The LDAT Constitutive equation

Summary:

The constitutive equation of LDAT is solved by the LDAT calculation engine, when an LDAT simulation of a configuration of one or more landfill processes takes place. The constitutive equation takes the form of a multi-component multi-phase flow model with a source term that is arranged to accommodate the landfill processes of dissolution, waste degradation, gas solubility, gas diffusion, chemical equilibrium, and the impact of temperature changes due to heat generation and transfer. The LDAT calculation engine solves the landfill process constitutive equations using a finite difference algorithm within a framework of rectangular representative elementary volumes. Sources of published information providing further details of this procedure are given.

Constitutive equation

The constitutive equation of LDAT is the analytical representation of landfill waste degradation and transport. It lies at the core of the LDAT landfill degradation and transport numerical model. It is the constitutive equation that is solved by the LDAT calculation engine, when an LDAT simulation of a configuration of one or more landfill processes takes place. The constitutive equation takes the form of a multi-component multi-phase flow model with a source term that is arranged to accommodate the landfill processes of dissolution, waste degradation, gas solubility, gas diffusion, chemical equilibrium, and the impact of temperature changes due to heat generation and transfer.

The software architecture of the LDAT numerical model is modularised with each sub-model representing one of these landfill processes. Explanations about how each of these sub-models is constructed and used are provided in the LDAT website – www.ldatmodel.com.

The liquid and gas flow sub-model is also described in detail in (White et al. 2015, White et al. 2014); the waste degradation, chemical equilibrium, and solid waste dissolution sub-models are described in (White and Beaven 2013).

As described in (White and Beaven 2013), LDAT solves the landfill process constitutive equations using a finite difference algorithm within a framework of rectangular representative elementary volumes (see Annexe to this Article for further details). The space occupied by the waste material in the landfill is represented by a three dimensional array of these elements. Each element e is surrounded by six neighbouring elements some of which may be boundary elements, Figure 1. The suffix em attached to a variable denotes that the variable is evaluated at the interface between the element and its neighbour m. In the case of vectors it also denotes direction, which is positive in the outward direction normal to the interface. A single suffix e attached to a variable denotes that the variable is evaluated at the centroid of element e.

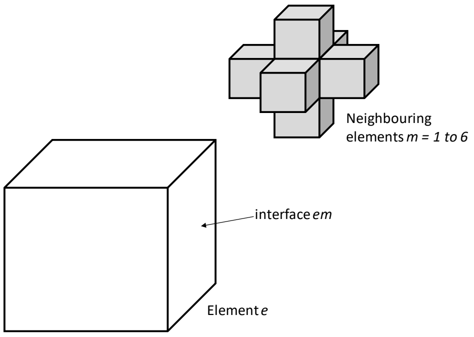


Figure 1 Finite difference three dimensional rectangular element.

Note: Each element has a number, in this case *e*. There are six interfaces with neighbouring elements. The interfaces are designated by the integer *m = 1 to 6*. The number of a neighbouring element *i* is obtained from an array mapping function *i = array_map(e, m)*.

The waste material is represented as the assembly of a number of component chemical compounds and species (chemical elements), each of which can exist in one or all of the three phases: solid, liquid, and gas. The conservation of the mass, $m^P_{e,n}$, of the n^{th} component chemical compound of the waste in phase P (solid, liquid or gas) in the context of a representative elementary volume, V_e, may be expressed by the following equation,

$$\frac{\partial m^P_{e,n}}{\partial t} = \rho^P_{e,n} G^P_{e,n} V_e - \sum_m \rho^P_{em,n} A^P_{em,n} v^P_{em,n} \qquad (1)$$

$m^P_{e,n} = \rho^P_{e,n} z^P_{e,n} V_e$ where $z^P_{e,n}$ is the component compound volumetric concentration or volume fraction (m^3/m^3) and $\rho^P_{em,n}$ is the component density (kg/m^3). For the gas and liquid phases $v^P_{em,n}$ is the interstitial flow velocity (m/day) of component n which is estimated using Darcy's equation after solving equation (1) for the pressure field. For the solid phase it represents the solid particle displacement velocity. $A_{em,n}$ is the area associated with the flow or displacement velocity $v^P_{em,n}$.

$G^P_{e,n}$ is the source term (m^3/day/m^3) associated with component n in phase P, and is composed of four parts, an abstraction/recharge term, a waste degradation term, a gas solubility/phase change term, and a diffusion/dispersion term, as shown by equation (2).

$$G^P_{e,n} = G^P_{abs,e,n} + G^P_{deg,e,n} + G^P_{sol,e,n} + G^P_{diff,e,n} \qquad (2)$$

In the process of solving equation (1), which ensures that mass is balanced, the condition $\sum_n z_{e,n}^P = 1$ is applied to preserve volumetric consistency. The individual concentrations of the compounds $z_{e,n}^P$ are related to the overall concentrations of the solid, liquid and gas phases z_e^S, z_e^L and z_e^G, and to the porosity of the solid phase ϕ_e by the relationships,

$$z_e^P = \sum_n z_{e,n}^P \text{ and } \sum_P z_e^P = 1$$

$$\phi_e = z_e^L + z_e^G,\ 1 - \phi_e = z_e^S$$

A detailed account of how the finite difference algorithm for the liquid and gas flow sub-model is developed and solved is given in (White et al. 2014). The sub-models which affect the source term, such as the bio-degradation and phase change algorithms, are used to evaluate $G_{e,n}^P$ in equation (2). Darcy's Law is substituted into equation (2) to obtain an equation based upon the pressure fields in the gas and liquid phases. Equation (2) is in fact a set of three groups of equations, one group for each phase, with each group containing the individual equations for each component n. By adding together the equations in the liquid and gas phase groups, and using the van Genuchten equation to evaluate the capillary pressure, which is the difference between the gas pressure and the liquid pressure, an overall equation for the gas pressure, p^G, may be obtained and solved numerically using a finite-difference approach to give the transient gas pressure field.

Once this has been done, the process is reversed to back calculate the liquid pressure field and the overall mass transfers in the system. The individual compound concentrations and masses can then be obtained from a further back calculation using the equations represented by equations (1) and (2).

White, J., Zardava, K., Nayagum, D. and Powrie, W. (2015) Functional relationships for the estimation of van Genuchten parameter values in landfill processes models. Waste Management 38(0), 222-231.

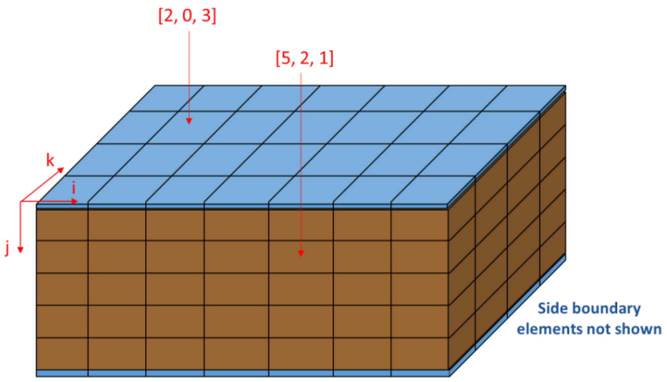
White, J.K. and Beaven, R.P. (2013) Developments to a landfill processes model following its application to two landfill modelling challenges Waste Management 33(10), 1969-1981.

White, J.K., Nayagum, D. and Beaven, R.P. (2014) A multi-component two-phase flow algorithm for use in landfill processes modelling. Waste Management 34(9), 1644-1656.

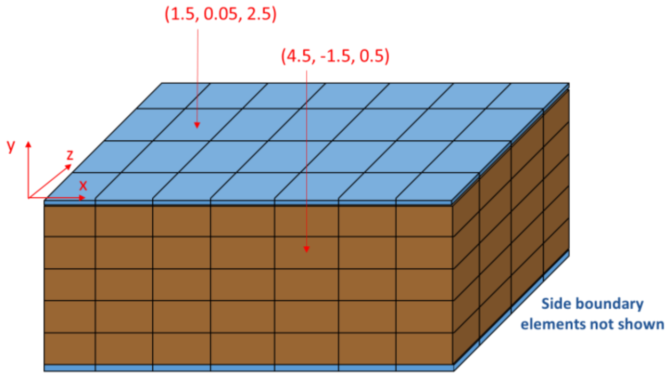
Annexe

The framework of rectangular representative elementary volumes

The solution of the LDAT constitutive equation is carried by a finite difference procedure within the context of a three-dimensional array of elements.



i, j, k index system



x, y, z co-ordinate system

The rows and columns of the elements in the model are identified using the indices i, j and k. These indices are used to identify the individual element 'Slices' as well. Thus element i, j, k is the element in row j of column (i, k). In the LDAT model editor you set the widths and depth of the elements vertical Slice by vertical Slice, and horizontal Slice (or layer), by horizontal Slice. This automatically sets the widths and depths of all of the elements contained by the Slice. This means that you only need to set the widths and depths of the Slices that intersect along a diagonal of the element array, in order to set the widths and depths of all of the elements in the array.

The index origin of the element array is at the nearest upper left hand corner of the element array. From here the zero based element integer indices i, j, and k increment across the element array in orthogonal directions, the j index being vertically downwards. LDAT uses an x, y, z orthogonal co-ordinate system to calculate the geometrical position of various points within the element array. The position of the centroid of each element is provided as part of the results output. The origin of the x, y, z system can also be set by using the LDAT model editor.

Note that LDAT assumes that the Boundary elements form an outer layer around the Active elements. In the Figures above the side boundary elements are not shown so the number of elements in the element array is 9 elements in the i direction, 7 in the j direction and 6 in the k direction. The origin element is the nearest left upper element and has i, j, k indices [0, 0, 0]. This element is not actually shown in the Figures as it is a side element. The boundary element [2, 0, 3] and the active element [5, 2, 1] are picked out.

In the example above, the element array origin in the (x, y, z) co-ordinate system has been set at (-0.1, 0.1, -0.1) and the active elements all have the unit size of 1m x 1m x 1m. The thickness of the boundary elements is 0.1 metres. This results in the centroid of element [2, 0, 3] being at (1.5, 0.05, 2.5), and the centroid of element [5, 2, 1] being at (4.5, -1.5, 0.5).