

# Computational Approaches to Simulation and Optimization of Global Aircraft Trajectories

Hok K. Ng<sup>1</sup>

*University of California, Santa Cruz, Moffett Field, CA 94035-1000*

Banavar Sridhar<sup>2</sup>

*NASA Ames Research Center, Moffett Field, CA 94035-1000*

**This study examines three possible approaches to improving the speed in generating wind-optimal routes for air traffic at the national or global level. They are: (a) using the resources of a supercomputer, (b) running the computations on multiple commercially available computers and (c) implementing those same algorithms into NASA's Future ATM Concepts Evaluation Tool (FACET) and compares those to a standard implementation run on a single CPU. Wind-optimal aircraft trajectories are computed using global air traffic schedules. The run time and wait time on the supercomputer for trajectory optimization using various numbers of CPUs ranging from 80 to 10,240 units are compared with the total computational time for running the same computation on a single desktop computer and on multiple commercially available computers for potential computational enhancement through parallel processing on the computer clusters. This study also re-implements the trajectory optimization algorithm for further reduction of computational time through algorithm modifications and integrates that with FACET to facilitate the use of the new features which calculate time-optimal routes between worldwide airport pairs in a wind field for use with existing FACET applications. The implementations of trajectory optimization algorithms use MATLAB, Python, and Java programming languages. The performance evaluations are done by comparing their computational efficiencies and based on the potential application of optimized trajectories. The paper shows that in the absence of special privileges on a supercomputer, a cluster of commercially available computers provides a good option for computing wind-optimal trajectories for national and global air traffic system studies.**

## I. Introduction

The desire to improve the efficiency of air traffic routes requires the computation of the fuel and time optimal trajectories between the city-pairs and comparison with the current baseline routes. The computation of the optimal trajectory depends on the aircraft dynamics model, fuel flow calculations, wind conditions and a cost function. Uncertainties in flight information, operations, and weather result in the requirement to generate large number of optimal aircraft trajectories to estimate range of safety, capacity, efficiency and environmental impacts based on various strategic plans and concepts.

There is extensive literature<sup>1-5</sup> on the computation of optimal trajectories for single or few aircraft involving 3-degrees of freedom (DOF) or 6-DOF models, different cost functions and wind conditions. Research at NASA has developed wind-optimal routes for domestic flights<sup>2</sup> and oceanic flights<sup>3-5</sup> using 3-DOF models, wind data, different cost functions and constraints. The MATLAB computation for generating wind-optimal routes using an Apple Mac Pro with dual 2.66 GHz 6-core Intel Xeon processors and 16 GB memory takes approximately 30 sec per trajectory depending on the distance between city-pairs. The generation of one set of wind-optimal routes per day in the National Airspace System and, assuming 30,000 flights, takes approximately 10 days. Based on the Mac Pro, a simulation of wind-optimal global air traffic routes of about 90,000 flights by the OAG<sup>6</sup> for 2014, will take more than a month on a desktop computer like the Mac Pro. It is clear that alternative computational approaches are needed to generate wind-optimal aircraft trajectories in order to support various trade-off studies involving global air

<sup>1</sup>Research Scientist, U.C. Santa Cruz, MS 210-8, Member AIAA.

<sup>2</sup>Senior Scientist for Air Transportation Systems, Aviation Systems Division, Fellow.

traffic.

This paper describes preliminary results in an effort to use the Pleiades supercomputer in the NASA Advanced Supercomputing (NAS)<sup>7</sup> facility at Ames Research Center for computing global air traffic simulation of wind-optimal routes. It also presents initial setups for running the same computations on a computer cluster in the Airspace Concept Evaluation System (ACES)<sup>8</sup> laboratory at NASA Ames Research Center. The evaluation of potential benefits is based on the computational efficiency gained from accessing NAS and ACES systems for the generation of wind-optimal routes worldwide for one day. The MATLAB wind-optimal code is also rewritten into the Java programming language with modified numerical techniques for further computational efficiency and being compatible with FACET<sup>9</sup>.

Section II describes the three potential computational approaches and a brief description of the applied trajectory optimization algorithm. Section III introduces the worldwide global flight schedules and global atmospheric data. Section IV compares the computational efficiency of using a Mac Pro, ACES computer cluster, and Pleiades supercomputer. Section V evaluates the accuracy of the MATLAB program and the FACET programs under a useful application of wind-optimal trajectories to assess the potential performance changes due to algorithm modifications made in FACET programs.

## II. Computational Approaches

The trajectory optimization algorithm implemented for this study adapts a practical optimization approach<sup>2-5</sup> by assuming a typical structure for an aircraft trajectory and focuses on optimizing direct operating cost during cruise when the time and fuel savings have the most impact. A typical aircraft trajectory consists of an initial climb, a steady-state cruise, and a final descent. Typically, the cruise portion is the longest by far. Here, aircraft performance is optimized for the cruise phase only. The aircraft optimal heading during cruise is the solution of the Zermelo problem<sup>1</sup> derived on a spherical Earth surface in the absence of constraints. The horizontal trajectory segments are optimized based on the objective cost values i.e. cost-to-go associated with optimal extremals i.e. potential optimal trajectories generated by forward or backward integrating the dynamical equations for optimal heading and aircraft motion from various points in the airspace. This computationally efficient algorithm searches for optimal solutions by combining calculus of variations and dynamic programming. The details of the optimization procedure are described in references<sup>[2, 4-5]</sup>.

The implementation of the trajectory optimization algorithms use MATLAB, Python, and Java programming languages. The aircraft trajectory optimization algorithm is originally implemented in MATLAB and is called the MATLAB M-Stage program in this paper. The MATLAB M-Stage program optimizes an aircraft trajectory in multiple stages; and is run on a Mac Pro with dual 2.66 GHz 6-core Intel Xeon processors and 16 GB memory for computing air traffic simulation of wind-optimal routes at the regional level. With minor modifications, the MATLAB M-Stage program is ready for parallel executions on NASA's ACES computer cluster. The NAS facility currently does not support multiple processors for MATLAB computations but it does for the Python language. For this purpose, the Python M-stage program is written based on MATLAB M-Stage program. Python is chosen for its quick implementation and less development efforts due to required numerical techniques readily available in Python numerical libraries. In addition to the computational enhancement through parallel processing on computer clusters, this study also re-implements MATLAB M-Stage program in Java programming language for reduction of computational time through algorithm modifications and for running with FACET on a Mac Pro. The integration of Java programs with FACET required relatively more development effort. However, it facilitates integration of the new features which calculate time-optimal routes between worldwide airport pairs in a wind field for use with existing FACET applications.

Section IIA outlines the processing of Python M-Stage program on NAS supercomputers. Section IIB summarizes the parallel computations of MATLAB M-Stage program on ACES computer cluster. Section IIC considers modifications to the optimization algorithms to reduce the computational time and describes the efforts related to FACET development.

### A. NASA Advanced Supercomputing Facility

The NAS facility encompasses several supercomputers that are available to users<sup>2</sup>. The primary compute system is Pleiades, which includes 163 racks with 11,176 total nodes containing 184,800 CPU cores. Pleiades contains three different types of processors: Westmere, Sandy Bridge, and Ivy Bridge. A submitted job will use only one type of processor. Both Westmere and Ivy Bridge nodes are used in this study. Sandy Bridge nodes were not used, as they do not contain enough memory. Another supercomputer used in this study is Merope, with 8 racks, 512 nodes, and 6,144 CPU cores. This system contains only Westmere nodes. Selection of processors with sufficient memory is

important because trajectory optimization for long-haul flights requires more memory. The results shown in this paper used Ivy Bridge nodes that each has twenty 2.8 GHz Intel Xeon processors with 3.2 GB memory per CPU core.

All of the supercomputers are available only through the submission of batch jobs. To prepare and submit jobs, users log into one of many front-end systems. Each supercomputer has a set of associated front-ends. The user edits and compiles the program on the front-end systems and can also execute the programs for debugging and testing. Pleiades offers a development queue for program development and testing and a normal queue for regular computations. The wait time, which is defined as the time period after a job submission and before its execution, is shorter for development queue that has a higher priority than the normal queue. However, it has limitations that only one job may be submitted to this queue per user each time and the maximum execution time is two hours to discourage over use.

The NASA's NAS environment includes three file systems that are relevant to the current study. The user home directories are accessible to a user logged onto a Pleiades or Merope Front End computer or to processes running on Pleiades or Merope. This file system contains all of the program source and binary files, and contains the default directory when the Python program is running on Pleiades or Merope. The nobackup file system is available to a user logged into a front-end system, a supercomputer, a Lou front end, or Lou. This is the common area for files generated by programs. The Lou file system is designed for long-term storage of files. Output files from the Python program are copied here for copying to local systems at the Aviation Systems Division and kept here for archival storage.

Compute jobs are submitted to the supercomputers by using the Portable Batch System (PBS) utility, developed at NASA Ames Research Center. With more than 10,000 instances of the program executing simultaneously, synchronization between the instances is necessary to ensure that all cases are processed, and each case is only processed once. The current study uses a Bourne shell script with PBS directives that specify the parameters for the batch job to synchronize the program instances.

## **B. NASA's ACES Computer Cluster**

A computer cluster in the ACES laboratory at NASA Ames Research Center is used for running the MATLAB M-Stage program for global trajectory optimization. The computational efficiency for this approach will be compared to that of NAS Supercomputers. The computer cluster consists of sixteen 2.3GHz 8-core Intel Xeon processors with 64 GB memory. The MATLAB Parallel Computing Toolbox is used for developing MATLAB programs that are ready for parallel executions on a multicore desktop computer. They are then scaled up using MATLAB Distributed Computing Server (MDCS)<sup>10</sup> that is setup on the computer cluster. MDCS supports running MATLAB programs in command line and through batch jobs. It also includes a built-in cluster job scheduler to manage multiple applications simultaneously running on the cluster computer.

For the current study, users log in to the client machine, which is connected to the MDCS running on the cluster computer, remotely from the local desktop. The parallel computing toolbox is installed on the client machine and computation jobs are submitted from the client node. The user can edit the MATLAB programs on the client node and can also execute the programs for debugging and testing.

A common file system can be built for data sharing between the client node and the cluster computer. In general, the client node contains all of the program source, data files, and output results for the MATLAB program. After completing all computations, the results can be copied from the client node to local computer systems.

## **C. FACET: Future ATM Concepts Evaluation Tool**

FACET is a simulation and analysis tool developed by NASA Ames Research Center to provide a simulation environment for concepts and technology development and evaluation to support future air traffic management systems<sup>9</sup>. FACET models system-wide en-route airspace operations at the U.S. national level and process over 5,000 aircraft on a single desktop computer. FACET was designed with a modular architecture to facilitate rapid prototyping of advanced ATM concepts. Each ATM concept application is implemented as an individual module linked to the core structure of FACET. This core provides modeling features (e.g., airspace and 4D trajectories) required for the evaluation of almost any ATM concept.

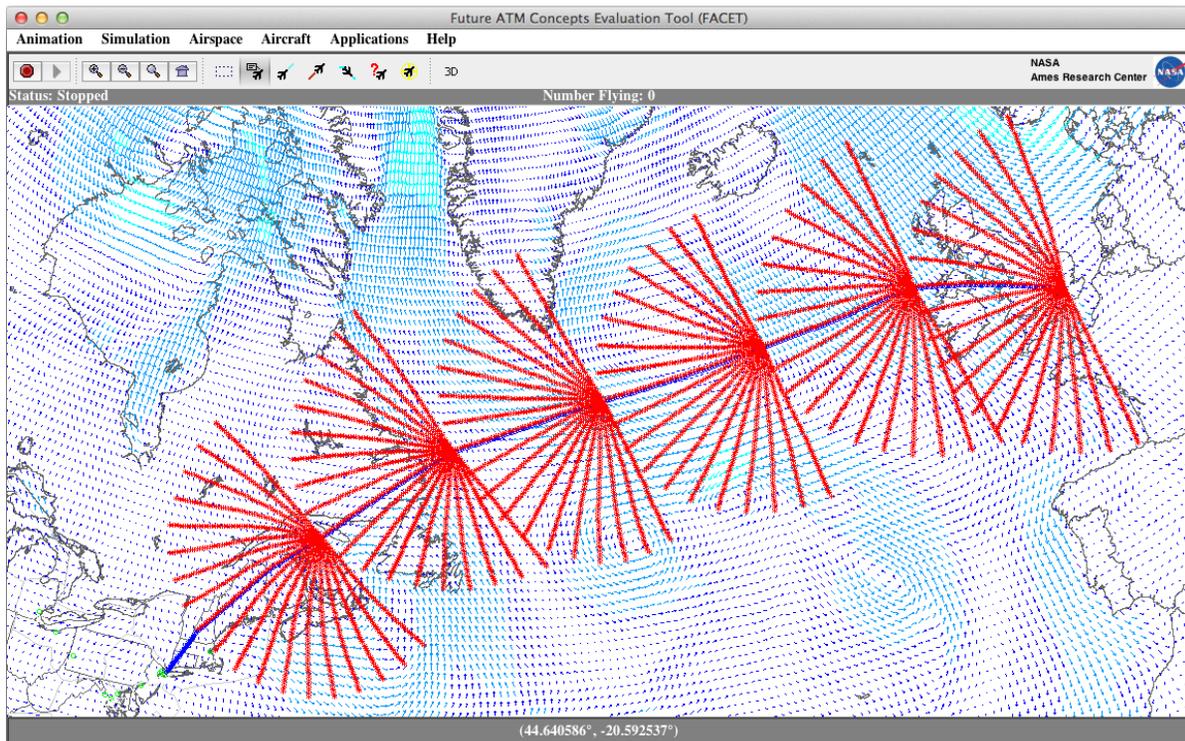
In addition to the utilizations of parallel processing on computer clusters for global trajectory optimization, this study also re-implements the MATLAB M-Stage program in Java programming language with modified numerical techniques to reduce the computational time. The development of Java programs for FACET required relatively more development effort, but it facilitates integration of the new features, which calculate time-optimal routes between worldwide airport pairs in a wind field, for use with existing FACET applications.

### 1. Aircraft Trajectory Simulation

FACET models four-dimensional (4D) aircraft trajectories in the presence of winds using round-earth kinematic equations. Aircraft can be flown along flight plan routes or direct (great circle) routes as they climb, cruise, and descend according to their individual aircraft-type performance models. Performance parameters are obtained from data table lookups. Heading and airspeed dynamics are also modeled. The airspace model includes geometric descriptions of Air Route Traffic Control Centers (ARTCCs or “Centers”), their sectors, Victor Airways and Jet Routes, as well as the locations of airports and fixes (navigation aids and airway intersections). For the purpose of simulating global aircraft trajectories, the current airspace domain in FACET was modified from US continental airspace to global scope. Global Forecast System (GFS) provides wind data over the globe. The worldwide commercial air traffic schedules describing aircraft type, departure time, city-pairs are provided by data supplied by OAG Worldwide LLC. The integration of GFS and OAG data enables the simulation of global aircraft trajectories.

### 2. Aircraft Trajectory Optimization

FACET integrates three distinct components (1) Data computation using the “C” programming language, (2) display of information through a GUI written in Java; and (3) API using the Java programming language. The FACET API is a set of routines for building software applications to support development of future ATM systems. In this study, two versions of the optimization algorithms are programmed in and implemented with the FACET API. They are called FACET M-Stage program and FACET 0-stage program, respectively. The FACET M-Stage program optimizes an aircraft trajectory in multiple stages and FACET 0-Stage program optimizes an aircraft trajectory in one step for further reduction of computational time. The original MATLAB M-Stage program applied interpolation techniques provided by MATLAB numerical library for the estimation of optimal aircraft headings and cost-to-go values at various locations along the potential optimal routes within the search space. FACET M-Stage program and FACET 0-Stage program are implemented using only self-implemented numerical techniques without using build-in third party numerical library.



**Figure 1. The wind-optimal trajectory depicted in blue line and the forward wind extremals depicted in red lines for a transatlantic flight from London to New York on August 19, 2014 calculated by FACET API.**

The FACET M-Stage program optimizes an aircraft trajectory in multiple stages and based on cost-to-go values along wind-optimal extremals. For illustration purpose, figure 1 depicts the horizontal wind-optimal trajectory computed by FACET API in the blue line for a transatlantic flight from London to New York on August 19, 2014.

The wind vectors at 36,000 feet are plotted in cyan and blue for illustration of wind directions and magnitudes. The forward wind-optimal extremals generated by forward integration using a range of different initial heading angles at start of each optimization stages are plotted in red. These forward extremals provide the minimum cost-to-go values from the starting point to any points along them. Using a range of different final heading angles at the destination, another collection of backward wind-optimal extremals is generated by backward integration. These backward extremals provide the minimum cost-to-go values to the destination from any points along them. Interpolation techniques are then applied to estimate the total cost-to-go values from any points in the covered airspace region using the cost-to-go values provided by the forward extremals and the backward extremals. Based on those, the wind-optimal trajectory is determined by minimizing the total cost-to-go values from the origin to the destination. The FACET 0-Stage program is a simplified version of FACET M-Stage program that optimizes an aircraft trajectory based on backward extremals only. The optimal initial aircraft heading is computed by interpolations using the closest available wind-optimal extremals at the origin.

Table 1 summarizes the initial performance evaluation of the three programs. It lists aircraft travel time of the wind-optimal trajectories computed by them for flights operating between five popular city pairs during August 19, 2014. It also lists the computational times when they are running on a Mac Pro with dual 2.66 GHz 6-core Intel Xeon processors and 16 GB memory. Note that the average computational time of the MATLAB M-Stage program is about 29.24 seconds in this example. The aircraft travel time of each wind-optimal trajectory computed by Matlab M-stage program is almost the shortest for the chosen city pairs except from New York to London. FACET M-Stage program provides comparable performance overall and reduces the average computational time of MATLAB M-Stage program by a factor of 15 to around 1-2 seconds. The FACET 0-Stage program offers the shortest computational times range between 0.5 and 0.9 seconds and provides reasonably close results except for the flight from New York to Paris. In this case, the program terminates without satisfying the convergence tolerance due to the approximations made for the program. The FACET 0-Stage program can potentially reduce the average computational time of MATLAB M-Stage program by a factor of 40.

A useful application of the wind-optimal trajectories is the assessment of potential aircraft travel time and fuel savings due to winds. Hence, the performances of the three programs are evaluated under this potential application in Section V based on a group of transatlantic flights during the month of January 2012.

**Table 1. Performance evaluations for the FACET M-Stage, FACET 0-Stage and MATLAB M-Stage programs.**

Origin-Destination	Aircraft Travel Time (minutes)			Computational Time (seconds)		
	FACET M-Stage	FACET 0-Stage	MATLAB M-stage	FACET M-Stage	FACET 0-Stage	MATLAB M-Stage
New York-London	350	355	351	2.0	0.91	25.1
Chicago-London	409	409	409	2.1	0.74	31.7
New York-Paris	371	Fail	366	1.6	0.61	28.5
New York-San Francisco	298	298	298	1.0	0.56	18.6
Dallas-London	488	491	488	2.3	0.76	42.3

### III. Flight and Atmospheric Data

In this study, global flight schedules are obtained from the OAG. Preliminary records from the OAG data file show that there are 96,682 active flights with 203 aircraft types flown for Thursday, June 26, 2014. The top 10 airports worldwide that have the most number of departures and arrivals are listed in Table 2. The ten most frequently flown aircraft types are shown in table 3. The global flight schedules and aircraft data acquired from OAG are applied for the preparation of simulating global trajectory optimization scenarios on different computing platforms.

The computation of aircraft trajectories uses atmospheric data such as wind and temperature provided by the Global Forecast System (GFS). GFS is a global numerical weather prediction computer model run by the National Oceanic & Atmospheric Administration four times a day. It produces forecasts up to 16 days, and produces a forecast for every 3<sup>rd</sup> hour for the first 180 hours, and after that, every 12 hours. The horizontal resolution is roughly equivalent to 0.5×0.5 degree latitude/longitude. GFS data has 64 unequally spaced vertical isobaric pressure levels ranging between 0.25-1000 mb, with enhanced resolution at low and high altitude.

For the purpose of trajectory optimization with respect to wind conditions for flights during cruise, the horizontal wind vectors are obtained by interpolations for 21 altitude layers between flight levels 240 through 440. The wind shear is pre-computed for every grid points and altitude layers in MATLAB and saved as a binary file.

**Table 2. Ten busiest airports during June 26, 2014.**

Rank	Airports	Total Departure and Arrivals	Rank	Airports	Total Departure and Arrivals
1	Chicago (KORD)	2,608	6	Beijing (ZBAA)	1,488
2	Atlanta (KATL)	2,387	7	Charlotte (KCLT)	1,436
3	Dallas (KDFW)	1,936	8	Frankfurt (EDDF)	1,371
4	Los Angeles (KLAX)	1,646	9	Houston (KIAH)	1,370
5	Denver (KDEN)	1,565	10	London (EGLL)	1,338

**Table 3. Ten most common aircraft types during June 26, 2014.**

Rank	Aircraft Types	Total	Rank	Aircraft Types	Total
1	Airbus A320	14,036	6	Airbus A321	3,764
2	Boeing 737-800	12,252	7	Embraer E-190	2,285
3	Boeing 737-700	6,861	8	Bombardier DH8D	2,222
4	Airbus A319	6,279	9	ATR 72	2,199
5	Bombardier CRJ-900	4,633	10	Embraer ERJ 140	1,974

#### IV. Efficiency of the Computing Platforms

This section presents the wind-optimal trajectories computed by the three computational platforms. It compares the computational efficiency for generation of global aircraft trajectories in wind fields using a Mac Pro, NAS supercomputer, and ACES computer cluster, respectively. The results are based on applying the optimal trajectory algorithm to calculate an aircraft trajectory in the presence of winds that minimizes the aircraft travel time. Typical cruise speed and altitudes are chosen based on Eurocontrol's Base of Aircraft Data Revision 3.6 (BADA)<sup>11</sup> given the aircraft type.

Wind-optimal aircraft trajectories are computed using global air traffic schedules from OAG for June 26, 2014. Although the flights were taken from the schedule for June 26, 2014, they were used in conjunction with the wind data for June 26, 2010. The wind data were chosen out of the set of available data but do not affect the results for examining run-time performance of the current study. The flight operations involving unknown airports are neglected. The optimal trajectories for flights that have expected travel time less than one hour are approximated by the great-circle route to reduce the total number of required computations. Only one optimal trajectory is calculated for the flights with the same origin and destination airports and departing within the same 6-hour weather data window because they have the same optimal trajectory, to eliminate the redundancy in the optimization process. This leaves a total of 34,123 flights for which optimized routes are calculated. Figure 2 plots the wind-optimal trajectories worldwide based on flight schedules for June 26, 2014 and wind data for June 26, 2010.

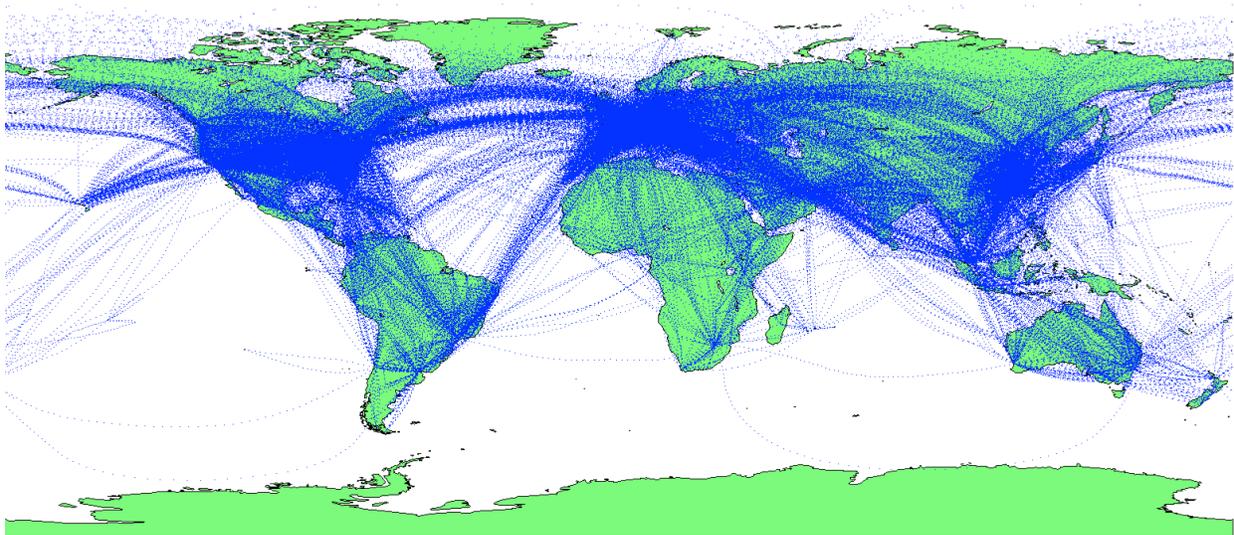
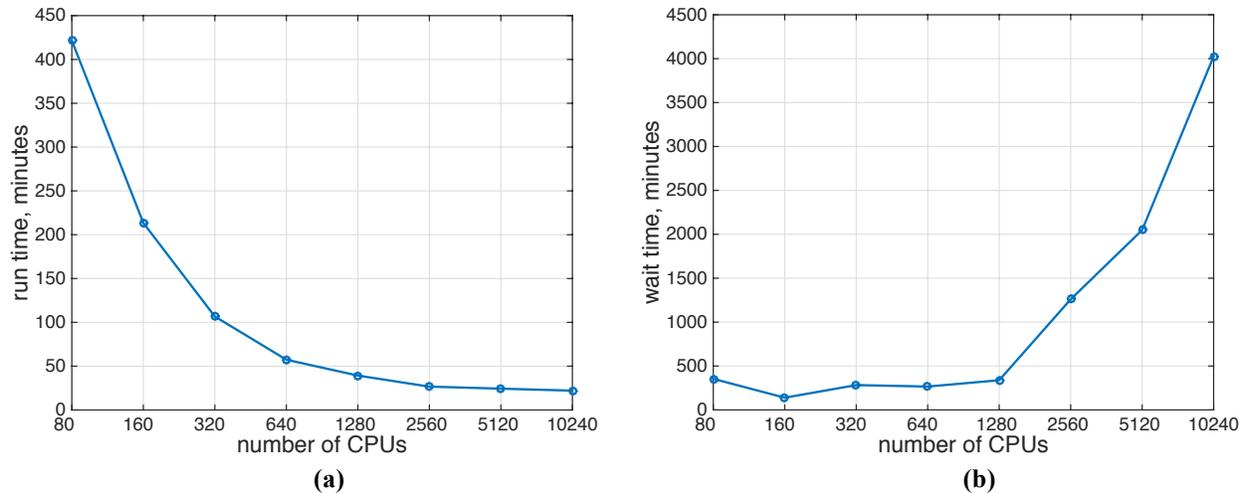
**Figure 2. Wind-optimal trajectories worldwide for June 26, 2010.**

Figure 3 presents the run time in (a) and wait time in (b) on Pleiades for computation of 34,123 optimal trajectories when running Python M-Stage program using various numbers of CPUs ranging from 80 to 10,240 CPUs. The values for run time and wait time are listed in Table 4. They are obtained based on a total of 33 jobs submitted between August 2014 and September 2014 with at least four jobs submitted for same number of CPUs. In this example, results show that the average run time on Pleiades for 34,123 optimal trajectories is about 22 minutes or i.e. 0.039 sec/trajectory when 10,240 CPUs are used. In this case, the average wait time in the queue after the job submission is about 4,027 minutes i.e. 2.8 days. The run times for using 2,560+ CPUs are similar and less than 30 minutes, but the wait times jump to above 20 hours. When 640 CPUs are used, the computations take about 57 minutes i.e. 0.1 sec/trajectory with the average wait time of 266 minutes. It is observed that the average turnaround time that includes run time and wait time is the shortest for the case when 640 CPUs are used due to a long wait period for requesting large number of CPUs.



**Figure 3. Run time and wait time on Pleiades with different number of CPUs employed for generation of the wind-optimal trajectories worldwide on June 26, 2010; Run time is Fig. 3a and wait time is Fig. 3b.**

When the computations for 34,123 optimal trajectories are performed by running MATLAB M-Stage program on the ACES cluster computer using 100 MATLAB workers, the computational time is about 453 minutes i.e. 0.8 sec/trajectory. This is a significant improvement over the previous desktop method, which would take around 12 days to complete. There is no wait time for this particular case since there is no other MATLAB job submitted to the computer cluster at the time. In this example, the computations done in NAS using 160 CPUs to 1280 CPUs provide the shorter turnaround times than the ACES Cluster.

Neglecting the wait time, the shortest estimated computational time for generation of wind-optimized global aircraft trajectories for one day is about 22 minutes when 10,240 CPUs on Pleiades are used. Including the wait time the shortest estimated turnaround time for the same computation is about 5.4 hours when 640 CPUs on Pleiades are used.

**Table 4. Computational performance on 3 platforms for different number of threads.**

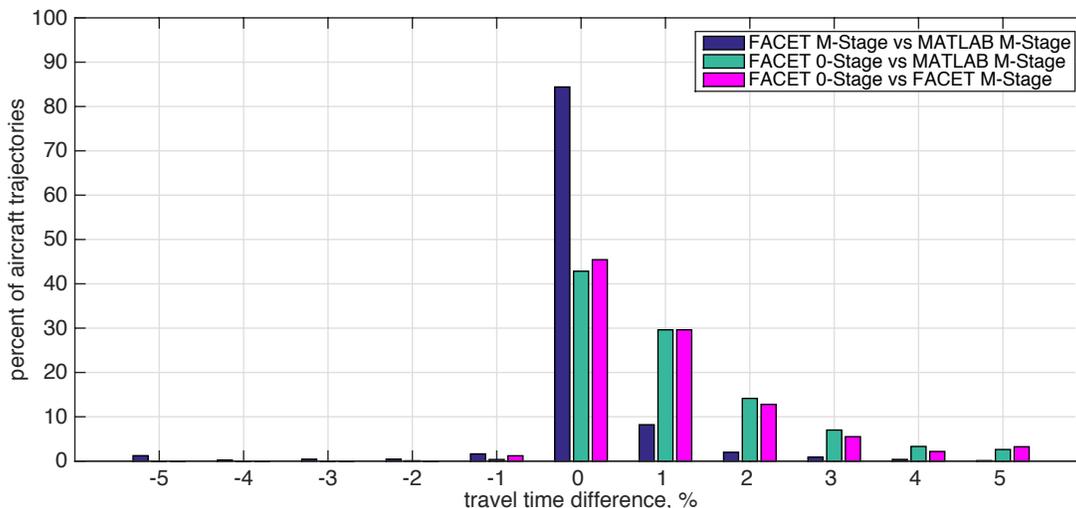
Algorithm	Platform	Threads	Run time/trajectory (seconds)	Total run time (hours)	Wait time (hours)
MATLAB M-Stage	Mac Pro	1	30	285.30	0
MATLAB M-Stage	ACES Cluster	100	0.80	7.55	0
Python M-Stage	NAS Pleiades	80	0.74	7.03	5.9
Python M-Stage	NAS Pleiades	160	0.38	3.57	2.3
Python M-Stage	NAS Pleiades	320	0.19	1.78	4.7
Python M-Stage	NAS Pleiades	640	0.10	0.95	4.4
Python M-Stage	NAS Pleiades	1,280	0.07	0.65	5.6
Python M-Stage	NAS Pleiades	2,560	0.05	0.45	21.0
Python M-Stage	NAS Pleiades	5,120	0.04	0.40	34.1
Python M-Stage	NAS Pleiades	10,240	0.04	0.37	67.1

The computational time can be further reduced by the application of FACET M-Stage program and FACET 0-Stage program that were written in Java with modified numerical techniques. As shown in Section IIC, the FACET M-Stage program reduces the average computational time of MATLAB M-Stage program by a factor of 15 on the Mac Pro; and FACET 0-Stage program can potentially reduce the average computational time of MATLAB M-Stage program by a factor of 40. In the future, Python M-Stage program running on NAS can be replaced by the FACET M-Stage program and FACET 0-Stage program. When considering only computational time, the computation of daily wind-optimized aircraft trajectories worldwide can take potentially 1.5 minutes for running FACET M-Stage program on Pleiades using 10,240 CPUs; and it will take about 33 seconds when running FACET 0-Stage program. The wait time for requesting a large number of CPUs on NAS could be reduced by negotiating a higher priority for the submitted jobs to be able to compute multiple air traffic scenarios for long periods of time.

## V. Accuracy of MATLAB and FACET Algorithms

Due to the potential computational enhancement gained from the FACET programs, this section compares the accuracy and the performance of the aircraft trajectories optimized by FACET M-Stage, FACET 0-Stage, and MATLAB M-Stage programs. It is assumed that the MATLAB M-Stage algorithm provides the highest accuracy with the most computational time. An important application for the wind-optimal aircraft trajectories is the assessment of potential impact of winds to flight path planning for air traffic worldwide. The impact of winds is greater for long-haul flights. The wind-optimal trajectories are generated for transatlantic air traffic for the entire month of January 2012. The flight schedules and aircraft data are provided by FAA's Traffic Flow Management System (TFMS). A total of 10,682 eastbound transatlantic flights and 11,023 westbound transatlantic flights are selected for the evaluation based on all the information needed to make the wind-optimal computations. The results are used to evaluate the performance of the three optimization programs by assessing the travel time of each wind-optimal trajectory for various airport pairs and aircraft types over this period.

The FACET 0-Stage program successfully calculates a total of 16,441 wind-optimal aircraft trajectories for the 9,541 eastbound flights and 6,900 westbound flights. These flights are selected for comparison with the wind-optimal trajectories calculated by MATLAB M-Stage program and FACET M-Stage program. Note that the convergence rate can be improved by increasing the backward wind extremals.



**Figure 4. Percent travel time difference between the aircraft trajectories computed by different approaches for transatlantic flights during January 2012.**

The blue bars in figure 4 plot the travel time differences of the wind-optimal trajectories computed by the FACET M-Stage program and the MATLAB M-Stage program and the evaluation of the two FACET implementations. About 84% of the wind-optimal trajectories computed by the two programs have the same travel time. Approximately 10% of the wind-optimal trajectories calculated by the FACET M-Stage program have 1%-2% travel time greater than those computed by MATLAB M-Stage program. The differences are due to the approximation of interpolations made for FACET M-Stage program and can be reduced by increasing the number of forward wind extremals. The FACET M-Stage program provides aircraft trajectories that have overall similar performance to those computed by MATLAB M-Stage program with computational speed approximately 15 times

faster.

The green bars and magenta bars present the results for the FACET 0-stage program compared with the MATLAB M-Stage and FACET M-Stage algorithms, respectively. About 43% of the wind-optimal trajectories computed by the FACET 0-Stage program and the MATLAB M-Stage program have the same travel time. Approximately 44% of the wind-optimal trajectories calculated by FACET 0-Stage program have travel time longer than those computed by MATLAB M-Stage by 1%-2%; and almost 14% have travel time greater than 3%. The 0-Stage algorithm has a bigger performance difference due to the simplification made using only backward wind extremals in addition to approximation of interpolation technique but with computation efficiency about 40 times faster than MATLAB M-Stage program. In this example, there is clearly a tradeoff that exists between computation speed and optimization results.

## VI. Conclusion

This paper provides three approaches for computing global air traffic simulation of wind-optimal routes. They are: (a) using the resources of a supercomputer, (b) running the computations on multiple machines and (c) using FACET. Wind-optimal aircraft trajectories are computed using global air traffic schedules for June 26, 2014. Neglecting the wait time, the shortest estimated computational time for the generation of wind-optimized global aircraft trajectories for one day is about 22 minutes when 10,240 CPUs on Pleiades are used. Including the wait time the shortest estimated turnaround time for the same computation is about 5 hours 24 minutes when 640 CPUs on Pleiades are used; the computation takes about 57 minutes i.e. 0.1 sec/trajectory with the average wait time of 267 minutes. When the same computation is performed on ACES computer cluster using 100 MATLAB workers, the turnaround time is about 7 hours 33 minutes i.e. 0.8 sec/trajectory with no other MATLAB job submitted to the computer cluster at the time. The computations done in NAS using 160 CPUs to 1280 CPUs provide the shorter turnaround time than the ACES cluster computer.

In the future, the computational time can be further reduced by the application of FACET M-Stage program and FACET 0-Stage program that were written in Java with modified numerical techniques. When considering only computational time, the computation of daily wind-optimized aircraft trajectories worldwide is estimated to take 1.5 minutes when running FACET M-Stage program on Pleiades using 10,240 CPUs; and the estimated computational time is about 33 seconds for running FACET 0-Stage program. FACET M-Stage program seems to provide the best balance between performance and run time for wind-optimal trajectory application. Thus, in the absence of special privileges on the supercomputer, a cluster of computers provides a good option for computing wind-optimal trajectories for national and global air traffic system trade-off studies.

## Acknowledgement

The authors acknowledge the contribution of Mr. Jim Gibson for the development of Python programs and initial processing of NAS supercomputer. They also wish to thank Mr. Alex Morando and Dr. Kapil Sheth for advice with integration of optimization programs and FACET API and Ms. Michelle Eshow, Dr. Gano Chatterji, and Mr. Dat Duong for the initial setup of MATLAB MDSCS.

## References

- <sup>1</sup>Bryson, A. E., and Ho, Y. C., *Applied Optimal Control*, Taylor and Francis, Levittown, PA, 1975.
- <sup>2</sup>Sridhar, B., Ng, H. K., Chen, N. Y., "Aircraft Trajectory Optimization and Contrail Avoidance in the Presence of Winds," AIAA Aviation Technology, Integration, and Operation Conference, September 2010, Fort Worth, TX.
- <sup>3</sup>Grabbe, S., Sridhar, B., Cheng, N., August 2006, "Central East Pacific Flight Routing," AIAA-2006-6773, Keystone, Colorado, *AIAA Guidance, Navigation, and Control Conference*.
- <sup>4</sup>Ng, H. K., Sridhar, B., Grabbe, S., and Chen, N., "Cross-Polar Aircraft Trajectory Optimization and the Potential Climate Impact," *IEEE/AIAA 30<sup>th</sup> Digital Avionics Systems Conference*, Oct. 2011.
- <sup>5</sup>Ng, H. K., Sridhar, B., and Grabbe, S., "Optimizing Aircraft Trajectories with Multiple Cruise Altitudes in the Presence of Winds," *AIAA Journal of Aerospace Information Systems*, 2014.
- <sup>6</sup>OAG Absolute Aviation Advantage, <http://www.oag.com>.
- <sup>7</sup><http://www.nas.nasa.gov/hecc/>.
- <sup>8</sup><http://www.aviationsystemsdivision.arc.nasa.gov/research/modeling/aces.shtml>.
- <sup>9</sup>Bilimoria, K., Sridhar, B., Chatterji, G., Sheth, K., and Grabbe, S., "FACET: Future ATM Concepts Evaluation Tool," 3<sup>rd</sup> USA/Europe Air Traffic Management R&D Seminar, Napoli, Italy, June, 2000.
- <sup>10</sup><http://www.mathworks.com/products/distriben/>.
- <sup>11</sup>"User Manual for the Base of Aircraft Data (BADA), Revision 3.6," Eurocontrol Experimental Center (EEC) Note No. 10/04, Project ACE-C-E2, Sept. 2004.