The Monotonic Lagrangian Grid for Fast Air-Traffic Evaluation

Carolyn Kaplan[1], Elaine Oran[2], Natalia Alexandrov[3]*, and Jay Boris[4]
Naval Research Laboratory, Washington DC 20375

**NASA Langley Research Center, Hampton VA 23681*

This paper describes the continued development of a dynamic air-traffic model, ATMLG, intended for rapid evaluation of rules and methods to control and optimize transport systems. The underlying data structure is based on the Monotonic Lagrangian Grid (MLG), which is used for sorting and ordering positions and other data needed to describe *N* moving bodies, and their interactions. In ATMLG, the MLG is combined with algorithms for collision avoidance and updating aircraft trajectories. Aircraft that are close to each other in physical space are always near neighbors in the MLG data arrays, resulting in a fast nearest-neighbor interaction algorithm that scales as *N*. In this paper, we use ATMLG to examine how the ability to maintain a required separation between aircraft decreases as the number of aircraft in the volume increases. This requires keeping track of the primary and subsequent collision avoidance maneuvers necessary to maintain a five mile separation distance between all aircraft. Simulation results show that the number of collision avoidance moves increases exponentially with the number of aircraft in the volume.

Nomenclature

N	=	the number of particles in the system
x, y, z	=	coordinates of the particles in 3D
i, j, k	=	indices of the particles in the data structure
Nx, Ny, Nz	=	the number of particles in directions x, y, z, respectively

I. Introduction

In order to develop strategies to design and control the increasingly complex aspects of air transportation, we must have a fast research tool for modeling various layers of the air-traffic system and, eventually, the layers of the entire multi-modal transport system. This is the second report on a dynamic global model to be used for simulating the air traffic flow as a platform for evaluating approaches to local and global air systems control and optimization. The model, based on an algorithm called the Monotonic Lagrangian Grid (MLG), can quickly sort, track, and update positions of more than 10000 aircraft, both on the ground (at airports) and in the air. The Air Traffic Monotonic Lagrangian Grid (ATMLG) can be used to evaluate new system concepts; for instance, as an aid in testing various control strategies, such as collision avoidance, separation assurance, and traffic flow management, as well as in evaluating the reaction of the system to local and global perturbations, such as weather. It could be used to determine the most efficient way to reroute air traffic after local conditions, such as thunderstorms, have propagated the local disturbance throughout the entire system.

In a previous paper,[1] we presented results from a series of proof-of-concept experiments to test two air traffic problems: collision avoidance and circumventing restricted zones. Those simulations showed the ability of the model to account for and minimize the impact of sudden blockages in flight paths, avoid near-collisions, and suggest

[1] Research Chemical Engineer, Laboratory for Computational Physics & Fluid Dynamics (LCP&FD), Code 6410, AIAA Associate Fellow.
[2] Senior Scientist for Reactive Flow Physics, LCP&FD, Code 6404, AIAA Fellow.
[3] Senior Research Scientist, Aeronautics Systems Analysis Branch, Mail Stop 442, AIAA Associate Fellow.
[4] Chief Scientist and Director, LCP&FD, Code 6400, AIAA Fellow.

alternative flight path routes. The results also showed that the MLG is a promising platform for investigating ways to maintain safe separation between many aircraft in a complex airspace. This is important for designing the Next Generation Air Transportation System (NGATS), in which the use of high altitude airspace will increase and aircraft will use automated separation assurance methods. In an automated operation, the goal of airspace management will be to manage complexity operations, where the complexity threshold is considered to be manageable if an additional aircraft can enter the airspace without causing a safety critical situation.[2] An aircraft entering the airspace should be able to choose its flight path through the airspace, and in a conflict situation, it should have at least one degree of freedom available to resolve any conflict.[2] If that one degree of freedom is not available, then that airspace must be considered full. This automated operation planned for NGATS is vastly different from the current air-transport system, in which the amount of air traffic in a given sector is limited by the workload of the controller. In this paper, we use the MLG to simulate air traffic flow in a 3-D volume to gain a better understanding of when a volume of airspace is controllable.

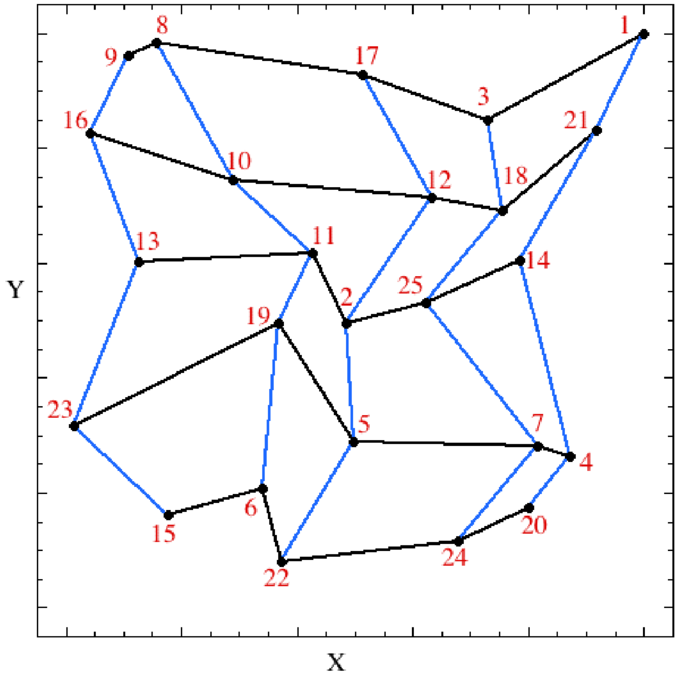
II. The Monotonic Lagrangian Grid

The underlying sorting and ordering algorithm and data structure is the MLG,[3-5] a free-Lagrangian data structure for storing the positions and other data needed to describe N moving bodies. The MLG has been used for two decades as the underpinning for various particle dynamics simulations, including molecular dynamics,[6] direct simulation Monte Carlo,[7] and exploratory missile defense[8-11] applications. A three-dimensional MLG data structure in Cartesian coordinates is defined by the constraints:

$$\begin{array}{llll}
x(i,j,k) \le x(i+1,j,k) & i=1,....,Nx-1; & j=1,.....,Ny; & k=1,.....,Nz \\
y(i,j,k) \le y(i,j+1,k) & i=1,....,Nx; & j=1,...,Ny-1; & k=1,....,Nz \\
z(i,j,k) \le z(i,j,k+1) & i=1,....,Nx; & j=1,.....,Ny; & k=1,.....,Nz-1,
\end{array} \qquad (1)$$

where N_x, N_y, and N_z are the number of objects or nodes (particles) in each direction., and $N = N_x \cdot N_y \cdot N_z$ is the total number of objects. Here, the meaning of "object" or "node" depends on the particular application. That is, for molecular dynamics simulations, a node may correspond to an atom; for direct simulation Monte Carlo applications, it corresponds to a group of molecules; for the present application, a node or object corresponds to an aircraft. Moreover, a node may be a dimensionless point or it may correspond to a physical object of any appropriate size. The constraints mean that each grid line in each spatial direction is forced to be a monotone index mapping. As an example, Figure 1 depicts a small two-dimensional (5x5) MLG. The MLG model developed here is three-dimensional in space; however we are showing a two-dimensional subset to explain the principles of the MLG. The image on the left shows the 25 MLG nodes (points in space) at their irregular spatial physical locations, while the table on the right shows the grid indices (in *i-j* space) of each node. Although the nodes are irregularly spaced (left figure), they are indexed regularly in the MLG by a monotonic mapping between the grid indices and the locations. Note that we generally index the vertical direction from bottom to top (as in a graph), rather than top to bottom (as in a display).

A node with three spatial coordinates has three indices in the MLG data arrays. Data relating to each node are stored in computer memory locations identified by these indices. Thus nodes that are close in physical space are always near neighbors in the MLG data arrays. A computer program based on the MLG data structure does not need to check *N-1* possible distances to find which nodes are close to a particular node. Rather, the indices of the neighboring nodes are automatically known because the MLG node indices vary monotonically in all directions with the Lagrangian node coordinates. For example, as shown in Figure 1, we automatically know that near neighbors of node 11 are nodes 16, 10, 12, 13, 2, 23, 19, 5, without having to check the distances of all 24 remaining nodes. The cost of most tracking algorithms using the MLG is dominated by the calculation of the interactions of nodes with their near neighbors, and thus the timing scales as *N*. For applications involving a large number of nodes, such as the air-traffic problem, knowing each node's nearest neighbors automatically, without having to check all *N-1* distances, results in significant computational savings.



5	9	8	17	3	1
4	16	10	12	18	21
3	13	11	2	25	14
2	23	19	5	7	4
1	15	6	22	24	20
J/I	1	2	3	4	5

Figure 1. An example of a two-dimensional MLG. *The figure shows a 2-D MLG (5x5) containing the x- and y-locations of 25 labeled nodes. The solid black (horizontal) lines show the x-links and the dotted blue (vertical) lines show the y-links. The table shows the regular grid indices of the nodes shown in the figure. That is, node-15 is indexed at i = 1, j = 1; node-6 is indexed at i = 2, j = 1, etc.*

When sorting nodes to MLG order, the simplest MLG algorithm sorts each axis individually. That is, for a 5x5 MLG, it sorts the first five points in the x-direction, then the next five points in the x-direction, etc. After all 25 points have been sorted in the x-direction, it then sorts all 25 points in the y-direction. As each axis becomes sorted, the sorting process may destroy monotonicity in the other axes. Therefore, the sorting process is repeated until all axes are monotonic. The MLG uses two sorting algorithms to put nodes in order. One is a bubble sort, in which each node is compared with the node immediately following it on that axis, and if they are not in monotonic order, their MLG position is switched. This bubble-sorting algorithm is most effective when the nodes are already partially pre-sorted. The other algorithm is a shell sort, in which each node is compared with one that is a half-axis length away, and if the two nodes are not in monotonic order, they are switched. This shell-sorting algorithm is best for completely random data.

III. Model

For a series of test scenarios, we have combined the MLG (for sorting and ordering the locations of aircraft) with new algorithms for collision avoidance and updating aircraft trajectories. The current model we test here consists of the following steps:

1. Assign each aircraft: an ID, an initial *x-, y-, z*-location, a final *x-, y-, z*-location, and an average speed;
2. Calculate an initial velocity vector for each aircraft, assuming a straight path between its initial and final destinations;
3. Put data (aircraft ID and current x-, y-, z-location) into the MLG data structure. The MLG is a two-dimensional array, where the first dimension corresponds to the number of sort variables, and the second dimension corresponds to the number of aircraft in the MLG. For this air-traffic application, the number of sort variables is four (x, y, z locations and ID number for each aircraft) and the number of aircraft is N, where $N = N_x \cdot N_y \cdot N_z$. Therefore, the MLG data structure is:

MLG (i, N), where $i = 1,\ldots,4$

$i = 1$: x-location
$i = 2$: y-location
$i = 3$: z-location
$i = 4$: aircraft ID

4. Sort aircraft into MLG order, based on x-, y- and z-location. Although the aircraft are sorted based only on physical location (so that we automatically know nearest neighbors), the aircraft IDs are included in the MLG data structure, so that we can cross-reference other information (velocity, flight time, final destination) for each aircraft. This additional information is not carried within the MLG, as it would increase memory requirements and slow down the sorting computation.
5. Perform diagnostics on the MLG to check for monotonicity and grid entanglement.
6. Avoid collisions as follows:
   Check nearest neighbors for potential collisions. For each node in the MLG, we check the current distance between it and its nearest two adjacent nodes in all three (x-, y- and z-) directions. If the distance between two nodes (aircraft) are within five miles of each other, then we project forward in time. The amount of time projected forward equals the amount of time needed for the slower aircraft to advance two miles. The locations of both aircraft are advanced by this same amount of time. If the projected distance between these two aircraft is less than the current distance, then the aircraft are considered to be moving toward each other, and the velocity vector of one of the aircraft is modified by 20 degrees. The 20 degree modification is made in the x-y plane only, so that the aircraft maintains the same altitude. This collision avoidance algorithm is illustrated in Figure 2.
7. Move aircraft, based on each aircraft's computed velocity vector. This is either the initial velocity vector (calculated in step #2) or the velocity vector that has been calculated in step #6 for collision avoidance.
8. At two minute intervals, recalculate the velocity vector for each aircraft, so that there is a straight path to their final *x-, y-, z-*locations. This step is periodically necessary because the aircraft have been detoured in step #6 to avoid collisions.
9. Increment the time variable, and return to step #4.

These nine steps described above represent the tracking strategy used in the tests described below. It would be straightforward to replace these with different algorithms for collision avoidance or aircraft separation maintenance, as appropriate.

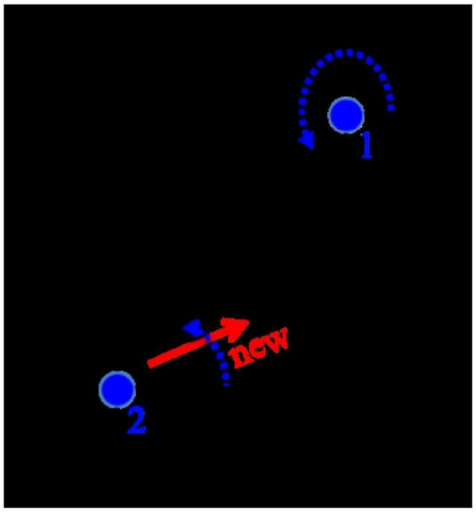


Figure 2. Schematic of collision avoidance maneuver algorithm between two aircraft that are within five miles of each other and moving towards each other. *The direction of the velocity vector for each aircraft (before the collision avoidance move) is shown by the solid black arrow, and the angle between that velocity vector and the positive x-axis is shown by the dotted blue arrow. The collision avoidance algorithm modifies the direction of one aircraft by 20 degrees, as shown by the red arrow, while maintaining the same velocity magnitude for that aircraft. The 20 degree modification is made in the x-y plane only, so that the aircraft maintains the same altitude.*

IV. Computational Experiments

We use the MLG to simulate air traffic flow in a 3-D volume to study the conditions under which the volume of airspace is controllable. One measure we can use to quantify the controllability of a volume of airspace is the number of potential collisions among aircraft within that volume. To try to quantify this, we conducted a series of tests that measured the number of initial and subsequent collision-avoidance moves needed to maintain a separation distance of five miles between *all* aircraft.

Consider a 500 mile x 500 mile x 100 mile volume of airspace with different numbers of aircraft, ranging from 125 to 8000, corresponding to MLGs of 5^3 to 20^3, respectively. Within that airspace, the aircraft have randomly chosen initial and final destinations, and their average speeds vary randomly between 400 to 500 miles per hour. Because the initial and final destinations have been chosen randomly, the flight times and the directions vary considerably. The simulations proceed until all aircraft have landed at their final destinations.

Figure 3 illustrates the MLG structure for 1000 aircraft at one point in time. For visual clarity, the 3-D image contains only every other aircraft on each axis. The 2-D image shows all 100 nodes in one x-y plane of the 3-D MLG, and also shows the ID number of each aircraft. The red, blue and green links represent the x-, y- and z-links, respectively, where the links represent the distance between each aircraft.

Figure 4 shows the locations of two approaching aircraft, at times *t1* through *t6*. The simulations use the collision avoidance maneuver (step #6) that prevents them from coming within five miles of each other at the same time. The result here is that one aircraft is redirected by 20 degrees in the x-y plane to separate them.

In each simulation, we tracked the number of collision-avoidance moves for each aircraft; this includes the initial maneuver when two aircraft first approach each other and any subsequent maneuvers required as a result of the initial maneuver. That is, in the collision avoidance process (step #6 above), after each aircraft is redirected to maintain a fixed separation distance, we then check to ensure that the modified trajectories do not create subsequent potential collisions. This process is repeated, as necessary, until all projected trajectories position the aircraft in locations that maintain the required separation distance. Figure 5 shows that the total number of collision avoidance maneuvers increases exponentially with the number of aircraft in the simulation.

One important metric to evaluate is the number of subsequent (secondary) potential collisions that arise after one aircraft is rerouted. Secondary maneuvers are considered to be all maneuvers required after the first redirection. Table 1 shows the maximum number of secondary collision avoidance maneuvers (at any one point in time) that were required to maintain this five mile separation distance. As expected, as the number of aircraft in the simulation increases, the number of required secondary maneuvers increases.

No. of Aircraft	Max No. of Secondary Maneuvers
125 (5**3)	3
216 (6**3	3
343 (7**3)	3
512 (8**3)	6
729 (9**3)	6
1000 (10**3)	6
1728 (12**3)	6
3375 (15**3)	6
5832 (18**3)	7
8000 (20**3)	7

Table 1. Maximum Number of Secondary Maneuvers for each simulation.

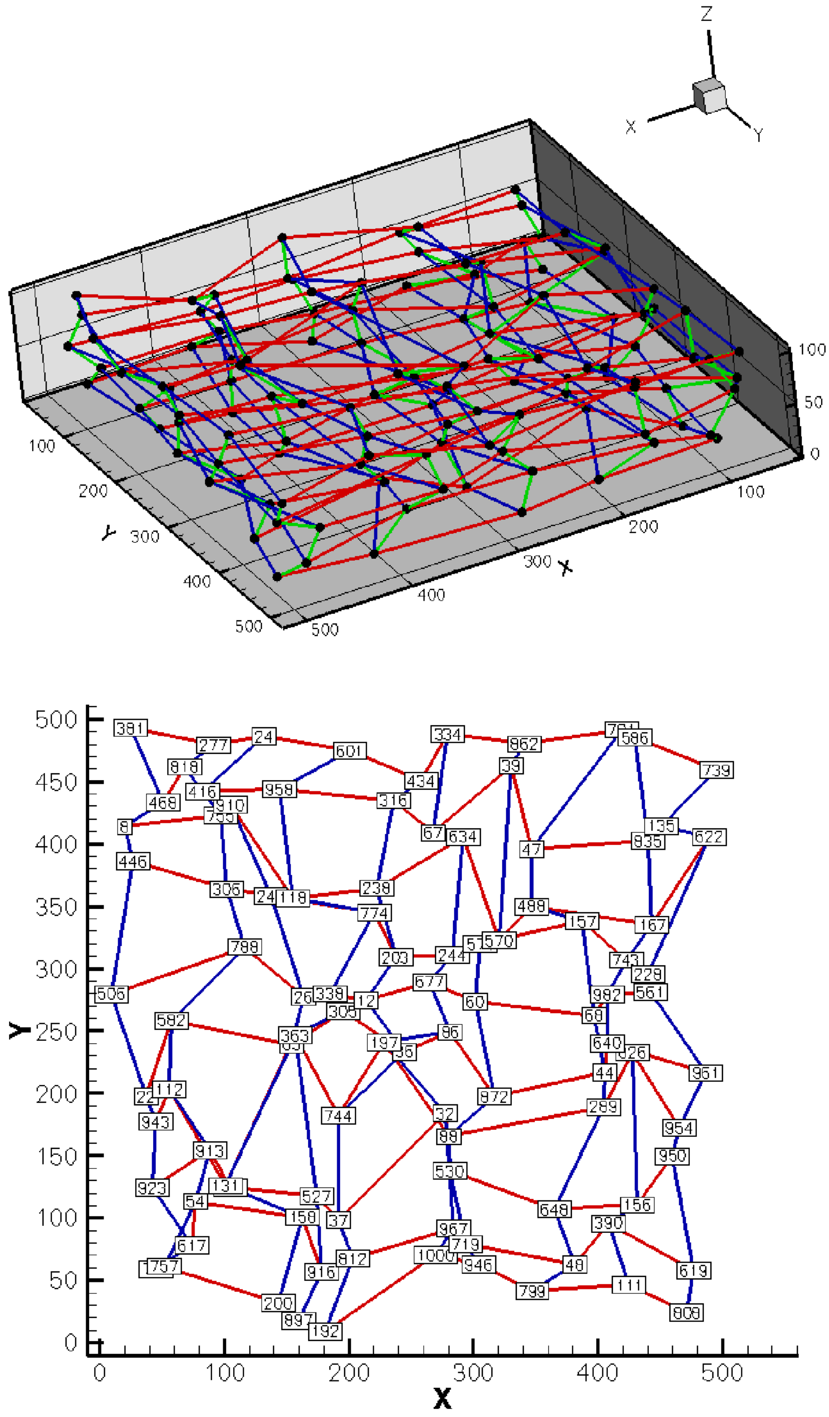


Figure 3. A 3-D MLG (top figure), and one 2-D plane (bottom figure) within that 3-D MLG, at one point in time, for a case involving 1000 aircraft. *(Top): The 3-D image shows every other node, on each axis, for visual clarity; that is, rather than showing all 1000 (10^3) nodes, it shows 125 ($10^3/8$) nodes so that the MLG structure is clearer. The black circles represent the MLG nodes (aircraft). The x-links are red, the y-links are blue, and the z-links are green. (Bottom): The 2-D image shows all 100 (10^2) nodes in one x-y plane of the 3-D MLG, and also shows the ID number of each node. The x-links are red and the y-links are blue.*

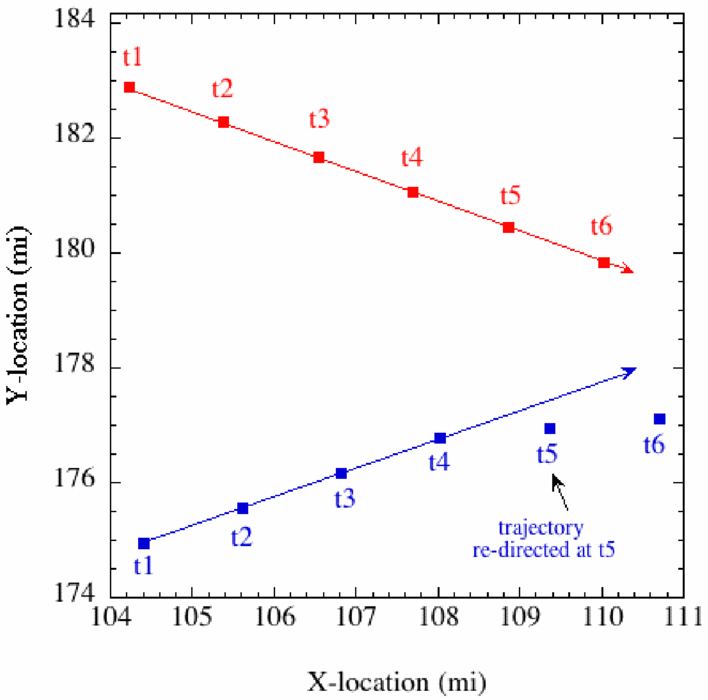


Figure 4. Simulation data showing collision avoidance maneuver between two aircraft. *The red and blue squares represent the location of each aircraft at specified times (t1, t2,.....), and the line through the squares corresponds to the intended horizontal trajectory (in the x-y plane) of that aircraft. As indicated in the collision avoidance algorithm discussion above, when two aircraft are within five miles of one another, and they are approaching one another, the direction of one aircraft is modified by 20 degrees (see Figure 2). As shown here, at t5, the blue aircraft is redirected to maintain the required separation distance from the red aircraft. Although the simulations are 3-D, the collision avoidance maneuvers are done in the x-y plane only, so that the aircraft maintains the same altitude.*

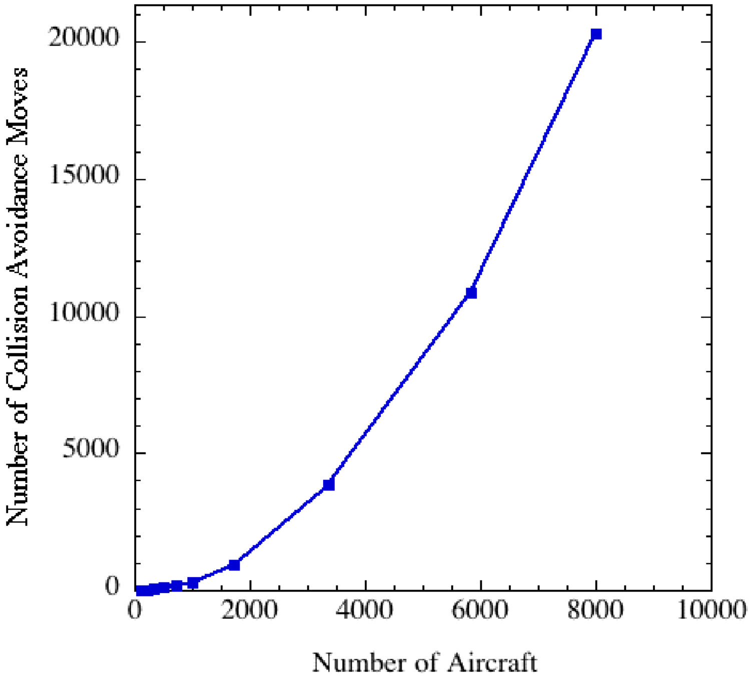


Figure 5. Total number of collision avoidance maneuvers as a function of number of aircraft in the simulation.

V. Discussion and Concluding Remarks

We have developed a dynamic global model for air traffic, ATMLG, to be used in a computational test-bed framework for evaluating approaches to global air systems control and optimization. The underlying algorithm used in the model is the MLG, which can quickly track and sort positions of many aircraft, both on the ground (at airports) and in the air. By using the fast sorting algorithm in the MLG, the simulations presented here (with 8000 aircraft) have used less than five minutes of CPU time, on a desktop Apple computer, to simulate five hours of physical time.

Our major target is the investigation of systemic phenomena in complex transport systems, starting with air transportation and eventually progressing to multimodal transportation. Of particular interest is predictive modeling that should enable active design of the system. Model development and assessment, especially in systems that include large numbers of interacting *autonomous* participants, requires massive computational experiments. Flexible, *fast* simulation, assisted by easy-to-use parallel and distributed computations, facilitates model building to a great extent. This is why, despite the availability of sophisticated air traffic simulators, such as FACET[12], we are interested in developing a fast simulator that can be easily incorporated into the overall computational framework for studying complex transport systems.[13] We plan to use the existing simulators, in addition to the MLG tool, for model validation.

An important aspect of the NGATS includes the increased use of automation for maintaining separation. Within this airspace, aircraft should be able to choose a route, and to have at least one degree of freedom of flexibility to resolve any conflict with other aircraft. In an effort to study these complexity operations and to better understand when a volume of airspace is manageable, we have conducted a series of numerical experiments to simulate air traffic flow in a 3-D volume, while keeping track of primary and subsequent collision avoidance maneuvers necessary to maintain a five mile separation distance between all aircraft. The simulation results show how the ability to maintain the required separation distance decreases as the number of aircraft in the volume increases.

In the near future, we plan to continue the development of ATMLG along several directions. Present computational experiments have been conducted with artificially populated airspace. The next steps will involve linking ATMLG with ETMS data, in order to have a more realistic view of moving aircraft. Collision avoidance maneuvers in the present system have been simplified. We now plan to include more sophisticated algorithms for detecting and preventing the loss of separation. Another direction is concerned with establishing quantified measures of airspace "fullness" or complexity and mathematical developments in support of establishing provable controllability, given specific algorithms for detecting and preventing the loss of separation.

Acknowledgments

The authors gratefully acknowledge support from NASA Langley Research Center (LaRC) under Interagency Agreement IA1-910 between LaRC and LCP&FD.

References

[1] Kaplan, C.R., Oran, E.S., Alexandrov, N., Boris, J.P., "The Monotonic Lagrangian Grid Particle Grid: A Fast Tracking Methodology for Air-Traffic Modeling,", AIAA Paper 2009-1635, American Institute of Aeronautics and Astronautics, Reston, VA, 2009.

[2] Kopardekar, P. Bilimoria, K. and Sridhar, B., "Initial Concepts for Dynamic Airspace Configuration," AIAA Paper 2007-7763, American Institute of Aeronautics and Astronautics, Reston, VA, 2007.

[3] Boris, J., "A Vectorized "Near Neighbors" Algorithm of Order N Using a Monotonic Logical Grid," *J. of Computational Physics*, **66**, 1 (1986).

[4] Picone, J.M., Lambrakos, S.G., Boris, J.P. and Jajodia, S., "Initial Comparison of Monotonic Logical Grid and Alternative Data Base Structures," NRL Memorandum Report 5860, Sept 30, 1986.

[5] Lambrakos, S.G. and Boris, J.P., "Geometric Properties of the Monotonic Logical Grid Algorithm for Near Neighbor Calculations," J. of Computational Physics, **73**, 183 (1987).

[6] Lambrakos, S.G., Nagumo, M., Boris, J.P., Oran, E.S., and Gaber, B., "Molecular Dynamics Simulation of a Lipid Bilayer Using Novel Monotonic Logical Grid (MLG) and Multiple Constraint Relaxation (MCR) Algorithms," Biophysical Journal, **51**, A440 (1987).

[7] Cybyk, B., Oran, E.S., Boris, J.P., and Anderson, J.D., "Combining the Monotonic Lagrangian Grid with a direct simulation Monte Carlo Model," J. of Computational Physics, **122**, 323 (1995).

[8] Boris, J.P., Picone, J.M., and Lambrakos, S.G., "BEAST: A High-Performance Battle Engagement Area Simulator/Tracker," NRL Memorandum Report 5908, December 31, 1986.

[9] Jajodia, S., Meadows, C.A., Boris, J.P., Picone, J.M., and Lambrakos, S.G., "NRL Research in Database Management for the SDI Battle Management System," NRL Memorandum Report 5921, January 30, 1987. Distribution limited.

[10] Picone, J.M., Boris, J.P., Lambrakos, S.G., Uhlmann, J. and Zuniga, M., "Near-Neighbor Algorithms for Processing Bearing Data," NRL Memorandum Report 6456, May 10, 1989.

[11] Kolbe, R.L., Boris, J.P. and Picone, J.M., "Battle Engagement Area Simulator/Tracker," NRL Memorandum Report 6705, October 8, 1990.

[12] Bilimoria, K., Sridhar, B., Chatterji, G., Sheth, K., Grabbe, S.: FACET: Future ATM Concepts Evaluation Tool, 3rd USA/Europe Air Traffic Management R&D Seminar, Napoli, Italy, June 2000.

[13] Alexandrov, N., Kincaid, R.K., Vargo, E.P., “Toward Optimal Transport Networks”, AIAA Paper 2008-5814, American Institute of Aeronautics and Astronautics, Reston, VA, 2008.