Proceedings of the 17th Biennial Waste Processing Conference
ASME 1996

PC WINDOWS FINITE ELEMENT MODELING OF LANDFILL GAS FLOW

Stephen R. Mull, Robert J. Lang, Samuel A. Vigil, Harold Cota
Civil & Environmental Engineering Department
California Polytechnic State University
San Luis Obispo

ABSTRACT

A two dimensional demonstration program, GAS, has been developed for the solution of landfill gas (LFG) flow problems on a personal computer (PC). The program combines a Windows^{TM} graphical user interface, object oriented programming (OOP) techniques, and finite element modeling (FEM) to demonstrate the practicality of performing LFG flow modeling on the PC.

GAS is demonstrated on a sample LFG problem consisting of a landfill, one gas extraction well, the landfill liner, cap, and surrounding soil. Analyses of the program results are performed for successively finer grid resolutions. Element flux imbalance, execution time, and required memory are characterized as a function of grid resolution.

INTRODUCTION

LFG is generated in a municipal landfill as a result of biological decay of the organic components of solid waste. The principal components of LFG are methane and carbon dioxide. Other trace gases, some toxic, make up the balance. Generation of LFG within any closed cell of a landfill varies with time, available moisture, and biodegradable organics present (Tchobanoglous et. al, 1993). Simulation of LFG flow is important because:

1) LFG migration off-site can pose a public health risk,
2) LFG diffusion from a landfill cap can be a significant source of air pollution,
3) LFG can be burned to generate economically significant amounts of electricity.

Flow of the generated gases, principally due to convection, may be modeled by the following mass continuity partial differential equation (PDE), where Φ is landfill porosity, C is gas concentration, t is time, ρ is gas density, $\bar{v}$ is the gas velocity vector, and G is the gas generation rate.

$$\Phi \frac{\partial C}{\partial t} = -\bar{\nabla} \bullet (\rho \bar{v}) + G \qquad (1)$$

Analytic and finite difference solutions of this equation have been developed (Young, 1989), (Sumadhu, 1995), however, they are limited. Analytic solutions are constrained to the separate treatment of gas flow within any homogeneous portion of the landfill. These solutions cannot simultaneously describe the generation and flow of gas inside a landfill, its migration through semi-permeable liners, and its dispersion throughout soil surrounding the landfill. Analytic solutions are also limited by the assumptions underlying their formulation and those of their boundary conditions.

Finite difference solutions are limited by the typical requirement for a dense, evenly spaced grid. This spacing requirement limits the geometric flexibility of any general purpose model using finite difference formulations. For a given degree of accuracy, the modeling and computing resources required by finite difference techniques often limit the numerical computations to a workstation or mainframe computer.

FINITE ELEMENT MODELING ON THE PC

To overcome the limitations of analytic and finite difference methodologies, finite element solution methods have been proposed to model landfill gas movement (Lang & Tchobanoglous, 1989b). FEM methods do not need uniform grid spacing, and physical properties can be defined on an element by element basis. Further, by implementing FEM processing using dynamic memory management techniques, existing memory can be more efficiently utilized.

The PC today is the most popular computing platform for practicing engineers. Windows is the most popular operating system for that platform. Because of this large potential user base, software development costs can be more broadly amortized. This makes software written for a PC Windows environment generally less expensive to develop and/or buy than software for other platforms.

To further amortize development costs, object oriented programming can be helpful. This modular programming approach has two important advantages. First, it is designed to facilitate re-use of previously developed software modules (objects), which can drastically reduce development time & hence software cost. Second, it can potentially increase the efficiency of memory utilization within a program, which can increase the accuracy of the numerical results generated by the software. OOP can increase memory utilization by using dynamic memory management. By dividing computations into independent segments, memory used to store intermediate results can be recovered and re-used for subsequent calculations. This increase in the efficiency of memory utilization facilitates more computing power for a given amount of RAM. Memory management of this type is used, for example, whenever windows or dialog boxes within a Windows environment are opened or closed.

GAS PROGRAM

GAS was written to evaluate the viability of applying the Windows graphical user interface, object oriented programming techniques, and finite element modeling to the PC. GAS presently exists as a two dimensional demonstration program. Written in Borland C++ v4.02 for a Windows 3.1 environment, GAS makes use of Borland's container classes and Object Windows Library. GAS contains 4,000 lines of source code, and references both Windows and Borland dynamic link libraries. GAS manages the user interface, defines and/or splits the grid, and applies boundary conditions. GAS interfaces with FEMS. Also written in C++, FEMS is a general purpose finite element engine and occupies 10,000 lines of source code.

FEMS evaluates equation (2), which was derived by combining equation (1) with Darcy's Law, the Ideal Gas Law, and then linearizing in p^2 (Lang, 1989a). In equation (2), M is the molecular weight of the gas, R is the universal gas constant, T is absolute temperature, μ is the dynamic viscosity of the landfill gas, and k is the intrinsic permeability of the solid through which the gas passes.

$$0 = \frac{M}{2\,RT\,\mu}\left(k_x\,\frac{\partial^2(p^2)}{\partial x^2} + k_y\,\frac{\partial^2(p^2)}{\partial y^2}\right) + G \qquad (2)$$

GAS and FEMS presume atmospheric pressure at the top of the cap and the surrounding soil, and no flux through the perimeter of the below grade surrounding soil. Potential attributable to extraction wells is allocated to the appropriate elements.

FEMS sets up and solves a large set of simultaneous linear equations, to enable the general solution of equation (2) for each point in the grid. The coefficient matrix of the equation set forms a symmetric banded diagonal matrix. Bandwidth reduction algorithms are used to reduce the storage required by the coefficient matrix from a potentially square matrix to a thin rectangular matrix, with no loss of information. This results in a significant reduction in memory required to solve a given size grid (Carey & Oden, 1984).

DEMONSTRATION CASE

The following two dimensional example problem was evaluated using GAS with the FEMS engine. A "Quarry" section template was selected from GAS. The quarry template is a pre-defined vertical cross-section of a below grade rectangular landfill. Figure 1 shows the Quarry section. The landfill material is surrounded below and on both sides by a liner and surrounding soil. On top of the landfill, a cap separates the fill material from atmosphere. The Quarry template may be scaled with the use of characteristic lengths. These lengths can be modified via menu commands and pop-up dialog boxes. The Quarry template was modified by changing the default lengths, changing the default permeabilities, and adding a single verticle well to form the model described in Table 1, and Figures 2 and 3. Table 1 documents the input physical properties to the GAS program.

Table 1
DEMONSTRATION CASE PHYSICAL PROPERTIES

Gas Generation Rate	76.25 E-9 Kg/(m3-sec)
Well Vacuum Pressure	-10 iwg [0.99842 atm]
Permeability Constant *	**Kg/(m-atm^2-sec)**
Cap (stratified clay)	1.3178 E-5
Liner (unweathered clay)	4.1180 E-9
Fill (horizontal)	8.2361 E-1
Fill (vertical)	2.1826 E-2
Surrounding Soil	3.2944 E-5

* M $k_{x,y}$ / 2 RTμ from equation (2)

The basic Quarry template shown in Figure 1 contains only 16 elements. In order to obtain an acceptably accurate solution a higher grid resolution is required. This grid resolution is easily obtained by splitting the basic grid via a "SplitGrid" menu command. Figure 2 shows the GAS main window with the Quarry grid split to a grid resolution of four. In response to a dialog box prompt, a higher grid resolution number, *n*, is specified. The GAS program automatically splits the basic template into the finer grid specified by the higher grid resolution number. Further, the split is "smart" in that horizontal elements are split into *n* horizontally adjacent sub-elements, vertical elements are split into *n* vertically adjacent sub-elements, and corner elements are not split. As Figure 2 shows, the remaining elements are split both horizontally and vertically into n^2 sub-elements.

GAS also supports the specification of multiple vertical or horizontal wells. The wells are modeled numerically as line sinks, and GAS automatically apportions their negative pressure among the appropriate elements, and within an element to the appropriate points.

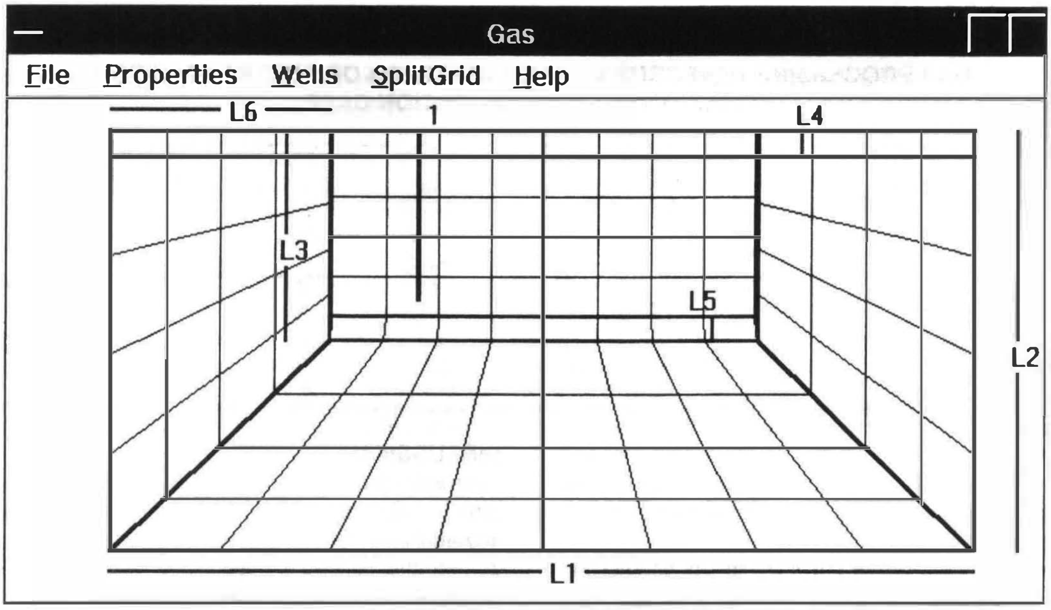

Figure 2. GAS MAIN WINDOW

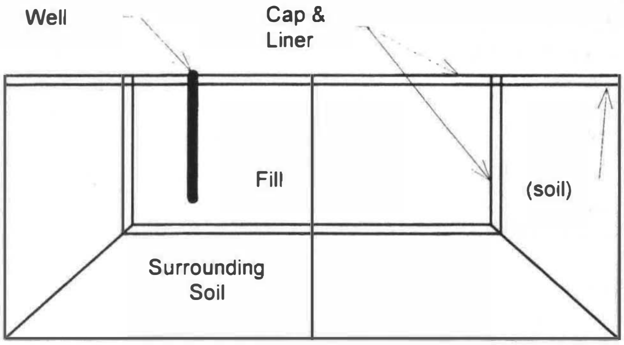

Figure 1. QUARRY TEMPLATE SECTION

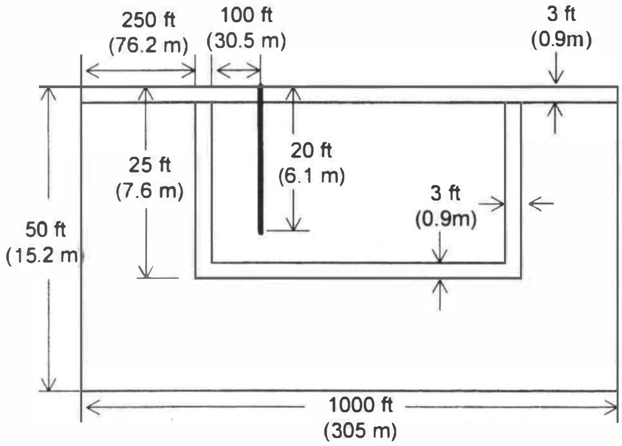

Figure 3. QUARRY TEMPLATE DIMENSIONS

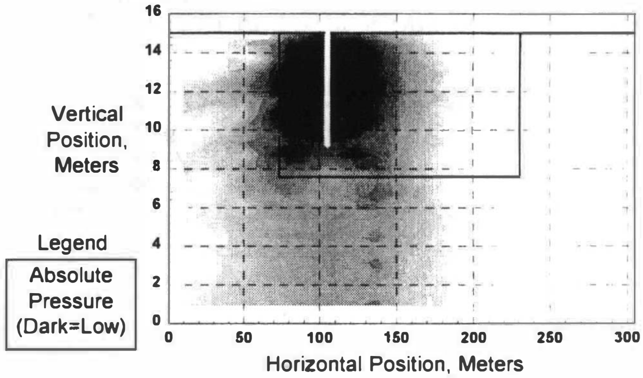

Figure 4
PRESSURE, QUARRY DEMONSTRATION CASE

For inspection, Figure 4 shows a potential plot for an intermediate resolution (GridRes = 4) case. Absolute pressure is plotted by position, using the same orientation as Figures 1, 2, and 3. As Figure 4 shows, the absolute pressure is reduced in the vicinity of the extraction well. In addition, the zone of low pressure extends further vertically than horizontally, which is consistent with the lower horizontal permeability of the fill material used in this example.

PERFORMANCE

After the definition of the basic template for the demonstration case, the grid was split to successively higher resolutions and re-run. Table 2 shows the results of those runs.

Since the finite element method assumes element flux balance and solves for potential by matching inter-element fluxes, a check on the original assumption is one measure of the success of the

Table 2
GAS PROCESSING PERFORMANCE AS A FUNCTION OF GRID RESOLUTION, QUARRY (2-D) DEMONSTRATION CASE

Grid Resolution Number	Number of Elements	Number of Points	Initial Band-width	Final Band-width	Grid Generation Time (sec)	Band-width Reduction Time (sec)	FEM Solution Time (sec)	Required Stiffness Matrix Coefficient Memory (KByte)	Mean Element Flux Imbalance (Kg/sec)
1	16	23	13	7	1.1	1.7	4.2	0.86	-3.98 E-6
2	42	53	43	9	1.6	2.5	6.9	2.4	-1.57E-6
4	130	149	135	13	2.7	8.1	17.3	9.0	-5.06E-7
8	450	485	471	21	9.8	60.1	59.0	44.7	-1.46E-7
16	1666	1733	1719	37	66.2	708.2	264.2	271	-4.17E-8

method. To evaluate the increase in accuracy concurrent with the increased memory requirements and increased processing time of higher grid resolution, mean element flux imbalance was used as an error estimate.

Figure 5 compares the change in (stiffness matrix coefficient) memory requirement, processing speed, and flux imbalance as a function of changing grid resolution. As Figure 5 shows, processing time increases faster than memory usage for increasing grid resolution. As expected, there is a continuous reduction in element flux imbalance as grid resolution increases. All execution time measurements were made with an Intel 80486-33 MHz processor.

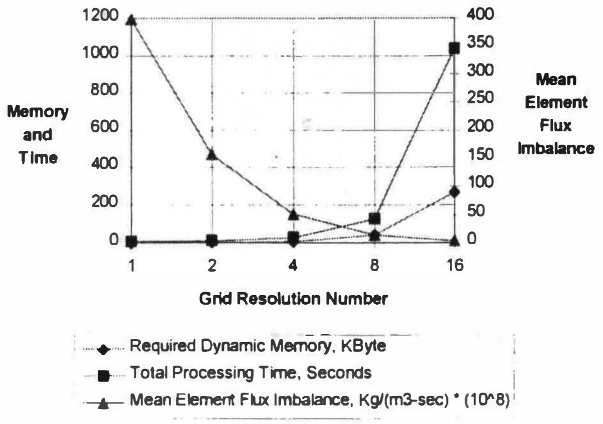

Figure 5
PROCESSING PERFORMANCE

The trade-off between memory, execution time, and accuracy depicted in Figure 5 and Table 2 is generally characteristic of both finite difference and finite element numerical estimation techniques. As the grid spacing becomes more dense, accuracy, required memory, and/or required processing time all increase. Characterizing this relationship for a particular type of application, software, and hardware combination is important to determining whether accurate results can be achieved in a reasonable amount of time on a PC.

DISCUSSION

Because of GAS's Windows user interface, comparison of different design assumptions and scenarios can be more easily accomplished. Well(s) location, spacing, depth, and vacuum can all be manipulated using dialog boxes instead of manually calculating boundary conditions and then editing ASCII text files. Similarly, cap and liner selection and gas generation rate can all be more easily manipulated. In addition to supporting the landfill shape & size manipulation discussed previously, GAS also supports two other geometry templates, one for a "Mound", shown in Figure 6, and one for a "Ravine", shown in Figure 7.

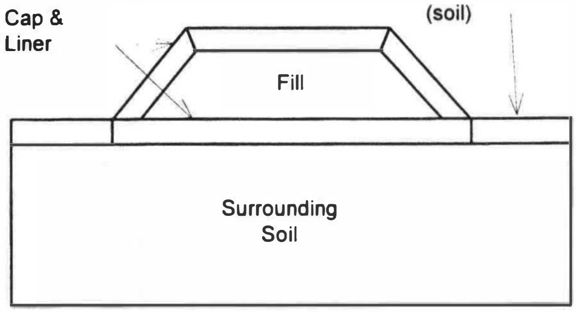

Figure 6. MOUND TEMPLATE SECTION

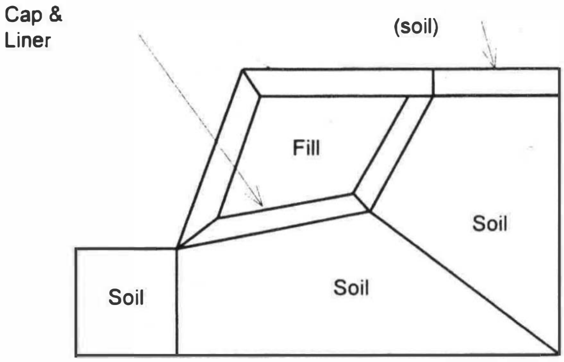

Figure 7. RAVINE TEMPLATE SECTION

2-D landfill simulations can be used, for example, to estimate extraction well radius of influence under different conditions, and to characterize as-built performance of different cap & liner materials.

Extension of GAS and FEMS to three dimensional modeling could enable the calibration of a finite element model to field test data for an existing landfill. By applying field data to a numerical model, the boundary conditions and constants used by the model can be estimated for the particular conditions of the landfill. Such data could include:

- permeability and gas generation rate estimates derived from tests of core samples,
- gas composition & flow data from test wells, and
- gas concentration data from surface tests.

The calibrated model could then be used to predict gas recovery, net gas emissions through the landfill cap (before & after installation of an extraction system), as well as any subsurface gas migration off site.

Extension of FEMS to three dimensions is conceptually straightforward, however it introduces other complications. Because 3-D FEM requires more memory, it introduces more complex memory management issues, including the design for the memory partitions used by Windows 3.1. An additional complication arising in 3-D FEM and not present in GAS is the extension of the Windows interface design to a 3-D grid. The Windows graphics required to enable 3-D grid visualization, navigation, and editing involve use of more complex Windows controls.

Windows programming can be tedious. To simplify programming, encapsulated libraries are supplied by a variety of vendors (Microsoft VBX, OCX, and MFC, Borland OWL). These libraries occupy substantial memory. Some components of these libraries can be dumped from RAM at run-time to make room for a larger FEM grid, however not all components can be discarded. This Windows memory overhead, at 4-8 MB, significantly dominates coefficient memory requirements for the two dimensional case.

CONCLUSIONS

The FEM method is capable of solving 2-D landfill gas problems on a PC to within the accuracy limitations imposed by available time and PC memory.

A Windows user interface substantially simplifies the effort required by the user to generate a FEM grid and apply the necessary boundary conditions, however a Windows interface imposes substantial memory overhead. This overhead is not a limiting factor for two dimensional modeling, but may become one for three dimensional modeling.

The benefit of combining OOP, finite element modeling, and a Windows graphical user interface on the PC is primarily the ability to simulate landfill gas movement easily with a PC. While this benefit is real, it does not come without increases in software complexity. Extension of object based finite element modeling to the PC is possible and can provide powerful and easy to use tools for environmental professionals.

REFERENCES

Carey, F.G., and Oden, J.T., 1984, *Finite Elements: Computational Aspects*, Prentice-Hall, New Jersey.

Lang, R.J., 1989, "Modeling the Movement of Trace Gasses in Municipal Solid Waste Landfills," Ph.D. Thesis, University of California, Davis.

Lang, R.J., and Tchobanoglous, G., 1989, "Modeling Emissions of Trace Gasses from Landfills," *Proceedings, 82nd Annual Meeting & Exhibition, Air & Waste Management Association*, Anaheim, Ca. Vol. 89-53.2.

Sumadhu, G.A., et al., 1995, "Gas Generation, Transport, and Extraction in Landfills," *Journal of Environmental Engineering*, Vol. 121, No. 1, pp. 33-44.

Tchobanoglous, G., Theisen, H., Vigil, S., 1993, *Integrated Solid Waste Management*, McGraw-Hill, NY, pp. 381-417.

Young, A. 1989, "Mathematical modeling of Landfill gas Extraction." *Journal of Environmental Engineering*, ASCE, Vol. 115, No. 6, pp. 1073-1087.