High Performance Computing for Flight Projects at JPL

Chris Catherasoo, Ph.D.
California Institute of Technology

HPC User Forum Meeting
Bruyères-le-Châtel, France
3-4 October 2011
Outline

- Introduction
  - JPL and its mission
  - Current flight projects
- HPC resources at JPL
  - Institutional HPC resources
  - HPC resources at NASA Ames
- Examples of HPC usage by flight projects
  - Entry, descent and landing simulations
  - The Phoenix Mars Lander radar ambiguity
  - Mars Science Laboratory supersonic parachute design
  - Juno planetary protection trajectory analysis
- Future work
  - Evolutionary computing
    - Low-thrust orbit optimization
Introduction
  - JPL and its mission
  - Current flight projects

HPC resources at JPL
  - Institutional HPC resources
  - HPC resources at NASA Ames

Examples of HPC usage by flight projects
  - Entry, descent and landing simulations
  - The Phoenix Mars Lander radar ambiguity
  - Mars Science Laboratory supersonic parachute design
  - Juno planetary protection trajectory analysis

Future work
  - Evolutionary computing
    - Low-thrust orbit optimization
Jet Propulsion Laboratory overview

- JPL is part of NASA and Caltech
  - Owned by NASA – a “Federally-Funded Research and Development Center” (FFRDC)
  - Operated by Caltech, under contract to NASA
- $1.7 billion business base
- 5,000 employees
- Site area: 0.75 km²
JPL’s mission for NASA is robotic space exploration

- Mars
- Solar system
- Exoplanets
- Space science
- Earth science
- Interplanetary network
Robotic explorers in space

Voyagers 1 and 2 in interstellar space

Cassini at Saturn

Mars Reconnaissance Orbiter

Two Mars Exploration Rovers

Stardust-NExT to comet Tempel 1

EPOXI to comet Hartley 2

Mars Science Laboratory

Dawn to asteroids Ceres and Vesta
Robotic remote sensing on earth

- Atmospheric Infrared Sounder (AIRS) provides monthly global temperature maps
- Jason provides global sea surface height maps every ten days
- Gravity Recovery and Climate Experiment (GRACE) provides monthly maps of Earth’s gravity
- QuikSCAT provides near global (90%) ocean surface wind maps every 24 hours
- Microwave Limb Sounder (MLS) provides daily maps of stratospheric chemistry
- Tropospheric Emission Spectrometer (TES) provides monthly global ozone maps
- Multi-angle Imaging Spectro Radiometer (MISR) provides monthly global aerosol maps
- CloudSat provides monthly maps of cloud ice water content
Outline

- Introduction
  - JPL and its mission
  - Current flight projects
- HPC resources at JPL
  - Institutional HPC resources
  - HPC resources at NASA Ames
- Examples of HPC usage by flight projects
  - Entry, descent and landing simulations
  - The Phoenix Mars Lander radar ambiguity
  - Mars Science Laboratory supersonic parachute design
  - Juno planetary protection trajectory analysis
- Future work
  - Evolutionary computing
    - Low-thrust orbit optimization
Institutional HPC resources

Dell Xeon Cluster
- 1024 3.2 GHz processors
- 2 GB per CPU distributed memory
- Myrinet interconnect

HP SFS File System
- 2 MDS/Admin servers
- 16 OSS servers
- Read / write > 2 GB/s
- 32 TBytes

Visualization Center
- Sony SRX-R110 projector
- 12’ x 7’ display
- Resolution: 4096 x 2160 (8 MPixels)

Dell Xeon Clusters
- 2 x 512 3.06 GHz processors
- 2 GB per CPU distributed memory
- Gigabit ethernet interconnect

SGI Altix 3700s
- 2 x 256 and 1 x 64 1.5 GHz processors
- 2 GB per CPU shared memory
- ccNUMA interconnect

Online Storage
- RAID6 system
- 1 PByte
HPC resources at NASA Advanced Supercomputing

- **SGI ICE cluster**
  - Total cores: 111,872
    - 4,544 Westmere (Xeon X5670) nodes
      - 2 six-core processors per node
    - 1,280 Nehalem (Xeon X5570) nodes
      - 2 quad-core processors per node
    - 5,888 Harpertown (Xeon E5472) nodes
      - 2 quad-core processors per node
  - Total memory: 188 TB
  - Infiniband DDR, QDR interconnect
    - 11-D hypercube topology

- **SGI Altix 4700 system**
  - Total cores: 4,608 (originally 10,240)
    - Four compute nodes
  - Total memory: 9 TB
  - NUMALink interconnect
    - Single-system image on each compute node
Outline

- Introduction
  - JPL and its mission
  - Current flight projects
- HPC resources at JPL
  - Institutional HPC resources
  - HPC resources at NASA Ames
- Examples of HPC usage by flight projects
  - Entry, descent and landing simulations
  - The Phoenix Mars Lander radar ambiguity
  - Mars Science Laboratory supersonic parachute design
  - Juno planetary protection trajectory analysis
- Future work
  - Evolutionary computing
    - Low-thrust orbit optimization
Outline

- Introduction
  - JPL and its mission
  - Current flight projects
- HPC resources at JPL
  - Institutional HPC resources
  - HPC resources at NASA Ames
- Examples of HPC usage by flight projects
  - Entry, descent and landing simulations
  - The Phoenix Mars Lander radar ambiguity
  - Mars Science Laboratory supersonic parachute design
  - Juno planetary protection trajectory analysis
- Future work
  - Evolutionary computing
    - Low-thrust orbit optimization
Spacecraft components in cruise configuration

- Cruise Stage
- Backshell
- Lander
- Heatshield
Entry, descent and landing

- Entry, Descent and Landing (EDL) is the sequence of events that brings a spacecraft safely to the surface of a planet.
- It consists of several phases:
  - Cruise stage separates before entering the atmosphere.
  - Entry phase:
    - Aerobraking – friction with the planetary atmosphere is used to slow the spacecraft from over 5,500 m/s to 500 m/s in about 220 s.
  - Descent phase:
    - Parachute braking – slows the spacecraft down to 100 m/s in 20 s.
  - Landing phase:
    - The parachute separates and spacecraft lands:
      - Retro rockets
      - Airbags
      - Sky crane
- For Mars, the EDL sequence takes about 7 minutes.
  - Signal time from Mars to Earth is about 10 minutes.
Typical entry, descent and landing sequence

- **Entry**
  - Deploy Supersonic Parachute
  - Heatshield Separation
  - Entry Balance Mass Jettison
  - Radar Activation and Mobility Deploy
  - MLE Warm-Up

- **Supersonic Parachute Descent**
  - h ≈ 10 km MSL
  - M = 2.0

- **Powered Descent**
  - Backshell Separation
  - h ≈ 800 m AGL

- **Sky Crane**
  - Cut to Four Engines
  - Rover Separation
  - Rover Touchdown

- **Flyaway**
  - 2000 m above MOLA areoid
Entry, descent and landing simulations

- EDL simulations are one of the most mission-critical HPC applications run at JPL
  - The simulations involve multi-body dynamics of the parachute, backshell and lander system
  - The EDL application is the “Program to Optimize Simulated Trajectories” (POST)
    - Application was developed at NASA Langley
    - Uses a 6 degrees of freedom modelling scheme
    - Inputs include spacecraft parameters, ambient atmospheric conditions and wind speeds
  - Monte Carlo simulations are performed to determine spacecraft entry, descent and landing characteristics to evaluate safety metrics for landing
  - EDL simulations are used to
    - Down-select landing sites, and to choose the final landing site
    - Apply final trajectory maneuver corrections to the spacecraft prior to cruise stage separation
EDL simulations used for JPL missions to Mars

- EDL simulations used successfully for
  - Mars Pathfinder
    - Landed: 4 July 1997
    - Length: 0.65 m; weight: 10.6 kg
  - Mars Exploration Rovers – twin rovers
    - Landed: 3 January 2004 and 24 January 2004 respectively
    - Length: 1.6 m; weight: 180 kg
  - Mars Phoenix Lander
    - Landed: 25 May 2008
    - Length: 1.5 m; weight: 350 kg

- Upcoming
  - Mars Science Laboratory
    - Launch: November 2011
    - Length: 2.7 m; weight: 950 kg
Outline

- Introduction
  - JPL and its mission
  - Current flight projects
- HPC resources at JPL
  - Institutional HPC resources
  - HPC resources at NASA Ames
- Examples of HPC usage by flight projects
  - Entry, descent and landing simulations
  - The Phoenix Mars Lander radar ambiguity
  - Mars Science Laboratory supersonic parachute design
  - Juno planetary protection trajectory analysis
- Future work
  - Evolutionary computing
    - Low-thrust orbit optimization
The Phoenix Mars Lander radar ambiguity

- The Phoenix Mars Lander was launched in August 2007
  - Mission was to explore the Martian polar region for evidence of water
- The lander used a radar to obtain ground-relative altitude and velocity during terminal descent
- Both helicopter field tests and simulations were used to validate the radar performance
- Analysis of the radar simulation data showed that the presence of the jettisoned heat shield could cause radar to lock on a range ambiguity
  - The radar was not locking on to the heat shield
  - Did not occur in the absence of the heat shield
The Phoenix Mars Lander radar ambiguity

- The radar erroneously reported an altitude that was much lower than the true lander altitude
  - Could not be distinguished from the expected altitude behavior
  - Would have caused the premature separation of the lander from the backshell
  - Result would have been catastrophic loss of the mission
- The problem was impossible to characterize analytically
  - Too many contributing parameters – lander altitude, heatshield range, heatshield radar cross-section, heatshield attitude, attitude rate
- With only eight months to go before launch, resolving this problem became a critical activity
- Hundreds of thousands of radar simulation runs were made to characterize the design space
  - Phoenix was given highest priority on all the laboratory’s clusters
Results were plotted on radar ambiguity maps

- Each dot is the result of a single simulation that took about 3.5 core hours to run
  - Gray: No target acquisition
  - Green: Radar correctly locked on the ground
  - Red: Radar incorrectly (ambiguously) locked on the ground
  - Cyan: Points at which radar begins making measurements
The Phoenix Mars Lander radar ambiguity

- The problem was resolved by
  - Delaying the start of the radar search
  - Modifying the radar Pulse Repetition Interval (PRI)
- The modified radar was field tested, and the updated radar-model simulation results were used to verify that the problem had been eliminated.

Problem

Resolution
Outline

- Introduction
  - JPL and its mission
  - Current flight projects
- HPC resources at JPL
  - Institutional HPC resources
  - HPC resources at NASA Ames
- Examples of HPC usage by flight projects
  - Entry, descent and landing simulations
  - The Phoenix Mars Lander radar ambiguity
  - Mars Science Laboratory supersonic parachute design
  - Juno planetary protection trajectory analysis
- Future work
  - Evolutionary computing
    - Low-thrust orbit optimization
The Mars Science Laboratory (MSL) will be launched in November 2011
  - Mission is to detect and study organic molecules on the surface of Mars

Will employ advanced entry, descent and landing techniques
  - 21.5 m diameter supersonic parachute
  - Powered descent vehicle with 8 Mars Landing Engines (MLEs)
  - Sky-crane tethered landing of rover
The Mars Science Laboratory
supersonic parachute design

- A Parachute Decelerator System (PDS) provides the most efficient means of slowing an entry vehicle from supersonic to subsonic speeds
  - Prepares the vehicle for safe heatshield separation
  - Prepares the vehicle for powered descent
    - Attitude and velocity

- MSL PDS characteristics
  - 21.5 m Viking-type Disk-Gap-Band type parachute
    - Viking parachute was 16.1 m
  - Similar capsule to parachute diameter scaling as Vikings
  - Deployed at Mach 2.3
  - Limits time above Mach 1.5 (~10s)
  - Modern materials – nylon, Kevlar and Technora
Supersonic parachute instability

- In 1960s, high altitude (~50 km) supersonic parachute tests were performed
  - Showed canopy instabilities at Mach Numbers above 1.5
    - Observed partial inflations and collapses of the parachute, termed “Area Oscillations”
    - The oscillations resulted in projected area and drag fluctuations
  - Resulting load spikes could damage the parachute canopy
The Mars Science Laboratory supersonic parachute design

- Scaling up from Viking to MSL
  - A simulation capability was developed to extrapolate the Viking data to the larger scale and with different materials
    - Alternative would have been expensive high-altitude tests
  - Aerodynamic and dynamic performance of the MSL parachute in the supersonic regime is determined from
    - Subscale wind tunnel testing
    - Computational simulations
The parachute simulation application

- Developed CFD, FEM and FSI tools to model the physics of interest
- Application was based on the Virtual Test Facility (VTF) toolkit
  - Originally developed at the California Institute of Technology for the Department of Energy
  - Further developed by University of Illinois and Cambridge University
- Uses a CFD solver coupled to FEM solver
  - Fluid is simulated using unsteady, compressible, large-eddy simulations in an Eulerian-Cartesian mesh ~ 50 million cells
  - Canopy is simulated using large-deformation thin-shell Kirchhoff-Love finite elements on a Lagrangian mesh ~ 10,000 elements
  - Four levels of adaptive mesh refinement are used for finer resolution as needed
- Fast level set techniques were used at the boundaries between the fluid and the solid
Computational qualification approach

- Validated the simulations using wind tunnel data from scaled model experiments
- A piecewise validation approach was used
  - Capsule only
  - Rigid parachute only
  - Capsule and rigid parachute
  - Capsule and flexible parachute
- Also validated against the archived experimental Viking parachute data from the 1960s
- Following validation, the code is being used to simulate the full-scale MSL parachute
  - Extrapolate to the MSL parachute size, materials and flight conditions
Simulations and results

Simulation domain is $[-3,5] \times [-1,1] \times [-1,1]$ m

- Initially run on a Dell-Myrinet Xeon cluster (64 to 206 processors)
- Subsequently run on an SGI Altix 3700 system with 96 processors allocated to the fluid and 4 to the solid

Simulation results showed that the parachute supersonic behaviour and performance were as expected
Outline

- Introduction
  - JPL and its mission
  - Current flight projects
- HPC resources at JPL
  - Institutional HPC resources
  - HPC resources at NASA Ames
- Examples of HPC usage by flight projects
  - Entry, descent and landing simulations
  - The Phoenix Mars Lander radar ambiguity
  - Mars Science Laboratory supersonic parachute design
  - Juno planetary protection trajectory analysis
- Future work
  - Evolutionary computing
    - Low-thrust orbit optimization
The Juno spacecraft was launched in August 2011
- The spacecraft will orbit Jupiter
- Mission is to understand Jupiter’s origins by measuring its gravity field, and exploring the Jovian atmosphere and magnetosphere
- Juno’s highly eccentric orbit could lead to potential impact with the Galilean satellites (Io, Europa, Ganymede and Callisto)
- These large icy moons are of particular interest for future exobiology and astrobiology exploration
- Potentially contain biological and/or organic materials
Juno planetary protection trajectory analysis

- Planetary protection requirements dictate that during its prime mission Juno must not collide with any of the Galilean satellites
  - Any collision would cause contamination that would jeopardize future explorations
- Juno’s planned mission is for one year, after which it will be de-orbited into Jupiter’s atmosphere
  - In case de-orbiting is unsuccessful, planetary protection requirements must be met for a further period of 150 years
Monte-Carlo techniques were employed to determine the collision probabilities

- Required the analysis of thousands of trajectory states for hundreds of years for each case
- The wall-clock time for a single trajectory propagation was about 10 hours
- On one CPU, a single case would have taken over a year to complete
- Were able to complete each Monte-Carlo run in less than 12 hours, instead of the estimated year

HPC enabled the investigation of many failure scenarios and potential baseline trajectories
Outline

- **Introduction**
  - JPL and its mission
  - Current flight projects

- **HPC resources at JPL**
  - Institutional HPC resources
  - HPC resources at NASA Ames

- **Examples of HPC usage by flight projects**
  - Entry, descent and landing simulations
  - The Phoenix Mars Lander radar ambiguity
  - Mars Science Laboratory supersonic parachute design
  - Juno planetary protection trajectory analysis

- **Future work**
  - Evolutionary computing
    - Low-thrust orbit optimization
Evolutionary computing seeks to mimic processes used in nature to optimize multi-parameter engineering designs

- Uses sophisticated biological operators
  - Selection
  - Mutation
  - Recombination

**Advantages**

- Enables larger design spaces to be explored than could be examined manually or by computational brute force
- Results have shown competitive advantages over human-created designs
Low-thrust orbit optimization

- Low-thrust ion propulsion is more efficient than chemical propulsion
  - Uses less propellant
  - Demonstrated on Deep Space 1 and currently on Dawn
- Requires different trajectory optimization techniques
  - Involves many revolutions and continuous thrust arcs
- Goal is to optimize the trade-off between propellant mass and flight time for orbit transfers
  - Thrust angles and thrust arcs are optimized with the Q-law
    - The Q-law is a specialized feedback control law
    - There are about 15 independent parameters to be optimized
  - In this work, the Q-law parameters are optimized using evolutionary algorithms
  - Evolutionary algorithms are amenable to parallel computing implementation
Low-thrust orbit optimization

Optimal circle-to-circle orbit transfers

- Flight-time optimal Edelbaum transfer
- Propellant optimal Hohmann transfer

Propellant Mass (kg)

Flight Time (days)

GA/SA driven Q-law
Nominal Q-law
Edelbaum
References

Thank you!

Questions and/or comments?

Contact information:
Chris.Catherasoo@caltech.edu
+1 626 395-3679