MASTER OF PHILOSOPHY, Modelling of Materials

Thursday 24 April 2003 **9 – 12**

MODELLING OF MATERIALS (1) – possible answers

SECTION A

1(a). A lattice is an imaginary array of points, each of which has the same environment and is related to the others by translational symmetry. A motif is an atom or a group of atoms that is associated with each lattice point. Placing a motif at each lattice point generates the crystal structure.

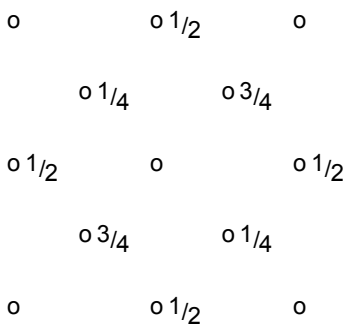

Fig.1. Projection of the diamond crystal structure along the z axis.

1(b). Metals: Anything which absorbs energy: lattice vibrations (phonons), magnetic order-disorder, electrons. However, electrons contribute very little since the Pauli exclusion principle prevents all but the ones located about kT of the Fermi level from participating in the energy absorption process (Note: the electronic contribution may be significant at low temperatures). There are extra degrees of freedom for polymers: chain curling and uncurling, molecular rotations and other vibration modes.

1(c). *Deterministic* computer models are those in which the future state of the system is completely determined by the initial conditions, for example in microcanonical (NVE) molecular dynamics. *Stochastic* models involve the use of random numbers, and thus the future state of the system is inherently unpredictable given the same initial conditions, for example in the Metropolis Monte Carlo method. Stochastic methods are usually distinguished by their reliance on a good quality source of random numbers and the fact that thermodynamic averages are constructed over configurational states uncorrelated in time. By contrast, deterministic methods do not need random numbers, and give thermodynamic averages over time. They also give rigorous information about dynamical processes in the system.

1(d). Change in density or electrical resistance during a phase transition (e.g. for studying pure systems); cooling curves (changes in cooling rates at a phase boundary); lattice parameters (using X-ray, neutron or electron diffraction); metallography; determination of composition using microanalytical techniques (EDX, WDX, EELS, etc.); thermodynamic data (heat capacity, enthalpy changes, measured using calorimetry, or DTA).

1(e). Common blocks are effectively an implicit mechanism for passing parameters. Passing data using parameters makes explicit which values are being passed, and which variables might be modified, at the point of the call; this self-documentation can help avoid error. Parameters are particularly well-suited to FORTRAN functions which mimic mathematical functions.

Unfortunately, if the parameter mechanism alone is used in any program of reasonable size, the number of parameters quickly becomes unwieldy; if the main program calls subprogram "X", which in turn calls subprogram "Y", and "Y" needs access to variables in the main program, these variables must be passed to, and declared within, "X" even though it may not use them. A FORTRAN compiler cannot check the numbers and types of parameters in programs which are

separately compiled, but mistakes in long parameter lists are actually very easy to make. Such errors are likely to produce incorrect results, rather than causing the program to crash; and can take a very long time to detect.

The types and sizes of the common blocks must be consistent in each subprogram, just as the types and numbers of parameters must match in both the called and calling code. (Exceptions to this rule for common blocks can hardly be regarded as modern good practice.) The great advantage of common blocks over parameter lists, however, is that a common block and its associated declarations can be duplicated easily, for example by using cut-and-paste with an editor, or use of 'include files'. This makes it much more likely that the common blocks will be correct, and much reduces the possibility of error. On the other hand, a subprogram (or even another subprogram called from it) can modify a variable without this being apparent from the call. This can again cause subtle and hard-to-find errors.

1(f). Consider a couple made from *A* and *B*. If the diffusion fluxes of the two elements are different ($|J_A| > |J_B|$) then there will be a net flow of matter past the inert markers, causing the couple to shift bodily relative to the markers. This can only happen if diffusion is by a vacancy mechanism. Place exchange mechanisms (such as ring diffusion) cannot produce a net flow of matter.

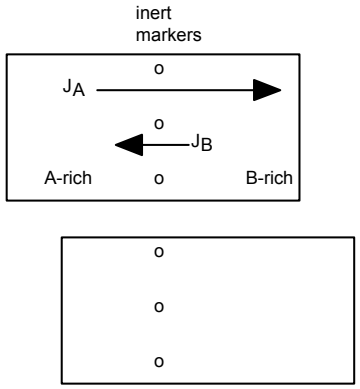

Fig.2. Diffusion couple with markers

1(g). The Metropolis algorithm can be summarised as follows:

1. Start with system in arbitrarily chosen state μ and evaluate the internal energy E_μ
2. Generate a new state ν by a small ergodic perturbation to state μ and evaluate E_υ
3. If $E_\upsilon - E_\mu < 0$, then accept the new state. If $E_\upsilon - E_\mu > 0$, then accept the new state with probability $\exp[-\beta(E_\upsilon - E_\mu)]$
4. Return to step 2 and repeat until equilibrium is achieved (i.e. thermodynamic state function of interest has converged to steady-state value when averaged over all configurations)

Because the Metropolis algorithm conserves particle number, volume and temperature, it can be applied to any atomistic system that obeys those constraints (i.e. constant volume and in thermal equilibrium with a heat reservoir). Examples include ideal gas or Lennard-Jones fluid, Ising magnet, or polymeric system. The method should not be applied to systems in which there is a change of state (e.g. from solid to liquid) or where the chemical potential is constant (e.g. osmotic equilibrium).

1(h). The purpose of coarse-graining is to remove as many degrees of freedom as possible from an atomistic system, whilst preserving certain universal aspects of its behaviour, in order to facilitate computer simulation over much larger length scales and longer time scales than would be possible by considering the whole system. As well as saving large amounts of CPU time, this procedure also reduces the amount of data produced to manageable levels by getting rid of any irrelevant detail. Of course, what is irrelevant depends on what information is required from the simulation, and considerable care must be taken to ensure that the coarse-grained model is not oversimplified. For example, in polymeric systems, a simple random-walk model is sufficient to reproduce the correct scaling relation (universal feature) for the end-to-end distance as a function of chain length for polymers in a melt or theta-solvent, but is unable to predict chain stiffness. There are hence many different levels of coarse-graining that can be applied to an atomistic system, and there is usually a compromise between amount of detail retained and level of precision obtained.

1(i). Liquid crystalline polymers (LCPs) contain rigid functional groups along their chain, giving rise to orientationally ordered mesophases which resemble those of small molecule liquid crystals. This makes the chain orientation very

sensitive to extensional strain fields during processing, allowing a much greater degree of chain alignment than in conventional polymers. Since polymers are very much stiffer in the axial direction than in the transverse direction, this leads to better mechanical properties in the finished article. Furthermore, LCPs give rise to lower die swell (caused by elastic recovery of chain configuration) as the degree of orientation is generally more uniform than in conventional polymers. Finally, LCPs tend to exhibit lower viscosity at high strain rates due to phenomena such as log-rolling and tumbling, which aids their mould-filling characteristics. Disadvantages are their high cost and poor mechanical performance at weld-line interfaces.

1(j). The atomic arrangement in a crystal can be altered either by breaking all the bonds and rearranging the atoms into an alternative pattern (*reconstructive* transformation), or by homogeneously deforming the original pattern into a new crystal structure (*displacive* transformation).

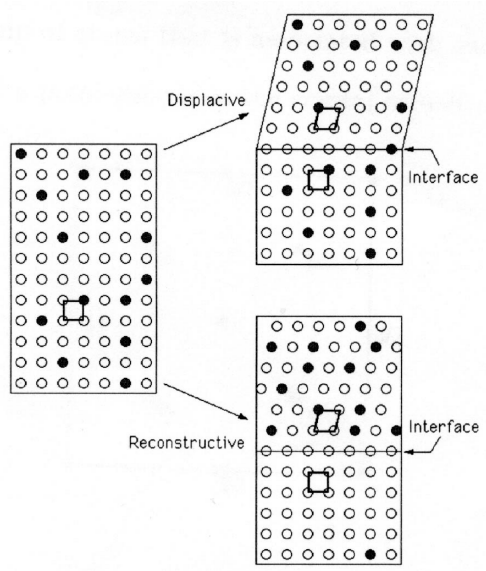

Fig.3. The main mechanisms of transformation. The parent crystal contains two kinds of atoms. The figures on the right represent partially transformed samples with the parent and product unit cells outlined in bold. The transformations are unconstrained in this illustration.

In the displacive mechanism the change in crystal structure also alters the macroscopic shape of the sample when the latter is not constrained. The shape deformation during constrained transformation is accommodated by a combination of elastic and plastic strains in the surrounding matrix. The atoms are displaced into their new positions in a co-ordinated motion. Solutes may be forced into the product phase, a phenomenon known as solute trapping.

It is the diffusion of atoms that leads to the new crystal structure during a reconstructive transformation. The flow of matter is sufficient to avoid any shear components of the shape deformation, leaving only the effects of volume change. There may be a composition change during a reconstructive transformation.

SECTION B

2. To avoid overfitting, the experimental data are divided into two sets, a *training* dataset and a *test* dataset. The model is produced using only the training data. The test data are then used to check that the model behaves itself when presented with previously unseen data. This is illustrated in Fig.4, which shows three attempts at modelling noisy data for a case where y should vary with x^3. A linear model (a) is too simple and does not capture the real complexity in the data. An overcomplex function such as that illustrated in (c) accurately models the training data but generalises badly. The optimum model is illustrated in (b). The training and test errors are shown schematically in (d); not surprisingly, the training error tends to decrease continuously as the model complexity increases. It is the minimum in the test error which enables that model to be chosen which generalises best to unseen data.

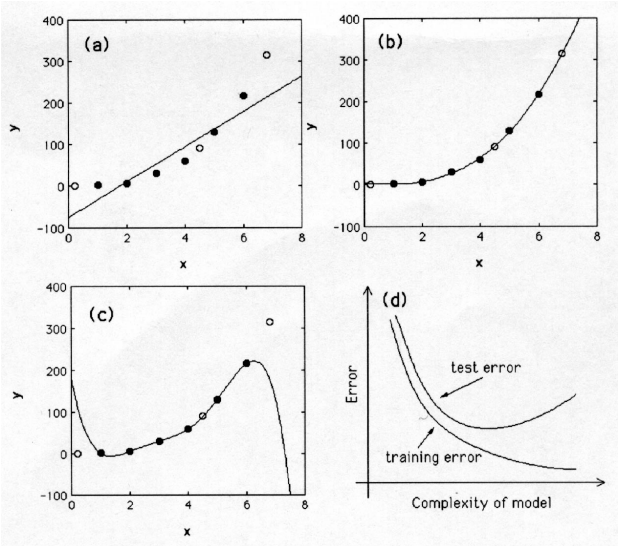

Fig.4. Test & training errors as a function of model complexity, for noisy data in a case where y should vary with x^3. Filled points represent training data, and circles the test data. (a) A linear function which is too simple. (b) A cubic polynomial with optimum representation of both the training and test data. (c) A 5th order polynomial which generalises poorly. (d) Schematic illustration of the variation in the test and training errors as a function of the model complexity.

In neural network analysis, the output y is a non-linear function of x_j, the function usually chosen being the hyperbolic tangent because of its flexibility. The exact shape of the hyperbolic tangent can be varied by altering the weights (Fig.5). Difficulty (c) in Fig. 4 is avoided because the hyperbolic function varies with position in the input space.

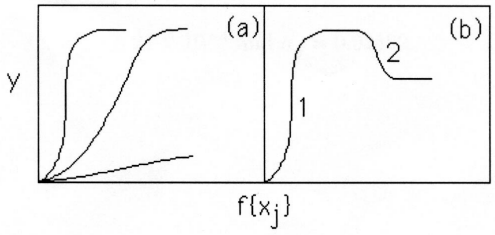

Fig. 5. (a) Three different hyperbolic tangent functions; the "strength" of each depends on the weights. (b) A combination of two hyperbolic tangents to produce a more complex model.

The off-diagonal terms are zero when there is no dependence of x_1 on x_2 and vice-versa.

$$\sigma_y^2 = \mathrm{x}^T \mathrm{V} \mathrm{x}$$

$$= (1\ 3)\begin{pmatrix} 2.7\times10^{-5} & 0 \\ 0 & 1.8\times10^{-4} \end{pmatrix}\begin{pmatrix} 1 \\ 3 \end{pmatrix} = 1.65\times10^{-3}$$

Therefore, for x = (1 3), $\sigma_y^2 = 1.65 \times 10^{-3}$ and $\sigma_y = 0.041$

3. In *normal grain growth*, the average grain size increases with time, and the structure remains geometrically self-similar. The grain diameter distribution is log-normal. The overall reason for grain growth is the reduction of grain-boundary area and thereby of total grain-boundary energy. The local driving force for boundary migration is the boundary curvature.

In a grain structure, the average boundary curvature is inversely proportional to the average grain diameter $\overline{D}$. The boundary velocity is proportional to the boundary curvature so that

$$\frac{\mathrm{d}\overline{D}}{\mathrm{d}t} \propto \frac{1}{\overline{D}} \qquad \text{Integrating:} \qquad \overline{D}^2 = \overline{D}_0^{\,2} + Kt$$

When $\overline{D}_0$ is small, the grain size evolution with time approximates to $\overline{D} \propto t^{1/2}$.

This analytical approach treats only the evolution of mean grain diameter, when the conditions for normal grain growth are satisfied. It does not work if the grain structure does not remain self-similar, notably in *abnormal grain growth*. It does not enable the computation of realistic grain structures, which have a random (stochastic), not regular, nature. It

does not permit the calculation of how grain structures evolve in arbitrary volumes (e.g. in the conductors on an integrated circuit). It does not allow other factors to be taken into account, such as variation in: grain-boundary energy, grain-boundary mobility (affected by solute drag, or Zener drag by second-phase particles), or temperature.

A *vertex model* approximates the grain-boundaries as straight lines (or planes in 3-D). The vertices move because of unbalanced interface tensions. This is computationally the simplest model, and is readily applicable to 3-D. It is not fully realistic because of the lack of boundary curvature, but is a reasonable first approximation.

A *Monte-Carlo model* divides the material into pixels. The pixels in one grain share the same order parameter, different from those in neighbouring grains. Pixels are selected at random and are allowed to change order parameter (with some probability) if by so doing they would lower their energy. The energy is determined by interaction with neighbouring pixels, and is lowest if these match in order parameter. A Monte-Carlo model is accurate only if the pixel size is much smaller than the grain size. It is therefore computationally intensive. It can suffer from anisotropy (influenced by the pixel lattice), but this can be reduced by including longer-range interactions (not just nearest-neighbour). Monte-Carlo models are readily applicable to 3-D.

In a *front-tracking approach*, the grain boundaries are modelled as a series of points. These points are all allowed to migrate, as are the vertices where boundaries meet, under the action of local interface tensions. This approach has the advantages that: arbitrary boundary curvatures can be treated; system non-uniformity from local factors such as stresses and temperature gradients can readily be taken into account. Front tracking is computationally intensive, however, and is difficult to apply in 3-D.

4. In finite element analysis functions of continuous quantities, such as temperature or displacements, can be approximated by piecewise approximations. Thus, a finite element representation of heat diffusion would divide space into small elements, each of which is approximated to be at a single temperature.
Analytical results:

Distance from quenched end, x (mm)	Time from start of quench, t (s)	$x/2\sqrt{at}$ (= dimensionless temperature)	Temperature (^{o}C)
10	20	0.256	232
10	100	0.114	114
10	500	0.051	62
10	1000	0.036	50
65	100	0.749	642
65	500	0.333	296
65	1000	0.235	215

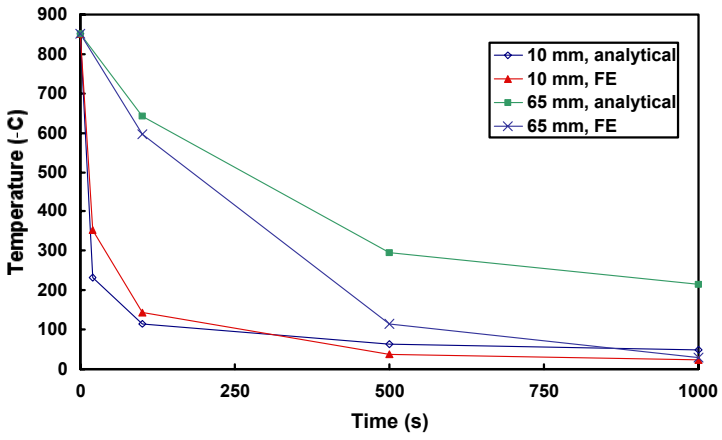

Fig. 6. T(t) for x = 10 and 65mm, analytical and FE solution.

Discrepancies:

x = 10mm – at short times, analytical solution predicts faster cooling due to assumption of perfect heat transfer as opposed to imperfect water quench; at long times, the FE solution reaches ambient sooner, due to finite length of bar (as opposed to semi-infinite assumption of analytical model).

x = 65mm – at all times, analytical solution predicts higher temperatures than FE solution; here the finite length of the bar dominates, as this position is fairly remote from the quench (so less sensitive to difference in heat transfer), but is more than halfway along the bar (so the continued supply of heat from x>120mm in the analytical solution slows the cooling rate considerably).

5. Add the Taylor series for $\mathbf{r}(t+\Delta t)$ and $\mathbf{r}(t-\Delta t)$:

$$\mathbf{r}(t+\Delta t) = \mathbf{r}(t) + \Delta t \cdot \frac{d\mathbf{r}(t)}{dt} + \tfrac{1}{2}\Delta t^2 \cdot \frac{d^2\mathbf{r}(t)}{dt^2} - \tfrac{1}{6}\Delta t^2 \cdot \frac{d^2\mathbf{r}(t)}{dt^2} + O(\Delta t^4)$$

$$\mathbf{r}(t-\Delta t) = \mathbf{r}(t) - \Delta t \cdot \frac{d\mathbf{r}(t)}{dt} + \tfrac{1}{2}\Delta t^2 \cdot \frac{d^2\mathbf{r}(t)}{dt^2} + \tfrac{1}{6}\Delta t^2 \cdot \frac{d^2\mathbf{r}(t)}{dt^2} + O(\Delta t^4)$$

$$\text{gives } \mathbf{r}(t+\Delta t) + \mathbf{r}(t-\Delta t) = 2\,\mathbf{r}(t) + \Delta t^2 \cdot \frac{d^2\mathbf{r}(t)}{dt^2} + O(\Delta t^4)$$

which can be re-arranged to the expression given in the question, recognizing that $\mathbf{a}(t) = \dfrac{d^2\mathbf{r}(t)}{dt^2}$.

Hence, dropping higher order terms gives an error proportional to Δt^4

Superior answers may include comment that this is despite the fact that the Verlet algorithm involves terms only of order Δt^2, i.e. naively one might expect an error proportional to Δt^3. However, as can be seen by the above procedure, the third order terms cancel leaving a smaller resultant error.

There are many other numerical algorithms for integrating the equations of motion, for example Gear predictor-corrector methods, which can have a higher precision than the Verlet method outlined above. This means, for a given time step length, the degree of fluctuations in the temperature or total energy will be smaller. However, the magnitude of the fluctuations is not the most important factor in an MD simulation, since any truncation errors will accumulate exponentially regardless of the precision of the numerical algorithm. More important is the fact that the temperature (NVT) or total energy (NVE) should not drift systematically from their mean values. It turns out that numerical methods that are time-reversible (strictly, they should also be *symplectic*, or phase-space volume preserving, but time-reversible will do) are guaranteed not to produce systematic drift. It can be seen from the expression derived in the first part of the question that this is true for the Verlet method, i.e. replacing Δt with $-\Delta t$ results in the expression being unchanged. So, although precision of Verlet method is not as good as some other numerical methods, it is computationally cheap to evaluate and results in low systematic drift and so is still often used for MD simulations.

The choice of time step length is crucial to achieving an efficient MD simulation. Ideally, it is desirable to make it as long as possible, so that the greatest amount of real time can be simulated for a given amount of computational effort. Too short a time step, and the phase space of the system will be sampled inefficiently. However, if the time step is made too long then the conserved quantities in the simulation (e.g. energy or temperature) may start to fluctuate wildly, eventually leading to catastrophic instability (‘blowing up’). This is because the trajectories of the particles are extrapolated into regions where the potential energy is very high, for example if the particles overlap. A general rule of thumb is that the time step should be comparable in magnitude to either the mean time between collisions in an atomic fluid, or one tenth of the fastest period of motion in a flexible macromolecule. However, the precise value of time step that can be used is very sensitive to the individual conditions under which the simulation is carried out, and must be tuned appropriately to maximise efficiency. The standard limits can be exceeded for macromolecules by freezing out the fast-varying degrees of freedom in the system. Methods for doing this discussed in lectures were constraint dynamics using SHAKE algorithm and rigid-body dynamics. Also, multiple time step (MTA) algorithms using a spatial or bonded neighbour lists can be employed to integrate each set of degrees of freedom with its own frequency. Furthermore, the system may be spatially or temporally coarse-grained by integrating equations of motion for the slow degrees of freedom in the equilibrated mean-field of the faster ones. To obtain maximum marks, answers should describe at least one of the above methods in outline form.

SECTION C

6. Consider the pure components A and B with molar free energies μ^o_A and μ^o_B respectively. The average free energy of a mixture of large lumps of A and B is simply:

$$G\{\text{mixture}\} = (1\text{-}x)\,\mu^o_A + x\mu^o_B \qquad (1)$$

where x is the mole fraction of B. It is assumed that the lumps are so large that the A and B atoms do not "feel" each other's presence via interatomic forces, and that the number of ways in which the mixture of powder particles can be arranged is not sufficiently large to affect configurational entropy. A blend of lumps which, for a mean composition x, obeys equation 1 is called a *mechanical mixture*. By contrast, the free energy of an ideal solution is smaller because of the contribution from configurational entropy.

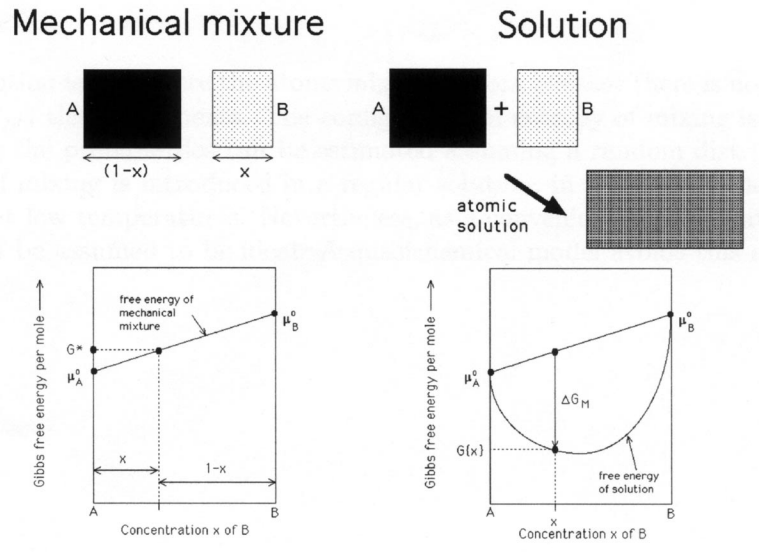

Fig.7. (a) The free energy of a mechanical mixture, where the mean free energy is simply the weighted mean of the components. (b) The free energy of an ideal atomic solution is always lower than that of a mechanical mixture due to configurational entropy.

An alloy can be created without melting, by violently deforming mixtures of different powders (Fig. 8). The dispersion-strengthened alloyed powders are then consolidated using hot-isostatic pressing and extrusion.

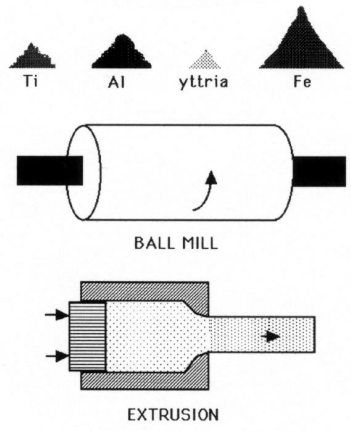

Fig. 8. The manufacture of mechanically alloyed metals for engineering applications. The elemental powders are milled together to produce solid solutions with uniform dispersions. This powder is consolidated by extrusion.

$\Delta G_M = -T\Delta S_M$. The curve will be symmetrical about $x = 0.5$, intersecting each of the vertical axes at $x = 0$ and $x = 1$ at finite points ($\Delta G_M = 0$) but infinite slopes.

Differentiation.

The infinite slopes are mathematical artifacts because the concentration x is treated as a continuous variable which can be as close to zero or unity as desired. Solutions in fact are discrete. Thus, when considering a solution consisting of N

particles, the concentration can never be less than $1/N$ since the smallest amount of solute is just one particle. The slope of the free energy curve will therefore be finite. Indeed, since the concentration is not a continuous variable, the free energy "curve" is not a curve, but is better represented by a set of points connecting the discrete values of concentration that are physically possible when mixing particles.

An ideal solution is one where the atoms mix at random because there is no enthalpy change (ΔH_M) on mixing the components. The configurational entropy of mixing (ΔS_M) is easily derived because the probabilities can be estimated assuming a random distribution of atoms. The enthalpy of mixing is introduced in a regular solution, in which case the atoms may not mix randomly at low temperatures. Nevertheless, as a convenient approximation, the entropy of mixing might be assumed to be ideal. A quasichemical model avoids this approximation.

7. Silicon has four valence electrons, and in order to have a complete electronic shell (which is the most stable configuration) it must gain (or lose) those four electrons. In covalent bonding atoms share electrons so that each atom can have a full shell. By sharing a single electron with each of four other silicon atoms, silicon can effectively gain four electrons and attain a full electron shell, and hence stability.

   Since the electrons are shared between atoms in a covalent bond, the valence electron density is concentrated in the bonds themselves - i.e. in the regions of space directly in between the atoms.

   A simple pair potential does not take into account the directional nature of the covalent bond. Silicon has a preferred bond angle, and this requires a three-body term in the interaction energy.

   For the silicon atom and two of its nearest neighbours, there are two pairs and one triplet, so the interaction energy is:

$$U = 2\left(-\frac{A}{r^6}+\frac{B}{r^{12}}\right)+\frac{2C(\cos(\gamma)+\frac{1}{3})^2}{r^9}$$

   where r is the bond length and γ the bond angle.

$$\frac{\partial U_3}{\partial r} = \frac{-18C(\cos(\gamma)+\frac{1}{3})^2}{r^{10}}$$

   C is positive, $\frac{\partial U_3}{\partial r} < 0$ *i.e.* U_3 is a repulsive potential.

   Since U_3 is parabolic in $\cos(\gamma) + {}^1/_3$, the minimum must be at $\cos(\gamma) = -{}^1/_3$ i.e. $\gamma = 109.5°$.

   Proof: At the optimal bond angle,

$$\begin{aligned} 0 &= \frac{\partial U_3}{\partial \gamma} \\ &= \frac{4C(\cos(\gamma)+\frac{1}{3})(-\sin(\gamma))}{r^9} \\ &= (\cos(\gamma)+\tfrac{1}{3})\sin(\gamma) \\ &\Rightarrow \sin(\gamma) = 0 \text{ or } \cos(\gamma) = -\tfrac{1}{3} \end{aligned}$$

   Clearly $\sin(\gamma) = 0$ means $\cos(\gamma) = 1$ so this is the maximum, and it is $\cos(\gamma) = -{}^1/_3$ which corresponds to the minimum.

   There's no point sketching for bond-angles less than 0° or greater than 180°. Important features of the sketch: minimum at $\gamma = 109.5°$; maximum at $\gamma = 0°$ where $\cos(\gamma) = 1$; maximum at $\gamma = 180°$ where $\cos(\gamma) = -1$

At the equilibrium bond length,

$$\begin{aligned} 0 &= \frac{\partial U}{\partial r} \\ &= 2\left(\frac{6A}{r^7} - \frac{12B}{r^{13}}\right) - \frac{18C(\cos(\gamma)+\frac{1}{3})^2}{r^{10}} \\ &= 2A - \frac{4B}{r^6} - \frac{3C(\cos(\gamma)+\frac{1}{3})^2}{r^3} \end{aligned}$$

and at equilibrium $r = R_0$, so

$$2A - \frac{4B}{R_0^6} - \frac{3C(\cos(\gamma)+\frac{1}{3})^2}{R_0^3} = 0$$

at the optimal bond angle, γ_{opt}, $\cos(\gamma) + {}^1/_3 = 0$

$$\Rightarrow 2A - \frac{4B}{R_0^6} = 0$$

$$\Rightarrow R_0 = \sqrt[6]{\frac{2B}{A}}$$

Important features of the sketch: U is large (tends to $+\infty$ for small r); slope is large and negative for small r; there is a minimum; slope is positive for large r, and tends to zero as r tends to $+\infty$.

The major advantage of this potential is that it is simple, and only includes two- and three-body terms. This means that it is cheap to compute, so simulations could be carried out on many atoms. It also favours the correct bond-angles for a tetrahedral arrangement of atoms, such as a silicon crystal.

There are several obvious weaknesses to the potential:

- it is based on a Lennard-Jones potential, which is designed to model Van der Waals interactions not covalent bonds, and so may not adequately describe the bond stretching or compressing
- the potential will always favour a tetrahedral arrangement of atoms, regardless of the hybridisation of the atom, so it will not be able to cope with sp^2 structures, etc.
- the model does not take into account the bond-order, i.e. how the strength of an atom's bonds changes as the number of bonds increases
- the bond-bending term U_3 is harmonic, and so cannot possibly reproduce anharmonic behaviour
- there is no real physical basis for the model, so the constants A, B and C must be fitted to a set of experimental data, and there is no hard guarantee that the model will work outside of that set
- it is a three-body potential, and so we might expect it to fail whenever there are many-body effects (e.g. electron delocalisation)
- the triplet term only models bonded atoms, and cannot model processes where bonds break or form dynamically

Amongst the improvements that could be made would be:

- change the polynomial behaviour to exponential behaviour, to better model the covalent bond
- introduce higher order terms to better account for bond-order, although this would come at considerable computational cost
- introduce bond-torsion terms, although these are higher-order terms and so are expensive to calculate
- change the angular dependence (U_3) to allow other hybridisations. One simple change would be to change $\cos(\gamma) + {}^1/_3$ to $\cos(\gamma) + {}^1/_h$ to allow studies of sp^h systems (where h is the hybridisation)
- use an *ab initio* method instead, such as a density functional approach