser (e.g. creation and rendering of bonds based on interatomic distance criteria etc.) will therefore become increasingly redundant.

3.1. Simple molecules

The following Perl script illustrates the flexibility and versatility of scripting within Materials Studio, e.g. re-centering, probing of atomic descriptors, e.g. colour, and coordinates using the Properties Explorer. At present only a basic CPK model has been completed as shown in Figure 5, although other representations reliant upon bonding descriptors, e.g. stick, ball-and-stick and cylinders are being developed.

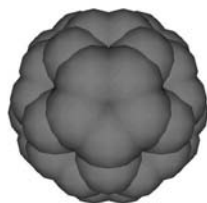


Figure 5: Buckminsterfullerene, C_{60}

```
#!/perl
use strict;
use MaterialsScript qw(:all);
my $doc = $Documents{"water.xsd"};
my $atoms = $doc->DisplayRange->Atoms;
my $string = ""; my $level = 0; my %def = ();
my $centroid = $doc->CreateCentroid($atoms);
$centroid->CentroidXYZ=(Point({X=>0,Y=>0,Z=>0}));
$centroid->Delete;
# VRML header ...
foreach my $atom (@$atoms) {
my ($red, $green, $blue) = atom_color($atom->Color);
my $element = $atom->ElementSymbol;
same("Transform");
same("");
right("translation ", $atom->X, " ", $atom->Y, " ",
$, $atom->Z);
same("children ");
same("[");
if (exists ($def{$element})) {
```

```
right("USE ", $element);}
else {
right("DEF ", $element, " Shape");
same("");
right("appearance Appearance");
same("");
right("material Material");
same("");
right("diffuseColor ", $red, " ", $green, " ", $blue);
left(")"); left(")");
same("geometry Sphere");
same("");
right("radius ", $atom->vdwRadius);
left(")"); left(")");
$def{$element}=1;}
left("]"); left("]");}
print $string;
sub atom_color {returns RGB atom colours}
sub right {adds 4 spaces}
sub left {removes 4 spaces}
sub same {prints $string with indents}
```

3.2. Periodic systems

As periodic cells can be infinitely extended, MaterialsScript API allows the user to filter and return either the contents of the asymmetric unit, unit cell, or the current display range, e.g. a $2 \times 2 \times 2$ repeat, as shown for sodium chloride (Figure 6). The scripting interface in particular allows the user to generate the scene graph without the need for hard-coding of any symmetry or space group data, unlike the current version of MV. Work currently in development include an investigation of the MaterialsScript API tools for the interconversion of primitive and conventional unit cells, and the rendering of unit cell vectors.

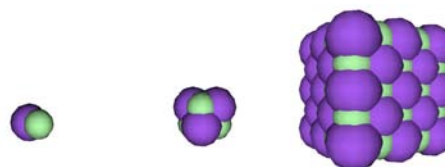


Figure 6: Sodium chloride: asymmetric unit (left), conventional cell (middle), $2 \times 2 \times 2$ repeat (right)

3.3. Molecular dynamics and vibrations

Modification of the Perl script described in Section 3.1 allows for the creation and export of a TimeSensor, Position Interpolator nodes, and all required VRML routing. The key-values for each atom are obtained by looping through each frame of the MD run. A section of the Perl script illustrating the parsing of the coordinates from each frame of the trajectory into corresponding PositionInterpolator nodes is shown below.

For an M frame trajectory with N atoms, $M \times N$ frames must be loaded in total, leading to high I/O overheads for larger systems, such as that shown below in Figure 7 (100 frames containing 84 atoms). Parsing of the trajectory from a single .arc file as described in Section 2.1 will dramatically increase I/O efficiency, but does not demonstrate the full capability of the MaterialsScript API. Nevertheless, the Perl script can be successfully employed to visualize molecular vibrations, changes in molecular geometry that occur upon geometry optimisation / energy minimisation and molecular dynamics trajectories.

```
foreach my $atom (@$atoms) {
    $i++;
    my $temp = "";
    same("DEF Move", $i, "PositionInterpolator");
    same("{}");
    right("key");
    same("{}");
    for (my $j = 0; $j <= $traj->NumFrames; $j++) {
        $temp = $temp.$j/$traj->NumFrames." ";
        right($temp);
        left("{}");
        same("keyValue");
        same("{}");
        $temp = $atoms->Item($i-1)->X." ".$atoms->Item($i-1)->Y." ".$atoms->Item($i-1)->Z." ";
        for (my $k = 1; $k <= $traj->NumFrames; $k++) {
            $traj->CurrentFrame = $k;
            my $strajatoms = $doc2->DisplayRange->Atoms;
            $temp = $temp.($strajatoms->Item($i-1)->X." ".$strajatoms->Item($i-1)->Y." ".$strajatoms->Item($i-1)->Z." ");
        }
        right($temp);
        $temp="";
        left("{}");
        left("{}");
    }
}
```

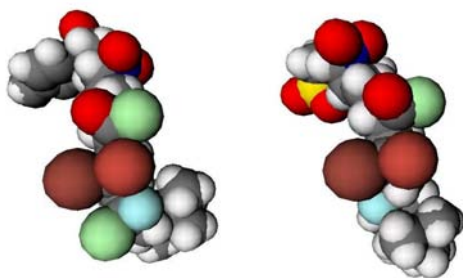


Figure 7: Snapshots of an MD trajectory: initial (left), mid-way (right)

4. Computational Steering

Computational steering refers to the process by which a scientist monitor and interacts with a running application, such as a molecular dynamics or molecular docking simulation. For large, complex systems such as molecular dynamics or molecular docking simulations, the chemist will often need to couple predictive modelling results and "on-line" visualization via steering middleware in order to navigate and explore results such as the evolution of the simulation, and to make informed steering decisions. The complexity of this visualisation can be tailored to suit available hardware (e.g. CAVE, powerwall, high-end workstation, laptop, PDA etc.).

The "human in the loop" aspect of computational steering is more flexible and efficient than the traditional non-interactive approach to simulations, in which text files describing the initial conditions and parameters for the course of the simulation are prepared, and the simulation submitted to a batch queue, to wait for sufficient resources to proceed. The simulation then proceeds entirely according to the prepared input file, outputting the results to disk for later examination. Checkpointing enables restart of the traditional calculation from a previous snapshot, e.g. in case of machine failure, or bad parameterisation.

The following examples demonstrate how computational steering principles can be applied to a number of simple scientific problems. In each example, a VRML scene graph consisting of sliders and / or buttons allows the user to visualize in real-time, the affect of steering the simulation by manipulation of various input parameters. These proof-of-concept studies illustrate approaches which may be employed to create custom molecular dynamics or docking simulations within the School of Chemistry.

4.1. Simple Newtonian Mechanics

Undergraduate students within the School of Chemistry attempt to steer the trajectory of a moving box via a slider as shown in Figure 8.

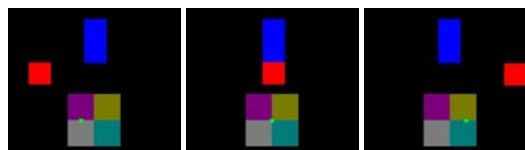


Figure 8: Steering of the red box: deceleration (left), constant velocity (middle), acceleration (right)

The motion of the larger blue cuboid is determined using a PositionInterpolator node, which linearly interpolates the motion, i.e. the object has constant velocity. The displacement, \vec{x} , of the red cube under a constant acceleration, \vec{a} , is given by:

$$\vec{x} = \vec{x}_0 + \vec{v}_0 t + \frac{1}{2} \vec{a} t^2$$

where \vec{x}_0 and \vec{v}_0 are the initial position and velocity. The *SFVec3f translation_changed* eventOut of the slider is routed into the *SFVec3f set_acceleration* eventIn of the *Script* node, allowing the **acceleration** field within the *Script* node to be interactively modified.

```
DEF acceleration Script {
  eventIn SFFloat set_fraction
  eventIn SFVec3f set_acceleration
  eventOut SFVec3f position
  field SFVec3f x0 -5 -2 0
  field SFVec3f v0 10 5 0
  field SFVec3f a 0 0 0
  url "javascript:
  function set_acceleration(value) {
  for (i = 0; i < 3; i++){
  a[i] = 20 * value[i];} function set_fraction(t){
  for (i = 0; i < 3; i++){
  position[i] = x0[i] + v0[i]*t + 0.5*a[i]*t*t;}"
```

The above application will be enhanced by using leapfrog, Euler or Verlet integration algorithms, which integrate the equation of motion for a series of particles under the influence of varying forces or accelerations within molecular dynamics simulations.

4.2. Particle-in-a-box

One of the simplest quantum mechanical models emphasizing quantisation is the particle in a 1-D box. The potential $U(x)$ is assumed to be zero inside the box of length L , and is infinite outside the box. The normalised wavefunction becomes:

$$\psi_n = \sqrt{\frac{2}{L}} \sin \frac{n\pi x}{L}$$

The probability of finding the particle between two points ($0 \leq aL \leq bL \leq L$) within the 1-D box can be calculated as:

$$\int_{aL}^{bL} \psi^2 d\tau = (b-a) + \frac{1}{2n\pi} [\sin 2n\pi a - \sin 2n\pi b]$$

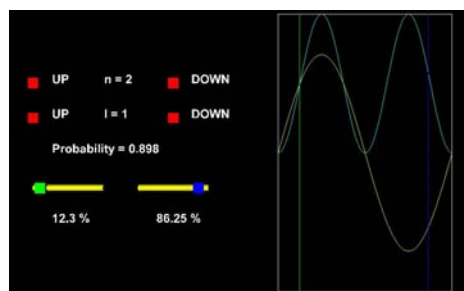


Figure 9: Simulation of a particle in a 1-D box. The wavefunction, ψ , and its probability density, ψ^2 are shown in yellow and cyan respectively. The integration limits are described by the green and blue lines and sliders.

Several Script nodes within the VRML scene graph (Figure 9) allow the user to steer the simulation of the particle in a 1-D box by varying the quantum number, n , the length of the box, L , and the two integration limits using the sliders. Upon any variation of the above four parameters, the scene graph automatically updates all graphical and textual information. The above methodology will be applied to 2-D and 3-D boxes, producing 3-D and 4-D plots respectively. Abstraction of the 4-D plot to a 3-D representation can be accomplished by application of a colour spectrum, without loss of essential information.

4.3. Molecular steering

One of the key requirements for molecular dynamics or molecular docking simulations is the description of the potential energy hypersurface on which the atomic nuclei move using a forcefield. Within a typical forcefield, the potential energy of the system can be expressed as a sum of valence (or bond), crossterm, and non-bond interactions, which are themselves comprised of different analytical expressions. A range of homo- and hetero-nuclear diatomic molecules such as H_2 or HF have been employed in this initial study, as the forcefield can be approximated using a simple CVFF harmonic function, $E = k(r - r_0)^2$, where k is the spring constant and r_0 is the reference bond [DORO*88]. The steering approach shown below (Figure 10) allows the user to interactively probe the harmonic potential energy function. The graphical approach also allows rapid appreciation of key concepts such reactivity and bond strength e.g. $N \equiv N$, $O = O$ and $H-H$; or the shallow potential for iodine.

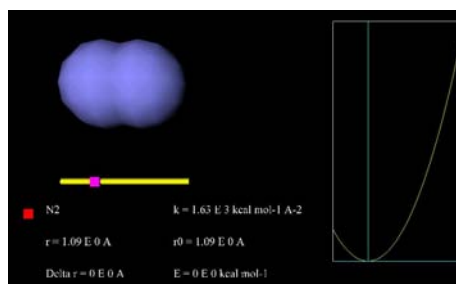


Figure 10: Harmonic approximation for the triple bond within the nitrogen molecule.

5. Summary

The development of novel molecular visualization and computational steering applications is crucial to rapid comprehension of key concepts, such as molecular dynamics, molecular docking, solid state structures etc. Although the subset of applications described above are pilot, proof-of-concept studies, the results bode well for future development, e.g. the development of novel steered molecular docking or minimisation tools within a virtual reality environ-

ment. The Scripting capabilities in Materials Studio currently show great potential in its ability to create novel, interactive, X3D and VRML-based scene graphs for molecular visualization. Enhancements such as specific prototype nodes, e.g. orbitals, multiple bonds, polymers, dendrimers and surfaces are planned.

fabrication of accurate molecular models. *J. Mol. Graph* 13, 2 (1995), 126–135.

[WC53] WATSON J. D., CRICK F. H. C.: Molecular structure of nucleic acids. *Nature* 171, 4356 (1953), 737–738.

References

- [BAB*95] BEURSKENS P. T., ADMIRAAL G., BEURSKENS G., BOSMAN W. P., DE GELDER R., ISRAEL R., SMITS J. M. M.: *DIRDIF-94 Manual*. University of Nijmegen, 1995.
- [CLZL06] CAI Y. Y., LU B., ZHENG J., LI L.: Immersive protein gaming for bio edutainment. *Simulation Gaming* 37 (2006), 466–475.
- [DJMH05] DAVIES R. A., JOHN N. W., MACDONALD J. N., HUGHES K. H.: Visualization of molecular quantum dynamics – a molecular visualization tool with integrated web3d and haptics. In *Proceedings of the 10th International Conference on Web3D Technology* (2005), pp. 143–150.
- [DORO*88] DAUBER-OSGUTHORPE P., ROBERTS V. A., OSGUTHORPE D. J., WOLFF J., GENEST M., HAGLER A. T.: Structure and energetics of ligand binding to proteins: E. coli dihydrofolate reductase-trimethoprim, a drug-receptor system. *Proteins: Struct., Function Genetics* 4 (1988), 31–47.
- [Dur05] DURIE B.: Doors of perception. *New Scientist* 185, 2484 (2005), 34–36.
- [GSSO06] GILLET A., SANNER M., STOFFLER D., OLSON A.: Immersive protein gaming for bio edutainment. *Simulation Gaming* 37 (2006), 466–475.
- [Hah05] HAHN T. (Ed.): *International Tables for Crystallography, Volume A: Space-group symmetry*. International Union of Crystallography, Springer, 2005.
- [KBD*58] KENDREW J. C., BODO G., DINTZIS H. M., PARRISH R. G., WYCKOFF H., PHILLIPS D. C.: A three-dimensional model of the myoglobin molecule obtained by x-ray analysis. *Nature* 181, 4610 (1958), 662–666.
- [OyPH*88] OUH-YONG M. M., PIQUE M. E., HUGHES J., SRINIVASAN N., BROOKS F. P.: Using a manipulator for force display in molecular docking. In *Proc. IEEE Robotics and Automation Conference* (1988), vol. 3, pp. 1824–1829.
- [Pol03] POLYS N.: Stylesheet transformations for interactive visualization: Towards a web3d chemistry curricula. In *Proceedings of the 8th International Conference on Web3D Technology* (2003), pp. 85–ff.
- [SBO*95] SKAWINSKI W. J., BUSANIC T. J., OFSIEVICH A. D., VENANZI T. J., LUZHKOV V. B., VENANZI C. A.: The application of stereolithography to the