NASA Ames Mars Global Climate Model (MGCM) Version 3.1 User Guide

Courtney Batterson[1,2], Richard Urata[1,2], Melinda Kahre[2], Amanda Brecht[2], Kathryn Steakley[2], John Wilson[2], Victoria Hartwick[1,2], Alex Kling[1,2], and Sonny Harman[2]

[1]Bay Area Environmental Research Institute, Moffett Field, CA
[2]NASA Ames Research Center, Moffett Field, CA

January 17, 2024

Contents

List of Figures

List of Tables

Part I

Quick Start Guide

Run MGCM 3.1 Out-of-the-Box on NAS

In this first part of the User Manual, we provide instructions for downloading MGCM 3.1, minimally modifying the runscript, and running the default simulation on the NASA Advanced Supercomputing (NAS) system. This process is parsed into three steps:

1. Download the model from GitHub
2. Compile the model
3. Run the model

> **NOTE:** These instructions assume you have installed all of the required software listed in Chapter 3 (Section 3.1) and that you are running the model on the NAS system. Additional steps are required for running the model on non-NAS systems (see Chapters 3 and 4).

Step 1: Download the model from GitHub

MGCM 3.1 is built by cloning (downloading) the NOAA-GFDL Atmospheric Model Version 4 (AM4) and installing the physics packages for Mars developed by the Mars Climate Modeling Center (MCMC) on top of it. These instructions guide the user through the process of cloning the AM4 model and the Mars physics package from GitHub, installing them in Pleiades, and then patching them together.

In a new terminal window, `ssh` into Pleiades (`pfe`), load the `pkgsrc` module, then clone AM4 into your preferred directory:

```
user@pfe:~$ module load pkgsrc/20XXQX # any available version
user@pfe:~$ git clone --recursive --branch 2021.03
↪ https://github.com/NOAA-GFDL/AM4.git
```

Change to the `AM4/src/` directory and delete the `ice_param` module:

```
user@pfe:~$ cd AM4/src/
user@pfe:~$ rm -rf ice_param
```

In `AM4/src/`, create and switch to `mars_branch`. Then, add the Mars physics submodules to AM4:

```
user@pfe:~$ git checkout -b mars_branch
user@pfe:~$ git submodule add https://www.github.com/nasa/AmesGCM.git
user@pfe:~$ git submodule add https://github.com/NOAA-GFDL/mkmf.git
user@pfe:~$ git submodule add https://github.com/NOAA-GFDL/FRE.git
```

Change to the `AM4/src/AmesGCM/patches/` directory and apply the patches that stitch the Mars physics and the AM4 model together:

```
user@pfe:~$ cd AmesGCM/patches/
user@pfe:~$ ./apply_patch.sh
```

Congratulations! You have successfully built MGCM 3.1 on the NAS system.

Step 2: Compile the model

Go back three directories from `AmesGCM/patches/` and change to the `AM4/exec/` directory. Compile the code there:

```
user@pfe:~$ cd ../../../exec/ # from AmesGCM/patches
user@pfe:~$ ./compile.archives
```

The compilation takes ~ 5 minutes. When it is complete, the compiled version of MGCM 3.1 will be stored under `exec.intel.mars3.1`.

Step 3: Run the model

To run MGCM 3.1, edit and submit the runscript through the PBS on NAS. To edit the runscript, open `fms_mars_default` (located in `AM4/exec/`) and modify the following PBS settings:

```
! Set the account to be billed:
PBS -W group_list=sXXXX
! Enter your email address:
PBS -M user@email.com
```

Save `fms_mars_default` and return to the terminal. Submit the run to the Pleiades supercomputing system from the command line with:

```
user@pfe:~$ qsub fms_mars_default # from the AM4/exec/ directory
```

That's it! You'll receive an email at the address you provided in the runscript when the simulation executes and when it completes. Barring any other changes in the runscript, the simulation will output files onto Lou (`lfe`) at:

```
/u/$USER/FV3/xanadu/am4_mars_runs/fms_mars_default/history/
```

Restart files are ported to your `nobackup/` directory on Pleiades:

```
/nobackup/$USER/FMS_MARS_runs/am4_mars_runs/ fms_mars_default/restart/
```

Part II

User Manual

Introduction

This document serves as a manual for obtaining and using the NASA Ames Mars Global Climate Model (MGCM) version 3.1. MGCM 3.1 simulates the Martian climate using an external finite-volume dynamical core to solve the equations of motion and internally developed physics packages for Mars to predict the global atmosphere given various planetary and physical parameters. A brief overview of the physics in MGCM 3.1 is included here. A more complete description will be included in later editions of the User Manual.

> **NOTE:** This public release of the MGCM (version 3.1) includes a reduced set of physics compared to internal versions that have been used in recent publications (e.g., Bertrand et al., 2020, Haberle et al., 2019 and Kahre et al., 2022). We will include more sophisticated physics in future public releases of the MGCM.

The MGCM 3.1 dynamical core is the NOAA-GFDL Finite-Volume Dynamical Core (Xanadu version), which is publicly available on the NOAA-GFDL GitHub Repository at https://github.com/NOAA-GFDL and described in Harris et al. (2021). The dynamical core integrates the fluid mechanical primitive equations in time over the globe. It also provides the horizontal grid framework for the model, which is described in greater detail in Chapter 1.

The physics in MGCM 3.1 have been implemented and tested by the Mars Climate Modeling Center (MCMC) and are a subset of those described in Haberle et al. (2019). Physical parameterizations in the model include: surface properties such as thermal inertia and albedo; a ground and subsurface temperature scheme; a planetary boundary layer (PBL) scheme that provides vertical diffusive mixing in the boundary layer and vertical diffusion throughout the atmospheric column; carbon dioxide sublimation/condensation physics; an analytically prescribed dust field; and a two-stream radiative transfer code based on correlated-k's. The MGCM utilizes Local True Solar Time (LTST), which means that the length of a day is constant throughout the year and that the sun is always the highest in the sky at local noon (LTST = 12.0). A more detailed description of the physics in MGCM 3.1 is provided in Chapter 1.

All software components in MGCM 3.1 were developed and supplied by the NASA Ames MCMC. Surface maps for thermal inertia, albedo, and the residual north polar cap boundary were created by team members and collaborators at Oregon State University. Dust optical properties were computed based on the indices of refraction supplied by Michael Wolff at Space Science Institute in Boulder, CO. All other input data was supplied by NASA.

Chapter 1

Model Description

MGCM 3.1 has a finite-volume dynamical core and a cubed-sphere grid, both of which were developed by NOAA-GFDL. The vertical grid has a hybrid sigma-pressure vertical coordinate that transitions from sigma near the surface to pure pressure near the model top. The altitude at the top of the model varies between ∼100–130 km depending on the vertical grid selected in the runscript. Various options for the vertical grid and how to implement them are discussed in Chapter 2. In this chapter, we provide a brief description of the horizontal and vertical grid structures in the model, and we summarize the physics included in MGCM 3.1.

1.1 The Dynamical Core

The MGCM has a mass-conserving finite-volume dynamical core that was developed at NOAA-GFDL based on the transport and shallow water algorithms from Lin and Rood (1996, 1997). The Lagrangian vertical discretization scheme is as described in Lin (2004) and the pressure gradient force is computed from a modified version of Green's theorem as described in Lin (1997). The model grid has the cubed-sphere geometry described in Lin and Putman (2007) that enables high horizontal resolution simulations, higher order finite-volume numerics, and grid nesting. The dynamical core can be run in hydrostatic or non-hydrostatic mode, but MGCM 3.1 only supports hydrostatic mode. In hydrostatic mode, the vertical grid is defined by a hybrid sigma-pressure coordinate system, which is terrain-following near the surface and transitions to pure pressure in the upper atmosphere.

1.1.1 Horizontal Grid Structure

The cubed-sphere grid, illustrated in Figure 1.1, is comprised of six tiles: one centered over each of the poles and four around the equator. This provides near-uniform global coverage, eliminating the singularity at the poles inherent to traditional latitude-longitude grids. Relatively uniform grids such as the cubed-sphere allow for efficient computation on massively parallel machines, making high-resolution simulations practical.

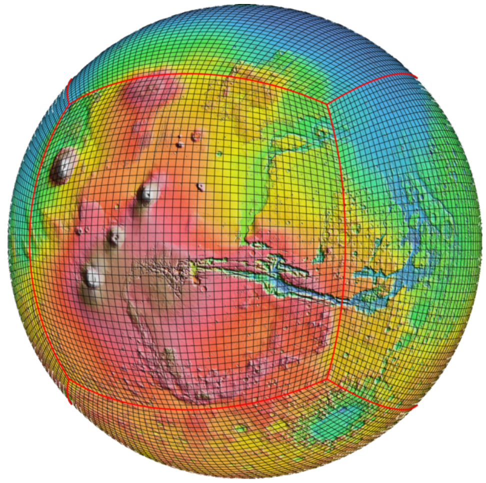

Figure 1.1: An illustration of the c48 (1.875°) cubed-sphere grid over Martian topography. Tile boundaries are drawn in red.

The horizontal resolution of the model depends on the number of grid cells there are on each tile, which is determined by the grid selected at initialization. The desired grid is selected by setting the `NCX` parameter in the runscript. For example, `NCX = 24` indicates a "c24" simulation with 24 grid cells in both the X and Y directions on each tile. The resulting horizontal resolution for `c24` is ∼3.75°, calculated as follows:

$$\text{resolution (degree)} = \frac{360^\circ}{\text{NCX} * 4} \tag{1.1}$$

Or, equivalently, ∼221.5 km, calculated by:

$$\text{resolution (km)} = \frac{2\pi r}{\text{NCX} * 4} \tag{1.2}$$

where the radius of Mars (r) is 3,386 km and the number "4" refers to the four tiles surrounding the equator. The MCMC has carried out simulations using MGCM 3.1 at c12, c24, c48, c96, and c192 resolutions. Currently, the MCMC supports running MGCM 3.1 at two of those resolutions: c24, the default resolution, and c48, a higher resolution. Descriptions of the horizontal grids and instructions for moving to c48 are included in Chapter 4.

> **NOTE:** The horizontal grid resolution is not *truly* uniform because the cubed-sphere grid necessitates that each tile be distorted to more closely resemble a sphere than a cube.

1.1.2 Vertical Grid Structure

MGCM 3.1 utilizes a hybrid sigma-pressure vertical coordinate, transitioning from sigma near the surface to pure pressure at the top of the atmosphere (Simmons & Burridge, 1981). The sigma coordinates are terrain following, which is useful for solid surfaces with large varying terrains such as Mars. The hybrid sigma-pressure vertical grid defining the layer interfaces is given by Equation 1.3:

$$P[k] = ak[k] + bk[k] * P_{sfc} \quad \text{for distinct } k \in \{1, \ldots, nlevs\} \tag{1.3}$$

where $nlevs$ is the number of vertical layers plus one, k is the layer index is, ak is the pressure component of the hybrid coordinate, and bk is the sigma component of the hybrid coordinate. ak has units of pressure in Pascal, while bk is unitless and varies between 0 and 1. The top of the model corresponds to $k = 1$, with k increasing as the layers approach the surface. The full pressure grid is constructed in the source code in `fv_eta.F90`, which references the ak and bk values for the desired vertical grid that are stored in:

```
AM4/src/GFDL_atmos_cubed_sphere/tools/fv_eta_mars.h
```

The default MGCM pressure grid has 56 layers. The pressure and approximate altitude (calculated using on a constant scale height of 10 km and a reference surface pressure of 705 Pa) of each layer midpoint in the 56-layer grid are listed in Table 1.1 for reference. The resolution of the 56-layer grid varies from ∼5 m near the surface to ∼4-5 km near the top of the model's physical domain (∼80-90 km or 1×10^{-1} Pa). There are 10 pure-pressure layers that start near ∼1 Pa. The top three layers above ∼80 km have lower vertical resolution, and they are ignored for analysis due to potential model top boundary condition effects.

Other vertical grids available in MGCM 3.1 include a 30- and a 37-layer hybrid sigma-pressure grid, as well as a 24-layer pure sigma grid for backward compatibility with the Legacy MGCM (Haberle et al., 2019). The pressure and altitude of the layers in all of the vertical grids are illustrated in Figure 4.2 in Chapter 4. Explicit pressures and altitudes for the other vertical grids are provided in:

```
AM4/src/AmesGCM/diagnostics/
```

Table 1.1: Pressures and altitudes of the layer midpoints in the default 56-layer vertical grid. Altitude is calculated using the reference surface pressure from the model grid (705 Pa) and a 10 km scale height.

Pressure [Pa]	Altitude [km]	Pressure cont. [Pa]	Altitude cont. [km]	Pressure cont. [Pa]	Altitude cont. [km]
(top) 0.003	122.9	35.2	30.0	474.3	4.0
0.013	108.8	44.5	27.6	502.1	3.4
0.038	98.2	55.5	25.4	528.5	2.9
0.076	91.3	68.3	23.3	553.5	2.4
0.13	86.1	83.0	21.4	576.7	2.0
0.21	81.2	99.6	19.6	598.1	1.6
0.34	76.4	118.2	17.9	617.5	1.3
0.53	71.9	138.6	16.3	634.8	1.0
0.82	67.6	161.0	14.8	650.1	0.81
1.2	63.4	185.1	13.4	663.2	0.61
1.9	59.4	210.7	12.1	674.4	0.44
2.8	55.3	237.8	10.9	683.5	0.31
4.1	51.4	266.1	9.7	690.8	0.20
6.0	47.7	295.3	8.7	696.3	0.12
8.4	44.3	325.2	7.7	700.2	0.07
11.7	41.0	355.5	6.8	702.8	0.031
15.8	38.0	385.8	6.0	704.2	0.011
21.0	35.2	415.9	5.3	704.8	0.002 (bottom)
27.4	32.5	445.5	4.6		

1.2 Physics: Current Mars

The Mars physics parameterizations in MGCM 3.1 are described here. Several of them are described in greater detail in Haberle et al. (2019) and future versions of the User Manual will include more detailed descriptions of the model physics.

1.2.1 Surface and Sub-Surface Properties and the CO_2 Cycle

In MGCM 3.1, model topography is determined from Mars Orbiter Laser Altimeter (MOLA) retrievals, and surface thermal inertia and albedo are derived from Viking and Mars Global Surveyor (MGS) Thermal Emission Spectrometer (TES) measurements. The south polar residual CO_2 ice cap (SPRC) is not explicitly prescribed in the model, but is instead represented by regions of high surface thermal inertia and albedo detected by TES. The depth and extent of the seasonal CO_2 ice caps are self-consistently determined by a surface energy balance that considers surface thermal inertia, surface albedo, topography, and the CO_2 ice albedo at the north and south poles. The prescribed albedos are chosen to produce a good fit to Viking data (Figure 1.2), which is a standard method for validating the annual mass variation due to the CO_2 cycle in MGCMs (Haberle et al., 2008). Surface and sub-surface temperatures are computed from a soil conduction calculation that implements a surface heat balance at the upper boundary and constant temperature at the lower boundary. Computed surface temperatures partially determine the amount of CO_2 frost on the ground. CO_2 condensation occurs as necessary to maintain the CO_2 frost point temperature.

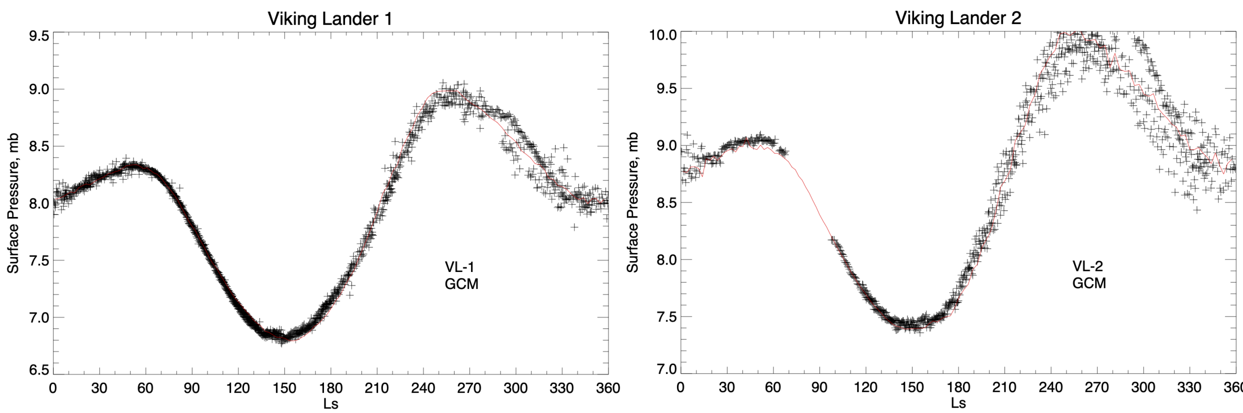

Figure 1.2: Daily mean observed (+) and simulated (red line) surface pressures at Viking Lander 1 (VL1; left) and VL2 (right). Simulated data has been interpolated and hydrostatically corrected to the locations of VL1 and VL2 for comparison.

1.2.2 The Planetary Boundary Layer (PBL)

Interactions between the surface and the atmosphere are represented by the PBL scheme, which predicts winds and temperatures throughout (approximately) the lowest scale height of the atmosphere. A level-2 Mellor and Yamada (1982) turbulence closure scheme predicts near-surface heat fluxes, and surface heat and momentum fluxes are handled according to Monin-Obukhov similarity theory. In addition to mixing in the boundary layer, the PBL scheme also provides diffusive mixing throughout the atmospheric column. A convective adjustment is performed every physical time step to remove any remaining superadiabatic temperature profiles, which ultimately results in more realistic near-surface air temperatures.

1.2.3 Radiative Transfer (RT)

The radiative transfer (RT) scheme is a 2-stream code that handles the radiative effects of airborne dust and gaseous CO_2. Gaseous CO_2 opacities are calculated from correlated-k tables (tabulated off-line), and Mie theory is used to compute the extinction efficiencies and scattering properties of the dust. Rayleigh scattering by CO_2 is directly calculated. The RT code predicts fluxes and flux divergences from the optical properties of dust aerosols, which are largely dependent on their effective particle size distributions. Radiative heating and cooling rates are then computed from those fluxes. More information about the RT scheme can be found in Chapter 2.

1.2.4 Atmospheric Dust

The spatially and temporally evolving dust in MGCM 3.1 can be prescribed (in the horizontal and vertical domains) or transported (self-consistently determined by the atmospheric tracer fields). These options are controlled through a set of user-defined parameters (specified in Chapters 2 and 4). Airborne dust is radiatively active at both visible and infrared wavelengths, but can be set to be radiatively inert. This section divides the discussion of the atmospheric dust options into two parts: the treatment of prescribed dust and the treatment of transported dust carried by the atmospheric tracer fields.

> **NOTE:** This public release of the MGCM does not currently include a water cycle or water-ice cloud microphysical schemes, but these will be included in a later public release of the MGCM.

Prescribed Dust

The prescribed dust option decouples the tracer field from the RT code. In this prescription, the spatially and temporally evolving pattern of dust seen by the RT is not determined by the tracer fields. Instead, the RT code ingests a horizontally and vertically prescribed dust field. There are multiple options for how dust can be prescribed in both the horizontal and the vertical. These options are described below.

Horizontal Prescription

In the horizontal, the user can choose one of three dust prescriptions: (1) a dust distribution that is constant in time and space (at a visible column optical depth defined by the user at a reference pressure), (2) a distribution that is determined by an inputted map that evolves in time, or (3) an analytic expression based on Montmessin et al. (2004) referred to in this document as the "MGS scenario". The expression for the MGS scenario is:

$$\begin{aligned}\tau_{north}(\theta, L_s) &= \tau_n + 0.5(\tau_{eq} - \tau_n)(1 - \tanh(4.5 - \theta/10)),\\ \tau_{south}(\theta, L_s) &= \tau_s + 0.5(\tau_{eq} - \tau_s)(1 - \tanh(4.5 - \theta/10))\end{aligned} \tag{1.4}$$

where τ_{north} and τ_{south} are given at a reference pressure of 7 mbar and τ_n, τ_s, and τ_{eq} are given by:

$$\begin{aligned}\tau_n &= 0.1\\ \tau_s &= 0.1 + 0.4(\cos(0.5(L_s - 250)))^{14}\\ \tau_{eq} &= 0.2 + 0.3(\cos(0.5(L_s - 250)))^{14}\end{aligned} \tag{1.5}$$

For a dust distribution determined by an inputted map, the dust maps are generally based on the Montabone et al. (2015) dust scenarios for various Mars years. These are binned to 3° x 3°

in space and daily in time, then regridded to the model resolution at runtime. A "background" dust scenario (also binned to 3° x 30° in space and daily in time) is available as well. This map is constructed from the dust scenarios of Montabone et al. (2015) by identifying the spatially and seasonally varying minimum value of the multi-year record (MY 29–34) to filter out the influence of episodic regional dust storms (Figure 1.3). The specific dust scenario maps available are listed and described in Section 2.3.

Vertical Prescription

In the vertical, the prescribed dust distribution can be defined by either a Conrath profile (Conrath, 1975) or a modified Conrath profile (Forget et al., 1999). The Conrath profile is given by:

$$q = q_{ref} \exp\left(\nu \left[1 - \frac{P_{ref}}{P}\right]\right) \tag{1.6}$$

where q is the mixing ratio at pressure P, q_{ref} is the mixing ratio at the reference pressure P_{ref}, and ν is the Conrath parameter. The modified Conrath profile is based on the Z_{max} parameter:

$$q = q_{ref} \exp\left(0.007 \left[1 - \left(\frac{P_{ref}}{P}\right)^{\frac{70}{Z_{max}}}\right]\right) \tag{1.7}$$

where q is the mixing ratio at pressure P, q_{ref} is the mixing ratio at the reference pressure P_{ref}, and Z_{max} is the maximum altitude of the dust provided in kilometers. Z_{max} varies in latitude and season in accordance with the following expression:

$$\begin{aligned} Z_{max}(\theta, L_s) &= 60 + 18(\sin(L_s - 158)) \\ &- (32 + 18\sin(L_s - 158))\sin^4\theta \\ &- 8\sin(L_s - 158)\sin^5\theta \end{aligned} \tag{1.8}$$

where θ is latitude (in degrees) and L_s is solar longitude. Figure 1.3 illustrates the dust height as defined by Z_{max}.

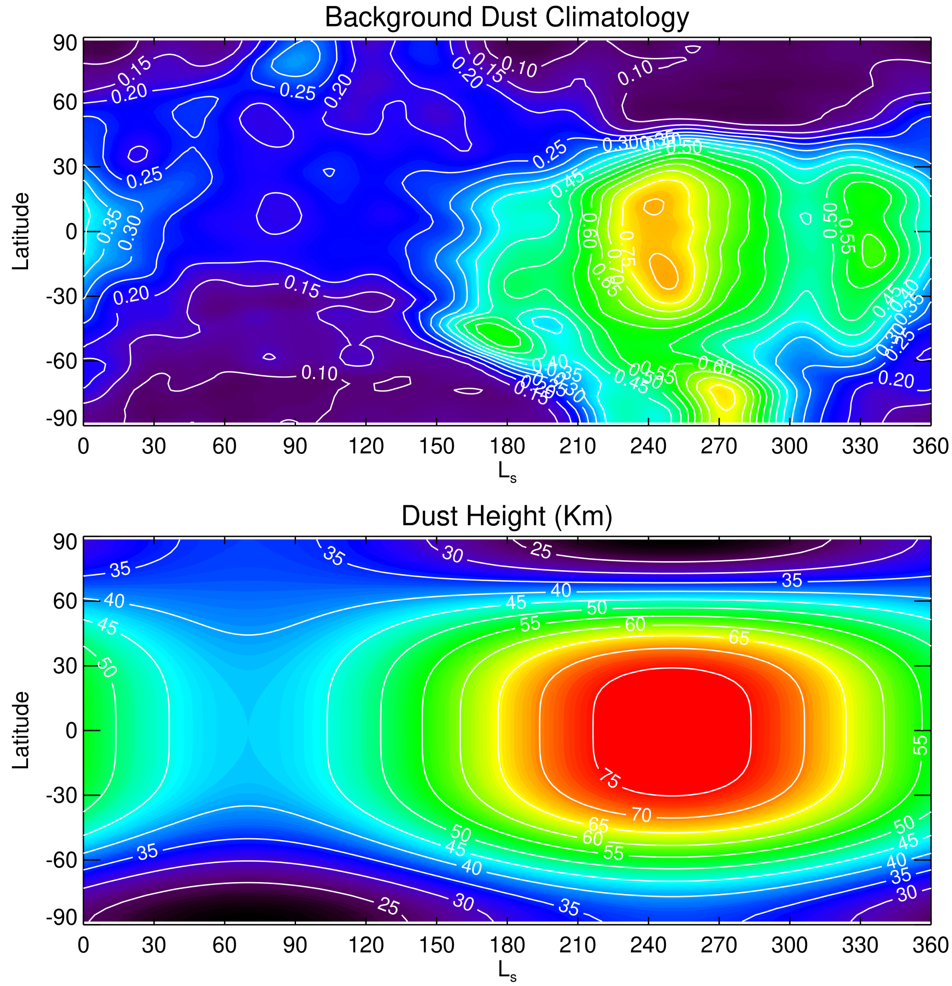

Figure 1.3: Background dust scenario (top) and dust height computed from Equation 1.8 (bottom).

Tracer-Carried (Transported) Dust

In MGCM 3.1, tracer-carried dust is transported horizontally and vertically by model-resolved winds, parameterized turbulent mechanical mixing, and size-dependent gravitational sedimentation (described below). Transported dust is treated using the moment method, described in detail in Haberle et al. (2019), wherein the actual particle size distribution is represented by the moments (mean, variance, etc.) of the distribution. Size-dependent aerosols assume to have a log-normal distribution:

$$n(r)dr = \frac{1}{r\sigma_0\sqrt{2\pi}} \exp\left[\frac{1}{2}\left(\frac{\ln(r/r_0)^2}{\sigma_0}\right)\right] \tag{1.9}$$

where $n(r)dr$ is the number of particles per unit volume with radii between r and $r + dr$, r_0 is the median radius of the particles, and σ_0 is the standard deviation the distribution. It follows that the number of particles N between r_{min} and r_{max} is:

$$N = N_0 \int_{r_{min}}^{r_{max}} n(r)dr = \frac{N_0}{2}\left[\text{erf}\left(\frac{\ln(r_{max}/r_0)}{\sqrt{2\pi\sigma_0}}\right) - \text{erf}\left(\frac{\ln(r_{min}/r_0)}{\sqrt{2\pi\sigma_0}}\right)\right] \tag{1.10}$$

where N_0 is the total number of particles. Thus, r_0, σ_0, and N_0 fully describe a log-normal distribution. r_0 can be expressed in terms of the mass (M_0) of particles per unit volume:

$$M_0 = \frac{4}{3}\pi\rho N_0 r_0^3 \exp(4.5\sigma_0^2) \tag{1.11}$$

where ρ is the particle's density. Solving for r_0 gives:

$$r_0 = \left(\frac{3M_0}{4\pi\rho N_0}\right)^{1/3} \exp(-1.5\sigma_0^2) \tag{1.12}$$

Thus, the particle size distribution for the transported dust is fully described by three parameters: M_0, N_0, and σ_0. This can be simplified further by setting σ_0 to a constant. In MGCM 3.1, $\sigma_0 = 0.6376$ for dust, which gives an effective variance of 0.5.

Sedimentation

Gravitational sedimentation in MGCM 3.1 is based on the Stokes-Cunningham relationship for particle fall velocity (v_f) with the slip correction for the thin Martian atmosphere. The fall velocity is given by:

$$v_f = \frac{2gr^2\rho_P}{9\vartheta_a}(1 + \alpha K_n) \tag{1.13}$$

where g is gravity, ρ_P the particle density, ϑ_a the dynamic viscosity of air, and α the Cunningham slip-flow correction given by:

$$\alpha = 1.246 + 0.42 \exp\left(\frac{-0.87}{K_n}\right) \tag{1.14}$$

Dust Lifting

There are multiple dust lifting methods available in MGCM 3.1. The lifting parameterizations can broadly be divided into two categories: map-tracking schemes and interactive lifting schemes. These options are explored below.

Dust Lifting: The Map-Tracking Method

The default setup for dust lifting is the map-tracking method. The tracking algorithm compares the model-predicted column dust tracer visible opacity at each grid point with its counterpart in the prescribed dust map or constant opacity. If the predicted opacity is less than the prescribed opacity, dust is lifted from the surface into the lowest model layer. In the default setup, gravitational sedimentation is the only sink for dust which can result in the overprediction of tracer-transported dust relative to the dust map, particularly in locations of strong dust storm activity. An additional option that alleviates this is an ad-hoc sink for the dust that ensures agreement with the dust map at all times, but note that it is not based on any physical process. See Section 4.3.4 for instructions for enabling this flag, and Appendix B for a table of the dust namelist parameters.

Dust Lifting: The Interactive Method

Fully interactive dust lifting occurs in the model through parameterizations of wind stress (to represent lifting by saltation) and dust devil dust lifting. While there are multiple schemes available for both wind stress and dust devil lifting in the MGCM, only one scheme of each type is supported and described here for the version 3.1 release.

Interactive Dust Lifting: Wind Stress

Of the multiple schemes available for wind stress lifting in MGCM 3.1, the only supported scheme is the one described in Kahre et al. (2008, 2015, 2006, 2005). This scheme is based on an empirical relationship between surface friction velocity and dust lifting in the Saharan Desert developed by Westphal et al. (1987). It was modified for Mars by Haberle et al. (2003) and implemented into fully interactive dust cycle simulations by Kahre et al. (2008, 2015, 2006, 2005). Instructions for implementing wind stress lifting can be found in Section 4.3.4.

In this scheme, the flux of dust (F_W) injected into the atmosphere depends on the surface wind stress (τ) and a wind stress threshold ($\tau*$) required for lifting:

$$F_W \, (kg \, m^{-2} \, s{-}1) = \alpha_W (2.3 \times 10^{-3})(\tau^2) \left(\frac{\tau - \tau*}{\tau*} \right) \tag{1.15}$$

where α_W is a tunable efficiency factor. Parameters are provided for wind stress lifting based on a threshold stress of 22.0 (m N m^{-2}) at c24, L56. Note that the tuning will likely change as resolution (horizontal and vertical) changes, as well as the lifted particle size, etc.

Interactive Dust Lifting: Dust Devils

Of the multiple schemes available for dust lifting by dust devils (convective vortices) in MGCM 3.1, the supported scheme is based on the thermodynamic theory of dust devils developed originally by Renno et al. (1998). The dust devil dust lifting parameterization was first implemented into a GCM by Newman et al. (2002a, 2002b) and has since been used by Basu et al. (2004, 2006) and Kahre et al. (2008, 2015, 2006, 2005). Instructions for implementing dust devil lifting can be found in Section 4.3.4.

In this scheme, the lifted dust flux by dust devils (F_D) depends on the sensible heat flux (F_s) and the depth of the planetary boundary layer (PBL) in the following manner:

$$F_D \, (kg \, m^{-2} \, s{-}1) = \alpha_D \cdot F_s \cdot (1 - b) \tag{1.16}$$

where α_D is a tunable efficiency parameter and the parameter b is defined as:

$$b = \frac{p_{sfc}^{\chi+1} - p_{con}^{\chi+1}}{(p_{sfc} - p_{con})(\chi + 1)\, p_{sfc}^{\chi}} \tag{1.17}$$

where p_{sfc} is the surface pressure, p_{con} is the pressure at the top of the PBL, and χ is the gas constant divided by the specific heat capacity at constant pressure. Dust is injected into the atmosphere when the vertical heat flux is positive, which results in dust lifting at almost all locations during daylight hours (except when CO_2 ice is present on the surface, which is a user-defined flag that can be optionally turned off).

1.2.5 Parameterized Orographic Gravity Waves

The orographic scheme from Palmer et al. (1986) is implemented to represent the effective drag imposed on the atmosphere by the deposition of horizontal momentum associated with the dissipation of vertically-propagating orographic gravity waves. The scheme effectively accounts for the reaction force of unresolved orography (at c24, c48, and c96 resolutions) on the resolved regional-scale winds, and the upward propagation of the associated momentum (Figure 1.4). This scheme was first employed in a Mars context by Collins, Lewis, and Read (1997).

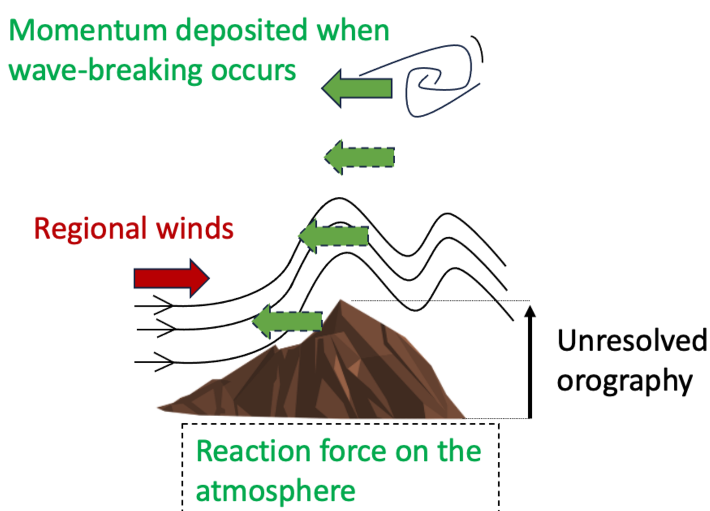

Figure 1.4: The physical processes represented by the parameterization for subgrid-scale orographic gravity wave drag.

The Palmer et al. (1986) scheme multiplies the variance of the unresolved orography by the low level winds (among other terms) to obtain the surface base flux for the orographic gravity wave. A coupling parameter (KAP) analogue to a characteristic horizontal wavelength is used as a tunable parameter for the scheme. It roughly reflects the grid-scale wavelength of the simulation, hence KAP $= 5 \times 10^{-6}$ (m^{-1}) is 200 km. Working upward through the atmospheric column, the stability of the layers to that gravity wave is evaluated. When saturation is detected and wave breaking occurs, momentum is transferred to the atmosphere in the form of a deceleration tendency (drag; m s^{-2}) applied to both the zonal and meridional winds. The implementation includes the vertical wavelength-dependent thermal damping rates from Eckermann et al. (2011). We use a thermal damping profile appropriate for a 10 km vertical wavelength in this parameterization.

1.2.6 Code Architecture

Figure 1.5 shows the order in which the physics routines are called in MGCM 3.1. All of the physics parameterizations are stored in the source code under:

```
AM4/src/AmesGCM/
```

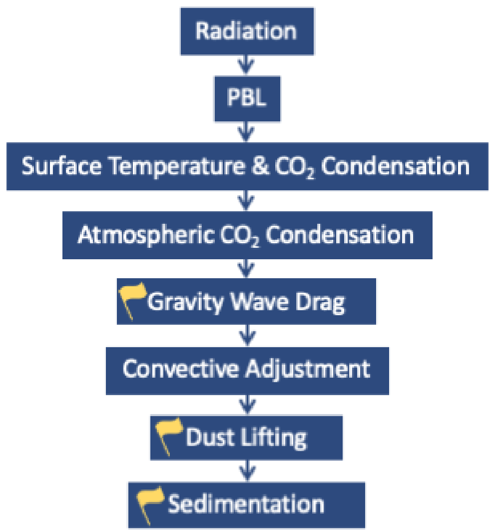

Figure 1.5: The order in which the physics modules are called in MGCM 3.1. Flags denote processes that can optionally be turned off in MGCM 3.1

The MGCM 3.1 physics module begins with the radiation driver and is followed by the PBL scheme. After that, surface and sub-surface temperatures as well as surface CO_2 condensation are calculated. Next, the atmospheric CO_2 condensation module condenses CO_2 from the atmosphere to maintain air temperature at or above the CO_2 condensation temperature. Optionally, the model can then do the orographic gravity wave parameterization. A convective adjustment is then performed to remove any unstable layers. Finally, dust processes of lifting and sedimentation are performed.

1.3 Physics: Early Mars

MGCM 3.1 can be used to simulate past Martian climates. These include more massive CO_2 atmospheres and hydrogen-rich atmospheres. The radiative transfer scheme in the MGCM includes an alternative correlated-k table that is appropriate for massive CO_2 atmospheres (see Table 2.2). Additionally, the collision induced absorption (CIA) scheme used in K. E. Steakley et al. (2023) is available in the MGCM to account for the infrared opacity produced by CO_2–CO_2 and/or CO_2–H_2

collisions in the atmosphere. For the early Mars configuration, radiative transfer is calculated over the 15 wavelength bands listed in Table 2.2.

Running simulations with commonly used early Mars parameters is straightforward to do from the runscript (see Section 4.3.6 for instructions). In the default early Mars configuration, the global surface pressure is set to 500 mb. The solar constant is reduced to represent the faint young Sun, which was only 75% of its present day luminosity 3.8 billion years ago (Ga) (Gough, 1981). The early Mars configuration specifies a circular orbit and present-day obliquity. Surface albedo, thermal inertia, and emissivity are set to constant values of 0.2, 250.0 (J m^{-2} $s^{-0.5}$ K^{-1}), and 1.0, respectively, instead of using surface maps for these values. There is no effect of subsurface ice accounted for in the thermal diffusivity calculation. When CO_2 ice is present, the surface albedo is 0.5 and emissivity is 0.85. These constants were chosen to be consistent with Forget et al. (2013), K. Steakley et al. (2019) and K. E. Steakley et al. (2023). Soil temperatures are initialized to warmer values in the early Mars configuration (220.0 K) than those for current Mars.

The default early Mars configuration in MGCM 3.1 produces a 2-year simulation of a massive CO_2 atmosphere with a 500 mb surface pressure. Output from this simulation is described in Section 6.2. Only limited early Mars physics are included in this release. MGCM 3.1 does not currently support running with a dust cycle or water cycle in the early Mars configuration. While this release does not include CO_2 clouds, a CO_2 cycle is included with atmospheric CO_2 condensation and instantaneous fall of condensed mass onto the surface. With this treatment of CO_2 condensation, if current Mars topography is included, the atmosphere collapses as large amounts of CO_2 condense onto the colder, high topography regions on the surface. For this reason, the default early Mars configuration uses a flat surface instead of current topography. Additional physics options for early Mars are in development and we anticipate that they will become available in subsequent MGCM releases.

Chapter 2

Model Input Files

This chapter describes the input file structure and content in MGCM 3.1. First, we present the input files for the surface fields and the RT scheme. Then, we describe the various dust scenarios that can be used to inform the prescribed dust field in MGCM 3.1, and we provide a discussion about the "restart" files that inform "warm" starts. Finally, we describe the field and diag tables, which are created in the runscript and define tracer fields and output variables, respectively.

2.1 Surface Fields

All of the input files for MGCM 3.1 discussed in this chapter are located in the `data/` directory at:

```
AM4/src/AmesGCM/data/
```

Included in this directory are the surface topography, albedo, emissivity, and thermal inertia files:

```
mars_topo_mola16.nc            # topography
mars_TES_albedo8_new.nc        # albedo
mars_sfc_emissivity8.nc        # emissivity
mars_thermal_inertia8_2011.nc # thermal inertia
```

Surface topography is $1/16°$ resolution MOLA data (Smith et al., 1999). Bare soil albedos and emissivities are $1/8°$ resolution MGS TES data (Putzig & Mellon, 2007). Thermal inertias are derived by Tyler and Barnes (2014) and have been mapped onto a $1/8°$ grid for ingestion into the MGCM. All surface fields are re-gridded to the resolution of the simulation at runtime (Wilson et al., 2007). Regions of high thermal inertia represent the SPRC, as the permanent CO_2 ice cap is not explicitly represented in the MGCM. The seasonal CO_2 ice caps in both the north and south grow and recede according to a surface energy balance that considers thermal inertia, surface albedo, topography, and the CO_2 ice albedo (0.65 for the northern ice and 0.43 in the south). The CO_2 ice albedos are namelist parameters that can be modified in the runscript. Their default values are chosen to produce a good fit to the Viking pressure cycle as shown in Figure 1.2. Finally, a distribution of subsurface ice is prescribed following a spatial pattern consistent with gamma ray spectrometer (GRS) measurements (e.g. Boynton et al., 2002). The necessity of subsurface ice for modeling the CO_2 cycle is described in Haberle et al. (2008).

2.2 Radiation Code

MGCM 3.1 uses a two-stream correlated-k radiation code to account for the radiative effects of CO_2 and H_2O gas and atmospheric aerosols (Haberle et al., 2019; Toon et al., 1989). Dust and water ice clouds can optionally be radiatively active in the radiation code, but we note that MGCM 3.1 does not support a water cycle, so dust is the only radiatively active aerosol in the model (see Chapter 4 for options). The radiation code for *current* Mars simulations is based on 12 wavelength bands: 7 in the visible and 5 in the infrared as shown in Table 2.1.

Table 2.1: 12-band spectral intervals for the radiation code.

Band	Wavelength Interval (µm)
Visible 1	4.50 – 3.24
Visible 2	3.24 – 2.48
Visible 3	2.48 – 1.86
Visible 4	1.86 – 1.31
Visible 5	1.31 – 0.80
Visible 6	0.80 – 0.40
Visible 7	0.40 – 0.24
Infrared 1	1000.0 – 60.0
Infrared 2	60.0 – 24.0
Infrared 3	24.0 – 12.0
Infrared 4	12.0 – 8.00
Infrared 5	8.00 – 4.50

The correlated-k and dust scattering property tables for the 12-band radiation code are available in the source code and listed below for reference.

```
CO2H2O_IR_12_95_INTEL             # correlated-k visible table
CO2H2O_V_12_95_INTEL              # correlated-k infrared table
Dust_vis_wolff2010_JD_12bands.dat # dust properties visible
Dust_ir_wolff2010_JD_12bands.dat  # dust properties infrared
```

For each of the twelve spectral intervals, correlated-k's were generated for a binary mixture of CO_2 and water vapor from a line-by-line code using the HITEMP database (Rothman et al., 2010) from HITRAN (Gordon et al., 2022) for CO_2 and a version of the Schwenke database (Schwenke, 1998) for H_2O. The absorption coefficients were sorted, reordered in g space (cumulative probability space), and stored in a look-up table for use in the MGCM.

An offline Mie code was used to compute the scattering properties (Q_{ext}, Q_{scat}, and g) of the dust based on the refractive indices of the dust from Wolff et al. (2009). For each spectral interval, the Planck-weighted (215 K in the infrared; 6,000 K in the visible) properties were computed for 20 mono-disperse populations with sizes ranging from 0.1–50 μm. Finally, dust is defined by a size distribution given by a user-defined effective radius and variance (0.5 in MGCM 3.1). The scattering properties of this (fixed) dust distribution are then computed using the visible and infrared wavelengths for dust in Table 2.1.

The radiation code can also be run with 15 wavelength bands instead of 12. In the 15-band version, there are 7 wavelength bands in the visible and 8 in the infrared as shown in Table 2.2.

Table 2.2: 15-band spectral intervals for the radiation code.

Band	Wavelength Interval (μm)
Visible 1	4.50 – 3.24
Visible 2	3.24 – 2.48
Visible 3	2.48 – 1.86
Visible 4	1.86 – 1.31
Visible 5	1.31 – 0.80
Visible 6	0.80 – 0.40
Visible 7	0.40 – 0.24
Infrared 1	1000.0 – 33.3
Infrared 2	33.3 – 18.2
Infrared 3	18.2 – 15.7
Infrared 4	15.7 – 14.2
Infrared 5	14.2 – 12.9
Infrared 6	12.9 – 10.3
Infrared 7	10.3 – 7.80
Infrared 8	7.80 – 4.50

The correlated-k and dust scattering property tables for the 15-band radiation code are available in the source code and listed below for reference. They are generated via the same methodology as the 12-band tables.

```
CO2H2O_V_2013_32                  # 15-band correlated-k visible table
CO2H2O_IR_2013_32                 # 15-band correlated-k infrared table
Dust_vis_wolff2010_JD_15bands.dat # 15-band dust properties visible
Dust_ir_wolff2010_JD_15bands.dat  # 15-band dust properties infrared
```

For *early* Mars simulations, separate, 15-band correlated-k tables appropriate for massive CO_2 atmospheres are available. These were introduced in K. Steakley et al. (2019) and used again in

K. E. Steakley et al. (2023). The visible and infrared correlated-k's for early Mars were produced using coefficients from the HITRAN 2012 database (Rothman et al., 2013) and use a sublorentzian line-shape with correction factors from Perrin and Hartmann (1989), and a line truncation of 500 cm^{-1} at pressures $>$10 mb to account for far line absorption. The correlated-k tables for early Mars are also available in the source code and are listed below for reference.

```
CO2H2O_V_15B_800K_v4     # early Mars correlated-k visible table
CO2H2O_IR_15B_800K_v4    # early Mars correlated-k infrared table
```

Collision-induced absorption (CIA) can also be accounted for in the radiation code for early Mars simulations. Both CO_2-CO_2 and CO_2-H_2 CIA treatment options are available. Wordsworth et al. (2010) and Forget et al. (2013) emphasize the importance of accounting for CO_2–CO_2 collisions when simulating massive CO_2 atmospheres (Forget et al., 2013; Wordsworth et al., 2010). The CO_2–CO_2 CIA coefficients used here are from Wordsworth et al. (2010) and the CO_2–H_2 CIA coefficients are from Turbet et al. (2020). The tables for CIA are available in the source code and are listed below for reference.

```
kbbar_8band_100_800.dat      # optical properties for CO2-CO2 CIA
kgbar_8band_100_800.dat      # optical properties for CO2-CO2 CIA
khbar_8band_T20_100_800.dat  # optical properties for CO2-H2 CIA
```

2.3 Dust Scenario Files

The dust scenarios that inform the horizontal dust distribution are defined by the `dust_scenarios` variable in the runscript. Instructions for further modifying the dust prescription are provided in Chapter 4. We limit discussion here to describing the dust scenario files that are included in the release of MGCM 3.1.

The dust scenario files for MGCM 3.1 are listed below.

```
DustScenario_MY24.nc        DustScenario_MY34.nc
DustScenario_MY30.nc        DustScenario_Background.nc
DustScenario_MY31.nc        DustScenario_MGS.nc
```

The dust scenarios are automatically interpolated to the model grid upon initialization so that every scenario works for any model resolution. Among the included files are dust absorption maps for Mars Year (MY) 24, MY 30, MY 31, and MY 34 from the Montabone dust climatology (Montabone et al., 2015), in which the normalized dust column is specified as:

```
dustcol(lon,lat,Ls)
```

and the vertical distribution of the dust is zonally uniform and given by:

```
zmax(lon,lat,Ls)
```

There is also a dust scenario file called `DustScenario_Background.nc` that was created to represent a typical annual dust cycle. It is a combination of the Montabone dust climatologies with additional processing that smooths out any short term dust events. It is therefore not specific to any particular Mars Year and thus represents a typical Mars dust cycle. Finally, there is a dust scenario proposed by Montmessin et al. (2004) called `DustScenario_MGS.nc` that has been used in multiple other studies.

2.4 Orographic Gravity Wave Files

The subgrid-scale orographic parameterization requires a representation of the unresolved orography at the chosen horizontal resolution. For each of the c24 (∼4°), c48 (∼2°), and c96 (∼1°) resolution cases, the mean elevation and the standard deviation against a $1/16°$ topographic map are pre-computed in the following input file:

```
palmer_drag_input_c24_c48_c96.nc
```

For example, for c24, the following input variables are read by the parameterization from the above file:

```
topo24:('lat', 'lon')=(1440, 2880), topography, resampled c24 [m]
SD24:  ('lat', 'lon')=(1440, 2880), standard deviation, resampled c24 [m]
```

2.5 Restart Files

At the end of every simulation, the model saves its current state in a series of files called "restart" files. These are tarred together and stored in the `restart/` directory on Pleiades:

```
/nobackup/$USER/FMS_MARS_runs/am4_mars_runs/ fms_mars_default/restart/
```

At the beginning of a warm start, the requested restart file is copied to the `INPUT/` directory and untarred so that the model can reference the saved state. Restart files contain data stored on the native grid in tile-specific files. Data is saved from the last timestep of the iteration and a 5-digit code in the tarred file name indicates the sol number of the data in the file. For example, a file named `00668.restart.tar` has data timestamped beginning from the last timestep of the 668^{th} day of a simulation. The variables stored in the restart files are only those required to reproduce the saved state of the simulation. Most notably, these include zonal wind (`u`), meridional wind (`v`), temperature (`T`), layer thickness (`delp`), and geopoential height (`phis`).

2.6 The Field Table

The field table is located in-line in the runscript and is used to define additional fields known as "tracers" in the simulation. MGCM 3.1 carries multiple (dust and water) tracers that are disconnected from any physics routines, are always zero, and are thus not supported in this release. These tracers are required for the radiation code and will be connected (with appropriate physics) and supported in a future public release of the model. Although MGCM 3.1 does not carry supported tracers, we provide a brief description of the field table here because it appears in the runscript.

Tracers are added to the field table by specifying their name (`'name'`), the module in which they are defined (`'module'`), their unit (`'units'`), and the name the tracer will have in the output file (`'longname'`). These parameters are formatted as follows:

```
'TRACER', 'module',      'name'
          'longname',    'user-defined-name-here'
          'units',       'user-defined-unit-here'
```

> **NOTE:** When running the model from a warm start, if the tracer `'name'` is not found in the restart files, then the tracer is initialized to zero.

2.7 The Diag Table

The diag table is where the output file types are defined and the variables to be written to those files are specified. The in-line diag table in the runscript lists the minimum fields that *must* be output by the model. There is also an "external" diag table (`diag_table.ext`) that is included because some users prefer to separate their added variables from those listed in the in-line diag table.

> **NOTE:** The external table (`diag_table.ext`) omits the section defining the output filetype. It is primarily used for specifying additional variables to write out.

The following subsections provide more detail about the two parts of the in-line diag table: defining the output files (Section 2.7.1) and assigning variables to those files (Section 2.7.2).

2.7.1 The Diag Table: Defining the Output Files

The first eight lines of the in-line `diag_table` define the output file names and the frequency at which output is stored in each of the files. MGCM 3.1 creates and stores variables in five netCDF files, but the user can add or remove files (or change the file names) if they so choose. The five files defined in the default runscript are:

- `grid_spec.nc`
- `fixed.nc`
- `atmos_daily.nc`

- `atmos_average.nc`
- `atmos_diurn.nc`

The `grid_spec.nc` file contains the cubed-sphere geometry information required for post-processing (e.g., for conversion to a regular latitude-longitude grid; see Chapter 5). Both `grid_spec.nc` and `fixed.nc` are invariant and only need to be written out once, but the model outputs the files every run iteration for consistency. The other three files (`atmos_*.nc`) store atmospheric variables that change over time. Each file stores output at different frequencies. The default settings are listed below.

```
! filename,       freq,  unit,  format, time_unit, time_name,
'grid_spec',       -1,  'hours', 1,     'days',     'time',
'fixed',           -1,  'days',  1,     'days',     'time',
'atmos_daily',      6,  'hours', 1,     'days',     'time',
'atmos_average', 5,  'days',  1,     'days',     'time',
'atmos_diurn',      5,  'days',  1,     'days',     'time',
```

From left to right, the columns above define the name of the output file (`filename`), the output frequency (`freq`), the output unit (`unit`), the file format (`format`), the time unit (`time_unit`), and time name (`time_name`). Complete descriptions of each of these column parameters is listed below.

1. `'filename'` – the name of the file that will store the fields
2. `'freq'` – an integer determining output frequency. If negative, output is written only once at the end of each iteration. If 0, output is written every physical time step (defined by `dt_atmos`)
3. `'unit'` – the time unit for the output frequency
4. `'format'` – should always be 1 for netCDF format
5. `'time_unit'` – the output time axis unit (`'years'` , `'months'` , `'days'` , `'hours'` , `'minutes'` , `'seconds'`)
6. `'time_name'` – the output time axis name (string must contain `'time'`)

Users may specify the following additional columns to more specifically define how often new input files are created and data are archived. These optional parameters are listed in column-order below.

7. `'new_file_freq'` – an integer defining how often a new file is created in `'new_file_unit'` time
8. `'new_file_unit'` – the unit for the new file frequency (e.g. `'days'` , `'hours'`)
9. `'start_time'` – a 6-integer string for writing the files in the format: “year, month, day, hour, minute, second”
10. `'file_dur'` – an integer setting the duration of the file period
11. `'file_dur_unit'` – the string unit for the file duration

The first part of the in-line diag table defines the output file types and archival frequencies. The next subsection describes the rest of the diag table, which specifies the variables to archive.

2.7.2 The Diag Table: Assigning Variables to the Output Files

The in-line diag table lists the minimum variables that are to be written to the output files. In the diag table, one variable is registered per line as follows:

```
'module_name', 'field_name', 'output_name', 'file_name', 'time_sampling',
↪  'time_method', 'spatial_opts', 'pack'
```

Descriptions for each of the above fields are provided in column order below.

- `'module_name'` – the submodule where the field is defined (e.g. `'dynamics'`, `'mars_physics'`, etc.)
- `'field_name'` – the field name as registered in the submodule above
- `'output_name'` – the field name as it will appear in the output file (customizable).
- `'file_name'` – the file in which to output the field (`'fixed'`, `'grid_spec'`, `'atmos_average'`, `'atmos_diurn'`, `'atmos_daily'`, etc.)
- `'time_sampling'` – the sampling frequency (`'all'` samples every time step)
- `'time_method'` – the output frequency for the field:
  - `.true.` for averaging every timestep over the requested interval [frequency]
  - `.false.` for instantaneous output
  - `'min'` for a minimum within an averaging interval
  - `'max'` for a maximum within an averaging interval
  - `'diurnal24'` for hourly output
  - `'diurnal12'` for output every 2 hours
- `'spatial_opts'` – the location from which the field is sampled (`'none'` for global output or `'lonmin lonmax latmin latmax Pmin Pmax'` [Pa] for output from a specific area)
- `'pack'` – the numeric type of the output values (`1` for double precision, `2` for float, `4` for 16-bit packed integers, and `8` for packed 1-byte)

The minimum required variables are pre-defined in the in-line diag table and listed in Tables 5.1–5.5 in Chapter 5. As a reminder, additional variables can be appended to either the in-line diag table or the external diag table, `diag_table.ext`.

Chapter 3

Obtaining the Model

The next two Chapters (Chapters 3 and 4) provide more detailed instructions for downloading, compiling, and running MGCM 3.1 than those provided in Part I of the User Manual. Some of the settings required to run the model in non-NAS environments are included here, but additional instructions for running the MGCM 3.1 on Mac OS can be found in Appendix A. MGCM 3.1 has yet to be tested on other systems and fully documenting the necessary changes continues to be a work in progress.

3.1 Software Requirements

MGCM 3.1 is built by installing the Mars physics package developed by the MCMC on top of the AM4 model from NOAA-GFDL. The MCMC provides GitHub "patches" that integrate the Mars physics paramaterizations with the AM4 software to convert the model into a Mars simulator. The required software for downloading and running MGCM 3.1 are listed below.

- **A Fortran Compiler** – MGCM 3.1 is primarily written in Fortran and includes some C, C++, HTML, and Shell scripts as well. As such, a Fortran compiler is necessary for running the model. We recommend the Intel Fortran Compiler **ifort**, but the GNU compilers **GCC** and **gfortran** are also supported.
- **netCDF4** – MGCM 3.1 input and output files are in netCDF format, which is recognized by most of the commonly-used scientific programming languages today including Python, MATLAB, IDL, GrADS, and netCDF-Fortran.
- **An MPI Distribution** – Installation of an MPI library is required for parallel processing.
- **Git** – MGCM 3.1 is released on GitHub and requires Git version 1.8.4 or later to install it.
- **Mars Physics Packages** – MGCM 3.1 physics are hosted on NASA's GitHub. Instructions for cloning the Mars physics packages are below.
- **NOAA-GFDL AM4** – The AM4 model provides the model infrastructure and the dynamical core for MGCM 3.1. Instructions for cloning the AM4 repository are below.

In the following sections, we provide instructions for cloning the NOAA-GFDL AM4 public release, cloning the MGCM physics packages, and patching them together to build MGCM 3.1.

The instructions in this chapter are specific to NAS and we therefore provide NAS-specific code for loading the relevant modules. Appendix A provides instructions for installing MGCM 3.1 on Mac OS. For non-NAS and non-Mac users, note that the instructions in both this chapter and Appendix A assume a Fortran compiler, the netCDF4 and MPI libraries, and Git are installed in your environment already.

3.2 Cloning the GitHub Repository

The physics packages for MGCM 3.1 are available on the NASA GitHub repository at https://www.github.com/nasa/AmesGCM. The AM4 dynamical core is available on the NOAA-GFDL GitHub at https://github.com/NOAA-GFDL/AM4.git. Begin by cloning the AM4 model to your preferred directory on Pleiades:

```
# load pkgsrc for git submodules, updating git version
user@pfe:~$ module load pkgsrc/20XXQX # use available version
user@pfe:~$ git clone --recursive --branch 2021.03
↪  https://github.com/NOAA-GFDL/AM4.git
```

> **NOTE:** The `module load` line is NAS-specific. It loads a module required for working with GitHub. Non-NAS systems may require a similar package to perform the installation. For installation on Mac OS, this line can be skipped.

Next, go into the `AM4/src/` directory and remove the `ice_param` module:

```
user@pfe:~$ cd AM4/src/
user@pfe:~$ rm -rf ice_param
```

Stay in the `src/` directory and create and switch to `mars_branch`:

```
user@pfe:~$ git checkout -b mars_branch
```

Add the Mars physics submodules to AM4:

```
user@pfe:~$ git submodule add https://www.github.com/nasa/AmesGCM
user@pfe:~$ git submodule add https://github.com/NOAA-GFDL/mkmf.git
user@pfe:~$ git submodule add https://github.com/NOAA-GFDL/FRE.git
```

Apply the patches to stitch together the AM4 model and the Mars physics:

```
user@pfe:~$ cd AmesGCM/patches
user@pfe:~$ ./apply_patch.sh
```

This creates a directory called `AM4/` containing the following:

```
analysis/
bin/
container/
exec/
run/
src/
README.md
```

The source code for the model is in the `src/` directory, and the Mars physics packages specifically are stored in `src/AmesGCM/`. The model is run from the `exec/` directory, which houses the default runscript (`fms_mars_default`).

All of the software required for building the model has been installed. Chapter 3 provides instructions for building and running MGCM 3.1 on NAS.

Chapter 4

Building and Running the Model

4.1 Building the Model

MGCM 3.1 is compiled by linking to the appropriate libraries and executing the compile script. This creates an executable that is then used to run the model. MGCM 3.1 can be run out-of-the-box on the NAS computing system using the `fms_mars_default` runscript as described in this chapter and in Part I. It can also be run on other systems, such as Mac OS (see Appendix A), and the default runscript can be modified at the user's discretion. In this section, we describe the process of compiling and running the model in more detail and include instructions for modifying the runscript.

4.1.1 Compiling MGCM 3.1

MGCM 3.1 is optimized for building and running on NAS. Compiling the model on Pleiades is straightforward. However, there are software requirements that must be met in order to run the model. These are listed and explained in Chapter 3 for your reference. This chapter assumes you have the necessary software installed already.

To build the model with the default settings, go to the `AM4/exec/` directory and compile the code using the provided script. If you have been following the instructions in Chapter 3, then you are likely still in the `patches/` directory:

```
AM4/src/AmesGCM/patches/
```

and you must go back three directories to `src/` and forward one directory into `exec/`:

```
user@pfe:~$ cd ../../../exec/
```

You can double-check that you are in `AM4/exec/` using `pwd`:

```
AM4/exec/
```

Execute the compile script from `AM4/exec/`:

```
user@pfe:~$ ./compile.archives
```

This creates a compiled version of the MGCM called `exec.intel.mars3.1` and MGCM 3.1 has been compiled with the default settings. Instructions for compiling with different settings are below. Instructions for running the model are in Chapter 4

Compile Options

Those who are utilizing NAS to run MGCM 3.1 will likely not need to change any settings in the compile script. However, modifications to the compile script are likely necessary for users running MGCM 3.1 in non-NAS environments. Specifically, the local compiler library and the netCDF and compiler directories need to be modified in order for the MGCM to run on non-NAS platforms. For Mac OS-specific instructions, see Appendix A. For all other systems, we provide a list of some of the options in the compile script that might need modification:

- `platform` – specifies the compiler to use (Intel and GNU are currently supported; Intel is suggested for NAS)
- `NETCDF` – the directory of the netCDF environment
- `NETCDFPATH` – also the directory of the netCDF environment
- `INTEL_LICENSE_FILE` – the directory containing the platform license
- `execdir` – the directory containing the compiled source code and the executable
- `execname` – the name of the executable

There are two additional options listed under `CPP_defs` that may have to be changed for non-NAS environments. These options are read by the compiler before compiling the code:

- `MARS_GCM` – [required] activates calls to Mars physics
- `DMARS_GDIAGS` – [optional] activates global sum diagnostics output (i.e., global mean surface pressure, global total water vapor or dust, etc.). This option slows the compilation time

Finally, the model can be compiled with optimizations for the specific CPU types available on NAS by defining the `optim` variable:

```
set optim = ""       ! leave blank for generic processors, or:
! for Pleiades:
set optim = bridge   ! for Sandy Bridge or Ivy Bridge processors
set optim = well     ! for Haswell or Broadwell processors
! for Aitken:
set optim = lake     ! for Cascade Lake processors on Aitken
set optim = rome     ! for Rome processors on Aitken
```

4.2 Running the Model

After compiling MGCM 3.1, the model can be run by submitting a runscript to the NAS PBS system. In the `AM4/exec/` directory, the default runscript `fms_mars_default` produces a simulation with minimal modifications. In this section, we provide instructions for modifying and submitting that runscript to the queue for processing. The following subsections are named according to the headers in `fms_mars_default` to make navigating the runscript and this chapter as straightforward as possible.

> **WARNING:** The MCMC currently supports running MGCM 3.1 on NAS and Mac OS only. The MCMC does not have experience running the model outside of these platforms and will likely not have answers to troubleshooting questions specific to non-NAS or non-Mac OS environments.

4.2.1 PBS Settings

The first several lines of the runscript define the PBS options for the job submission. There are a few settings that must be modified to reflect the user submitting the job:

```
-W group_list=sXXXX ! the account to be billed
-M user@email.com   ! email for job status updates
```

Other PBS options that are commonly modified in the runscript include the queue (`-q`) the job will be submitted to, the number of nodes (`-l select`) and processors (`ncpus`) requested, the node type (`model`), and the amount of walltime (`-l walltime`) the job needs. The default values for these options are listed below.

```
-q normal ! the queue
-l select=3:ncpus=28:model=bro ! 3 broadwell nodes, 28 processors/node
-l walltime=08:00:00 ! HH:MM:SS
```

> **NOTE:** More information on choosing the right number of processors for a job are in Section 4.2.2.

Complete documentation for these and additional PBS settings can be found on the NAS website at https://www.nas.nasa.gov/hecc/support/kb/portable-batch-system-(pbs)-overview_126.html

4.2.2 Execution Variables

The next part of the runscript defines the model grid structure and associated time step (δt). It also defines the locations of the input and output file directories and identifies the working directory. The variables in this section are listed in Table 4.1 in the order in which they appear in the runscript.

Table 4.1: Execution Variables and Descriptions

Variable	Default Value	Description
`name`	`fms_mars_default`	output file directory
`scriptname`	`$cwd/$name`	runscript path
`classdir`	`am4_mars_runs`	parent output directory
`workdir`	`/nobackup/$USER/FMS_MARS_runs` ↪`/$classdir/$name`	model execution directory
`datadir`	`/nobackup/$USER/FMS_MARS_data`	input file directory
`platform`	`intel`	compiling environment (intel or GNU)
`TILE_LAYOUT`	`3,4`	layout of the processor. Required number of CPUs: `6*TILE_LAYOUT1*TILE_LAYOUT2`
`model_executable`	`$cwd/exec.$platform.am4` ↪`/FMS_MARS.x`	executable path
`homedir`	`$cwd`	executable directory
`NCX`	`24`	horizontal resolution. Default: 24 grid cells per cube face column/row (≈4x4°)
`DTA (dt_atmos)`	`924`	atmospheric (physics) time step; decreases with higher resolution
`NKS (k_split)`	`1`	number of vertical remapping iterations per `DTA`; increases with higher resolution
`NNS (n_split)`	`4`	number of advective time steps per `NKS`; increases with higher resolution
`NPZ`	`56`	vertical grid to reference
`NPZ_RST`	`0`	number of vertical layers in restart file (“`0`” if `NPZ_RST=NPZ`)

NOTE: The variables `TILE_LAYOUT`, `DTA`, `NKS`, and `NNS` are unix variables that set the namelist variables `layout`, `dt_atmos`, `k_split`, and `n_split`, respectively, later in the runscript. They appear together at the top of the runscript for ease of use.

We *highly* recommend that the `name` parameter matches the name of the runscript itself. This way, the runscript and the output directory to which output files are written share the same name. Likely the only time `name` should match a pre-existing output directory name is when you want to extend a run out longer (i.e. perform a *continuation* warm start, see Section 4.2.3).

WARNING: Setting `name` to an existing output directory name may overwrite pre-existing output files in that directory.

We support two horizontal resolutions in this release: c24 and c48 (defined by `NCX` in Table 4.1). The former (c24) is a $\sim$3.75° simulation, and the latter (c48) is a $\sim$1.875° simulation. The physics (atmospheric) time step (`dt_amos`) is set by `DTA` and must divide evenly into the length of the Mars day as defined in the model (88,704 seconds). Generally, `DTA` should be decreased when running with higher horizontal resolution. We have found `DTA = 924` (default) or `DTA = 462` are good choices for the lower-resolution simulations we have tested (c24 and c48).

The advective (dynamical) time step is calculated by the model and given by:

```
advective δt = total dynamical δt = dt_atmos/(k_split*n_split)
```

where `k_split` is the number of vertical remapping iterations per physics timestep (`dt_atmos`) and `n_split` is the number of advective time steps per `k_split`. The vertical remapping conservatively regrids the Lagrangian layers onto a set of "Eulerian" reference coordinates that many of the physics paramaterizations use (Harris et al., 2021). This remapping resolves distortions in the vertical layers that might otherwise lead to stability problems. Ideally, `n_split` should be distinctly larger than `k_split`. The advective time step is governed by the Courant Friedrichs-Le (CFL) condition and is thus dependent on model resolution.

NOTE: As a general rule, `n_split` should be distinctly larger than `k_split`

The total number of processors (CPUs) required to run the model is calculated automatically later in the runscript, but it is an important number for the user to be able to calculate so that they can request enough computing power in the PBS settings to run the MGCM. The total number of CPUs required for a run is six times the layout of the processor (`6*TILE_LAYOUT1*TILE_LAYOUT2`). The layout of the processor is given by `TILE_LAYOUT` which accepts two numbers indicating the number of CPUs to use on each processor in the X and Y directions (see Table 4.1). By default, `TILE_LAYOUT = 3,4` which means the number of CPUs used per cube face is twelve ($3*4 = 12$) and the total number of CPUs required is seventy-two ($12*6 = 72$).

NOTE: A caveat: `TILE_LAYOUT` is constrained by the resolution (`NCX`) of the simulation. Specifically, both of the numbers provided in `TILE_LAYOUT` must be less than `NCX/3`. In the default setup, `TILE_LAYOUT = 3,4`, `NCX = 24`, and `24/3=8` is larger than both integers in `TILE_LAYOUT`.

The total number of CPUs requested must be greater than or equal to the number of CPUs required by the model (72 CPUs in the default setup). PBS options vary by machine, but for NAS systems the user specifies the node type, which usually the Pleiades Broadwell Nodes (`model=bro`), and the number of nodes to use. Each node type has a fixed number of processors on each node (28 for Broadwell, `ncpus=28`) so the user will likely have to request more processing power than is necessary for a run. For the default simulation, the user must request at least three Broadwell nodes (`select=3`) in order to have enough processors to run the model, which amounts to a total of 84 processors requested (three nodes with 28 processors each):

```
! Requesting 3 broadwell nodes and 28 processors/node = 84 processors
-l select=3:ncpus=28:model=bro # default setting
```

4.2.3 Time Set-Up

This section of the runscript defines the simulation length and number of iterations. This is also where the user initializes the model from a "cold" or a "warm" start. Table 4.2 below lists the runtime variables.

Table 4.2: Runtime variables and descriptions

Variable	Default Value	Description
`dayslist`	`668 668`	length (Mars days) of each model integration
`num_executions`	`$dayslist`	number of times the model is run
`RUNTYPE`	`0`	selects a cold or warm start: initialize from scratch ("cold"; 0), continue from previous run ("continuation" 1), or restart from a different run ("warm"; 2)

`restartfile`	`/nobackup/$USER`
	`↪/FMS_MARS_runs`
	`↪/am4_mars_runs`
	`↪/fms_mars_default`
	`↪/restart`
	`↪/00668.restart.tar`
		path to the restart files if `RUNTYPE=2` (warm start from a file)

The first variable in Table 4.2, `dayslist`, accepts a list of numbers representing the number of Mars Days of data to be written to each output file. The second variable, `num_executions`, sets the number of iterations of the simulation. One set of output files will be generated for each iteration of the the model. The default setting has `num_executions` equal to `dayslist`, which means the number of entries in `dayslist` indicates the number of output files to be created. In other words, `dayslist` defines the run iteration. In the default case, `dayslist = (668 668)` so the model will perform two run iterations and write two sets of output files, each with 668 days (1 MY) of data. See Chapter 5 for more information on output file structures.

The `RUNTYPE` variable determines whether or not the model will be initialized from a "cold" or a "warm" start. A cold start initializes the model on day zero whereas a warm start initializes the model from a previous simulation. There are two types of warm starts. In a "continuation" warm start (`RUNTYPE=1`), the model is initialized from where it left off. This setting is useful if you want to re-submit a runscript and have the simulation run out further. In a general warm start (`RUNTYPE=2`), the model initializes from the restart files specified by `restartfile`. This setting allows you to specify not only *which* simulation will initialize the model, but *when* during that simulation to initialize the model.

> **WARNING:** When warm-starting from a specified file (`RUNTYPE=2`), the untarred contents of the designated restart tar file will overwrite the files in `workdir/INPUT/`. Also, if the model dates are the same, subsequent history and restart files in `history/` and `restart/` will be overwritten as well. A guaranteed way to avoid this is to rename the runscript and, therefore, the output directory (`name` from Table 4.1).

4.2.4 Initial Conditions

The parameters in this section define the input files for the dust scenario. These parameters are listed in Table 4.3 and additional information about the dust scenario is provided below.

Table 4.3: Initial Conditions

Variable	Default Value	Description
`RESET_DATE`	`0`	logical: 1 resets the time variable for every output file, 0 does not
`dust_scenarios`	`DustScenario_` `↪Background.nc`	the dust scenario(s) to cycle through
`APPEND_EXTERNAL_` `↪DIAGTABLE`	`0`	logical: 1 reads in external diag table, 0 does not
`diagtable_ext`	`$homedir/diag_table.ext`	the external diag table to use

The `dust_scenarios` parameter accepts one or more dust scenario files. The dust scenario files included in MGCM 3.1 are listed in Chapter 2 and stored in the source code `data/` directory:

```
AM4/src/AmesGCM/data/
```

Broadly, there are two ways to inform the dust scenario:

Option 1: Specify one file to be referenced throughout the simulation.

```
! Option 1 Example:
set dust_scenarios  = (DustScenario_Background.nc)
```

Option 2: Specify a list of files for the model to iterate through throughout the simulation.

```
! Option 2 Example:
set dust_scenarios  = (DustScenario_MY30.nc DustScenario_MY31.nc)
```

More specifically, listing one file after `dust_scenarios` (**Option 1**) references the specified dust scenario for every iteration of the run. **Option 1** is the default setting. Listing multiple files after `dust_scenarios` (**Option 2**) initializes the dust scenario from the first file listed, then the model then cycles through the rest of the files with each run iteration. **Option 2** is especially useful for simulating multi-year dust scenarios. For example, if the model is initialized with the following settings:

```
dayslist       = (668 668)
num_executions = $dayslist
dust_scenarios = (DustScenario_MY30.nc DustScenario_MY31.nc)
```

then the simulation will reference the MY30 dust scenario for the first year of the run and the MY31 dust scenario for the second year of the run.

WARNING: The model will move on to the next dust scenario file in the list even if it has not simulated through a full year. In the example above, if `dayslist = (50 50)` then the first 50 days would reference the MY30 scenario and the next 50 days would reference the MY31 scenario.

4.2.5 Remaining Sections

After the dust scenario is set up, the section "Set Up Directory Structure" cleans the working directory, creates (or modifies, if pre-existing) the subdirectories that will hold the input, restart, output, and ASCII files, and copies the model executable to the working directory. The two sections after that define the in-line diag and field tables. This is also where the external diag table is referenced if `APPEND_EXTERNAL_DIAGTABLE = 1`.

The final modifiable section of the runscript is the namelist. The default value for each namelist variable is provided in `fms_mars_default` along with a description of the variable and its range of values. Some of the most commonly modified namelist settings are discussed in the next section (Section 4.3).

After the namelist, the rest of the runscript initializes the run. Unless the user is running MGCM 3.1 on a non-NAS environment, there are no other parts of the runscript that need modifying by the user.

WARNING: When running MGCM 3.1 on non-NAS environments, the user must take care to ensure all file paths throughout the runscript are updated for compatibility with their environment.

4.3 Customizing the Runscript

The NASA Ames MCMC supports limited customization of MGCM 3.1. This section provides instructions for modifying some of the default settings in the MGCM, including simulating the Martian atmosphere at higher resolution, changing the vertical grid, running with flat topography (a "billiard ball" run), and modifying the dust prescription.

4.3.1 Moving to Higher Resolution

There are two supported horizontal resolutions in MGCM 3.1: c24 and c48. As a reminder, the resolution is set by the `NCX` variable in the runscript. Increasing the resolution typically requires decreasing the value of `nord`, defined in the namelist under `&fv_core_nml`. `nord` sets the order of the del operator for the numerics: applied once (`nord=1`), the del operator is del-2; applied twice (`nord=2`), it is del-4; and applied thrice (`nord=3`), it is del-8. When `nord=3` (default), the runscript automatically changes `nord` if `NCX` >c24. However, this automatic setting can be overwritten if necessary by setting `nord` manually in `&fv_core_nml`. The recommended values for `nord` are listed in Table 4.4.

Table 4.4: Horizontal Grid Options

`NCX`	`nord`	Resolution (degrees)	Resolution (km)
`c24`	`3`	∼4°	∼221.5 x 221.5
`c48`	`2`	∼2°	∼110.8 x 110.8
`c96`	`1`	∼1°	∼55.4 x 55.4
`c192`	`1`	∼0.5°	∼27.7 x 27.7

Generally, increasing model resolution from c24 to c48 is straightforward and requires only the two modifications to the runscript:

1. Change the resolution: set `NCX=48`
2. Increase `k_split` and `n_split`: set `NKS=2` and `NNS=6`

The second modification increases the temporal resolution for stability. We recommend increasing the frequency of the vertical remapping loop (`k_split`) and the advection calculations (`n_split`) before decreasing the physics timestep (`dt_atmos`) or the radiative time step (`rad_calc_intv`). For the example above, we find that the default simulation at 2° horizontal resolution (c48) usually runs if `k_split=2` and `n_split=6`.

The physics time step `dt_atmos` is changed by setting `DTA`, which is nominally set to 924 seconds. This setting should be sufficient for most c24 and c48 simulations. The only constraint on `DTA` is that it must divide evenly into the length of the Mars day as it is defined in the model (88,704 seconds).

4.3.2 Changing Vertical Grids

Changing the vertical resolution of the model is simple and requires setting just one variable: `NPZ`. This variable is listed under "Execution Variables" in the runscript and is nominally set to the 56-layer vertical grid:

```
set NPZ  = 56   # number of vertical layers
```

The 56-layer grid was described in detail in Chapter 2. Equation 1.3 used to derive the vertical grids is also in Chapter 2. To change the vertical grid, simply set `NPZ` to the number of layers in the desired grid (either 56, 37, 30, or 24). The vertical grids included in the MGCM 3.1 are listed in order from highest to lowest resolution below. They are also illustrated in Figures 4.1 and 4.2.

- **56-layer grid (default)**: The uppermost layer midpoint is located at 0.003 Pa.
- **37-layer grid**: Lower resolution than L56, but more layers are defined above 0.1 Pa (see Figure 4.1). The uppermost layer midpoint is located at 0.005 Pa.
- **30-layer grid**: Lower resolution than L56 and L37. Has the lowest model top. The uppermost layer midpoint is located at 0.035 Pa.
- **24-layer grid**: Lowest resolution grid option. This grid is used in the Legacy version of the MGCM. The coordinate system is all sigma, with the top layer boundary located at $\sigma = 0$. The uppermost layer midpoint is located at 0.011 Pa.

All of the vertical grids in MGCM 3.1 have a lowest-layer midpoint between 2–5 m. Note that it is possible to change the vertical resolution of a simulation during a warm start. To do so, set `NPZ` to the desired vertical grid and `NPZ_RST` to the number of layers in the restart file (i.e. the number of layers in the previous simulation).

WARNING: Be sure to reset `NPZ_RST=0` for future runs.

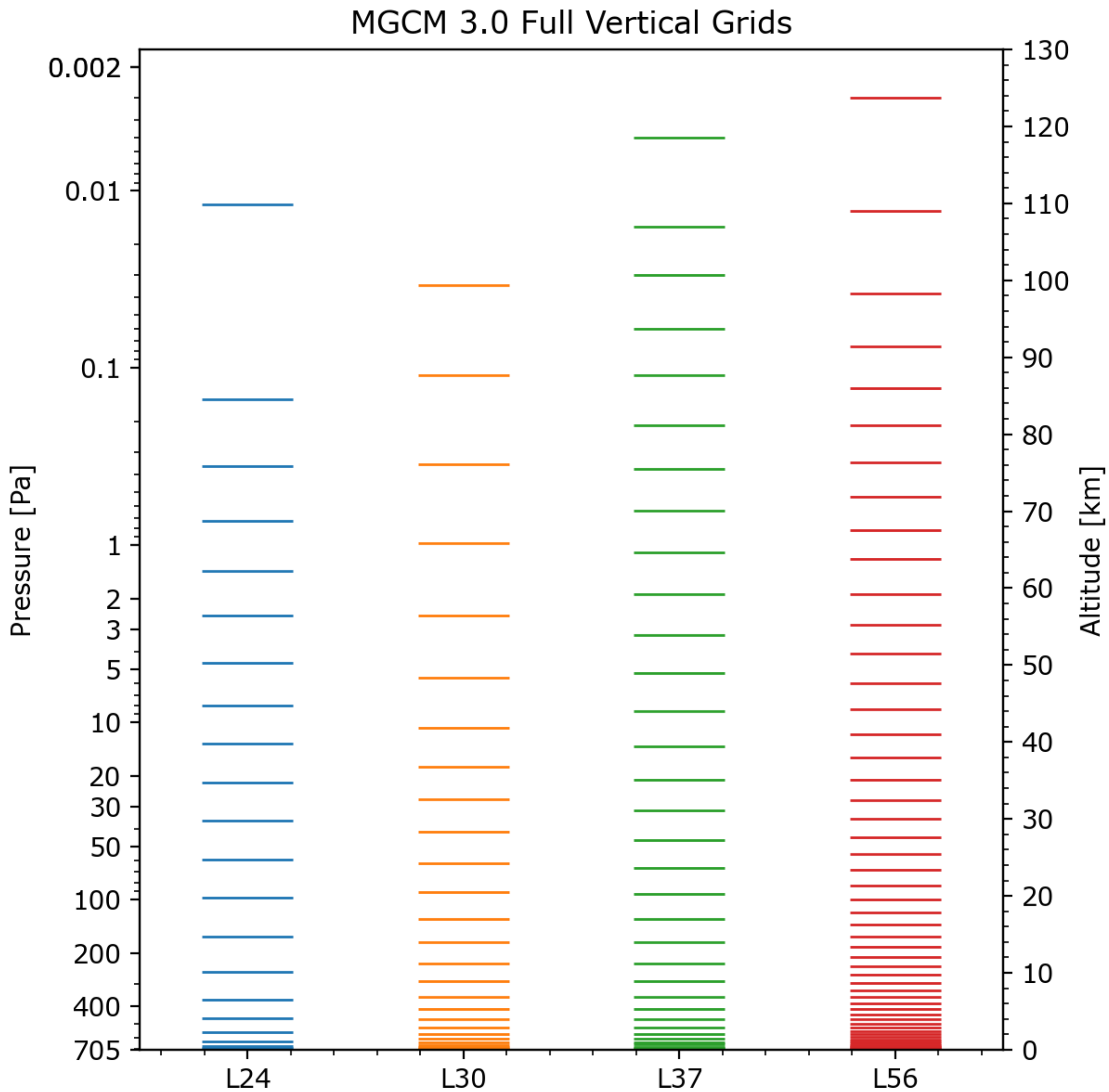

Figure 4.1: Midpoints of the vertical layers in the four vertical grids included in MGCM 3.1. From left to right, the uppermost layer midpoint is at 0.011 Pa (L24), 0.035 Pa (L30), 0.005 Pa (L37), and 0.003 Pa (L56).

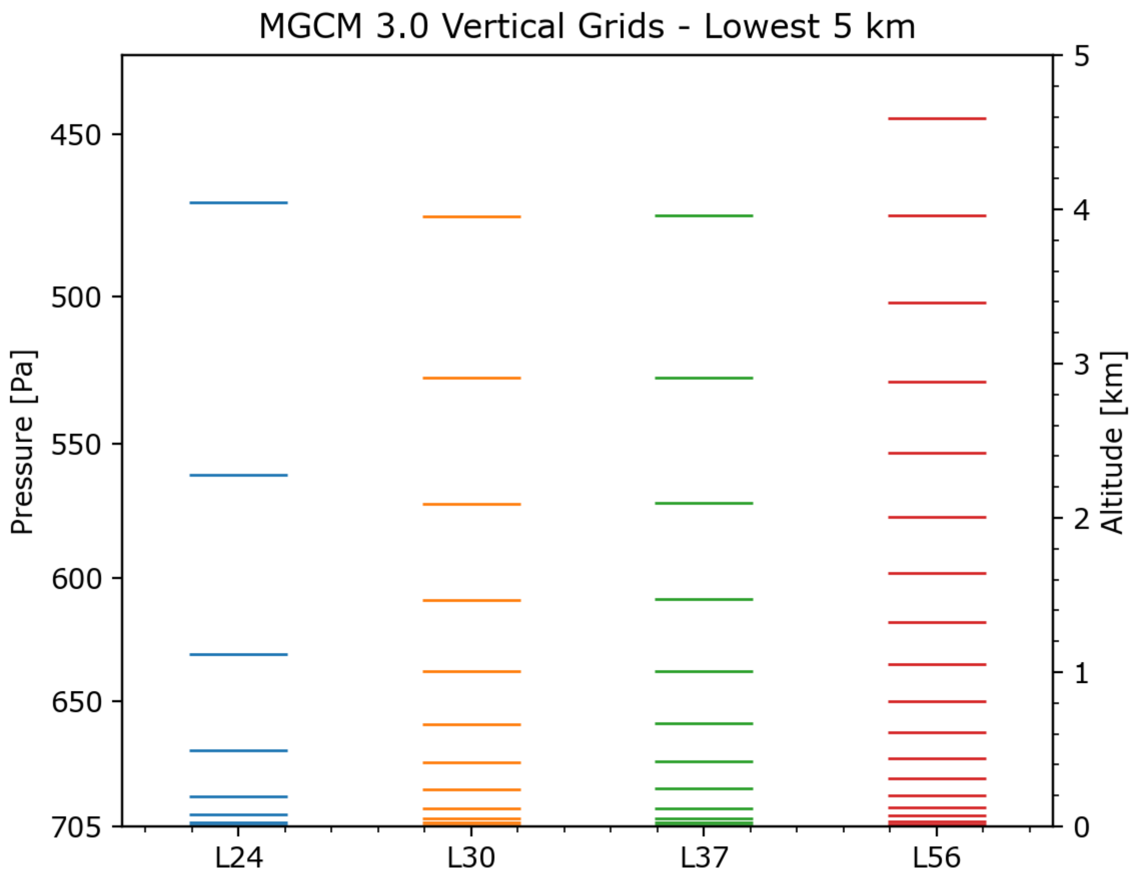

Figure 4.2: Midpoints of the vertical layers below 5 km in the four vertical grids included in MGCM 3.1.

4.3.3 Topographical Options

By default, MGCM 3.1 is initialized with a topographic map from MOLA with 1/16° (∼3.75 km) resolution. The model can be run with a smooth surface (i.e. a "billiard ball" run) simply by changing `test_case` from "11" to "10" in the `&test_case_nml` namelist in the runscript:

```
&test_case_nml
    test_case    = 11   ! Topography type. 11= Mars; 10= flat
```

> **WARNING:** Only two topographical types are supported in MGCM 3.1: `test_case=10` and `test_case=11`.

4.3.4 Configuring Dust Options

There are a number of customization options for the dust in MGCM 3.1 in addition to the dust scenarios introduced in Chapter 4 (Section 4.2.4). This section lists some configurations for both prescribed and transported dust that are commonly used in the model.

In the MGCM, "prescribed" dust means that the radiative transfer (RT) code is informed by dust fields that are defined in the horizontal and vertical by maps or analytic expressions (not by the dust tracer fields). Alternatively, users can choose to make the dust that is transported in the tracer fields radiatively active. In this "transported" dust case, the RT code is informed by the evolving dust tracers.

MGCM 3.1 defaults to a prescribed dust scenario in which the radiative transfer code reads in a dust scenario file that informs the column dust opacity and distributes the dust vertically according to a modified Conrath profile. In this default case, radiatively inert dust is lifted and transported. To toggle between the prescribed and transported dust scenarios, set the `radactive_dust_TOG` flag under `&ames_rtmod_nml`. Specifically, `radactive_dust_TOG = 3` is the default setting (the prescribed dust scenario). For transported dust, set `radactive_dust_TOG = 1`.

> **NOTE:** If a transported dust scenario is **not** radiatively active, the dust can still be lifted and transported inertly (i.e., radiatively passive dust).

In this section, we provide example dust scenario set-ups for both prescribed and transported dust. We also provide instructions for running MGCM 3.1 with interactive dust lifting, or dust lifted by surface winds and dust devils.

Prescribed Dust Options

The default dust setup in MGCM 3.1 is shown below. It is a prescribed dust scenario in which the RT code references the background dust scenario file (informing the column dust opacity) and the vertical distribution is set by a modified Conrath vertical profile:

```
!                          Prescribed Dust: Default Setup
!*****************************************
!            Aerosols Namelists
!*****************************************
&aerosol_util_nml
/
&aerosol_nml
    do_inpt_dust_cycle = T, ! Read the dust scenario from an input file.
                            ! Skipped if optical_depth_inpt <= 0.
    conrath_type  = 1,      ! Conrath vertical extent.
/
&dust_update_nml
    background = T          ! Tracer field references background dust
                            ! scenario file.
/
!*****************************************
!       Radiative Transfer Namelists
!*****************************************
&ames_rtmod_nml
    radactive_dust_TOG = 3, ! Prescribed dust informs RT. Transported dust
                            ! is inert.
/
```

To increase the effective radius of the dust, add the variable `reff_fixed` to the namelist `&aerosol_util_nml` and set its value as in the example below. For example, setting a 2.5 μm effective radius would require:

```
!                Prescribed Dust with Modified Effective Radius
!*****************************************
!             Aerosols Namelists
!*****************************************
&aerosol_util_nml
    reff_fixed    = 2.5,     ! Effective radius of the dust for RT code.
/
&aerosol_nml
    do_inpt_dust_cycle = T, ! Read the dust scenario from an input file.
                            ! Skipped if optical_depth_inpt <= 0.
    conrath_type   = 1,      ! Conrath vertical extent.
/
&dust_update_nml
    background = T           ! Tracer field references background dust
                             ! scenario file.
/
!*****************************************
!        Radiative Transfer Namelists
!*****************************************
&ames_rtmod_nml
    radactive_dust_TOG = 3, ! Prescribed dust informs RT. Transported dust
                            ! is inert.
/
```

To use a constant Conrath vertical profile instead of a modified Conrath parameter (as in the default setup, set `conrath_type=0`, add the Conrath parameter (`conrath`) to the aerosol namelist (`&aerosol_nml`), and set it to the desired value (0.003 is the default):

```
!           Prescribed Dust with a Constant Conrath Vertical Profile
!*****************************************
!             Aerosols Namelists
!*****************************************
&aerosol_util_nml
/
&aerosol_nml
    do_inpt_dust_cycle = T, ! Read the dust scenario from an input file.
                            ! Skipped if optical_depth_inpt <= 0.
    conrath_type   = 0,      ! Conrath vertical extent.
    conrath = 0.003,         ! Value of the Conrath parameter.
/
&dust_update_nml
    background = T           ! Tracer field references background dust
                             ! scenario file.
/
!*****************************************
!        Radiative Transfer Namelists
!*****************************************
&ames_rtmod_nml
```

```
    radactive_dust_TOG = 3, ! Prescribed dust informs RT. Transported dust
                            ! is inert.
/
```

To use a constant tau for the dust, toggle off `do_inpt_dust_cycle`, which informs the model to skip reading in the dust scenario file, add `optical_depth_inpt` to the aerosol namelist (`&aerosol_nml`) and set it to the desired dust opacity (5 in this example). The vertical profile of the dust is set by `conrath_type`, which must be zero for a constant Conrath profile (of the user's choosing) when using a constant dust tau.

```
!                    Prescribed Dust with a Constant Tau
!*****************************************
!            Aerosols Namelists
!*****************************************
&aerosol_util_nml
/
&aerosol_nml
    do_inpt_dust_cycle = F, ! Read the dust scenario from an input file.
                            ! Skipped if optical_depth_inpt <= 0.
    conrath_type  = 0,      ! Conrath vertical extent.
    conrath = 0.003,        ! Value of the Conrath parameter.
    optical_depth_inpt = 5  ! tau for the dust.
/
&dust_update_nml
    background = T          ! Tracer field references background dust
                            ! scenario file.
/
!*****************************************
!       Radiative Transfer Namelists
!*****************************************
&ames_rtmod_nml
    radactive_dust_TOG = 3, ! Prescribed dust informs RT. Transported dust
                            ! is inert.
/
```

WARNING: When reverting from a constant tau dust, be sure to reset `do_inpt_dust_cycle` to "True" and `optical_depth_inpt` to zero or delete the line altogether. If you reset `do_inpt_dust_cycle` but leave `optical_depth_inpt` equal to a nonzero number, the MGCM will read in the dust scenario file **AND** add a constant tau on top of it.

Transported Dust Options

The simplest transported dust setup is the default setup described in Section 4.3.4. In that case, the transported dust is radiatively inert and its horizontal distribution is informed by the same dust scenario file that informs the RT code. The dust scenario file instructs the model to "lift" or inject dust into the lowest model layer in an attempt to match the input dust map. Once airborne, the dust is transported by model winds or pulled out of suspension by gravitational sedimentation.

To make the transported dust radiatively active, set `radactive_dust_TOG = 1`:

```
!                      Radiatively Active Transported Dust
!*****************************************
!             Aerosols Namelists
!*****************************************
&aerosol_util_nml
/
&aerosol_nml
    ! Namelist settings ignored when radactive_dust_TOG = 1.
    ! RT informed by settings in &dust_update_nml instead.
/
&dust_update_nml
    background = T            ! Tracer field AND RT reference background
                              ! dust scenario file.
/
!*****************************************
!       Radiative Transfer Namelists
!*****************************************
&ames_rtmod_nml
    radactive_dust_TOG = 1,   ! Transported dust informs RT and is
                              ! radiatively active.
/
```

In the same manner, transported dust can be "lifted" to match a constant background optical depth. As is the case for dust lifted to match a background dust map, dust is injected into the lowest model level, advected by model winds, and removed via gravitational sedimentation. This configuration is enabled by setting the flag `opac_from_aerosol` in `dust_update_nml` to true and is only compatible with the constant tau scenario for the prescribed dust option. The user must also toggle off `do_inpt_dust_cycle` and set a desired background column optical depth using `optical_depth_inpt`. For radiatively active transported dust, set `radactive_dust_TOG = 1`.

```
!                      Transported Dust with a Constant Tau
!*****************************************
!             Aerosols Namelists
!*****************************************
&aerosol_util_nml
/
&aerosol_nml
   do_inpt_dust_cycle = F, ! Read the dust scenario from an input file.
                           ! Skipped if optical_depth_inpt <= 0.
   conrath_type   = 0,     ! Conrath vertical extent.
   conrath = 0.003,        ! Value of the Conrath parameter.
   optical_depth_inpt = 5  ! tau for the dust.
/
&dust_update_nml
   background = T          ! Tracer field AND RT reference background
                           ! dust scenario file.
   opac_from_aerosol = T   ! Match transported dust to the constant
                           ! column opacity.
```

```
/
!****************************************
!       Radiative Transfer Namelists
!****************************************
&ames_rtmod_nml
    radactive_dust_TOG = 1, ! Transported dust informs RT and is
                            ! radiatively active.
/
```

Instead of informing the dust scenario with an input map, the MGCM can lift dust interactively via wind stress and dust devil lifting. To use interactive dust lifting (with both wind stress and dust devil dust lifting), disable the flag for reading in the background dust scenario (`background`) and enable the flag for interactive lifting (`interact`), both in `dust_update_nml`:

```
!                Transported Dust with Interactive Dust Lifting
!****************************************
!            Aerosols Namelists
!****************************************
&aerosol_util_nml
/
&aerosol_nml
    ! Namelist settings ignored when radactive_dust_TOG = 1.
    ! RT informed by settings in &dust_update_nml instead.
/
&dust_update_nml
    background = F          ! Ignore background dust scenario file.
    interact = T            ! Lift dust interactively. Informs tracer
                            ! field AND RT.
/
!****************************************
!       Radiative Transfer Namelists
!****************************************
&ames_rtmod_nml
    radactive_dust_TOG = 1, ! Transported dust informs RT and is
                            ! radiatively active.
/
```

The size distribution of the transported dust is defined by an effective radius of 2.0 μm by default. Changing the size distribution is relatively straightforward. As in the second example from the prescribed dust cases, the namelist parameter to adjust is under `&aerosol_util_nml` and begins with `reff_`. However, users must take care to modify the namelist parameter corresponding to the type of lifting being done. The following three examples show how to set the effective radius of the transported dust to 2.5 μm for all three types of lifting: (1) the background dust scenario, (2) an interactive wind-stress lifting scenario, and (3) an interactive dust devil lifting scenario.

For a transported dust case where the lifting scheme is informed by the background dust scenario or other input dust map, set `reff_backgd`. For an effective radius of 2.5 μm:

```
!                   Transported Dust, Modified Effective Radius
!                            Map-Tracking Dust Lifting
!*****************************************
!             Aerosols Namelists
!*****************************************
&aerosol_util_nml
    reff_backgd = 2.5e-6     ! For transported dust lifted to map.
/
&aerosol_nml
    ! Namelist settings ignored when radactive_dust_TOG = 1.
    ! RT informed by settings in &dust_update_nml instead.
/
&dust_update_nml
    background = T           ! Tracer field AND RT reference background
                             ! dust scenario file.
/
!*****************************************
!       Radiative Transfer Namelists
!*****************************************
&ames_rtmod_nml
    radactive_dust_TOG = 1,  ! Transported dust informs RT and is
                             ! radiatively active.
/
```

For a transported dust case with wind stress lifting, set `reff_ws`. For an effective radius of 2.5 μm:

```
!                   Transported Dust, Modified Effective Radius
!                                Wind Stress Lifting
!*****************************************
!             Aerosols Namelists
!*****************************************
&aerosol_util_nml
    reff_ws = 2.5e-6         ! For transported dust lifted by wind stress.
/
&aerosol_nml
    ! Namelist settings ignored when radactive_dust_TOG = 1.
    ! RT informed by settings in &dust_update_nml instead.
/
&dust_update_nml
    background = F           ! Ignore background dust scenario file.
    interact = T             ! Lift dust interactively. Informs
                             ! tracer field AND RT.
/
!*****************************************
!       Radiative Transfer Namelists
!*****************************************
&ames_rtmod_nml
    radactive_dust_TOG = 1,  ! Transported dust informs RT and is
                             ! radiatively active.
/
```

For a transported dust case with dust devil lifting, set `reff_dd`. For an effective radius of 2.5 μm:

```
!                   Transported Dust, Modified Effective Radius
!                                Dust Devil Lifting
!*****************************************
!             Aerosols Namelists
!*****************************************
&aerosol_util_nml
    reff_dd = 2.5e-6        ! For transported dust lifted by dust devils.
/
&aerosol_nml
    ! Namelist settings ignored when radactive_dust_TOG = 1.
    ! RT informed by settings in &dust_update_nml instead.
/
&dust_update_nml
    background = F          ! Ignore background dust scenario file.
    interact = T            ! Lift dust interactively. Informs
                            ! tracer field AND RT.
/
!*****************************************
!        Radiative Transfer Namelists
!*****************************************
&ames_rtmod_nml
    radactive_dust_TOG = 1, ! Transported dust informs RT and is
                            ! radiatively active.
/
```

When the transported dust is informed by the background dust scenario or other input dust map, the model injects dust into the atmosphere to match the input dust map. The default setting allows airborne dust to fall out of suspension by gravitational sedimentation. This naturally results in simulated dust scenarios with varying degrees of similarity to the input dust map. To forcibly remove dust from the atmosphere (in addition to gravitational sedimentation), users can toggle on the `sink_bd` flag under `&dust_update_nml`:

```
!           Forcibly Removing Transported Dust from the Atmosphere
!*****************************************
!             Aerosols Namelists
!*****************************************
&aerosol_util_nml
/
&aerosol_nml
    ! Namelist settings ignored when radactive_dust_TOG = 1.
    ! RT informed by settings in &dust_update_nml instead.
/
&dust_update_nml
    background = T          ! Tracer field AND RT reference background
                            ! dust scenario file.
    sink_bd = T             ! Forcibly remove excess dust from the
                            ! atmosphere to match the map.
/
```

```
!*****************************************
!       Radiative Transfer Namelists
!*****************************************
&ames_rtmod_nml
    radactive_dust_TOG = 1, ! Transported dust informs RT and is
                            ! radiatively active.
/
```

Interactive Dust Lifting

The interactive dust lifting schemes described thus far are tuned for the default horizontal and vertical resolutions of the model (C24, L56). Interactive lifting schemes are sensitive to resolution, both vertical and horizontal, as well as lifted particle sizes. Altering any of these settings will likely require adjusting the threshold wind stress (`threshold_stress`) as well as the wind stress and dust devil tuning parameters (`alfa` and `alfa_dda`, respectively) under the `&dust_update_nml` namelist:

```
!                  Tuning the Interactive Dust Lifting Scheme
!*****************************************
!            Aerosols Namelists
!*****************************************
&aerosol_util_nml
/
&aerosol_nml
    ! Namelist settings ignored when radactive_dust_TOG = 1.
    ! RT informed by settings in &dust_update_nml instead.
/
&dust_update_nml
    background = F           ! Ignore background dust scenario file.
    interact = T             ! Lift dust interactively. Informs tracer
                             ! field AND RT.
    alfa = 0.0065            ! Wind stress tuning parameter.
    threshold_stress = 0.022 ! Threshold wind stress for lifting [N/m2].
    alfa_dda = 1.4e-10       ! Dust devil tuning parameter.
/
!*****************************************
!       Radiative Transfer Namelists
!*****************************************
&ames_rtmod_nml
    radactive_dust_TOG = 1, ! Transported dust informs RT and is
                            ! radiatively active.
/
```

4.3.5 Orographic Gravity Wave Parameterization

The orographic gravity wave subgrid-scale parameterization is disabled by default. The default runscript appears as follows:

```
!                     Default Orographic Gravity Wave Setup
!*****************************************
!             Physics Namelists
!*****************************************
&mars_physics_nml
    GW_drag_TOG = 0.,   ! Orographic gravity wave parameterization OFF.
/
```

Using the parameterization in a simulation is as simple as setting the flag to 1:

```
!             Enabling the Orographic Gravity Wave Paramaterization
!*****************************************
!             Physics Namelists
!*****************************************
&mars_physics_nml
    GW_drag_TOG = 1.,   ! Orographic gravity wave parameterization ON.
/
```

This will enable the orographic gravity-wave subgrid-scale parameterization with its default settings. Further tuning of the scheme can be achieved by modifying the `KAP` parameter under the `&palmer_drag_nml` namelist as follows:

```
!              Tuning the Orographic Gravity Wave Paramaterization
!*****************************************
!             Physics Namelists
!*****************************************
&mars_physics_nml
    GW_drag_TOG = 1.,   ! Orographic gravity wave parameterization ON.
/
&palmer_drag_nml
    KAP=5e-6,           ! Coupling parameter in meters.
/
```

The orographic gravity wave scheme will automatically detect the horizontal resolution of the model, however, the scheme is only compatible with three resolutions: C24, C48, and C96.

4.3.6 Early Mars Setup

MGCM 3.1 supports running with surface pressures of up to 10 bar as that is the upper limit of the correlated-k table in the model. The default early Mars case has a 500 mb surface pressure, a 95% CO_2 atmosphere with no hydrogen, and CO_2-CO_2 collision-induced absorption (CIA):

```
!                            Default Early Mars Setup
!*****************************************
!             Execution Variables
!*****************************************
set PREF = 5.0e4,            ! Global mean surface pressure (Pa).
```

```
!****************************************
!             Physics Namelists
!****************************************
&mars_physics_nml
    sponge_pbottom = 125.,   ! Rayleigh drag for PREF = 5.0e4 (500 mb).
    rflevel = 13.,           ! Rayleigh drag for PREF = 5.0e4 (500 mb).
/
!****************************************
!        Radiative Transfer Namelists
!****************************************
&ames_rtmod_nml
    do_cia = T,              ! Do collision-induced absorption.
    cia_co2 = 0.95,          ! molar concentration of CO2 in atmosphere.
    cia_h2 = 0.0,            ! molar concentration of H2 in atmosphere.
/
```

The Rayleigh drag namelist parameters `sponge_pbottom` and `rflevel` default to values compatible with a 500 mb atmosphere. The `sponge_pbottom` variable defines the pressure (Pa) above which Rayleigh damping is applied, and the `rflevel` variable defines the inflection point in the damping function calculation. We recommend applying Rayleigh damping to the top few atmospheric layers when running an early Mars case. Because both of these values are given in Pascal instead of altitude or layer number, altering the default surface pressure will require adjusting the Rayleigh Drag parameters accordingly. Recommended values for `sponge_pbottom` and `rflevel` for four early Mars-like cases are listed in Table 4.5 below.

Table 4.5: Parameters for Rayleigh drag for early Mars surface pressure scenarios.

Surface Pressure	`sponge_pbottom`	`rflevel`
100 mb	20	2
500 mb	125	13
1 bar	250	25
2 bar	500	50

For example, to run with a 2 bar atmosphere, set `PRES` to 2.0×10^5 Pa, `sponge_pbottom` to 500 Pa and `rflevel` to 50 Pa:

```
! ----------------- Early Mars Setup for a 2-bar Atmosphere -----------------
!****************************************
!             Execution Variables
!****************************************
set PREF = 2.0e5,           ! Global mean surface pressure (Pa).

!****************************************
!             Physics Namelists
!****************************************
```

```
&mars_physics_nml
    sponge_pbottom = 500.,   ! Rayleigh drag for PREF = 2.0e5 (2 bar).
    rflevel = 50.,           ! Rayleigh drag for PREF = 2.0e5 (2 bar).
/
!*****************************************
!       Radiative Transfer Namelists
!*****************************************
&ames_rtmod_nml
    do_cia = T,              ! Do collision-induced absorption.
    cia_co2 = 0.95,          ! molar concentration of CO2 in atmosphere.
    cia_h2 = 0.0,            ! molar concentration of H2 in atmosphere.
/
```

Users may need to decrease the timestep (`DTA`) in order to run an early Mars case because of the more massive atmosphere and the addition of CO_2 condensation. Our recommendation is to decrease the timestep to 77 seconds when running with surface pressures (`PRES`) greater than or equal to 1 bar.

The early Mars configuration also supports collision-induced absorption (CIA) warming from a mixed CO_2–H_2 atmosphere. To run with 10% H_2 and 90 % CO_2, for example, set the `cia_co2` and `cia_h2` variables to the molar concentrations of CO_2 and H_2, respectively:

```
!                Early Mars Setup with Collision-Induced Absorption
!*****************************************
!            Execution Variables
!*****************************************
set PREF = 5.0e4,            ! Global mean surface pressure (Pa).

!*****************************************
!            Physics Namelists
!*****************************************
&mars_physics_nml
    sponge_pbottom = 125.,   ! Rayleigh drag for PREF = 5.0e4 (500 mb).
    rflevel = 13.,           ! Rayleigh drag for PREF = 5.0e4 (500 mb).
/
!*****************************************
!       Radiative Transfer Namelists
!*****************************************
&ames_rtmod_nml
    do_cia = T,              ! Do collision-induced absorption.
    cia_co2 = 0.9,           ! molar concentration of CO2 in atmosphere.
    cia_h2 = 0.1,            ! molar concentration of H2 in atmosphere.
/
```

The sum of the molar concentrations of CO_2 and H_2 must be less than or equal to 1.0. The default values are 0% H_2 and 95% CO_2 as Mars' atmosphere is currently 95% CO_2 and 5% trace gases. I think this is a good place to add a note about what the atmospheric constituents are if the sum is <1

Chapter 5

Model Output and Post-Processing

The MGCM outputs netCDF files that are stored in a compressed (`.tar`) format. NetCDF is an interface for storing and accessing geophysical data and it is coupled with a library that enables interaction with the interface. The netCDF library defines a machine-independent format for representing scientific data so that, together, the interface, library, and format support the creation, access, and sharing of scientific data. NetCDF was developed at the Unidata Program Center in Boulder, Colorado. Free resources for netCDF are available on the Unidata website at http://www.unidata.ucar.edu/software/netCDF.

> **NOTE:** netCDF files are self-descriptive. Users can look at the stored data directly from the command line using `ncdump`, which is a netCDF function that writes an ASCII representation of file contents to the screen. Documentation for `ncdump` is available at http://www.bic.mni.mcgill.ca/users/sean/Docs/netCDF/guide.txn_79.html. The MCMC-developed CAP also has a function that can be used to inspect netCDF files (see Chapter 8).

On NAS, the MGCM outputs data to both Pleiades and the Lou mass storage system (for long-term data storage). The main output directory is on Pleiades at:

```
/nobackup/$USER/path/to/FMS_MARS_runs/am4_mars_runs/$runscript_name/
```

This directory includes output files ("history" files), restart files, log files (useful for diagnosing model issues), and input files. The runscript is also copied here for archiving, along with the `diag_table`, `field_table`, and model namelist (`input.nml`). An additional backup of these files is copied over to Lou and stored in the `history`, `restart`, and `ascii` directories under:

```
FV3/xanadu/am4_mars_runs/$runscript_name/
```

The `history/` directory contains files output at the end of each run iteration (recall that the run iteration is specified by `DAYSLIST` in the runscript). The tarred file names begin with a 5-digit date indicating the first sol number of the data in the file. For example, the default run defines `dayslist = (668 688)`, so output is separated into two files: the first begins on sol 0 and ends on sol 668 (`00000.nc.tar`), and the other begins on sol 668 and ends on sol 01334 (`00668.nc.tar`).

MGCM 3.1 data is archived on the native grid and stored in the file corresponding to the tile the data came from. The location of the tiles comprising the cubed-sphere grid is shown in Figure 5.1. Tile 3 is centered over the north pole, Tile 6 over the south pole, and Tiles 1, 2, 4, and 5 wrap around the globe centered at the equator. For the default diag table settings, the model will output 30 files each run iteration: six output files (one for every tile) for each of the five netCDF output file types defined in the diag table (`atmos_average` , `atmos_daily` , `atmos_diurn` , `atmos_fixed` , `grid_spec`). The model will also output restart files as described in Chapter 2.

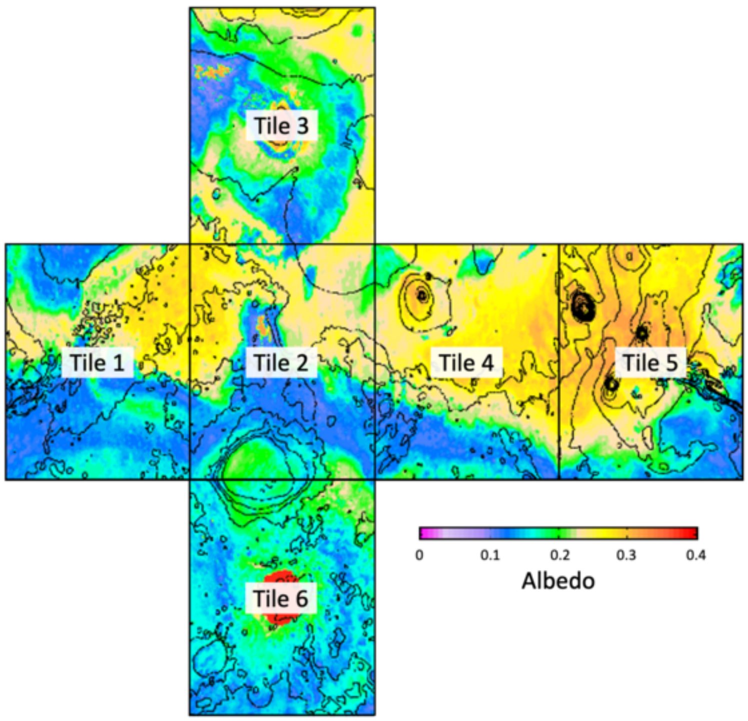

Figure 5.1: The tiles comprising the cubed-sphere grid in MGCM 3.1. Surface albedo is color-filled.

The 30 output files that are created each run iteration can be accessed by untarring one of the `XXXXX.nc.tar` files in the `history/` directory. Untarring a file reveals the following:

```
atmos_average.tile1.nc  atmos_daily.tile5.nc  fixed.tile3.nc
atmos_average.tile2.nc  atmos_daily.tile6.nc  fixed.tile4.nc
atmos_average.tile3.nc  atmos_diurn.tile1.nc  fixed.tile5.nc
atmos_average.tile4.nc  atmos_diurn.tile2.nc  fixed.tile6.nc
atmos_average.tile5.nc  atmos_diurn.tile3.nc  grid_spec.tile1.nc
atmos_average.tile6.nc  atmos_diurn.tile4.nc  grid_spec.tile2.nc
atmos_daily.tile1.nc    atmos_diurn.tile5.nc  grid_spec.tile3.nc
atmos_daily.tile2.nc    atmos_diurn.tile6.nc  grid_spec.tile4.nc
atmos_daily.tile3.nc    fixed.tile1.nc        grid_spec.tile5.nc
atmos_daily.tile4.nc    fixed.tile2.nc        grid_spec.tile6.nc
```

The variables in each of these files were defined in the diag table (Section 2.7) and are listed in Tables 5.1–5.5 below. The `grid_spec` file contains information about the latitude and longitude values for each boundary on a tile (Table 5.1). This file is used to re-grid the data to traditional latitude-longitude coordinates as described in the next section.

Table 5.1: Default output variables in `grid_spec` files

Variable	Units	Description
`grid_lon`	degrees E	(2D) the boundary longitude of each grid cell on the tile
`grid_lat`	degrees N	(2D) the boundary latitude of each grid cell on the tile

5.1 Installing the Post-Processing Routines

Untarring and re-gridding data in the average, daily, and diurn files to global latitude-longitude coordinates can be done in one step using a modified version of NOAA-GFDL's `FRE-NCtools` toolkit. After the data is re-gridded, the files can be interpolated to a standard pressure vertical coordinate also using an MCMC-modified version of `FRE-NCtools`. In this section are instructions for downloading `FRE-NCtools` and applying MCMC-developed patches to the toolkit to modify the re-gridding and pressure-interpolating functions for compatibility with MGCM 3.1 output.

NOTE: These instructions include snippets of code specific to the NAS System. However, these NAS-specific options usually refer to path names and can be modified for use on any system.

Begin by cloning `FRE-NCtools` from NOAA-GFDL's github to your preferred directory on Lou:

```
user@lfe:~$ git clone --recursive https://github.com/NOAA-GFDL/FRE-NCtools
```

Next, copy `srcFRE.patch` from the MGCM 3.1 source code on Pleiades to your home directory on Lou:

```
user@lfe:~$ scp USER@pfe:AM4/src/AmesGCM/patches/srcFRE.patch .
```

NOTE: Change `USER` to your NAS username and modify the path to `AM4` as necessary. Refer to the NAS High-End Computing Capability website for help with moving files between Pleiades and Lou (https://www.nas.nasa.gov/hecc/support/kb/).

Next, load the following modules on Lou. These enable interfacing with GitHub and installing the NOAA-GFDL post-processing distribution, `FRE-NCtools`.

```
user@lfe:~$ module purge
user@pfe:~$ module load pkgsrc/20XXQX # use available version
user@lfe:~$ module load comp-intel/2020.4.304 # or newer
user@lfe:~$ module load mpi-hpe/mpt.2.26
user@lfe:~$ module load hdf4/4.2.12
user@lfe:~$ module load hdf5/1.8.18_mpt
user@lfe:~$ module load netcdf/4.4.1.1_mpt
```

> **NOTE:** If you are installing `FRE-NCtools` on Pleiades, you will also need to load the GCC module with `module load gcc`.

Confirm that the proper modules are loaded by typing `module list` on the command line. This should show:

```
Currently Loaded Modulefiles:
  1) pkgsrc/20XXQ                5) szip/2.1.1
  2) comp-intel/2020.4.304       6) hdf5/1.8.18_mpt
  3) mpi-hpe/mpt.2.26            7) netcdf/4.4.1.1_mpt
  4) hdf4/4.2.12
```

Set the paths to netCDF, HDF5, and the config file using the appropriate syntax for your shell. The config file will be copied over in the `nas/` subdirectory in the next step.

For `csh` or `tcsh` users:

```
user@lfe:~$ setenv netCDF_HOME /nasa/netCDF/4.4.1.1_mpt
user@lfe:~$ setenv HDF5_HOME /nasa/hdf5/1.8.18_mpt
user@lfe:~$ setenv CONFIG_SITE
↪  /u/$USER/FRE-NCtools/site-configs/nas/config.site
```

For `bash` users:

```
user@lfe:~$ export netCDF_HOME=/nasa/netCDF/4.4.1.1_mpt
user@lfe:~$ export HDF5_HOME=/nasa/hdf5/1.8.18_mpt
user@lfe:~$ export CONFIG_SITE=
↪  /u/$USER/FRE-NCtools/site-configs/nas/config.site
```

> **NOTE:** Change `USER` to your NAS username and modify the path to `FRE-NCtools` as necessary.

Copy the `nas/` subdirectory containing the config file from the MGCM source code on Pleiades to `site-configs/`:

```
user@lfe:~$ scp -r USER@pfe:AM4/src/AmesGCM/diagnostics/nas/config.site
↪  FRE-NCtools/site-configs/.
```

WARNING: If you did not install `FRE-NCtools` in your home directory on Lou, then you have to change the path for `PREFIX` in the config file (`nas/config.site`) so that it points to your installation of `FRE-NCtools`.

Now go into `FRE-NCtools/` and apply the patch:

```
user@lfe:~$ cd FRE-NCtools
user@lfe:~$ git checkout -b mars_branch && git apply --reject --whitespace
↪  =fix ../srcFRE.patch
```

Reconfigure the toolkit:

```
user@lfe:~$ autoreconf -i
```

Now, build the toolkit. Create a `build/` directory in `FRE-NCtools`, and make the installation with `make`:

```
user@lfe:~$ mkdir build && cd build
user@lfe:~$ ../configure
user@lfe:~$ make && make install
```

Finally, copy the re-gridding and pressure-interpolation wrappers from the MGCM 3.1 source code on Pleiades to `FRE-NCtools/bin/`:

```
user@lfe:~$ scp USER@pfe:AM4/src/AmesGCM/diagnostics/runpinterp.csh
↪  /u/USER/FRE-NCtools/bin/.
user@lfe:~$ scp USER@pfe:AM4/src/AmesGCM/diagnostics/cinterp_script.csh
↪  /u/USER/FRE-NCtools/bin/.
```

WARNING: If you did not install `FRE-NCtools` in your home directory on Lou, then you have to modify three paths in the files you just copied over to reflect the directory `FRE-NCtools` is in:

- Modify the path to `pinterp.sh` in `runpinterp.csh`
- Modify the environment path (`setenv PATH`) in `runpinterp.csh`
- Modify `regridpath` path in `cinterp_script.csh`

Congratulations! You have installed the re-gridding and pressure-interpolation routines and modified them as appropriate for working with MGCM 3.1 output. The following two sections provide step-by-step instructions for regrinding and pressure-interpolating MGCM data.

5.2 Re-gridding Tiled Data

Begin by going into the `history/` directory for a simulation. For the default simulation (on NAS), the path is:

```
user@lfe:~$ cd FV3/xanadu/am4_mars_runs/fms_mars_default/history/
```

Note the resolution (`NCX`) of the data and the date code in the first part of the tarred file name. Pass these parameters to `cinterp_script.csh` as shown below. For the default simulation, `NCX=24` and the output files are `00000.nc.tar` and `00668.nc.tar`. To re-gridding output from the second year of the simulation, stay in the `history/` directory and do:

```
user@lfe:~$ ~/FRE-NCtools/bin/cinterp_script.csh -n c24 -d 00668
```

> **NOTE:** Modify the path to `FRE-NCtools` as needed.

When `cinterp_script.csh` is finished, it will have re-gridded the average, daily, and diurn files from each of the tiles onto a latitude-longitude grid. The re-gridded files are:

- `00668.fixed.nc`
- `00668.atmos_daily.nc`
- `00668.atmos_average.nc`
- `00668.atmos_diurn.nc`

Recall that the variables defined in the diag table determine the contents of these files. The minimum variables output into each of the files are listed in Tables 5.2–5.5 below for your reference.

> **NOTE:** Before you pressure-interpolate the data to a standard pressure grid as described here, the output files will reflect that the vertical coordinate is `pfull`. `pfull` is a time-invariant **reference** pressure grid that has the same number of layers as the model vertical grid. It is **not** the actual pressure at any specific location. It is a reference pressure grid based on a zero elevation pressure level of 705 Pa. The actual pressure at a given layer midpoint will vary depending on the local surface pressure and the thermal structure of the atmosphere.

Table 5.2: Default output variables in `fixed` files

Variable	Units	Description
`bk`	none	the vertical coordinate sigma value
`ak`	Pa	the pressure part of hybrid coordinate
`lon`	degrees East	longitude
`phalf`	mb	the pressure of the layer interface
`lat`	degrees North	latitude
`grid_yt_bnds`	degrees North	the cell boundary latitude
`grid_xt_bnds`	degrees East	the cell boundary longitude
`zsurf`	m	surface height
`thin`	mks	surface thermal inertia
`alb`	none	surface albedo
`emis`	none	surface emissivity
`gice`	none	GRS ice

Table 5.3: Default output variables in `atmos_average` files

Variable	Units	Description
`bk`	none	the vertical coordinate sigma value
`ak`	Pa	the pressure part of hybrid coordinate
`pfull`	mb	the reference pressure of the layer midpoint
`time`	days	number of sols since the start of the run
`average_T1`	days	start time for averaging period
`average_T2`	days	end time for averaging period
`average_DT`	days	length of averaging period
`lon`	degrees East	longitude
`phalf`	mb	the pressure of the layer interface
`scalar_axis`	none	the aggregating dimension (for netCDF)
`lat`	degrees North	latitude
`grid_yt_bnds`	degrees North	the cell boundary latitude
`grid_xt_bnds`	degrees East	the cell boundary longitude
`time_bnds`	days	time axis boundaries
`areo`	degrees	areocentric longitude
`ps`	Pa	surface pressure
`ucomp`	m s^{-1}	zonal wind
`vcomp`	m s^{-1}	meridional wind
`temp`	K	temperature
`ukd`	m s^{-1}	lowest-layer U velocity
`vkd`	m s^{-1}	lowest-layer V velocity
`tkd`	K	lowest-layer temperature
`stress`	N m^{-2}	surface wind stress
`snow`	kg m^{-2}	surface CO_2 ice
`precip`	kg m^{-2} dt^{-1}	amount of surface CO_2 ice originating from the atmosphere per time step
`ts`	K	surface temperature
`t05`	K	50 Pa temperature
`taudust_VIS`	op	visible dust opacity
`dustref`	optical depth per Pa	visible dust opacity per Pa

Table 5.4: Default output variables in `atmos_daily` files

Variable	Units	Description
`bk`	none	the vertical coordinate sigma value
`ak`	Pa	the pressure part of hybrid coordinate
`areo`	degrees	areocentric longitude
`ps`	Pa	surface pressure
`ts`	K	surface temperature
`scalar_axis`	none	none
`lat`	degrees North	latitude
`lon`	degrees East	longitude
`phalf`	mb	the pressure of the layer interface
`time`	days	number of sols since the start of the run
`grid_yt_bnds`	degrees North	the cell boundary latitude
`grid_xt_bnds`	degrees East	the cell boundary longitude

Table 5.5: Default output variables in `atmos_diurn` files

Variable	Units	Description
`bk`	none	vertical coordinate sigma value
`ak`	Pa	pressure part of hybrid coordinate
`areo`	degrees	areocentric longitude
`ps`	Pa	surface pressure
`ts`	K	surface temperature
`time`	days	number of sols since the start of the run
`average_T1`	days	start time for averaging period
`average_T2`	days	end time for averaging period
`average_DT`	days	length of averaging period
`time_of_day_24`	hours	hour of day
`scalar_axis`	none	none
`lon`	degrees East	longitude
`phalf`	mb	the pressure of the layer interface
`time_of_day_edges_24`	hours	time of day at the bin edges
`lat`	degrees North	latitude
`grid_yt_bnds`	degrees North	the cell boundary latitude
`time_bnds`	days	time axis boundaries
`grid_xt_bnds`	degrees East	the cell boundary longitude

5.3 Pressure-Interpolating Re-gridded Data

After untarring and re-gridding the output, `runpinterp.csh` can be used to pressure-interpolate each of the three `00668.atmos_*.nc` files onto a standard pressure grid. Note the names of the files to interpolate and call `runpinterp.csh` for each file as below:

```
user@lfe:~$ ~/FRE-NCtools/bin/runpinterp.csh -d 00668 -f atmos_average
user@lfe:~$ ~/FRE-NCtools/bin/runpinterp.csh -d 00668 -f atmos_daily
user@lfe:~$ ~/FRE-NCtools/bin/runpinterp.csh -d 00668 -f atmos_diurn
```

NOTE: Modify the path to `FRE-NCtools` as needed.

This creates three pressure-interpolated files ending in `*_pstd.nc` in the directory:

- `00668.atmos_daily_pstd.nc`
- `00668.atmos_average_pstd.nc`

- `00668.atmos_diurn_pstd.nc`

The interpolation script defaults to a 48-layer grid ranging from 10^3 Pa near the surface to 10^{-5} Pa near the top. The layers are listed in order from the top of the atmosphere to the layer nearest the surface below. Given in Pa, the standard pressure layers are:

```
1.0e-5   3.0e-5   5.0e-5   1.0e-4   3.0e-4   5.0e-4   3.0e-3   5.0e-3
1.0e-2   3.0e-2   5.0e-2   0.1      0.2      0.3      0.5      1
2        3        5        7        10       20       30       50
70       1.0e+2   1.5e+2   2.0e+2   2.5e+2   3.0e+2   3.5e+2   4.0e+2
4.5e+2   5.0e+2   5.3e+2   5.5e+2   5.9e+2   6.0e+2   6.3e+2   6.5e+2
6.9e+2   7.0e+2   7.5e+2   8.0e+2   8.5e+2   9.0e+2   9.5e+2   1.0e+3
```

The pressure-interpolated files contain all the variables in the diag table(s). The vertical dimension name for every 4D variable should reflect the change from the native grid (`pfull`) to the standard pressure grid (`pstd`).

Chapter 6

Default Simulation Description

The default runscripts for both the current and early Mars versions of the MGCM produce two-year long simulations of the Martian atmosphere at approximately 240 km ($\sim 4°$) horizontal resolution. This chapter provides figures and descriptions showing what the current and early MGCM-simulated atmospheres look like after running the default current (`fms_mars_default_v3.1`) and early Mars (`fms_earlymars_500mb_v3.1`) runscripts. If you run the default simulation as described in Part I, re-grid and pressure-interpolate your files as described in Chapter 5, and use the Ames Community Analysis Pipeline (CAP) to produce the default set of plots as described in Chapter 8, you can create the same figures and compare them to those shown here. These plots are also available in the source code:

```
AM4/src/AmesGCM/diagnostics/Model_Release_Diagnostics.pdf
```

6.1 Current Mars Simulation

The first eight sets of multi-panel plots (Figures 6.1–6.8) show latitude versus pressure zonal mean cross-sections of atmospheric temperature (K; top left), zonal (east-west) wind (m s^{-1}; top right), meridional (north-south) wind (m s^{-1}; bottom left), dust visible opacity (left), and dust mass mixing ratio (ppm; right) for the four cardinal seasons: $L_s = 0°$, 90°, 180°, and 270°. Figure 6.9 shows L_s (season) versus latitude of zonal mean surface temperature (K; top left), zonal mean surface CO_2 ice (kg m^{-2}; top right), and zonal mean surface stress (mN m^{-2}; bottom left). Figure 6.10 shows zonal mean column-integrated visible dust optical depth (normalized to 610 Pa; top left) and zonal mean column-integrated dust mixing ratio (g m^{-2}; top right) as well as the zonal mean dust lifting (bottom left) and deposition (bottom right) rates (micrometers per day). Figure 6.11 shows the annual cycle of the area-weighted global mean visible dust optical depth normalized to 610 Pa. Similarly, Figure 6.12 shows the annual cycle of global mean surface temperature (K; top left) and surface pressure (Pa; top right) and the annual global mean surface temperature (K; bottom left).

At $L_s = 0°$ (northern hemisphere spring equinox; Figure 6.1), the atmospheric thermal structure is nearly symmetric about the equator. In the low latitudes, zonal mean temperatures maximize near the surface in the tropics just over 220 K and decrease with altitude (decreasing pressure) to less than 130 K at approximately 0.5 Pa. Warm air ($\sim$160 K) extends upwards towards the poles in both hemispheres due to compressional heating from the nearly symmetric hadley cell circulation (not shown). Cold air ($\sim$130 K) resides at $\sim$10 Pa over both poles. The wind fields are consistent with this thermal structure. Westerly zonal winds (positive contours, top right panel, Figure 6.1) exist in both hemispheres, peaking at $\sim$100 m s^{-1} in the southern hemisphere and at slightly stronger than that in the northern hemisphere. Easterlies (negative contours, top right panel, Figure 6.1) exist in the tropics and subtropics, peaking at $\sim$125 m s^{-1} aloft. The meridional winds show a strong overturning circulation that is basically symmetric about the equator, with strong southerlies (northward flow) in the northern hemisphere ($\sim$30 m s^{-1}) at $\sim$0.1 Pa, strong northerlies (southward flow) in the southern hemisphere ($\sim$25 m s^{-1}) at 0.1 Pa, and weak opposite flow near the surface. The dust field shows the highest dust loading in the lower atmosphere at low latitudes (opacities >0.4 and mixing ratios of $\sim$5 ppm).

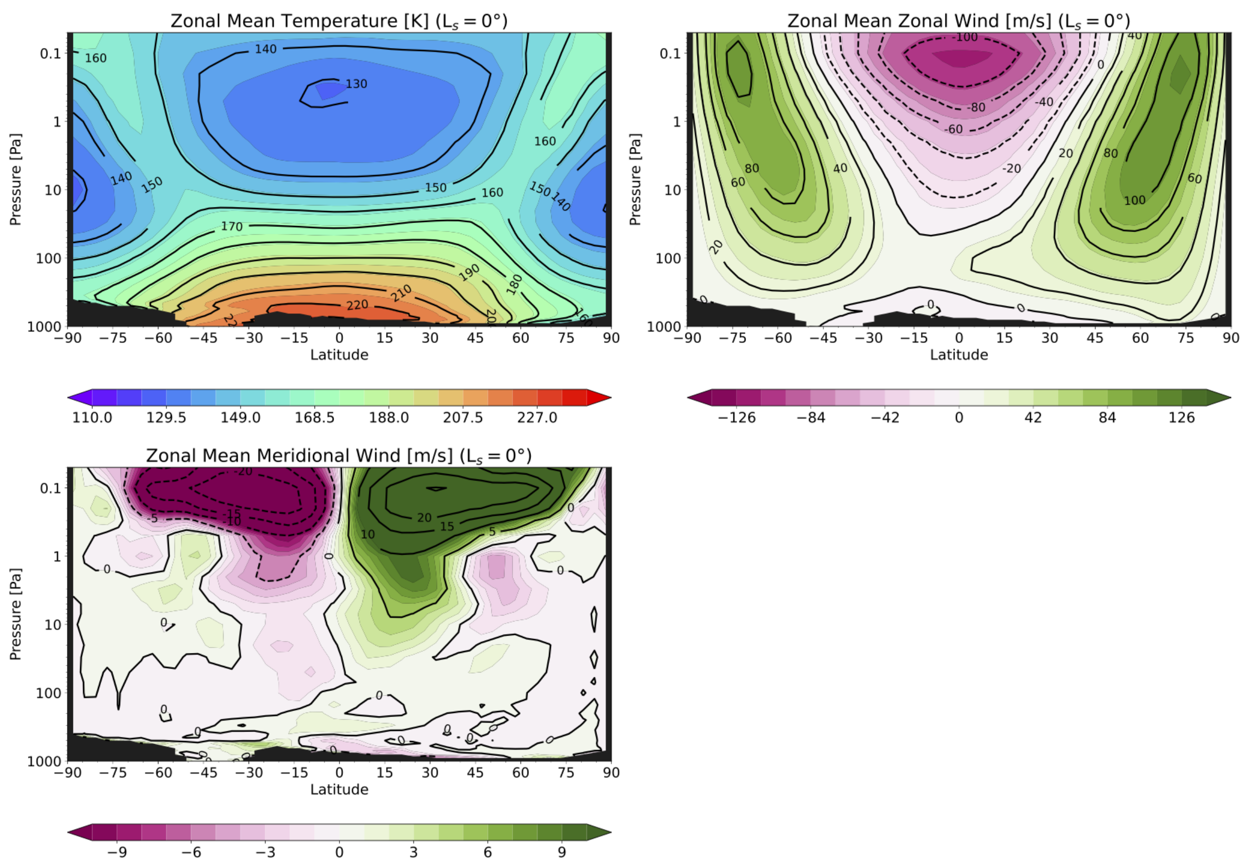

Figure 6.1: Zonal mean temperature (top left), zonal wind (top right), and meridional wind (bottom left) at $L_s = 0°$ from the default current Mars simulation (`fms_mars_default_v3.1`). Created with CAP (Chapter 8).

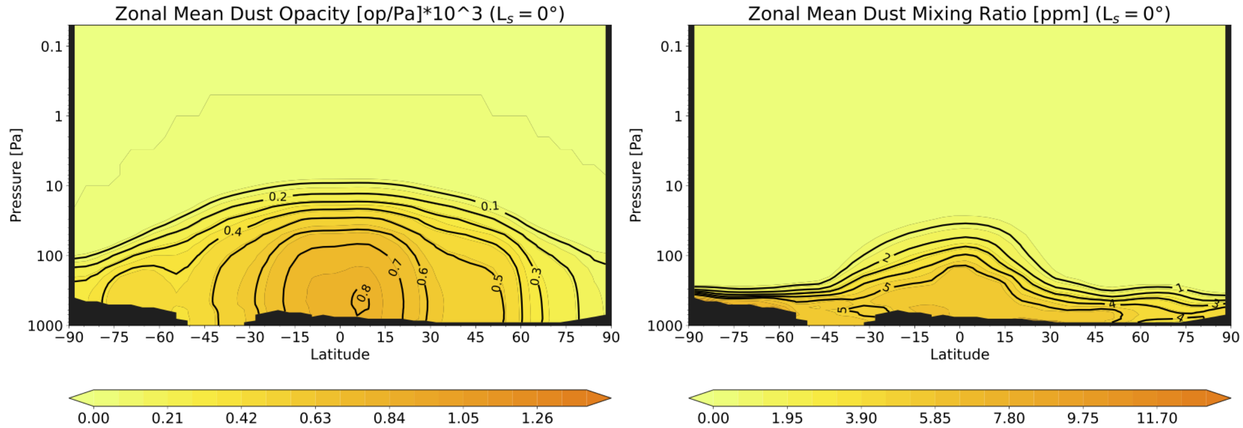

Figure 6.2: Zonal mean visible dust opacity (left) and mixing ratio (ppm; right) at $L_s = 0°$ from the default current Mars simulation (`fms_mars_default_v3.1`). Created with CAP (Chapter 8).

At $L_s = 90°$ (northern hemisphere summer solstice; Figure 6.3), peak temperatures reside in the northern hemisphere near the surface (∼230 K). The coolest temperatures are over the south (winter) pole at ∼1 Pa (∼120 K). Weak polar warming occurs due to the descending branch of the Hadley cell in the southern hemisphere. There is a layer of warm air at the top of the plot that peaks at ∼160 K in the high northern latitudes. The zonal wind field shows a westerly jet that peaks at ∼100 m s^{-1} in the southern hemisphere and easterlies throughout a large part of the northern hemisphere (peaking at ∼75 m s^{-1} at about 15° N). The meridional wind field shows multiple peaks in northerly (southward) flow: two at ∼20° S (∼10 Pa and ∼0.5 Pa, both ∼12 m s^{-1}), and one at the top of the plot at ∼25° N, peaking over 14 m s^{-1}). Southerly (northward) flow is evident very near the surface at the low latitudes. The dust field shows the highest dust loading near the surface over the equator (opacities >0.4 and mixing ratios >5 ppm).

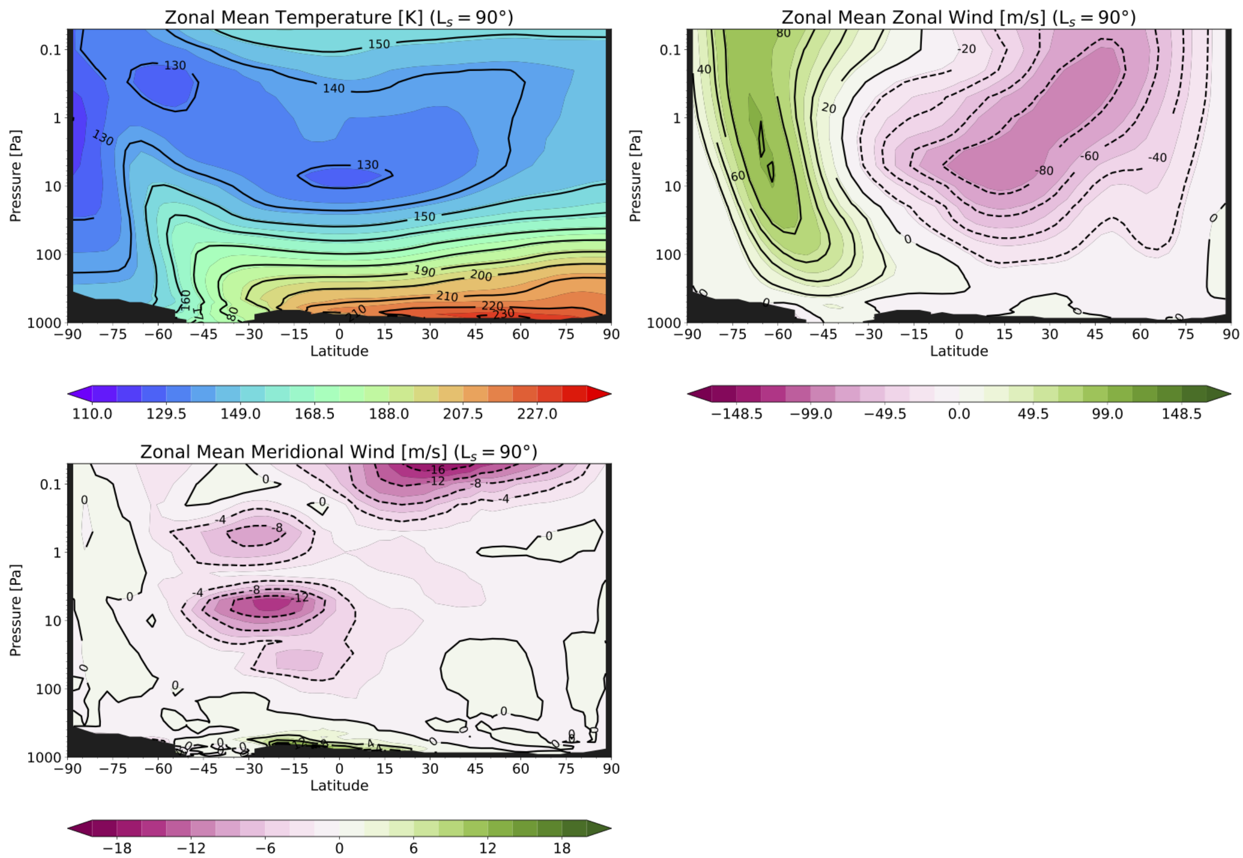

Figure 6.3: Zonal mean temperature (top left), zonal wind (top right), and meridional wind (bottom left) at $L_s = 90°$ from the default current Mars simulation (`fms_mars_default_v3.1`). Created with CAP (Chapter 8).

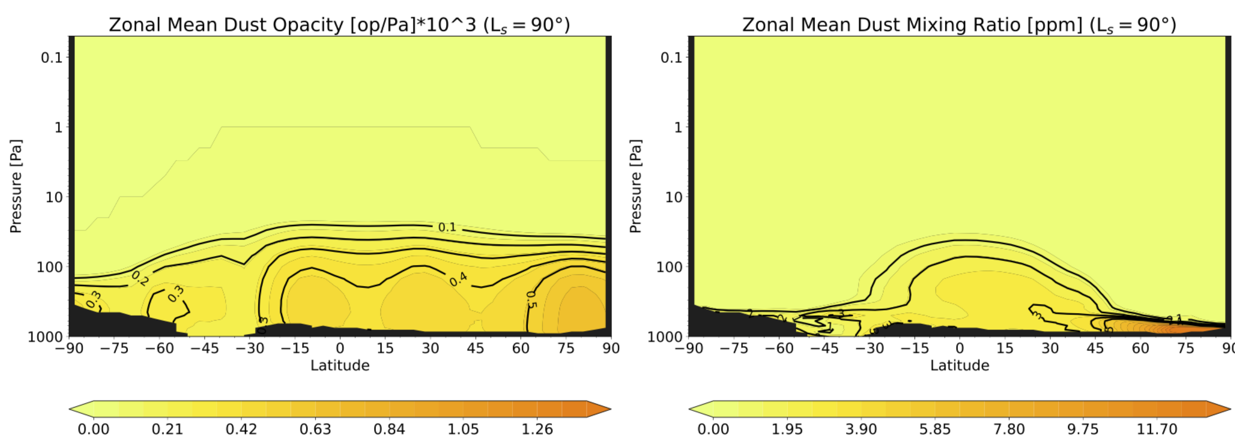

Figure 6.4: Zonal mean visible dust opacity (left) and mixing ratio (ppm; right) at $L_s = 90°$ from the default current Mars simulation (`fms_mars_default_v3.1`). Created with CAP (Chapter 8).

$L_s = 180°$ (northern hemisphere autumnal equinox; Figure 6.5), is similar in many ways to

$L_s = 0°$ (Figure 6.1). In the low latitudes, zonal mean temperatures maximize near the surface in the tropics just over 230 K and decrease with altitude (decreasing pressure) to less than 130 K at approximately 0.5 Pa. Warm air ($\sim$170 K) extends upwards towards the poles in both hemispheres due to compressional heating of a nearly symmetric Hadley cell circulation (not shown). Cold air ($\sim$130 K) resides at $\sim$10 Pa over both poles. Westerly zonal winds exist in both hemispheres, peaking at over 100 m s^{-1}. Easterlies (negative contours, top right panel, Figure 6.5) exist in the tropics and subtropics, peaking at $\sim$125 m s^{-1} aloft. The meridional winds show a strong overturning circulation that is basically symmetric about the equator, with strong southerlies (northward flow) in the northern hemisphere, and strong northerlies (southward flow) in the southern hemisphere (peaking at $\sim$25 m s^{-1} at 0.1 Pa), and weak opposite flow near the surface. The dust field shows the highest dust loading in the lower atmosphere at southern middle latitudes (opacities >0.6 and mixing ratios of $\sim$12 ppm).

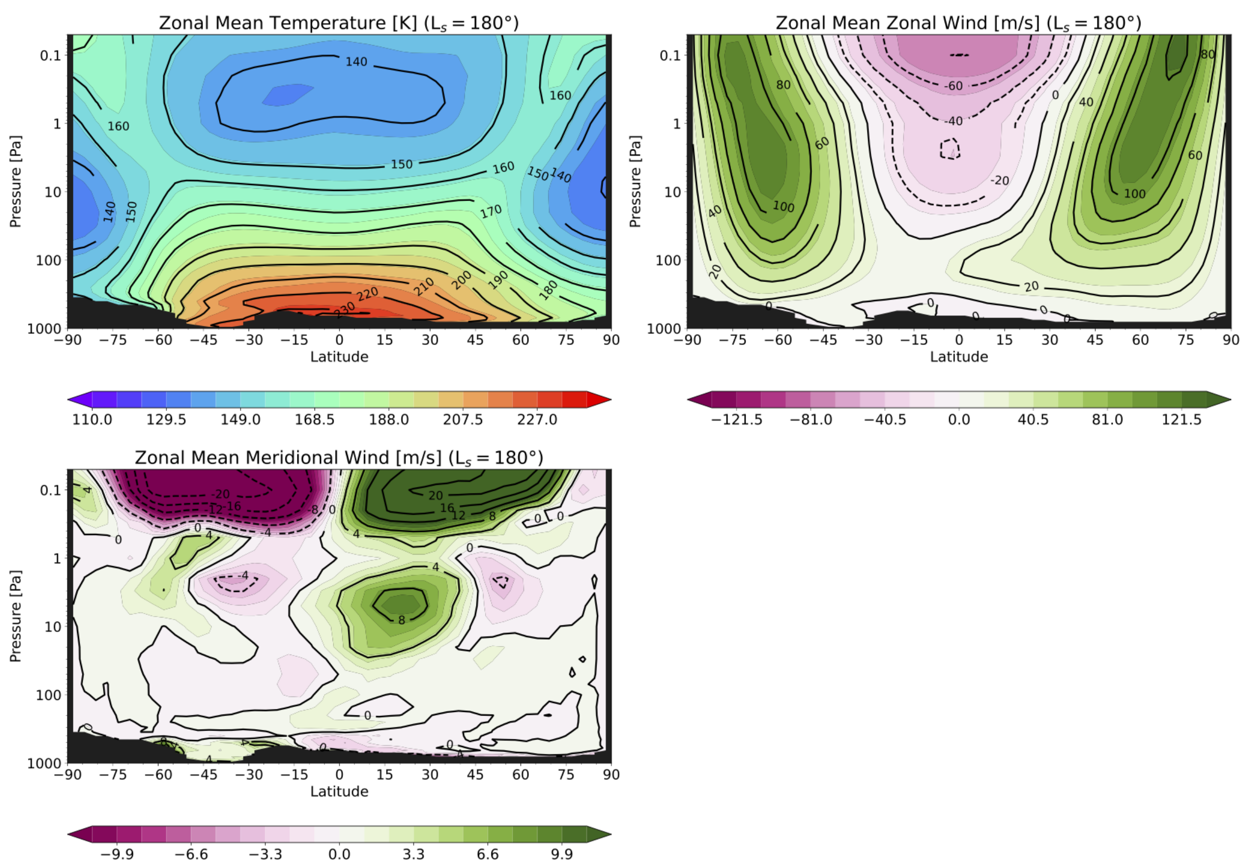

Figure 6.5: Zonal mean temperature (top left), zonal wind (top right), and meridional wind (bottom left) at $L_s = 180°$ from the default current Mars simulation (`fms_mars_default_v3.1`). Created with CAP (Chapter 8).

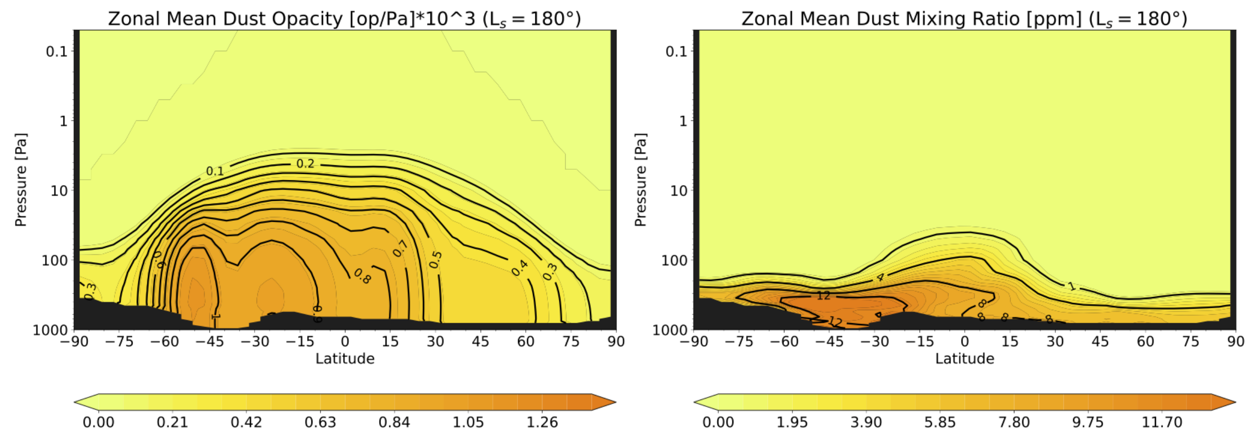

Figure 6.6: Zonal mean visible dust opacity (left) and mixing ratio (ppm; right) at $L_s = 180°$ from the default current Mars simulation (`fms_mars_default_v3.1`). Created with CAP (Chapter 8).

At $L_s = 270°$ (northern hemisphere winter solstice; Figure 6.7), peak temperatures reside in the southern hemisphere near the surface (∼240 K). The coolest temperatures are over the south (winter) pole at ∼100 Pa (∼140 K) and in the low latitudes aloft (less than 135 K at ∼2 Pa). Strong polar warming occurs due to the descending branch of the Hadley cell in the northern hemisphere. The zonal wind field shows a westerly jet that peaks at ∼100 m s^{-1} in the northern hemisphere and easterlies throughout a large part of the southern hemisphere (peaking at ∼150 m s^{-1} at about 15 S). The meridional wind field shows strong (∼40 m s^{-1}) southerly (northward) flow ∼2 Pa. Northerly (southward) flow is evident very near the surface at the low latitudes. The dust field shows the highest dust loading near the surface at the highest southern latitudes (opacities >0.9 and mixing ratios >12 ppm) and throughout the tropics and subtropics (>0.7 and mixing ratios >8 ppm).

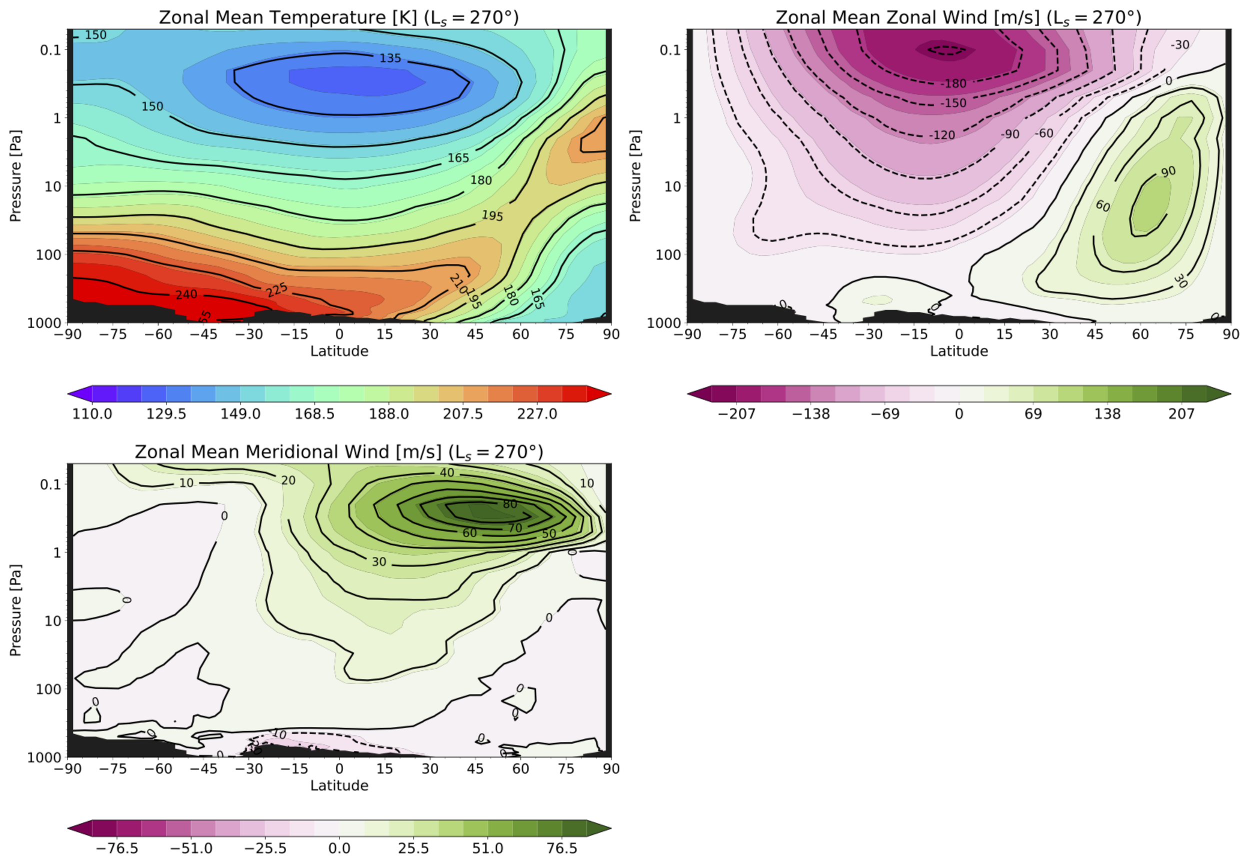

Figure 6.7: Zonal mean temperature (top left), zonal wind (top right), and meridional wind (bottom left) at $L_s = 270°$ from the default current Mars simulation (`fms_mars_default_v3.1`). Created with CAP (Chapter 8).

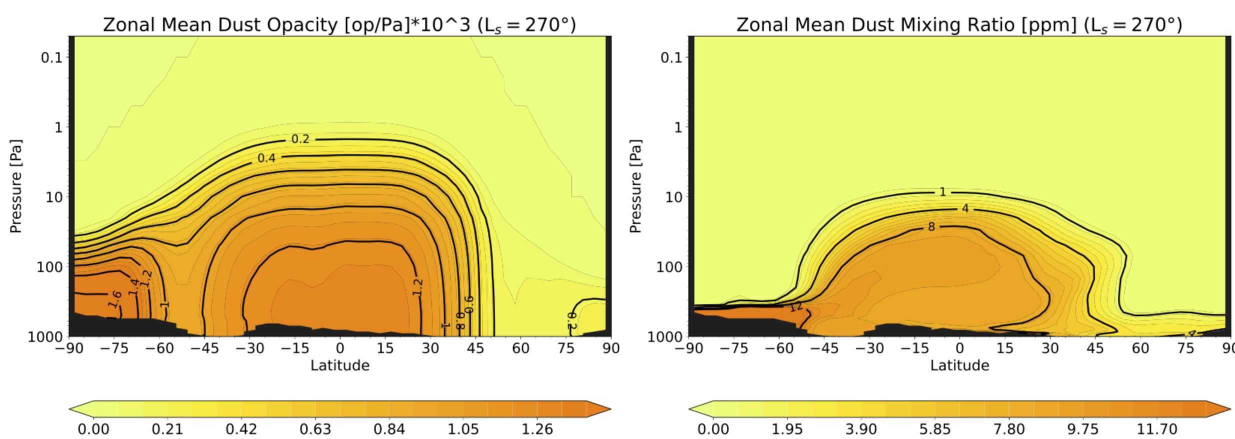

Figure 6.8: Zonal mean visible dust opacity (left) and mixing ratio (ppm; right) at $L_s = 270°$ from the default current Mars simulation (`fms_mars_default_v3.1`). Created with CAP (Chapter 8).

The annual zonal mean surface/column-integrated plots in Figures 6.9 and 6.10 show the full annual cycle. For the surface fields in Figure 6.9, zonal mean surface temperatures maximize during local summer (~230 K in the north during northern summer and ~250 K in the south during southern summer) and minimize during local winter (~150 K, which is controlled by the CO_2 saturation temperature). The low latitudes show a less pronounced seasonal cycle. The surface CO_2 ice field shows that CO_2 ice condenses during local autumn and sublimates during local spring, with maximum values of ~600 kg m^{-2} and more than 800 kg m^{-2} in the north and south, respectively. Surface stresses maximize along the growing and receding caps CO_2 caps due to cap-edge circulations and in the southern subtropics during southern summer due to strong winds associated with the (Coriolis) deflection of the lower branch of the Hadley cell. Surface stresses tend to be strongly controlled by topography and have large diurnal variations.

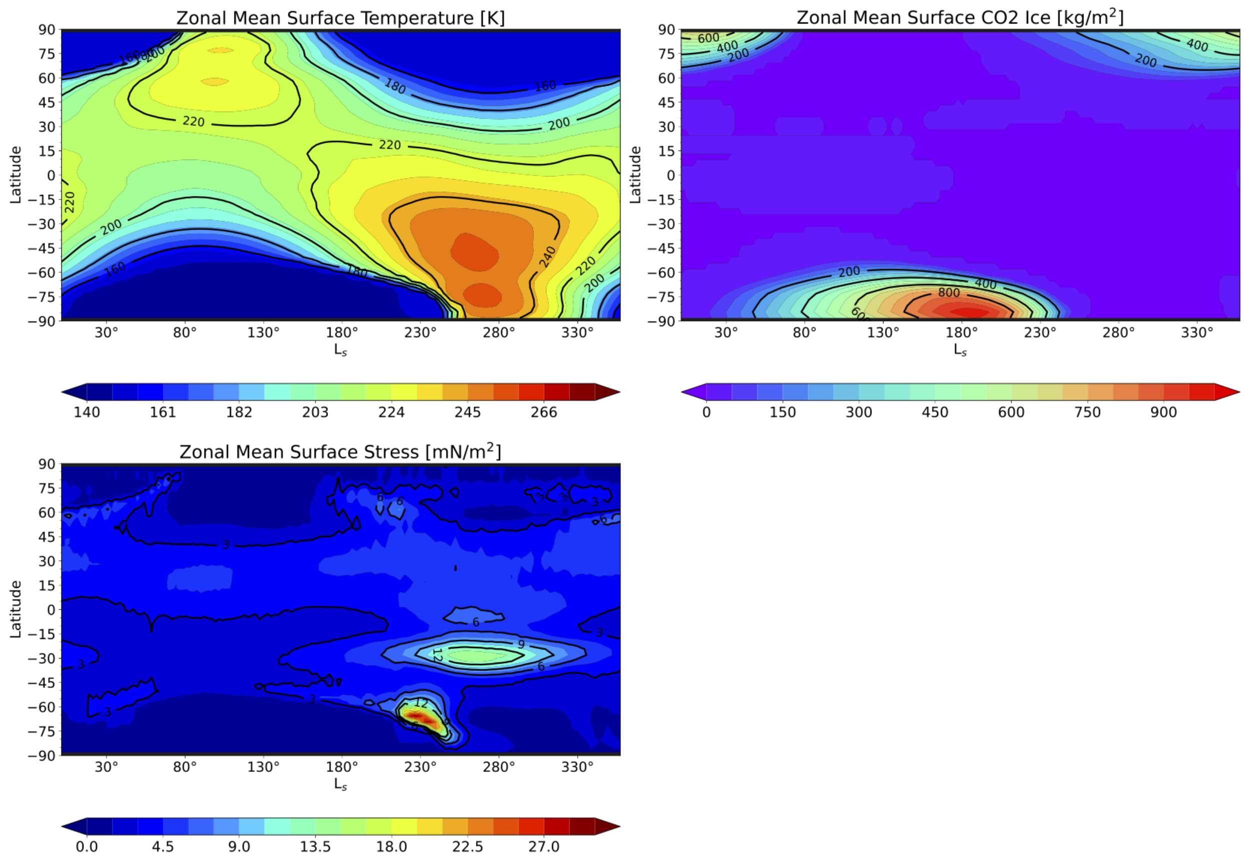

Figure 6.9: Zonal mean surface temperature (top left), CO_2 ice (top right), and surface stress (bottom left) for the second MY in the default current Mars simulation (`fms_mars_default_v3.1`). Created with CAP (Chapter 8).

Annual information regarding the dust tracers in the default simulation is shown in Figure 6.10. The zonal mean visible dust optical depth field (top left) shows that the model is forced with a background dust optical depth of ~0.2–0.3 during the first half of the year (which results in ~400 10^{-6} kg m^{-2} of column integrated dust; top right), with increased dust loading during the second half of the year. The pre- and post-solstitial increases in dust loading peak in the low latitudes at ~0.9 and ~0.7, respectively, which produces ~2000 10^{-6} and ~1200 10^{-6}kg m^{-2}, respectively, of column integrated dust mass.

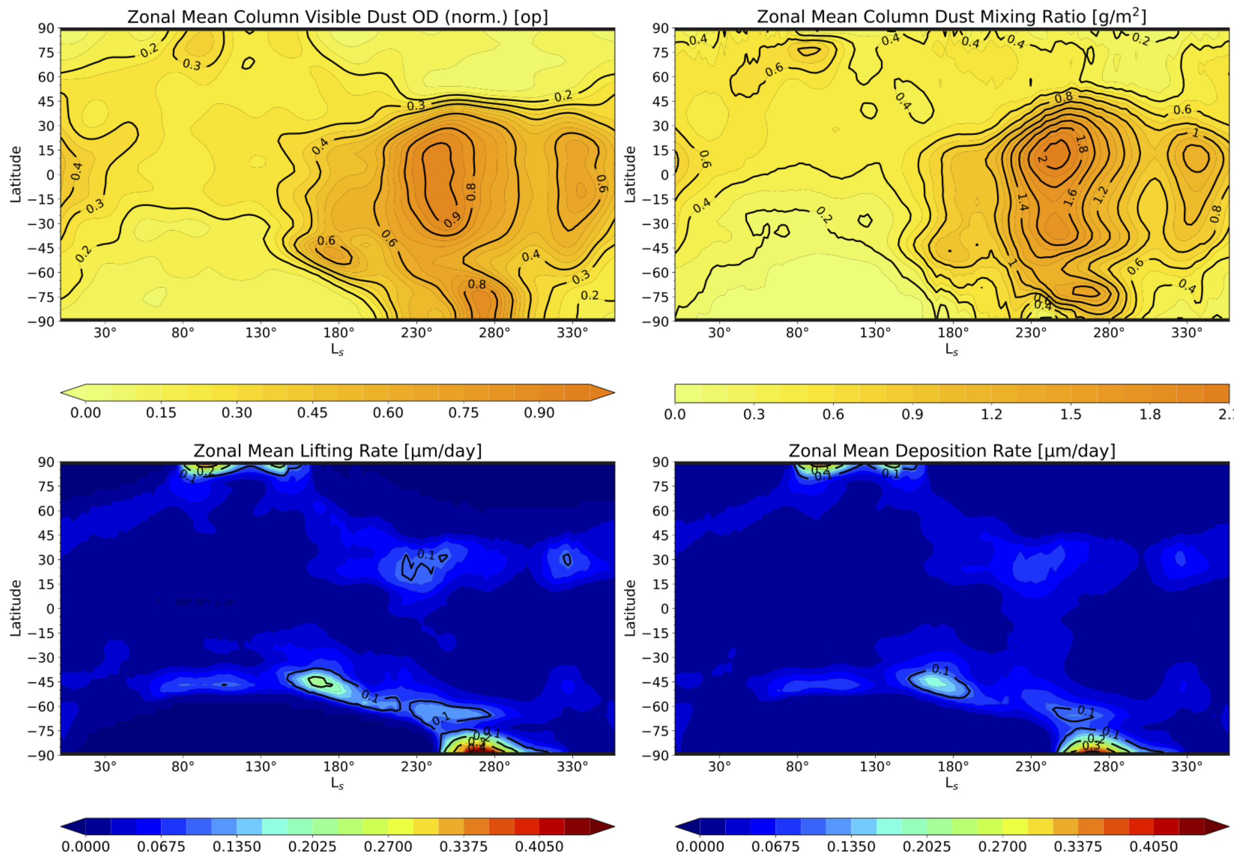

Figure 6.10: Zonal mean visible dust optical depth (top left), column-integrated mixing ratio (top right), and zonal mean dust lifting (bottom left) and deposition (bottom right) rates for the second MY in the default current Mars simulation (`fms_mars_default_v3.1`). Created with CAP (Chapter 8).

Global mean visible dust optical depth varies (Figure 6.11) annually in a manner consistent with the zonal mean vs season plot shown above in Figure 6.10. A global mean background dust haze of ∼0.25 is simulated during northern spring and summer, with elevated dust optical depth during norther fall and winter. The pre- and post-solstice peaks produce global mean visible dust optical depths of ∼0.65 and ∼0.5, respectively.

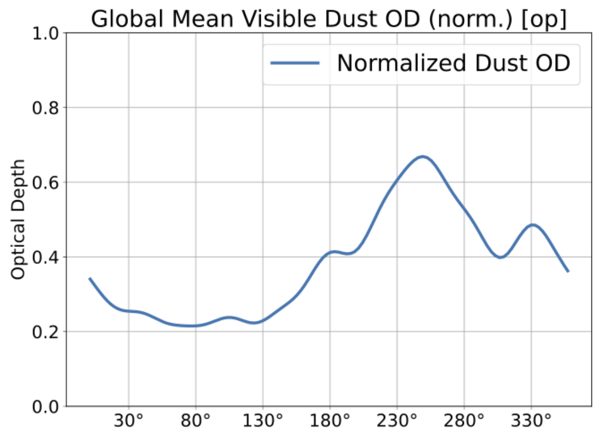

Figure 6.11: Annual global mean visible dust optical depth for the second MY in the default current Mars simulation. Created with CAP (Chapter 8).

Global mean surface temperature varies annually between ∼197 and ∼219 K (Figure 6.12). Global mean surface pressure varies between ∼537 and ∼685 Pa over the course of the year and results from exchange of CO_2 between the atmosphere and the north and south seasonal ice caps. The bottom left panel shows a map of annual mean surface temperature. Annual mean surface temperatures show the cold poles and reflect the zonally asymmetric surface albedos and thermal inertias at lower latitudes.

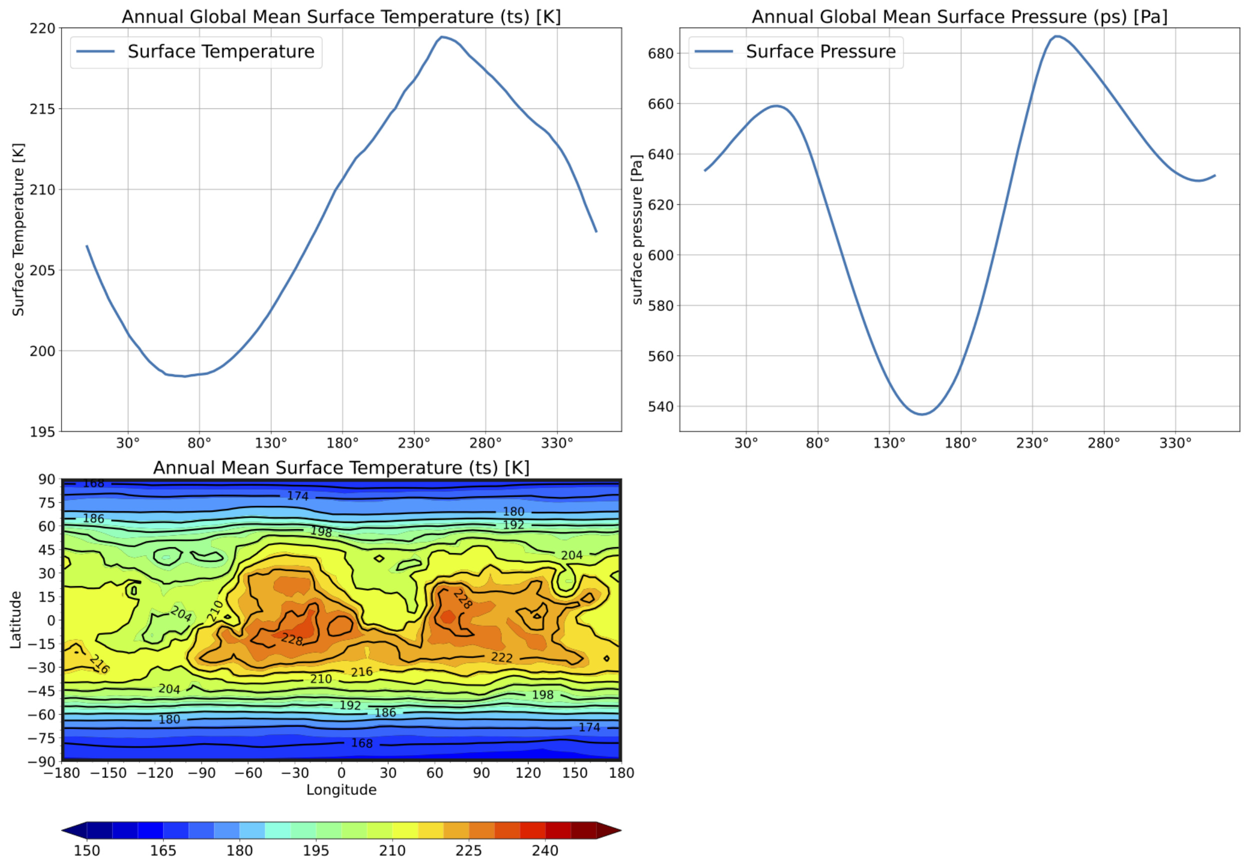

Figure 6.12: Annual global mean surface temperature (left) and surface pressure (right), and annual mean surface temperature (bottom) for the second MY in the default current Mars simulation. Created with CAP (Chapter 8).

6.2 Early Mars Simulation

The default early Mars configuration includes a 500 mbar CO_2 atmosphere and flat topography, among other treatments described in Section 1.3. Here we show the default plots that result from that simulation. The first four sets of multi-panel plots (Figures 6.13–6.16) show latitude versus pressure zonal mean cross-sections of atmospheric temperature (K), zonal (east-west) wind (m s^{-1}), and meridional (north-south) wind (m s^{-1}) for the four cardinal seasons: $L_s = 0°$, 90°, 180°, and 270°. Figure 6.17 shows L_s (season) versus latitude of zonal mean surface temperature (K; top left), zonal mean surface CO_2 ice (kg m^{-2}; top right), and zonal mean surface stress (N m^{-2}; bottom left). Figure 6.18 shows the annual global mean surface temperature (K; top left) and surface pressure (Pa; top right) and the annual zonal mean surface temperature (K; bottom left).

Simulation results at $L_s = 0°$ (northern hemisphere spring equinox) are shown in Figure 6.13. Note in this first figure and subsequent figures in this subsection, that the vertical y-axis range reflects the larger surface pressure for early Mars compared to current Mars, as well as the flat surface topography. Unlike present-day Mars at this $L_s = 0°$ season, there is some asymmetry about the equator in the temperature fields, as the southern hemisphere remains slightly warmer from the preceding summer. This asymmetry is also reflected in the zonal wind field at this period, with stronger westerlies in the northern hemisphere compared to those in the southern hemisphere. Peak westerly zonal wind speeds reach only ~25 m s^{-1} and are located near 300 Pa. The peak easterly zonal winds at this season are at a lower altitude in this simulation compared to present-day Mars. Peak meridional wind speeds are also much smaller in this early Mars configuration compared to the present day; there are only weak southerlies (northern flow) at altitude ~10 Pa in the northern hemisphere. At the equator, there is northerly flow close to the surface and southerly flow above that, but still in the lower atmosphere. In the more massive atmosphere environment, there is less latitudinal variation in atmospheric temperatures, and this smaller difference in equatorial vs polar temperatures drives a weak circulation compared to that of present day Mars.

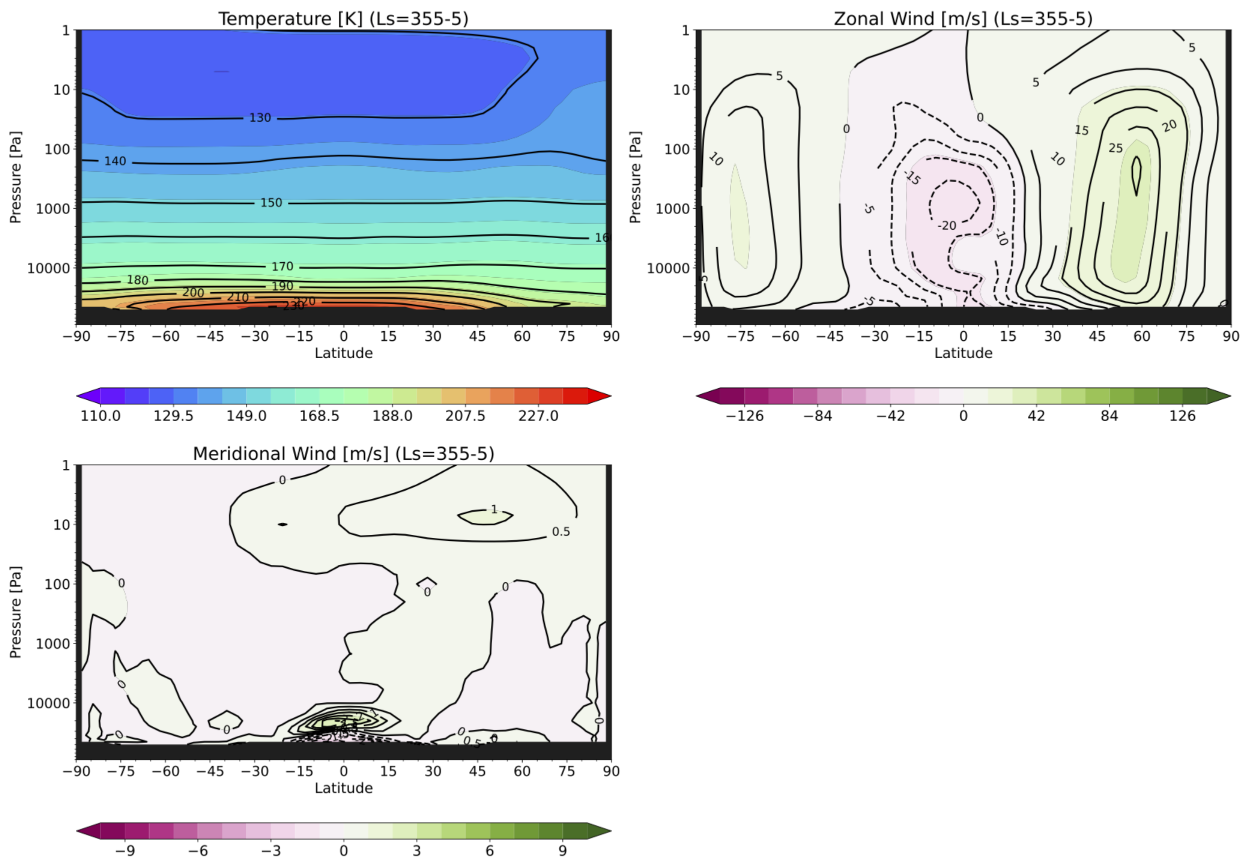

Figure 6.13: Zonal mean temperature (top left), zonal wind (top right), and meridional wind (bottom left) at L$_s$ = 0° from the default early Mars simulation (`fms_mars_default_v3.1`). Created with CAP (Chapter 8).

At L$_s$ = 90° (northern hemisphere summer; Figure 6.14), atmospheric temperatures reach a maximum of ~230 K near the surface in the northern hemisphere. The region of peak temperatures lies from ~25 to 60° N in latitude and the warmer near-surface temperatures do not extend to polar latitudes as they do for current Mars at this season. Minimum temperatures fall below ~125 K around 45° N at 4 Pa. In the zonal mean wind field, there are easterly winds between ~20° S and 45° N in latitude with westerly winds south of 20° S and north of 45° N. Easterly peak wind speeds reach -25 m s^{-1} around 10° N in latitude and 10^4 Pa in altitude. Westerly winds in the southern hemisphere peak above 30 m s^{-1} around 45° S in latitude, and in the northern hemisphere reach 20 m s^{-1} around 75° N in latitude. Meridional winds are primarily northerly, with local maxima at 10 Pa in altitude and ∼30° S in altitude as well as closer to the surface at the equator.

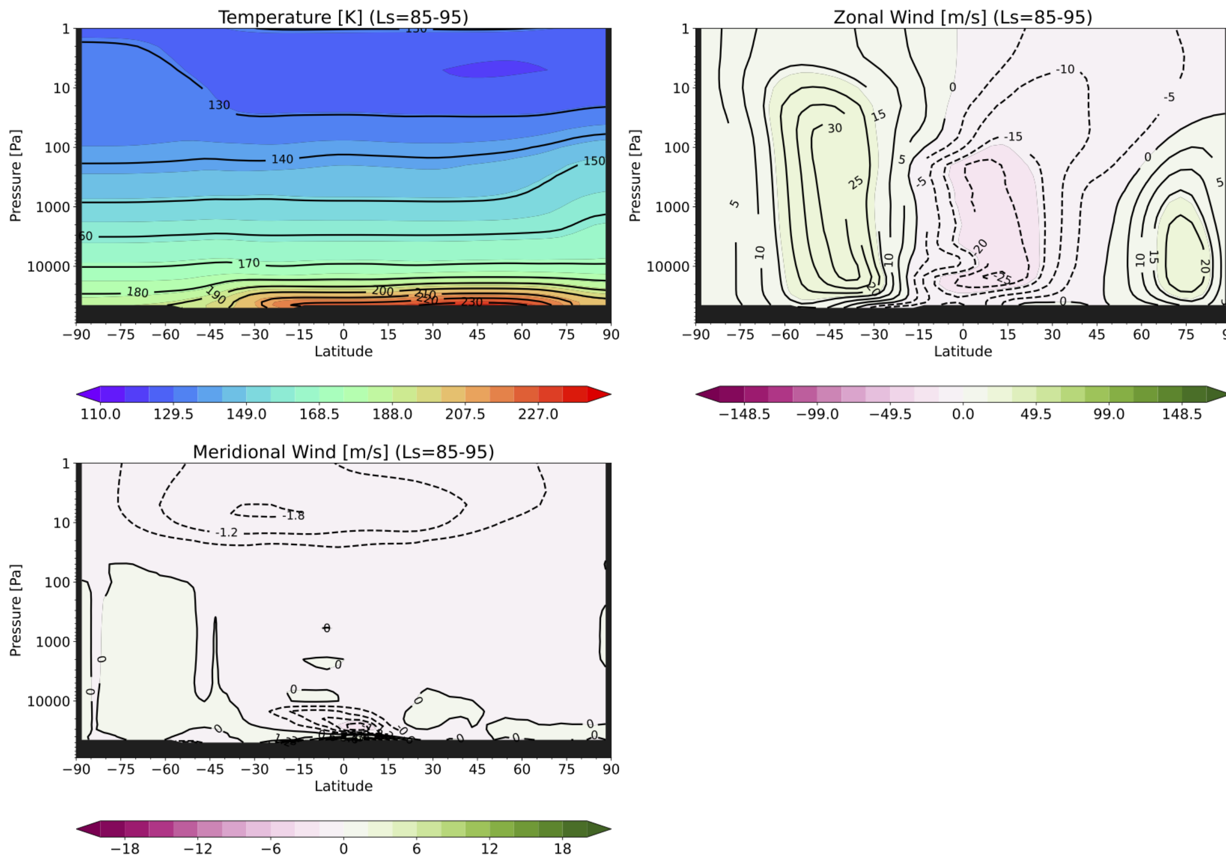

Figure 6.14: Zonal mean temperature (top left), zonal wind (top right), and meridional wind (bottom left) at $L_s = 90°$ from the default early Mars simulation (`fms_mars_default_v3.1`). Created with CAP (Chapter 8).

At $L_s = 180°$ (northern hemisphere autumnal equinox; Figure 6.15), atmospheric temperatures peak around 230 K near the surface just north of the equator. As in the $L_s = 0°$ season, atmospheric temperatures are not symmetric about the equator, but still somewhat offset with slightly warmer temperatures in the northern hemisphere, lagging from the summer season. The coolest temperatures are at ∼ 5 Pa in altitude centered over latitude 45° N. In the zonal wind field, easterly winds reside centered over the equator with peak wind speeds around 1000 Pa. As in the $L_s = 0°$ season, the peak windspeed of the easterlies over the equator is located at a lower altitude compared with present-day Mars. There are westerly winds south of ∼20° S in latitude with peak wind speeds around 30 m s^{-1} and also in the northern hemisphere north of 45° latitude, with peak wind speeds only reaching ∼15 m s^{-1}. The meridional wind field includes northerlies at altitude (∼1–10 Pa) in the southern hemisphere, northerlies around 2000 Pa at the equator, and southerlies near the surface at the equator.

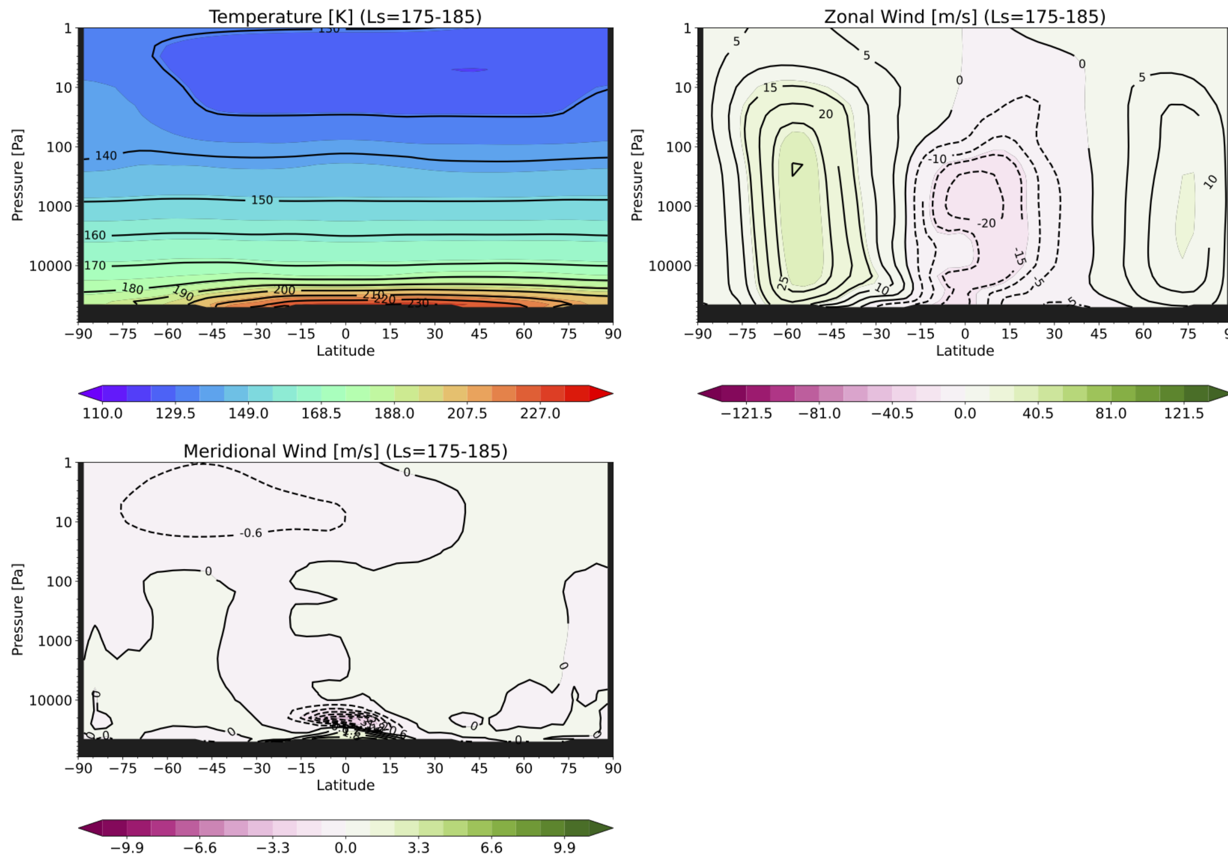

Figure 6.15: Zonal mean temperature (top left), zonal wind (top right), and meridional wind (bottom left) at $L_s = 180°$ from the default early Mars simulation (`fms_mars_default_v3.1`). Created with CAP (Chapter 8).

At $L_s = 270°$ (northern hemisphere winter solstice; Figure 6.16), peak atmospheric temperatures are near the surface around 45° S in latitude, and minimum atmospheric temperatures are at the same latitude around 4 Pa in altitude. In the zonal mean wind field, easterly winds reside mainly in the southern hemisphere between 15° N and 45° S in latitude with peak wind speeds located fairly close to the surface around 20,000 Pa. There are westerly winds in the northern hemisphere north of 15° in latitude that reach 30 m s^{-1} and also in the southern hemisphere south of 45° S with peak wind speeds that exceed 20 m s^{-1}. The meridional wind field includes southerlies with local peaks at altitude above 10 Pa, and also near the surface at the equator.

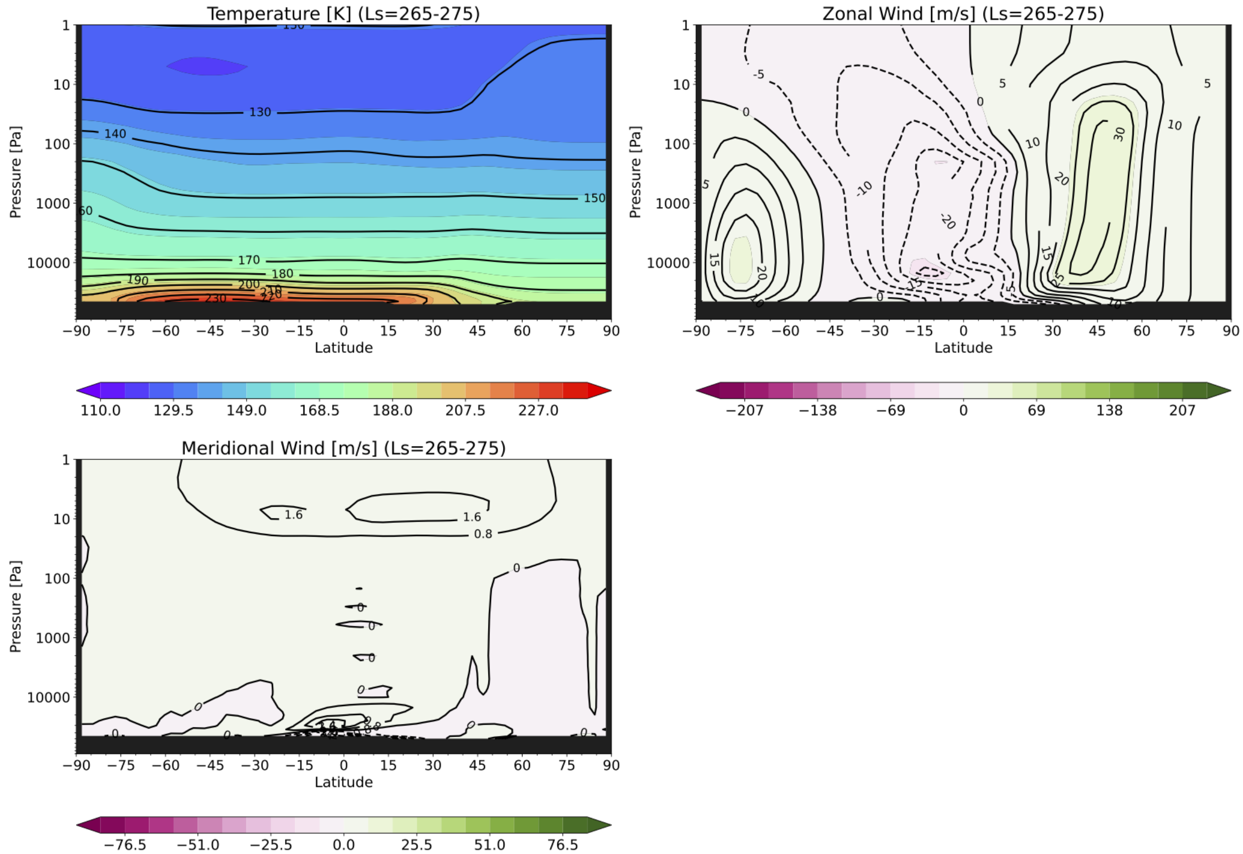

Figure 6.16: Zonal mean temperature (top left), zonal wind (top right), and meridional wind (bottom left) at $L_s = 270°$ from the default early Mars simulation (`fms_mars_default_v3.1`). Created with CAP (Chapter 8).

Zonal mean surface and column-integrated plots are shown in Figure 6.17. Zonal mean surface temperatures peak in the northern hemisphere around $L_s = 110°$ between 45° N and 90° N in latitude, and in the southern hemisphere around $L_s = 290°$ between 45° and 60° S. In the winter seasons, polar temperatures are fixed to the CO_2 condensation temperature, so temperature trends match the growth and recession of the CO_2 seasonal polar caps. The seasonal CO_2 ice caps extend to latitudes ± 50° in each winter hemisphere, and CO_2 surface ice quantities exceed 800 kg m-2 over a large latitude region. Zonal mean surface stress peaks close to the equator in the winter hemisphere, with a comparatively smaller maxima in the summer hemisphere each season around ±30° latitude. Maxima are also seen along the retreating edge of the CO_2 ice cap in the spring time leading into summer in each hemisphere.

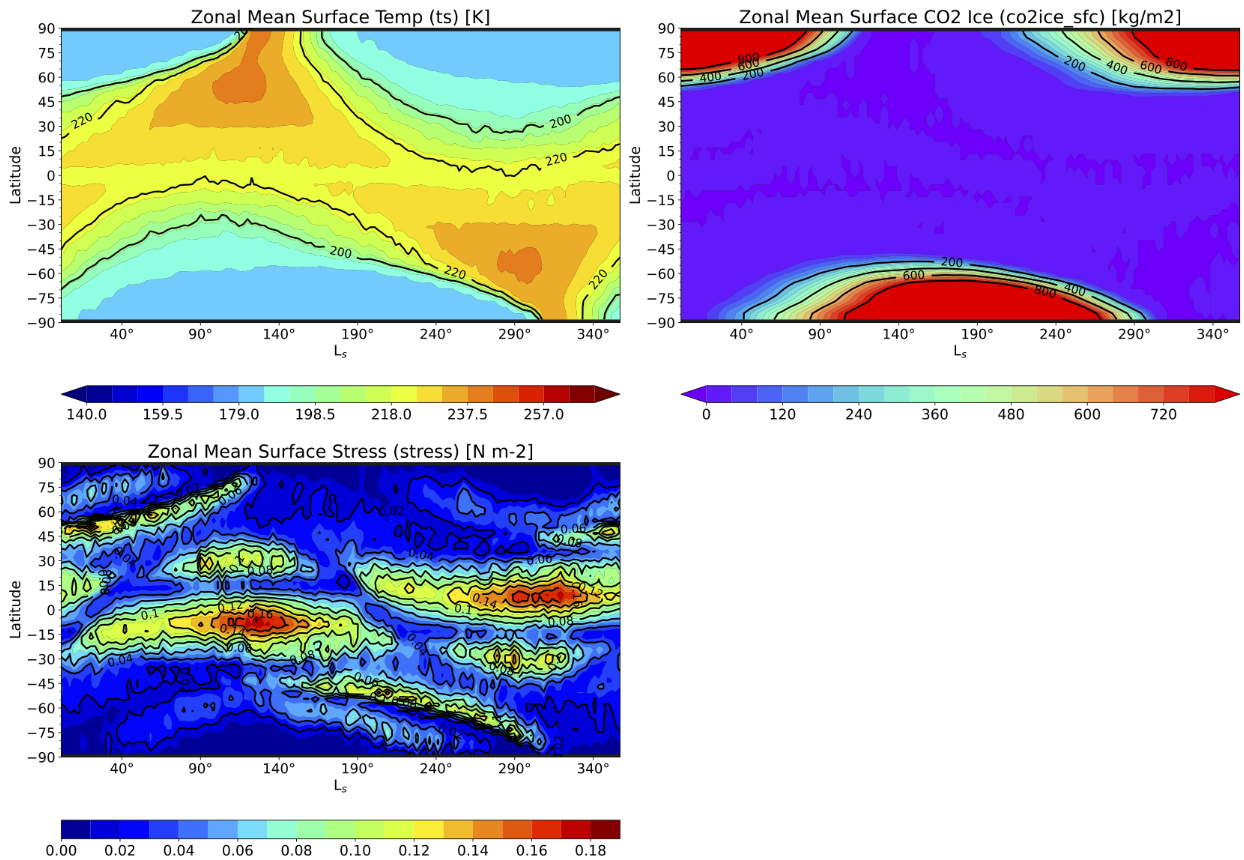

Figure 6.17: Zonal mean surface temperature (top left), CO_2 ice (top right), and stress (bottom left) for the second MY in the default early Mars simulation (`fms_mars_default_v3.1`). Created with CAP (Chapter 8).

Global mean surface temperature varies annually between ∼213 and 216 K (Figure 6.18). Global mean surface pressure varies between 495 and 496 mbar over the course of the year. The symmetric double peak variation in the temperature and pressure cycle here is the result of the circular orbit in this simulation, which means the seasonal variation in surface temperature and the CO_2 cycle is very similar between the northern and southern hemispheres. The bottom left panel shows a map of annual mean surface temperature. Annual mean surface temperatures are symmetric about the equator as topography is flat and eccentricity is zero. Annual mean temperatures range from ∼190 K at the poles to 222 K at the equator.

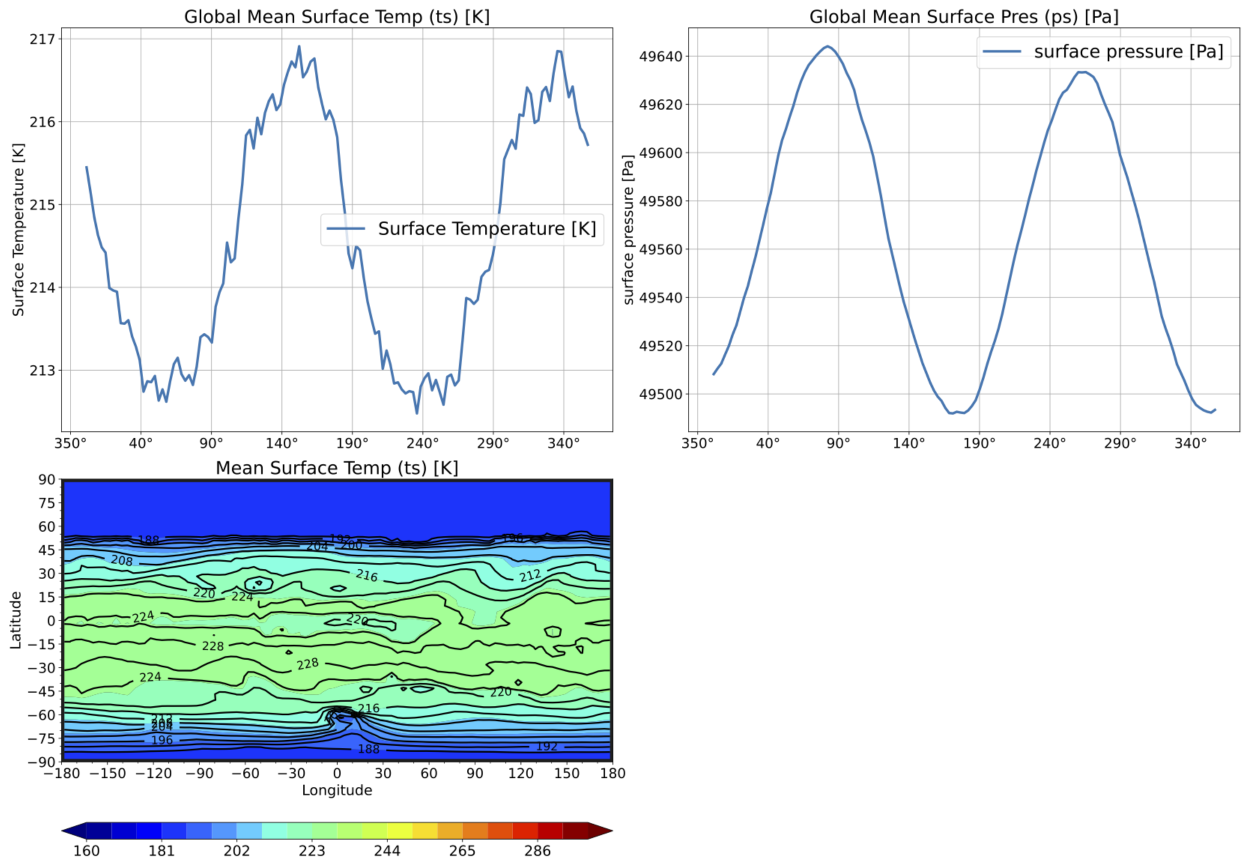

Figure 6.18: Annual global mean surface temperature (left) and surface pressure (right), and annual mean surface temperature (bottom) for the second MY in the default early Mars simulation. Created with CAP (Chapter 8).

Chapter 7

Troubleshooting

In this chapter, we address some of the common issues that arise when running MGCM 3.1, including compilation problems, runtime errors, and model stability issues. This is *not* intended to be an exhaustive list of potential problems or solutions, but it *is* intended to be a resource for users to reference when the model is not operating as expected.

7.1 Compilation Errors

When the model appears to crash before even starting to run, it is possible that the model was not compiled before the runscript was submitted for execution. To confirm that this is the problem, check the `fms_mars_out.*` file in the `workdir/`, which is on Pleiades at:

```
/nobackup/$USER/FMS_MARS_runs/am4_mars_runs/$RUNSCRIPT_NAME/
```

If failing to compile the model is the issue, then the contents of `fms_mars_out.*` will appear as follows:

```
./FMS_MARS.x: No such file or directory
MPT ERROR: could not run executable.
```

There is a simple fix for this: run the compile script before re-submitting the runscript for execution. For instructions on compilation, see Chapter 4.

7.2 Runscript Errors

When running MGCM 3.1 on NAS, a common reason that a simulation might fail to run all the way through the requested `dayslist` is because the model ran out of walltime. To diagnose this problem, open the `*.o*` file in the `AM4/exec/` directory. The `*.o*` file is named after the runscript, so for the default simulation you will look for a file like `fms_mars_default.o*` in `AM4/exec/`. The requested walltime is listed **on line 5** and the amount of walltime used is listed at the end under `Job Resource Usage Summary`. If the requested walltime is less than the walltime used, then the simulation did not complete before running out of walltime.

If some history files were created during the simulation, warm start the simulation and adjust `dayslist` to reflect the length of the simulation remaining. You can simply perform a "continuation" warm start by setting `RUNTYPE = 1` in the runscript and the simulation will pick up from the last generated restart file. You may also warm start the model from a specific restart file by setting `RUNTYPE = 2` and specifying the path to and name of the restart file under `restartfile`. Restart files for a simulation can be found in the `restart/` subdirectory:

```
/nobackup/$USER/FMS_MARS_runs/am4_mars_runs/$RUNSCRIPT_NAME/restart/
```

Other errors relating to the runscript setup include incorrect file paths to the input or output directories and incorrect namelist variables (typos). Here is a list of items to check:

1. Double check for typos in the namelist variables in your runscript by comparing them to the variables listed in the default runscript `fms_mars_default`.
2. Confirm that the options selected for each namelist item are of the correct type (boolean, float, integer).
3. Confirm that the various directories defined in the runscript exist **and contain the relevant files**.
4. Ensure that all namelists have the proper syntax:
   (a) Commas appear after all but the last variable in a namelist.
   (b) A backslash appears at the end of every namelist.
   (c) All namelist names begin with an ampersand (&)

If the model cannot find an input file because either the filepath is incorrect or the file does not exist, an error will show in `fms_mars_out.*` in the `workdir` (directory path provided above). For example, if the model cannot locate the topography input file, the following fatal error appears in `fms_mars_out.*`:

```
FATAL from PE   4: surfdrv: mars_topo not found in INPUT
```

If the model cannot find or does not recognize a namelist variable, the following fatal error appears in `fms_mars_out.*`:

```
FATAL from PE   35: check_nml_error in fms_mod: Unknown
                    namelist, or mistyped namelist variable
                    in namelist aerosol_util_nml, (IOSTAT = 19)
```

This error actually points to the namelist throwing the error, in this case, `aerosol_util_nml`. The error is often caused by one of three things:

1. A variable in the indicated namelist (`aerosol_util_nml`, in this case) is misspelled
2. The requested variable does not exist

3. The variable does exist but is listed under the wrong namelist in the runscript. The namelist listed in the error is the inappropriate namelist for the variable (`aerosol_util_nml` in the example above).

7.3 Model Execution Errors

Most other errors likely stem from the model execution. Model instabilities, violating the CFL condition, and calling physics modules that do not exist or are not supported in the current model release are examples of problems that can arise during model execution. While there should be no instability issues when running the model with the default runscript, if certain settings are changed, the user might inadvertently push the model into a regime where the model is unstable or CFL conditions are violated. Here we provide some solutions.

7.3.1 Model Stability Issues

To diagnose errors relating to model stability, check the `fms_mars_out.*` file in the `workdir` (again, listed above). Check for anomalies in the values of variables listed below:

```
Total surface pressure (Pa)      UA_top
UA                               VA
TA                               OM
dummy_tracer                     h2o_vapor
tracer1                          tracer2
tracer3                          dst_mass_mom
dst_num_micro                    ice_mass_mom
ice_num_micro                    cor_mass_mom
vap_mass_mom                     ice_cloud
Sol                              sec
ZS                               PS
```

If the vertical or horizontal resolution is increased without also increasing the model time step, the CFL condition may be violated. If this is the case, errors will typically manifest as unstable temperatures and winds that often become `NAN` just before crashing. Again, check for anomalies in the values of variables listed above, especially in the temperature and wind fields. There are two solutions to this problem:

Solution 1: Increase `n_split` and (possibly) `k_split` (by defining `NNS` and `NKS`).

Solution 2: Decrease the physics time step `dt_atmos` (by defining `DTA`).

Instructions for modifying these parameters are in Chapter 4 Section 4.3. There is no single correct answer to solving problems relating to time stepping. However, there are some implementation best-practices that we share here.

In general, we recommended attempting **Solution 1** and then, if that does not do the trick, **Solution 2**. Beyond that, how you implement the above solutions depends primarily on the resolution of your simulation. For the default horizontal (`c24`) and vertical (56-layer grid) resolution, try increasing only `n_split` first. Increasing `k_split` is generally most useful in cases when the vertical grid has more than 56 layers.

> **NOTE:** Increasing `k_split` is a diffusive process, and it is recommended that this number be as low as possible. There is no theoretical limit to the number of horizontal advection calculations (`n_split`) the model can do, so users should feel free to increase this value until stability is achieved.

Reducing the physics time step `dt_atmos` (**Solution 2**) is only recommended in cases where diabatic heating tendencies are expected to significantly change on timescales shorter than the default setting. Sometimes, running with too large of a time step results in a segmentation fault, which appears in the `fms_mars_out.*` file like so:

```
MPT ERROR: Rank 70(g:70) received signal SIGSEGV(11).
```

When decreasing the time step, best practice is to start by cutting the time step in half. Recall that the time step must evenly multiply into the length of the day. This is why the decision was made to use 88,704 seconds to define the solar day in MGCM 3.1: it allows for a greater number of time step options.

7.3.2 Calling Inactive Modules

The model will exit if flags are toggled to activate physics modules that have not been released. In this case, the model will communicate which physics module it attempted and failed to activate in the `fms_mars_out.*` file. For example, calling the microphysics module in MGCM 3.1 throws the following error:

```
FATAL from PE    56: dust_update_init: The null version of
                   dust_update_init should not be called.
```

7.3.3 BUS Error

When warm-starting the model on the NAS system, occasionally a "BUS" error will arise. This will appear in the `fms_mars_out.*` file like so:

```
MPT ERROR: Rank 23(g:23) received signal SIGBUS(7).
```

This can occasionally indicate that NAS had some internal issue unrelated to the runscript or the model execution, and the model exited as a result. As in the procedure for Model Stability Issues above, check the `fms_mars_out.*` file in the `workdir` for anomalies in the values of variables listed in that section. If no anomalies can be found, which would point to some other issue, it may be the rare case of a NAS issue. In our experience, simply re-submitting the same runscript again with no changes solves the problem.

7.4 Useful Metrics

When diagnosing issues with the model, it is often useful to know what season the simulated atmosphere was in when the model exited. This can be roughly determined by the Sol number when the model exited. This can be found in the last entry of `Sol = XXX` in the `fms_mars_out.*` file. Table 7.1 below shows the approximate sol number and corresponding L_s for a three year simulation.

Table 7.1: Approximate Sol number and corresponding L_s for a 3-year simulation.

L_s	Sol YR 1	Sol YR 2 (+668)	Sol YR 3 (+1336)
0°	0	668	1336
15°	29	697	1365
30°	60	728	1396
45°	93	761	1429
60°	125	793	1461
75°	159	827	1495
90°	192	860	1528
105°	225	893	1561
120°	257	925	1593
135°	287	955	1623
150°	317	985	1653
165°	345	1013	1681
180°	371	1039	1707
195°	397	1065	1733
210°	421	1089	1757
225°	445	1113	1781
240°	468	1136	1804
255°	491	1159	1827
270°	514	1182	1850
285°	538	1206	1874
300°	562	1230	1898
315°	586	1254	1922
330°	612	1280	1948
345°	639	1307	1975

NOTE: Sol can be larger than 668 because it continuously counts the number of Mars days from the beginning of the simulation.

Chapter 8

The Community Analysis Pipeline (CAP)

CAP is a user-friendly, command line data analysis and visualization tool designed to interface with output from the NASA Ames MGCM. At the time of writing, both the NASA Ames Legacy MGCM and MGCM 3.1 are supported in CAP. The purpose of this chapter is to provide an overview of some of the most useful CAP utilities for interacting with MGCM 3.1 output. A full description of CAP, including detailed installation instructions, a user tutorial, and definitions for every executable and function are available on GitHub at https://github.com/NASA-Planetary-Science/AmesCAP.

CAP is Python-based software that standardizes the post-processing effort by providing executables that perform file manipulations and create diagnostic plots from the command line. CAP enables users of almost any skill level to process and plot MGCM data. CAP is designed to be modular so that users are free to selectively integrate CAP's functions into their own analysis routines to the extent they see fit. For example, a user could post-process and plot MGCM output exclusively with CAP or a user could employ their own post-processing routine and then use CAP to plot the data. Brief descriptions of each of the CAP executables are listed below.

- `MarsFiles` provides tools for file manipulations, including code designed to create binned, averaged, and time-shifted files from MGCM output, perform tidal analysis that decompose variables into tidal harmonics, and filter data using a low or high band-pass filter.
- `MarsVars` provides tools relating to variable operations such as adding and removing variables, and performing column integrations. Variable operations include (but are not limited to) density, potential temperature, and mass stream function, to name a few. For more variable options, refer to the tutorial link provided above.
- `MarsInterp` performs a vertical interpolation from the reference (`pfull`) layers to standard pressure (`pstd`), altitude (`zstd`), and altitude-above-ground levels (`zagl`) using a default or user-provided vertical grid.
- `MarsPlot` is a plotting routine that accepts a modifiable template (`Custom.in`) containing a list of plots to create. It is useful for creating plots from MGCM output quickly and is designed specifically for use with the MGCM output files.

The plotting routine, `MarsPlot`, has several useful functions including averaging (e.g., zonal, global values), data reduction (e.g., specific altitude or latitude range), comparison between simulations (difference plots), simple variable operations (such as scaling), and data selection across multiple files. The `Custom.in` file provides templates for all of the typical meteorological 1D and 2D plots (e.g., time series, longitude/latitude cross-sections, and vertical profiles). It additionally provides some degree of flexibility, such as adjusting axes, color scales, or building multi-panel plots, but it is not intended to be fully customizable.

8.1 Plotting with CAP

CAP was used to create the figures from the default simulation shown in Chapter 6. Here we provide instructions specifically for using CAP to re-create those figures using output from your own simulations. These instructions assume CAP is installed on the system hosting MGCM 3.1 output files. See the installation instructions on GitHub if necessary.

The general process for plotting with `MarsPlot` is illustrated in Figure 8.1 and can be summarized as follows:

1. Open and modify `Custom.in` using a text editor
2. Pass `Custom.in` to `MarsPlot` to generate plots
3. View the plots in `Diagnostic.pdf`

When defining new plots in `Custom.in`, we recommend copy-pasting (or deleting) the example code blocks provided in the template. Be sure set any copied code blocks to `TRUE` so that `MarsPlot` knows to include the plot in the final PDF. If the name of the `Custom.in` file is not changed, the resulting PDF will be called `Diagnostic.pdf`. If the name of `Custom.in` is changed, then the name of the PDF will match the renamed `Custom.in` file. For example, renaming the template to `MyPlots.in` creates `MyPlots.pdf`.

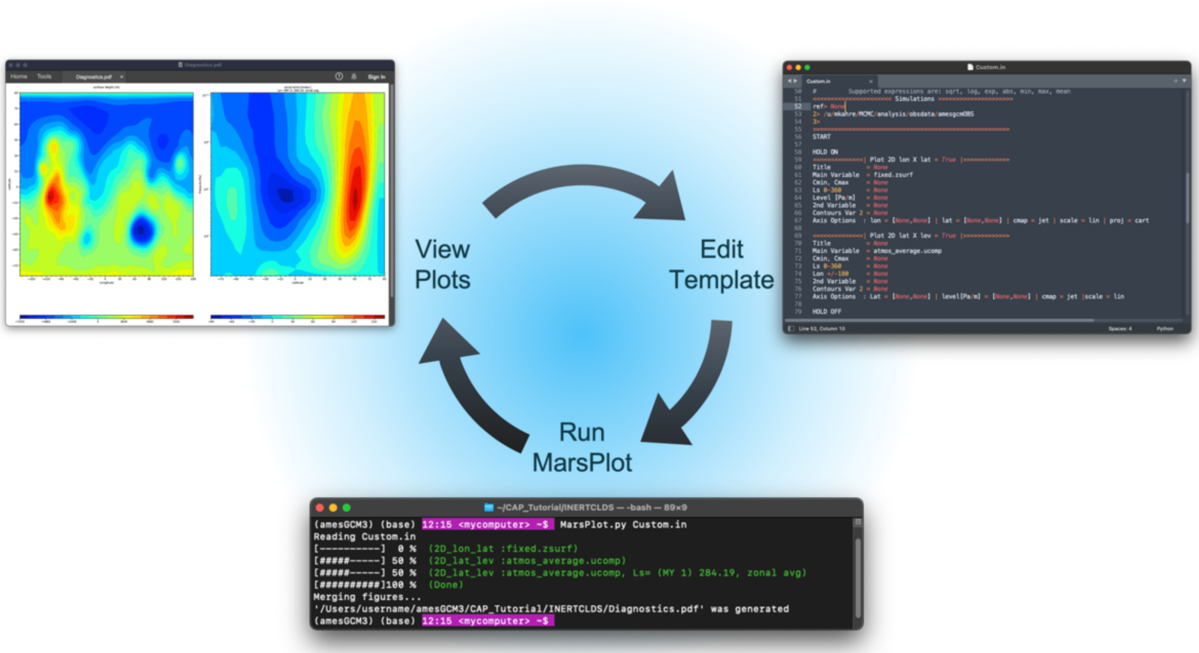

Figure 8.1: The `MarsPlot` cycle. Edit Template: Desired plots are defined in `Custom.in`; Run `MarsPlot`: `Custom.in` is passed into `MarsPlot` via the command line; View Plots: Plots are created in `Diagnostics.pdf`.

The plots shown in Chapter 6 were created using `Model_Release_Diagnostics.in`, which is included in the MGCM source code at:

```
AM4/src/AmesGCM/diagnostics/Model_Release_Diagnostics.in
```

To use `Model_Release_Diagnostics.in`, make a copy of the file and move it to your output directory:

```
user@lfe:~$ scp -r USER@pfe:AM4/src/AmesGCM/diagnostics/
↪  Model_Release_Diagnostics.in .
```

> **NOTE:** The secure copy command above contains NAS-specific syntax. Modify as necessary for non-NAS environments.

To make the plots in `Model_Release_Diagnostics.in` on NAS, you have to first install CAP. The NAS-specific installation is explained on the CAP GitHub at https://github.com/NASA-Planetary-Science/AmesCAP/blob/master/docs/NAS_INSTRUCTIONS.txt.

Once you have installed CAP, initiate the CAP virtual environment. The syntax for this could vary slightly depending on your package manager (`pip` or `conda`) and terminal shell (e.g. `bash`, `csh`, `tcsh`), but is typically performed with:

```
source ~/AmesCAP/bin/activate.csh
```

Then, pass `Model_Release_Diagnostics.in` to `MarsPlot` to create the figures:

```
MarsPlot.py Model_Release_Diagnostics.in
```

All of the figures will appear in a single PDF called `Model_Release_Diagnostics.pdf` in your current directory. You can use `Model_Release_Diagnostics.in` to verify that the default MGCM simulation performs as expected by comparing your plots to the ones provided in this User Guide. You may also find it useful to use `Model_Release_Diagnostics.in` for other simulations.

Appendix A

Running MGCM 3.1 on Mac OS

Installing and running MGCM 3.1 is currently supported on NAS. Building and running the model has also been explored using GNU libraries on an 8 core MacBook Pro using Mac OS. We cannot guarantee that the model can be used easily on other platforms, as this depends on the availability of necessary libraries and how easy it is to link to them, but we can provide instructions for installing and running the MGCM on Mac OS. A compile script and GNU template specific to Mac OS environments are provided in `AmesGCM/build_run/` to get you started.

A.1 Minimum Requirements

As outlined in Section 3.1, the minimum requirements for building MGCM 3.1 are:

- **A Fortran Compiler** – As stated in Section 3.1, the preferred compiler for MGCM 3.1 is the Intel Fortran Compiler (ifort). However, a GNU compiler (GCC or gfortran) can suffice, but they require GCC version 9 or later for some flags.
- **netCDF4** – MGCM 3.1 input and output files are in netCDF format. This requires a NetCDF4 library (netcdf.mod) with MPI and Fortran compatibility as well as has HDF5 libraries.
- **An MPI Distribution** – Installation of an MPI library is required for parallel processing. MGCM 3.1 requires at least six processors to run (one processor per tile) and an MPI distribution. Acceptable MPIs include Open MPI, HPE (MPI), MPICH, among others, as long as the distribution has binaries (mpi.mod) created using the compiler that will be used for compiling MGCM 3.1.
- **Git** – MGCM 3.1 requires Git version 1.8.4 or later for installation.
- **Mars Physics Packages** – Instructions for cloning the Mars physics packages are in Section 3.2.
- **The NOAA-GFDL AM4 Model** – Instructions for cloning the AM4 repository are in Section 3.2.

A.2 Installing Dependencies

The installation instructions for the GNU Compiler, MPI Distribution, and netCDF libraries use Macports. Macports is a great tool for downloading and installing libraries on Mac OS and it is available at https://www.macports.org/install.php. Follow the installation instructions for Macports there before moving on.

Install the GNU Compiler GCC

We recommend installing GCC as your GNU Compiler. GCC is available on OSX from Apple Developer Tools (https://developer.apple.com/download/all/?q=developer%20tools), but we recommend using Macports to install it. Install GCC from the command line with:

```
user@local:~$ sudo port install gcc
```

Install the MPI Distribution OpenMPI

OpenMPI is our preferred MPI Distribution for Mac OS. To install OpenMPI using Macports:

```
user@local:~$ sudo port install openmpi-gcc11
```

Install netCDF4 and HDF5

Finally, install netCDF and its dependencies using Macports:

```
user@local:~$ sudo port install netcdf
user@local:~$ sudo port install netcdf-cxx
user@local:~$ sudo port install netcdf-fortran
```

NOTE: At this point, you should install MGCM 3.1 according to the instructions in Section 3.2 if you have not already done so.

A.3 Compile MGCM 3.1 on Mac OS

Now that the environment for compiling and running MGCM 3.1 has been set up, settings internal to MGCM 3.1 that enable the model to run on Mac OS with the GCC compiler can be configured. To set up the model to run on Mac OS, copy the GNU template file and the compile script to the appropriate directories and modify the runscript as described below.

Copy over the GNU Template File

Begin by copying the GNU template file for Mac OS (`mkmf.template.osx`) to `AM4/bin/`:

```
user@local:~$ cp ~/AmesGCM/build_run/bin/mkmf.template.osx ~/AM4/bin/.
```

The GNU template file looks like this:

```
# /AM4/bin/mkmf.template.osx
CPPFLAGS = -I$(NETCDF)/include -I$(MPI_INC)

MARS_CPP = -DSPMD -Duse_libMPI -Duse_netCDF -DINTERNAL_FILE_NML -DCUBE_CORE

FFLAGS = $(CPPFLAGS) $(MARS_CPP) -w -fdefault-real-8 -O2 -c -fopenmp -
↪  fbacktrace -ffree-line-length-none fcray-pointer -fno-range-check -
↪  fdefault-double-8 -Waliasing -fallow-invalid-boz -fallow-argument-
↪  mismatch

CC = $(GCC_PATH)/mpicc
FC = $(GCC_PATH)/mpif90
LD = $(GCC_PATH)/mpif90
AR = ar cr
LDFLAGS = -L$(NETCDF)/lib -L$(MPI_LIB) -lnetcdff -lnetcdf -fopenmp -lmpi
ARFLAGS =
CFLAGS = -D__GFORTRAN $(CPPFLAGS) $(MARS_CPP) $(FFLAGS)
```

NOTE: Part of the `FFLAGS` line is only available in GCC version 9 or newer: `-fallow-invalid-boz -fallow-argument-mismatch`.

These settings should not need adjusting for running on Mac OS, however, we have encountered some issues when compiling the model using GNU compilers on NAS that might be relevant for troubleshooting issues on Mac OS. We outline three of those issues here.

First, the GCM references some Fortran-compiled versions of the MPI and NetCDF libraries (i.e., mpi.mod or netcdf.mod). These must be compiled with gfortran in order to be compatible with the the gfortran-compiled version of MGCM 3.1. On NAS, the default versions of the MPI and NetCDF modules were compiled with ifort and therefore are not compatible with MGCM 3.1 if compiled with gfortran.

Second, these compile options set under `FFLAGS` are confirmed to work for GNU compilers on NAS and should also work for Mac OS:

```
-w -fdefault-real-8 -g -O1 -c -fopenmp -fbacktrace -ffree-line-length-none -
↪  fcray-pointer -fno-range-check -fdefault-double-8 -Waliasing -fallow-
↪  invalid-boz -fallow-argument-mismatch
```

Finally, the lines `AR = ar cr` and `ARFLAGS =` are flags that allow the model to create archives (or mini libraries of compiled code) that the compiler can link to when generating the full executable. To avoid creating archives, set the `-DMARS_GDIAGS` flag in `compile.osx` as described below.

Copy over the Mac OS Compile Script

In order to compile MGCM 3.1 on Mac OS, the Mac-specific compile script `compile.osx` must be copied over to `AM4/exec/` from `AmesGCM/build_run/`:

```
user@local:~$ cp ~/AmesGCM/build_run/compile.osx ~/AM4/exec/.
```

The paths to the netCDF, GCC, and MPI libraries are set in the compile script. If you downloaded GCC from Apple Developer Tools instead of Macports, you need to comment out the line for Macports and un-comment the line for developer tools:

```
# AM4/exec/compile.osx
setenv NETCDF "/opt/local"
setenv NETCDFPATH "/opt/local"
setenv GCC_PATH "/usr/bin"          # if downloaded with developer tools
# setenv GCC_PATH "/opt/local/bin"  # if downloaded with macports
setenv MPI_INC "/opt/local/include/openmpi-gcc11"
setenv MPI_LIB "/opt/local/lib/openmpi-gcc11"
```

If you used Macports for all of the installations as recommended, then you do not neeed to change any settings in the compile script. To compile the code, run the executable from the `AM4/exec/` directory:

```
user@pfe:~$ ./compile.osx
```

The compilation takes ~ 5 minutes. When it is complete, the compiled version of MGCM 3.1 will be stored under `exec.osx.mars3.1`.

Modify the Runscript

There a few modifications that need to be made to the default runscript `fms_mars_default` in order for MGCM 3.1 to run successfully on Mac OS. First, comment out or delete lines 610–622, which appear in `fms_mars_default` as:

```
end

#------ Backup output to Lou (NAS only)
pbs_release_nodes -j $PBS_JOBID -a
ssh -q pfe "ssh -q lou mkdir -p $ASCII_DIR"
ssh -q pfe "ssh -q lou scp -q $workdir/ascii/*  $ASCII_DIR/."
ssh -q pfe "ssh -q lou mkdir -p $RESTART_DIR"
ssh -q pfe "ssh -q lou scp -q $workdir/restart/*.restart.tar  $RESTART_DIR/."
ssh -q pfe "ssh -q lou mkdir -p $HIST_DIR"
ssh -q pfe "ssh -q lou scp -q $workdir/history/*.nc.tar  $HIST_DIR/."
ssh -q pfe "ssh -q lou scp -q $homedir/$name  $ASCII_DIR/."

exit 0
```

These settings are NAS-specific archival specifications that are not needed for running on Mac OS. The PBS settings at the top of the runscript are ignored, so it is not necessary to delete them.

Next, set the directory you wish to run MGCM 3.1 in under `workdir`. The model will output files to this directory as well.

```
set workdir = /$USER/FMS_MARS_runs/$classdir/$name
```

Finally, specify Mac OS as the platform and set the layout of the processors to 1 by 1:

```
set platform = osx
set TILE_LAYOUT = 1,1
```

Those are the minimal settings that need modification to run on Mac OS. For more customization options, see See Section 4.2.

A.4 Run MGCM 3.1 on Mac OS

To run MGCM 3.1 on Mac OS, simply execute `fms_mars_default` from `AM4/exec/`:

```
user@pfe:~$ ./fms_mars_default
```

That's it! The model will run in the directory you specified under `workdir`. The output files will be located in a subdirectory called `history/` and restart files will be located in `restart/`.

> **NOTE:** To run the model in the background, add an ampersand (&) to the end of the execution line (i.e., `./fms_mars_default &`.

Appendix B

Dust Namelist Parameters

The following pages provide a list of namelist options for the dust scenarios that appear in or can be added to the runscript. Their default values are listed alongside descriptions of the parameter and its dependencies.

Table B.1: Dust Namelist Parameters

Module	Variable	Default Value	Description
`aerosol_util_nml`	`do_15band`	`T`	Flag: Do 15-band aerosol optics. Must match `ames_15band`.
`aerosol_util_nml`	`do_moment_dust`	`T`	Flag: Do moment dust lifting. F = no moment dust.
`aerosol_util_nml`	`dust_map_scale`	`3.67`	Infrared to Visible scaling factor for inputted dust map.
`aerosol_util_nml`	`reff_backgd`	`2.50e-06`	Effective radius for transported dust lifted to map [m].
`aerosol_util_nml`	`reff_dd`	`2.50e-06`	Effective radius for transported dust lifted by dust devils [m].
`aerosol_util_nml`	`reff_fixed`	`2.50e-06`	Effective radius of the dust for RT code [m].
`aerosol_util_nml`	`reff_ws`	`2.50e-06`	Effective radius for transported dust lifted by wind stress [m].
`aerosol_nml`	`conrath`	`0.003`	Value of the Conrath parameter.
`aerosol_nml`	`conrath_type`	`1`	Flag: Conrath vertical extent. 0 = Conventional Conrath top, 1 = Conrath with scaled ztop.
`aerosol_nml`	`do_inpt_dust_cycle`	`T`	Flag: Read the dust scenario from an input file. Skipped if `optical_depth_inpt` $\leq$0. If `optical_depth_inpt` is nonzero, MGCM reads the dust scenario file + constant Tau.
`aerosol_nml`	`optical_depth_inpt`	`0.0`	Constant Tau for the dust. If `background` = T, then this dust is added on top of the dust scenario map.

Table B.2: Dust Namelist Parameters Continued

Module	Variable	Default Value	Description
`dust_update_nml`	`alfa`	`0.0065`	Wind stress tuning parameter.
`dust_update_nml`	`alpha_dda`	`1.20e-10`	Dust devil tuning parameter.
`dust_update_nml`	`assim_t_thresh`	`215.0`	Surface temperature $<$ threshold [K], no background constant or map-tracked dust lifting.
`dust_update_nml`	`background`	`T`	Flag: Tracer field reference background dust scenario file.
`dust_update_nml`	`dd_co2_ice_thresh_n`	`1`	If CO_2 ice depth $>$ threshold [m], no interactive dust devil lifting in the northern hemisphere.
`dust_update_nml`	`dd_co2_ice_thresh_s`	`1`	If CO_2 ice depth $>$ threshold [m], no interactive dust devil lifting in the southern hemisphere.
`dust_update_nml`	`dd_h2o_ice_thresh`	`1`	If H_2O ice depth $>$ threshold [m], no interactive dust devil lifting.
`dust_update_nml`	`dda`	`T`	Flag: Dust devil (convective) lifting. Requires `interact` = T. Informs tracer field, informs RT if `radactive_dust_tog` = 1.
`dust_update_nml`	`dgdm_type`	`1`	Flag: Type of daily global dust map for assimilation. 1 = visible, 2 = infrared.
`dust_update_nml`	`interact`	`F`	Flag: Lift dust interactively. Requires `interact` = T. Informs tracer field, informs RT if `radactive_dust_tog` = 1.
`dust_update_nml`	`kfix`	`0,0`	Choice of levels where dust is injected. 0,0 by default, or PBL injection.

Table B.3: Dust Namelist Parameters Continued

Module	Variable	Default Value	Description
dust_update_nml	no_assim_over_caps	T	Flag: T = do not use assimilation over the ice caps.
dust_update_nml	opac_from_aerosol	F	Flag: Background dust lifting to match prescribed dust scenario with constant opacity (Tau). Req. do_inpt_dust_cycle = F, optical_depth_inpt $\neq 0$.
dust_update_nml	optd_thresh	100	Stop lifting if dust column opacity > threshold (only for interactive threshold wind stress lifting scheme).
dust_update_nml	sink_bd	T	Flag: Forcibly remove dust to match input dust scenario map.
dust_update_nml	sinkscale	1	Scale the sink with the dust scenario by this factor.
dust_update_nml	stress_lift	T	Flag: Use wind stress lifting.
dust_update_nml	tauscale	1.0	Scale the background constant or map-tracked dust from the scenario by this factor.
dust_update_nml	threshold_stress	0.022	Threshold wind stress for lifting [N/m2].
dust_update_nml	ws_co2_ice_thresh_n	1	If CO_2 ice depth > threshold [m], no wind stress lifting in the northern hemisphere.
dust_update_nml	ws_co2_ice_thresh_s	1	If CO_2 ice depth > threshold [m], no wind stress lifting in the southern hemisphere.
dust_update_nml	ws_h2o_ice_thresh	1	If H_2O ice depth > threshold [m], no wind stress lifting.
ames_rtmod_nml	radactive_dust_tog	3	Flag: Toggle for radiatively active dust. 0 = clear, 1 = transported moment dust scheme (defined by dust_update_nml flags; ignores aersosol_nml flags), 2 = not supported, 3 = prescribed fixed (defined by aerosol_nml flags). Prescribed dust informs RT, transported dust is inert.
ames_rtmod_nml	ames_15band	T	Flag: Use 15-band RT. F = use 12-band RT.

References

Basu, S., Richardson, M. I., & Wilson, R. J. (2004). Simulation of the Martian dust cycle with the GFDL Mars GCM. *J. Geophys. Res.*, *109*(E11006). doi: 10.1029/2004JE002243

Basu, S., Wilson, R. J., Richardson, M. I., & Ingersoll, A. P. (2006). Simulation of spontaneous and variable global dust storms with the GFDL Mars GCM. *J. Geophys. Res.*, *111*(E09004). doi: 10.1029/2005JE002660

Bertrand, T., Wilson, R. J., Kahre, M. A., Urata, R., & Kling, A. (2020). Simulation of the 2018 global dust storm on Mars using the NASA Ames Mars GCM: A multi-tracer approach. *Journal of Geophysical Research*, *125*(7). doi: 10.1029/2019JE006122

Boynton, W. V., Feldman, W. C., Squyres, S. W., Prettyman, T. T., Brückner, J., Evans, L. G. ., & Shinohara, C. (2002). Distribution of hydrogen in the near surface of Mars: Evidence for subsurface ice deposits. *Science*, *297*(5578), 81-85. doi: 10.1126/science.1073722

Collins, M., Lewis, S. R., & Read, P. L. (1997). Gravity wave drag in a global circulation model of the martian atmosphere: Parameterisation and validation. *Advances in Space Research*, *19*. doi: 10.1016/S0273-1177(97)00277-9

Conrath, B. J. (1975). Thermal structure of the Martian atmosphere during the dissipation of the dust storm of 1971. *Icarus*, *24*, 36-46. doi: 10.1016/0019-1035(75)90156-6

Eckermann, S. D., Ma, J., & Zhu, X. (2011). Scale-dependent infrared radiative damping rates on mars and their role in the deposition of gravity-wave momentum flux. *Icarus*, *211*, 429-442. doi: 10.1016/j.icarus.2010.10.029

Forget, F., Hourdin, F., Fournier, R., Hourdin, C., Talagrand, O., Collins, M., … Huot., J.-P. (1999). Improved general circulation models of the martian atmosphere from the surface to above 80 km. *Journal of Geophysical Research*, *104*, 24155-24175. doi: 10.1029/1999JE001025

Forget, F., Wordsworth, R., Millour, E., Madeleine, J. B., Kerber, L., Leconte, J., … Haberle, R. M. (2013). 3d modelling of the early martian climate under a denser CO2 atmosphere: Temperatures and CO2 ice clouds. *Icarus*, *222*(1). doi: 10.1016/j.icarus.2012.10.019

Gordon, I., Rothman, L., Hargreaves, R., Hashemi, R., Karlovets, E., Skinner, F., & ... Yurchenko, S. (2022). The HITRAN2020 molecular spectroscopic database. *Journal of Quantitative Spectroscopy and Radiative Transfer*, *277*. doi: 10.1016/j.jqsrt.2021.107949

Gough, D. O. (1981). Solar interior structure and luminosity variations. *Solar Physics*, *74*, 21-34. doi: 10.1007/BF00151270

Haberle, R. M., Forget, F., Colaprete, A., Schaeffer, J., Boynton, W. V., Kelly, N. J., & Chamberlain, M. A. (2008). The effect of ground ice on the martian seasonal co_2 cycle. *Planetary and Space Science*, *56*, 251-255. doi: 10.1016/j.pss.2007.08.006

Haberle, R. M., Kahre, M. A., Hollingsworth, J. L., Montmessin, F., Wilson, R. J., Urata, R. A.,

… Schaeffer, R., James (2019). Documentation of the NASA/Ames Legacy Mars Global Climate Model: Simulations of the present seasonal water cycle. *Icarus*, *333*, 130-164. doi: 10.1016/j.icarus.2019.03.026

Haberle, R. M., Murphy, J. R., & Schaeffer, J. (2003). Orbital change experiments with a Mars general circulation model. *Icarus*, *161*(1), 66-89. doi: 10.1016/S0019-1035(02)00017-9

Harris, L., Chen, X., Putman, W., Zhou, L., & Chen, J.-H. (2021). A scientific description of the GFDL finite-volume cubed-sphere dynamical core. *NOAA technical memorandum OAR GFDL*. doi: 10.25923/6nhs-5897

Kahre, M. A., Hollingsworth, J. L., Haberle, R. M., & Murphy, J. R. (2008). Investigations of the variability of dust particle sizes in the martian atmosphere using the nasa ames general circulation model. *Icarus*, *195*(2). doi: 10.1016/j.icarus.2008.01.023

Kahre, M. A., Hollingsworth, J. L., Haberle, R. M., & Wilson, R. J. (2015). Coupling the Mars dust and water cycles: The importance of radiative-dynamic feedbacks during northern hemisphere summer. *Icarus*, *260*, 477-480. doi: 10.1016/j.icarus.2014.07.017

Kahre, M. A., Murphy, J. R., & Haberle, R. M. (2006). Modeling the Martian dust cycle and surface dust reservoirs with the NASA Ames General Circulation Model. *Journal of Geophysical Research*, *111*(E06). doi: 10.1029/2005JE002588

Kahre, M. A., Murphy, J. R., Haberle, R. M., Montmessin, F., & Schaeffer, J. (2005). Simulating the Martian dust cycle with a finite surface dust reservoir. *Geophysical Research Letters*, *32*(20). doi: 10.1029/2005GL023495

Kahre, M. A., Wilson, R. J., Haberle, R. M., Harman, C. E., Urata, R. A., Kling, A., … Bertrand, T. (2022, June). Update and status of the Mars Climate Modeling Center at NASA Ames Research Center. In *Seventh international workshop on the mars atmosphere: Modelling and observations.* Paris, FR. doi: 2022mamo.conf.1101K

Lin, S.-J. (1997). A finite-volume integration method for computing pressure gradient force in general vertical coordinates. *Quarterly Journal of the Royal Meteorological Society*, *123*, 1749-1762. doi: 10.1002/qj.49712354214

Lin, S.-J. (2004). A "vertically lagrangian" finite-volume dynamical core for global models. *Monthly Weather Review*, *132*, 2293-2307. doi: 10.1175/1520-0493(2004)132⟨2293:AVLFDC⟩2.0.CO;2

Lin, S.-J., & Putman, W. M. (2007). Finite-volume transort on various cubed-sphere grids. *Journal of Computational Physics*, *227*, 55-78. doi: 10.1016/j.jcp.2007.07.022

Lin, S.-J., & Rood, B. R. (1996). Multidimensional flux-form semi-lagrangian transport schemes. *Monthly Weather Review*, *124*, 2046-2070. doi: 10.1175/1520-0493(1996)124⟨2046:MFFSLT⟩2.0.CO;2

Lin, S.-J., & Rood, B. R. (1997). An explicit flux-form semi-lagrangian shallow-water model on the sphere. *Quarterly Journal of the Royal Meteorological Society*, *123*, 2477-2498. doi: 10.1002/qj.49712354416

Mellor, G. L., & Yamada, Y. (1982). Development of a turbulence closure model for geophysical fluid problems. *Reviews of Geophysics*, *20*(4), 851-875. doi: 10.1029/RG020i004p00851

Montabone, L., Forgét, F., Millour, E., Wilson, R. J., Lewis, S. R., Cantor, B., … Wolff, M. J. (2015). Eight-year climatology of dust optical depth on Mars. *Icarus*, *251*, 65-95. Retrieved from http://www-mars.lmd.jussieu.fr/mars/dust_climatology/ doi: 10.1016/j.icarus.2014.12.034

Montmessin, F., Forget, F., Rannou, P., Cabane, M., & Haberle, R. M. (2004). Origin and role of water ice clouds in the martian water cycle as inferred from a general circulation model. *Journal of Geophysical Research: Planets*, *109*. doi: 10.1029/2004JE002284

Newman, C. E., Lewis, S. R., Read, P. L., & Forget, F. (2002a). Modeling the Martian dust cycle, 1. Representations of dust transport processes. *Journal of Geophysical Research: Planets*, *107*(E12). doi: 10.1029/2002JE001910

Newman, C. E., Lewis, S. R., Read, P. L., & Forget, F. (2002b). Modeling the Martian dust cycle 2. Multiannual radiatively active dust transport simulations. *Journal of Geophysical Research: Planets*, *107*(E12). doi: 10.1029/2002JE001920

Palmer, T. N., Shutts, G. J., & Swinbank, R. (1986). Alleviation of a systematic westerly bias in general circulation and numerical weather prediction models through an orographic gravity wave drag parametrization. *Quarterly Journal of the American Meteorological Society*, *112*, 1001-1039. doi: 10.1002/qj.49711247406

Perrin, M. Y., & Hartmann, J. M. (1989). Temperature-dependent measurements and modeling of absorption by CO2-N2 mixtures in the far line-wings of the 4.3µm CO2 band. *Journal of Quantitative Spectroscopy and Radiative Transfer*, *42*(4), 311-317. doi: 10.1016/0022-4073(89)90077-0

Putzig, N. E., & Mellon, M. T. (2007). Apparent thermal inertia and the surface heterogeneity of Mars. *Icarus*, *191*, 68-94. doi: 10.1016/j.icarus.2007.05.013

Renno, N. O., Burkett, M. L., & Larkin, M. P. (1998). A simple thermodynamical theory for dust devils. *Journal of the Atmospheric Sciences*, *55*(21), 3244-3252. doi: 10.1175/1520-0469(1998)055⟨3244:ASTTFD⟩2.0.CO;2

Rothman, L., Gordon, I., Barber, R., Dothe, H., Gamache, R., Goldman, A., … Tennyson, J. (2010). HITEMP, the high-temperature molecular spectroscopic database. *Journal of Quantitative Spectroscopy and Radiative Transfer*, *111*, 2139-2150. doi: 10.1016/j.jqsrt.2010.05.001

Rothman, L., Gordon, L., Babikov, Y., Barbe, A., Benner, D. C., Bernath, P., & ... Wagner, G. (2013). The HITRAN2012 molecular spectroscopic database. *Journal of Quantitative Spectroscopy and Radiative Transfer*, *130*. doi: 10.1016/j.jqsrt.2013.07.002

Schwenke, D. (1998). New H2O rovibrational line assignments. *Journal of Molecular Spectroscopy*, *190*. doi: 10.1006/jmsp.1998.7603

Simmons, A. J., & Burridge, D. M. (1981). An energy and angular-momentum conserving vertical finite-difference scheme and hybrid vertical coordinates. *Monthly Weather Review*, *109*, 758-766. doi: 10.1175/1520-0493(1981)109⟨0758:AEAAMC⟩2.0.CO;2

Smith, D. E., Zuber, M. T., Solomon, S. C., Phillips, R. J., Head, J. W., Garvin, J. B., … Duxbury, T. C. (1999). the global topography of Mars and implications for surface evolution. *Science*, *284*(5419), 1495-1503. doi: 10.1126/science.284.5419.1495

Steakley, K., Murphy, J., Kahre, M., Haberle, R. M., & Kling, A. (2019). Testing the impact heating hypothesis for early Mars with a 3-D global climate model. *Icarus*, *330*, 169-188. doi: 10.1016/j.icarus.2019.04.005

Steakley, K. E., Kahre, M. A., Haberle, R. M., & Zahnle, K. J. (2023). Impact induced H2-rich climates on early Mars explored with a global climate model. *Icarus*, *394*. doi: 10.1016/j.icarus.2022.115401

Toon, O. B., Mckay, C. P., Ackerman, T. P., & Santhanam, K. (1989). Rapid calculation of radiative heating rates and photodissociation rates in inhomogenous multiple scattering atmospheres. *Journal of Geophysical Research*, *94*(D13), 16287-16301. doi: 10.1029/JD094iD13p16287

Turbet, M., Boulet, C., & Karman, T. (2020). Measurements and semi-empirical calculations of CO2+CH4 and CO2+H2 collision-induced absorption across a wide range of wavelengths and temperatures. application for the prediction of early Mars surface temperature. *Icarus*, *346*. doi: 10.1016/j.icarus.2020.113762

Tyler, D., & Barnes, J. R. (2014). Atmospheric mesoscale modeling of water and clouds during northern summer on Mars. *Icarus*, *237*, 388-414. doi: 10.1016/j.icarus.2014.04.020

Westphal, D. L., Toon, O. B., & Carlson, T. N. (1987). A two-dimensional numerical investigation of the dynamics and microphysics of Saharan dust storms. *Journal of Geophysical Research: Atmospheres*, *92*(D3), 3027-3049. doi: 10.1029/JD092iD03p03027

Wilson, R. J., Neumann, G. A., & Smith, M. D. (2007). Diurnal variation and radiative influence of Martian water ice clouds. *Geophysical Research Letters*, *34*(2). doi: 10.1029/2006GL027976

Wolff, M. J., Smith, M. D., Clancy, T. R., Arvidson, R., Kahre, M. A., Seelos IV, F., ... Savijärvi, H. (2009). Wavelength dependence of dust aerosol single scattering albedo as observed by the Compact Reconnaissance Imaging Spectrometer. *Journal of Geophysical Research: Planets*, *114*. doi: 10.1029/2009JE003350

Wordsworth, R., Forget, F., & Eymet, V. (2010). Infrared collision-induced and far-line absorption in dense CO2 atmospheres. *Icarus*, *210*(2), 992-997. doi: 10.1016/j.icarus.2010.06.010