A rapid method for the computation, comparison and display of molecular volumes

Regine S. Bohacek and Wayne C. Guida

Pharmaceuticals Division, CIBA-GEIGY Corporation, Summit, NJ, USA

This paper presents a method for the rapid computation of approximate molecular van der Waals volumes and their subsequent display. The procedure relies on bit representation of individual volume elements that are mapped into an array that stores the total molecular volume. Our method differs from previously described algorithms in its use of bit-encoded templates that define atomic van der Waals radii. For each atom in the molecule, for which the volume is to be computed, the relevant template is mapped into a bit array with an offset corresponding to the appropriate atomic position. Bit-wise Boolean operations can be used for volume comparisons (e.g., common volume and excluded volume). An algorithm for the graphical display of the molecular surface encompassing the computed volumes is also described. The speed of the method enables users to perform volume computations in a reasonable period of time with VAX-class computers on molecules containing as many as several hundred atoms.

Keywords: molecular volume, graphics of molecular surface, volume mapping, bit maps, 3-D molecular shape, mesh surface

INTRODUCTION

Numerous methods have been described in the literature for determining molecular volumes. Intuitively, one might imagine a method in which each atom in the molecule is represented by a sphere whose radius is the van der Waals radius of the particular type of atom. The envelope defined by the set of interlocking atomic spheres would define the outer molecular surface, and calculating the total volume of all the spheres, with appropriate correction for the overlap volume, would define the total molecular volume. This approach has, in fact, been used successfully for molecular volume calculations.[1] In a related approach that has been described by Connolly,[2] either the van der Waals volume or the solvent-excluded volume can be computed from the appropriate enclosed molecular surface, since its area has been previously determined by an analytical procedure. In a numerical procedure that employs sampling techniques,[3] the molecule is embedded in a larger, predetermined envelope. The interior of this envelope is sampled with a large number of probe points, and the number of points inside the molecule is recorded. The ratio of points inside the molecule to the total number of points, multiplied by the volume of the predetermined envelope, yields the molecular volume.

An ingenious method for bit mapping of volume elements for molecular surface calculations[4] and molecular volume calculations[5] was described several years ago and has been extended recently by Stouch and Jurs.[6] In this method, the molecule is embedded in a virtual cube that is subdivided into cubic volume elements, and a bit array is used to map the individual volume elements. Each cubic volume element is tested to determine whether it is interior or exterior to the molecule. If it is interior, the appropriate bit in the bit array is turned on (i.e., set to 1). Integration (i.e., counting all the "on" bits) yields the total molecular volume, and the molecular surface can be computed by searching for boundary bits. The utility of this method lies in the bit encoding of the three-dimensional (3D) molecular shape, since volume comparisons can be accomplished readily via bit-wise Boolean operations available in many computer languages or their extensions.

Molecular volume comparisons are an important consideration in the active analog approach to drug design,[7] and the interactive, rapid computation, visualization and comparison of molecular volumes provides a valuable tool for the medicinal chemist. Thus, we sought to extend the algorithm described above, which relies on bit encoding of molecular volumes to increase its speed, and to incorporate it into a molecular modeling software package. We chose to incorporate into the MacroModel program[8] our algorithms for the rapid computation and display of molecular volumes, since the program provides an interface in which one can combine user-written subroutines with the main program. Versions 2.0 and 2.5 of MacroModel contain our volume algorithms, which were written in VAX-extended FORTRAN 77.[9]

Address reprint requests to Dr. Bohacek at the Research Department of the Pharmaceuticals Division, CIBA-GEIGY Corporation, Summit, NJ 07901, USA.
Received 14 November 1988; accepted 16 March 1989

ALGORITHMS

Our algorithm provides a simple technique that eliminates distance calculations, the most time-consuming aspect of numerical molecular volume computations. In the method described below, a set of atomic volume templates for each atom type is generated, and the molecular volume is constructed from these templates. The predetermined templates are stored as a disk file and are read by the program during initialization of the volume calculations. Even though distance calculations are required to generate the templates, the calculations are performed only once, since the results have been filed for future reference.

Each template can be envisioned as a virtual cube that circumscribes a sphere whose radius is the van der Waals radius of the given atom type (Figure 1a). In order to produce the templates, a 3D grid is constructed with dimensions exceeding the van der Waals diameter of the specific atom type. The positioning of this grid relative to the origin is arbitrary, but the grid for each atom type is positioned in the same way. Starting at the center of the grid, each grid point, which defines a cubic volume element, is tested to see if the distance between that point and the center point is less than or equal to the van der Waals radius. In this manner, all interior and exterior points are located. The coordinates corresponding to each grid point, and hence each cubic volume element, are then represented by a single bit within a computer word. If a grid point is an interior point, the bit is set to 1; otherwise, it remains 0 (Figure 1b).

The string of bits that encodes the atomic volume can be represented by an integer variable. Thus, a triply subscripted INTEGER*4 array (named "atombox" in our program) is employed for the bit map of the templates corresponding to the various atom types. One dimension is used to reference the specific atom type. Another dimension corresponds to the z direction (i.e., the individual slabs that define each xy plane as shown in Figure 1b). The third dimension is used for the bit representation of each cubic volume element within the xy plane. The integer array is loaded at run time from the disk file. To conserve disk space, only one-quarter of the sphere is stored on disk, and the complete sphere is constructed by symmetry during initialization of the program.

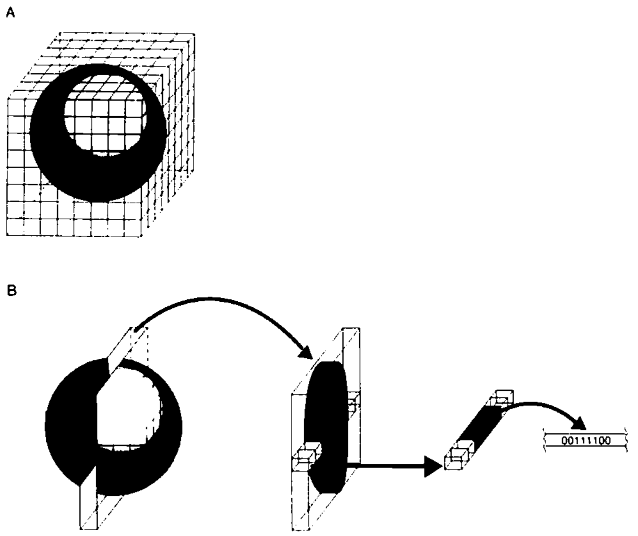

Figure 1. a) A cube, in which an atomic van der Waals sphere has been embedded, subdivided into discrete volume elements. Each element corresponds to one bit within a computer word. b) A plane taken from the cube which depicts how the volume of the sphere is coded at the bit level: Volume elements within the van der Waals sphere are represented by "on" bits; those exterior to the sphere are left "off"

Table 1. van der Waals radii for atom templates[a]

Atom type	0.1 Å grid	0.2 Å grid[b]
Hydrogen	1.2	1.2
Carbon	1.7	1.6
Oxygen, Florine	1.4	1.4
Nitrogen	1.6	1.6
Chlorine, Sulfur, Phosphorus	1.8	1.8
Bromine	2.0	2.0
Iodine, Silicon	2.1	2.0

[a] Based on van der Waals radii used by Gavezzotti.[3] [b] For the 0.2 Å grid, each volume element is 0.2 × 0.2 × 0.2 Å; therefore, the radii are rounded to the nearest 0.2 Å

Since the number of bits required to store volumes increases with increasing molecular size, we use two different templates, depending upon the size of the molecule. For smaller molecules ($\leqslant$ ca. 100 atoms), we use a template with a cubic volume element size of 0.001 Å^3 (i.e., a grid size of 0.1 Å). For larger structures ($>$ ca. 100 atoms), we use a grid size of 0.2 Å. Thus, for larger structures, each volume element corresponds to a volume of 0.008 Å^3; for smaller ones, as previously mentioned, it corresponds to 0.001 Å^3. For convenience, and due to the inexact nature of these calculations, we have constructed templates corresponding to the approximate volume of seven different sizes of atoms. Table 1 shows the van der Waals radius used to construct each template. Since the templates are read from a disk file, templates corresponding to any atomic radius could be used by modifying this file. Of course, a larger number of templates could be constructed, thereby increasing the accuracy of the calculations for some atom types. However, the accuracy of our calculations should be adequate for molecular modeling applications where a high degree of accuracy is not generally required, such as active analog drug design.[7]

Once the templates have been read from disk, the volume calculation is relatively straightforward. An INTEGER*4 array named "molebox" is used to store the bits that map a rectangular grid that encloses the molecule in a manner analogous to the bit map of the templates described above. The molebox array is a doubly subscripted array, and, as with the templates, a bit map is constructed so that one dimension corresponds to the z direction, whereas the other dimension is used to map each cubic volume element within an xy plane. Thus, the Cartesian coordinates for each atom in the molecule can be associated with a bit location within the molebox array. During initialization, all the bits in molebox are set to 0. Upon execution of the volume calculation, the bit corresponding to the Cartesian coordinates for a given atomic center is located, and then the template for that type of atom is selected. Beginning at the atomic center, the bit pattern of the template is copied into the molebox array at the appropriate position. Thus, all the bits corresponding to the volume elements within the van der Waals radius of each atom are set to 1 within the molebox array, with an offset corresponding to the location of each atom in Cartesian space. Since the van der Waals spheres encoded in the templates overlap, some of the on bits get overwritten with on bits. However, this procedure, which wastes CPU cycles, is more than compensated for, since it does not require testing to determine whether volume elements are disposed interior or exterior to the molecule. Figure 2 diagrammatically illustrates the algorithm.

Several subroutines (written in VAX assembly language and callable from within the object library of MacroModel) are used for bit copy from the template for the particular atom type into the molebox array. These specialized routines allow for the extraction of a specified number of bits from within an integer array and placement of a specified number of bits into an integer array. Similar functionality is provided by the VAX FORTRAN library subroutines IBITS and IBSET.

Boolean operations are readily performed by using two (or more) INTEGER*4 arrays to store the rectangular boxes and their associated molecular volumes. Thus, if Molebox1 and Molebox2 correspond to two arrays that store the bit maps of two distinct molecules, a loop that operates on each array element of these two arrays with the logical AND operator (i.e., Molebox1(I) .AND. Molebox2(I)) produces a new bit map that stores the volume common to the two molecules. Similarly, the .OR. operator affords the total volume occupied by both structures. Our program incorporates these and other, more complex Boolean operations.

For the graphical representation of the surface enclosing the molecular volume, we have chosen to use a mesh representation. However, no matter what representation is employed, the surface points must be previously determined. In the molecular bit maps described above, the surface points are represented by those on bits that possess at least one adjacent off bit. In order to search the molecular bit maps for such bits, we use an algorithm[10] that we refer to as the MAZE algorithm, since it employs the left-most-looking rule, a well-known rule for escaping from a maze: "always go left." Figure 3 illustrates the way in which this algorithm operates. For convenience of discussion, a two-dimensional grid is shown in Figure 3; the algorithm, of course, works on bit maps that encode successive planes of a 3D grid. The shaded area represents the interior, each cell of which is mapped as an on bit. All other cells are "off." The process begins when a surface bit is located somewhat arbitrarily. Since the rectangular box extends beyond the molecular boundaries, one typically starts at the origin and proceeds (in a linear direction in the bit map) until the first on bit is encountered. Arrow 1 repre-

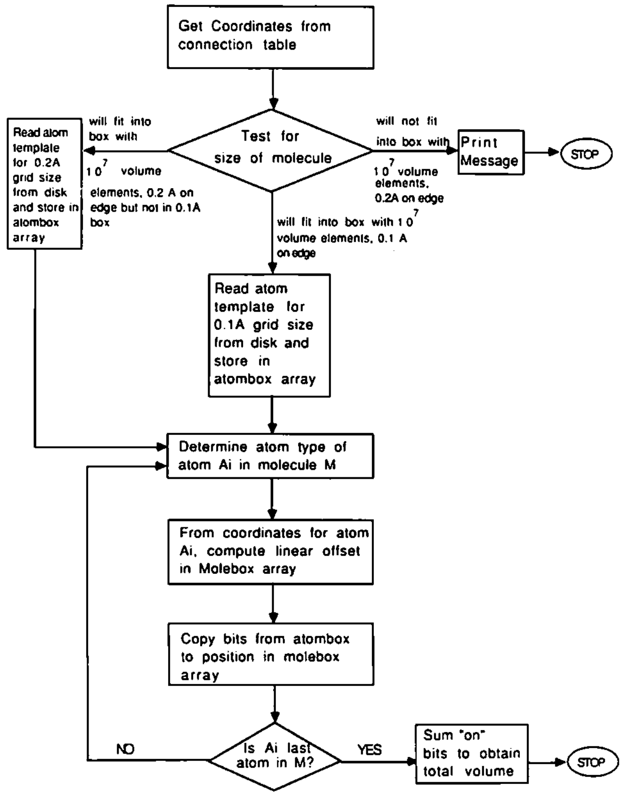
Get Coordinates from connection table
Test for size of molecule
will fit into box with 10^7 volume elements, 0.2 A on edge but not in 0.1A box
Read atom template for 0.2A grid size from disk and store in atombox array
will not fit into box with 10^7 volume elements, 0.2A on edge
Print Message
STOP
will fit into box with 10^7 volume elements, 0.1 A on edge
Read atom template for 0.1A grid size from disk and store in atombox array
Determine atom type of atom Ai in molecule M
From coordinates for atom Ai, compute linear offset in Molebox array
Copy bits from atombox to position in molebox array
Is Ai last atom in M?
NO
YES
Sum "on" bits to obtain total volume
STOP

Figure 2. Flow chart for the bit-encoded molecular volume algorithm

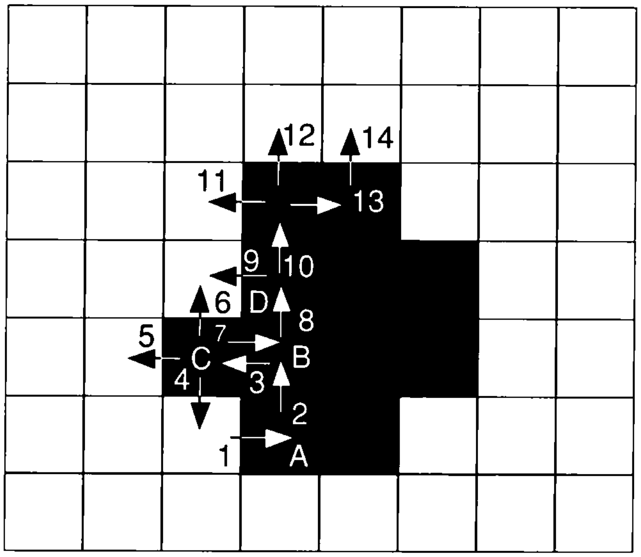

Figure 3. Determination of surface bits with the MAZE algorithm

sents the first step, and the cell reached, cell A, is "on." A move is then made to the left into another on cell, cell B, as depicted by arrow 2. Another left move, as shown by arrow 3, again reaches an on cell, cell C. However, the next left move, shown by arrow 4, locates an off cell, and hence cell C has been identified as a surface cell. Its position is stored, a step backward into cell C is made and another left turn is made (arrow 5). Again an off cell is located, so another return to cell C is made, followed by another left turn (arrow 6). The third off cell is located, which surrounds cell C. Another return followed by another left turn (arrow 7) locates an on cell (cell B), and two more left turns (arrows 8 and 9) are required to locate cell D as the next surface cell. Arrows 10–14 show a continuation of the process. The basic strategy is to continue turning left into on cells until an off cell is reached. The previous cell is therefore a surface cell and its position is recorded. A return to the previous cell is done, and the process is repeated. The search is concluded when the original cell is located again. The advantage of this procedure is that the entire search takes place at the molecular surface boundary, and hence interior points need not be sampled.

RESULTS

In order to assess the relative speed of our algorithm for computing molecular volumes, we undertook a comparison of CPU times for calculating several representative structures. Table 2 compares CPU times required to compute volumes for a variety of molecules using our algorithm versus the one previously described by Stouch and Jurs.[6] CPU times have been normalized to approximate MicroVAX II CPU time equivalents by comparing the speed, in MIPS, of the machine that performed the calculations with the speed of a MicroVAX II (0.9 MIPS). As demonstrated by the entries in Table 2, our algorithm is quite rapid. In fact, it is fast enough to allow the computation of molecular volumes for small proteins, such as crambin,[11] on a MicroVAX II–class machine in less than one CPU minute.

Although the CPU times reported in Table 2 do not account for the time required to compute the surface points with the MAZE algorithm, the speed of the MAZE algorithm relative to the volume algorithm is quite rapid. For example, when we calculated the CPU time required to construct the molecular bit map, sum the on bits, and execute the MAZE algorithm for locating the surface points for hydrocortisone, we found that the MAZE algorithm required only 2% of the total CPU time.

To determine the relative accuracy of our algorithm for calculating molecular volumes, we decided to compare volumes obtained using our procedure with those obtained from various other procedures. Of course, factors such as the van der Waals radii used in the different methods will influence the outcome of such a comparison. Table 3 compares volumes computed with our method to the ones calculated by the methods of Stouch and Jurs,[6] Pearlman[12] and Gavazotti.[3] As is shown, the volumes calculated by our algorithm compare favorably with those obtained by the other methods. They agree quite well, generally within a few percent, with the volumes obtained via the accurate method of Pearlman.[12]

Since the orientation of the molecule relative to the grid in which it is embedded might be expected to affect the computed volume, we determined the effect of orientation on the volumes calculated by our method for several molecules, such as benzoic acid and hydrocortisone. In general, the error in the volume computation due to reorientation of the molecule within the grid is not appreciable. For example, when the volume was calculated for hydrocortisone in 10 different orientations relative to the grid, the mean was 341.9 Å^3 with the 0.1 Å grid and 348.2 Å with the 0.2 Å grid, with a

Table 2. Comparison of CPU times required for the calculation of van der Waals molecular volumes[a]

Compound	Number of atoms	Method of Stouch and Jurs[b]	Present study[c]
Ethane	8	9.4	2.3
Cyclohexane	18	50.2	4.7
Trans-decaline	28	106.7	7.6
Trans, anti, trans-perhydrophenanthrene	38	192.2	9.4
Hydrocortisone (cortisol)	56	491.4	15.0
Crambin[d]	327	—	25.6

[a] CPU seconds corrected so that the times reported are in MicroVax II CPU equivalents. The CPU times given are those required for the calculation of the bit map and subsequent summation of the "on" bits to produce the total molecular volume. [b] Calculations performed on a Prime 750 with a grid size of 0.25 Å. [c] Calculations performed on a VAX 8800 with a grid size of 0.1 Å. [d] Coordinates obtained from Brookhaven Protein Data Bank, entry: 1CRN; grid size 0.2 Å

Table 3. Comparison of van der Waals molecular volumes[a]

Compound	Method: Stouch and Jurs[b]	Pearlman[c]	Gavazzotti	Present study[d]
Ethane	45.38	45.29	44.63	45.4
Methanol	36.57	36.52	34.89	37.6
Cyclohexane	101.41	101.37	99.1	101.3
Trans-decaline	156.97	157.24	—	158.4
Hydrocortisone (cortisol)	340.56	340.78	—	341.9

[a] In Å^3, exact values will, of course, be dependent on van der Waals radii employed for the computation. [b] Calculations performed using a 0.25 Å grid. [c] Data from reference 6. [d] Calculations performed using a 0.1 Å grid

standard deviation of 0.9 for the .1 Å grid and 2. for the .2 Å grid.

One application of the volume mapping procedure is, as previously stated, for active analog drug design.[7] Thus, calculating excluded volume maps and determining receptor essential volumes can be performed readily with our algorithm by using the bit-wise Boolean operators. In another molecular design application, the overlap volume between an enzyme and its inhibitor can be computed. Color Plate 1 depicts the inhibitor thiorphan[13] positioned in the active site of thermolysin. The coordinates for thermolysin were obtained from the Brookhaven Protein Data Bank,[14] and the active site was defined as those residues that possess at least a single atom within 8 Å from any atom of thiorphan. The molecular volume depicted in Color Plate 1 is a close-up view of the volume of the active site residues. Color Plate 2 shows the common volume (obtained as the Boolean AND) of the inhibitor volume and the active site volume. As expected for this relatively potent inhibitor, the only common volume is where hydrogen bonds occur between the ligand and the enzyme and where the ligand is coordinated to zinc. For computer-assisted molecular design, this technique affords a graphically useful tool to visualize where poor fit, due to van der Waals overlap between the cavity and the ligand, is likely to occur.

Color Plate 3 shows the molecular volume for the entire crambin[11] molecule. In our implementation of the algorithm, a structure this large requires use of the 0.2 Å grid. Nonetheless, the complete molecular volume is computed quite rapidly (see Table 2), and the resulting mesh surface exhibits acceptable detail.

SUMMARY

We have presented a method for the rapid computation and display of molecular van der Waals volumes. The method is sufficiently rapid that molecular volumes of structures as large as small proteins can be computed and displayed in a reasonable length of time on VAX-class machines. We believe that this technique will extend the utility of volume calculations for computer-assisted molecular design.

ACKNOWLEDGEMENT

We thank Dr. Richard Franks of AT&T Bell Laboratories for numerous stimulating discussions concerning development of these algorithms. We are grateful for assistance with the PS300 graphics programming given to us by Mlle. Veronique Eyraud of CIBA–GEIGY, Basel. We also thank Dr. Frank Clark of CIBA–GEIGY, Summit, for many helpful suggestions and Dr. Mark Erion of CIBA–GEIGY, Summit, for constructing the thermolysin computer model.

REFERENCES

1 Pavini, R., and Ranghino, G. A method to compute the volume of a molecule. *Computers & Chemistry* 1982, **6**, 133–135

2 Connolly, M.L. Computation of molecular volume. *J. Am. Chem. Soc.* 1985, **107**, 1118–1124

3 Gavezzotti, A. The calculation of molecular volumes and the use of volume analysis in the investigation of structured media and of solid-state organic reactivity. *J. Am. Chem. Soc.* 1983, **105**, 5220–5225

4 Pearl, L.H., and Honegger, A. Generation of molecular surfaces for graphic display. *J. Mol. Graphics* 1983, **1**, 9–12

5 Marsili, M., Floersheim, P., and Dreiding, A.S. Generation and comparison of space-filling molecular models. *Computers & Chemistry* 1983, **7**, 175–181

6 Stouch, T.R., and Jurs, P.C. A simple method for the representation, quantification, and comparison of the volumes and shapes of chemical compounds. *J. Chem. Inf. Comput. Sci.* 1986, **26**, 4–12

7 Marshall, G.R., Barry, C.D., Bosshard, H.E., Dammkoehler, R.A., and Dunn, D. A. The conformational parameter in drug design: the active analog approach, in *Computer Assisted Drug Design*, Olson, E.C., and Christofferson, R.E., Eds., ACS Symposium 112: Washington, D.C. 1979, 205–226

8 Mohamadi, F., Richards, N.G.J., Guida, W.C., Liskamp, R., Lipton, M., Caufield, C., Chang, G., Hendrickson, T., and Still, W.C., Macromodel — an integrated software system for modeling organic and bioorganic molecules using molecular mechanics. *J. Comp. Chem.*, submitted for publication

9 MacroModel is available from Professor W. Clark Still, Department of Chemistry, Columbia University, New York, NY, 10027, USA

10 Gonzalez, R.C., and Wintz, P. *Digital Image Processing*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1977, 253–262

11 Coordinates for crambin were obtained from the Brookhaven Protein Data Bank, entry: 1CRN, Hendrickson, W.A., Teeter, M.M. *Nature* 1981, **290**, 107–113

12 Pearlman, R.S., in *Physical Chemical Properties of Drugs*, Yalowsky, S.H., Sinkula, A.A., Valvani, S.C., Eds. Marcel Dekker, NewYork, 1980, 321–347

13 Benchetrit, T., Fournie-Zaluski, M.C., and Roques, B.P. Relationship between the inhibitory potencies of thiorphan and retrothiorphan enantiomers on thermolysin and neutral endopeptidase 24.11 and their interactions with the thermolysin active site by computer modelling. *Biochem. and Biophys. Research Comm.* 1987, **147**, 1034–1040

14 Coordinates for thermolysin were obtained from the Brookhaven Protein Data Bank, entry: 5TLN, Holmes, M.A., and Matthews, B.W. Binding of hydroxamic acid inhibitors to crystalline thermolysin suggests a pentacoordinate zinc intermediate in catalysis. *Biochem.* 1981, **20**, 6912–6920 and modified for mercaptan inhibitors, Monzingo, A.F., and Matthews, B.W. Structure of a mercaptan — thermolysin complex illustrates mode of inhibition of zinc proteases by substrate — analogue mercaptans. *Biochem.* 1982, **21**, 3390–3394