Kamodo：简化模型数据的访问与利用

Rebecca Ringuette[1,2], Lutz Rastaetter[2], Darren De Zeeuw[2,3], Asher Pembroke[4], 和 Oliver Gerland IV[4]。[1]*ADNET Systems Inc.*，美国马里兰州贝塞斯达市 *Rockledge Dr. 6720B* 号 *504* 室，邮编 *20817*，[2]美国国家航空航天局戈达德太空飞行中心社区协调建模中心，美国马里兰州格林贝尔特，邮编 *20771*，[3]美国天主教大学，美国华盛顿特区 *Michigan Ave. N.E. 620* 号，邮编 *20064*，[4]*Ensemble Government Services LLC*，美国马里兰州海亚茨维尔市 *Buchanan St. 4005* 号，邮编 *20781*。

摘要

为解决日球物理学领域缺乏易用软件、难以简化模型数据利用的问题，美国国家航空航天局戈达德太空飞行中心的社区协调建模中心（CCMC）开发了基于 Kamodo 的模型无关方法，让用户能够在工作流程中轻松访问和利用模型数据。该方法通过抽象处理繁多的文件格式以及专用网格插值的复杂性，大幅降低了社区访问和利用模型数据的门槛，同时为用户的工具箱增添了令人振奋的新功能。本文介绍模型数据的直接接口——模型读取器，并简要说明其用法。此外，本文还详细介绍了纳入自定义插值代码的计划方案，并介绍专用可视化开发的当前进展。CCMC 正在将 Kamodo 作为美国国家航空航天局官方开源软件进行维护，以促进并鼓励社区协作。

第 *1* 节：引言

目前，能够让用户在无需深入了解模型数据格式和数据插值细节的情况下使用模型数据的日球物理学软件资源并不多。用户通常只能通过文献中针对少数模型编写的自定义代码来访问这些模型，但这些代码通常不公开，使用各种编程语言编写，且语法往往不统一。一个值得一提的例子是 CCMC 的可视化与分析代码库。CCMC 维护着一个包含多种语言代码的大型代码库，可提供通用的插值和可视化功能，这些功能可通过 CCMC 网站及其上的应用程序使用。该代码库包含一些针对特定模型的自定义可视化功能，与第4节中介绍的类似。然而，该代码库并未公开，其中部分代码依赖专有软件许可证。这些代码也无法直接提供对模型数据的功能性访问。Kameleon[1], 是 CCMC 先前开发的一款模型访问代码，曾尝试解决这些问题。不过，Kameleon 软件主要以 C语言和部分 Python 编写，因此一些用户使用起来较为不便，贡献者也难以为其添加对新模型和新功能的支持。

[1] https://ccmc.gsfc.nasa.gov/tools/kameleon/

另一个例子是 SpacePy[2], ，这是一个以 Python 编写的开源软件包，可访问 SWMF/BATSRUS 模型的输出数据，但不易扩展以支持其他模型输出（Morley 等，2011；Toth 等，2007）。PYSAT[3] 软件包是一个例外，目前可有限访问电离层领域中两个模型的输出，其中包括基本的穿越探测能力[4] （Stoneback 等，2018）。CCMC 团队正与 pysat 开发者合作，通过连接 Kamodo 来扩展其功能，并已实现与 pysat 及所有 PyHC 核心软件包的互操作性（日球物理学Python社区：Barnum 等，2022；Polson 等人（2022）；Ringuette 等人（2022））。Kamodo 现已成为这些核心软件包之一。

为了解决模型数据的访问和使用难题，我们正在开发一种不依赖特定模型的方法，让用户能够使用 Kamodo 与 CCMC 托管的所有模型数据进行交互。Kamodo 是专为应对这一挑战而设计的独特 Python 软件包（Pembroke 等，2022年）。最终产品将提供简单的访问和使用函数调用，支持多个领域的日球物理学模型数据，包括 CCMC 托管的热层、电离层、磁层、日球层和太阳物理学模型的输出。这些功能包括通过模型读取器直接与模型数据交互、卫星穿越探测功能，以及其他软件工具；无论选择何种模型数据，这些工具都能使用相同或几乎相同的函数调用。

此处讨论的所有功能目前均可通过美国国家航空航天局的 Kamodo GitHub 代码库获取，代码采用开源许可[6]。第二个链接 Kamodo-core 提供 Kamodo 的核心功能，包括函数注册、组合、单位转换、自动绘图以及远程过程调用（RPC）基础设施。第一个链接托管了专用于空间天气的模型和数据读取器、卫星穿越探测功能、定制可视化函数，以及为满足 CCMC 用户需求而设计的多种其他功能。Kamodo 的这一部分依赖 Kamodo-core 库，并对其中定义的 Kamodo 类进行子类化。Kamodo-core 可通过 pip[7] 安装。安装完整的 Kamodo 软件还需执行其他步骤，具体说明见下方链接的 NASA GitHub 仓库（脚注 6 中的第一个链接）。

本文重点介绍模型数据的直接接口，即模型读取器，以及其功能，并将其与模型数据所缺少的基础设施进行比较。随着模型库不断扩充，部分模型读取器将需要专用的插值和坐标转换代码。因此，我们还将说明如何在现有结构中集成此类代码，同时遵循我们保持语法与模型无关的标准。对

[2] https://spacepy.github.io/[3] https://pysat.readthedocs.io/en/latest/index.html[4] https://pysatmodels.readthedocs.io/en/latest/index.html[5] https://heliopython.org/，https://ror.org/012prn105[6] https://github.com/nasa/Kamodo 和 https://github.com/nasa/kamodo-core[7] https://pypi.org/project/kamodo/

模型数据集的分析也至关重要地依赖于可用的可视化功能，而这些功能又需要专门开发高级可视化。我们列举了几个为此目的开发的高级可视化示例。虽然我们会在本文总结中简要介绍与模型无关的穿越分析及其他功能，但对于这些正在开发的功能，我们将留待未来工作中进一步深入探讨。

我们将在第2节介绍 Kamodo。随后，第3节将概述模型读取器的用法和结构，其中包括一个专门介绍自定义插值的小节。第4节介绍标准可视化功能，并展示目前提供的若干自定义可视化。第5节将展示模型读取器的跨模型能力，说明如何通过简单的语法，轻松绘制并比较多个模型中的同一变量。最后，我们将在第6节总结本文的工作。

第*2*节：什么是 *Kamodo*？

Kamodo 是一款开源软件，最初由 CCMC 开发，旨在专门解决如何简化对托管于该中心的日球物理学模型数据的访问与使用这一复杂问题。虽然模型数据的访问与使用本身就是一个复杂的问题，完全值得为此开发独立的软件包，但 Kamodo 的设计也考虑了更广泛的应用场景，例如数据函数化、单位转换、函数复合、自动生成交互式绘图以及模型数据比较。这些功能使用 Python 构建，并采用 SymPy[8] 和 Plotly[9]，具体如 Pembroke 等人（2022）所述（Python：Rossum 和 Boer（1991）；SymPy：Meurer 等人（2017）；Plotly：Plotly Technologies Inc.（2015））。得益于其优良的软件设计，Kamodo 也足够灵活，可用于日球物理学领域内外。Pembroke 等人（2022）以及 Kamodo 核心文档[10]介绍了这些功能及其他核心能力和令人振奋的新进展。

为 CCMC 托管的所有类型模型数据输出开发上述功能，是一项艰巨的任务（CCMC 托管的部分模型示例见图1）。我们的开发计划是为日球物理学各个领域中的部分模型创建模型读取器，并以这些模型读取器的结构为基础开发各种分析功能（例如前文提到的飞越分析）。截至本文撰写时，我们提供了十二种电离层-热层模型输出的模型读取器（图1中的青绿色类别）、两种磁层模型输出的模型读取器（同图中的黄色类别），以及一种大气-电离层领域的模型输出（WACCM-X，未在图1中显示）。具体而言，该库包含的

[8] https://www.sympy.org/en/index.html[9] https://plotly.com/[10] https://ensemblegovservices.github.io/kamodo-core/

Kamodo 支持的模型输出包括 CTIPe（耦合热层-电离层-等离子体层电动力学模型，Codrescu 等，2008）、IRI（国际参考电离层模型，Bilitza，2018）、GITM（全球电离层热层模型，Ridley 等，2006）、OpenGGCM（仅全球磁层输出；开放式地球空间大气环流模型，Raeder 等，2001）、SWMF（电离层电动力学和磁层输出；空间天气建模框架，Toth 等，2007）、AMGeO（地球空间观测同化制图，AMGeO 合作组，2019）、TIEGCM（热层-电离层电动力学大气环流模型，Qian 等，2013）、WACCM-X（扩展至热层和电离层的全大气社区气候模型，Liu 等，2018）、DTM（阻力温度模型，Bruinsma，2015）、SuperDARN（超大双极光雷达网，包含两种输出类型，Cousins 和 Shepherd，2010）、Weimer（Weimer，2005）、ADELPHI（AMPERE——高纬电离层电动力学特性推导模型，Robinson 等，2021），以及 WAM-IPE 模型输出（全大气模型与电离层-等离子体层电动力学模型的耦合输出，Maruyama 等，2016 和 Fang 等，2022）。

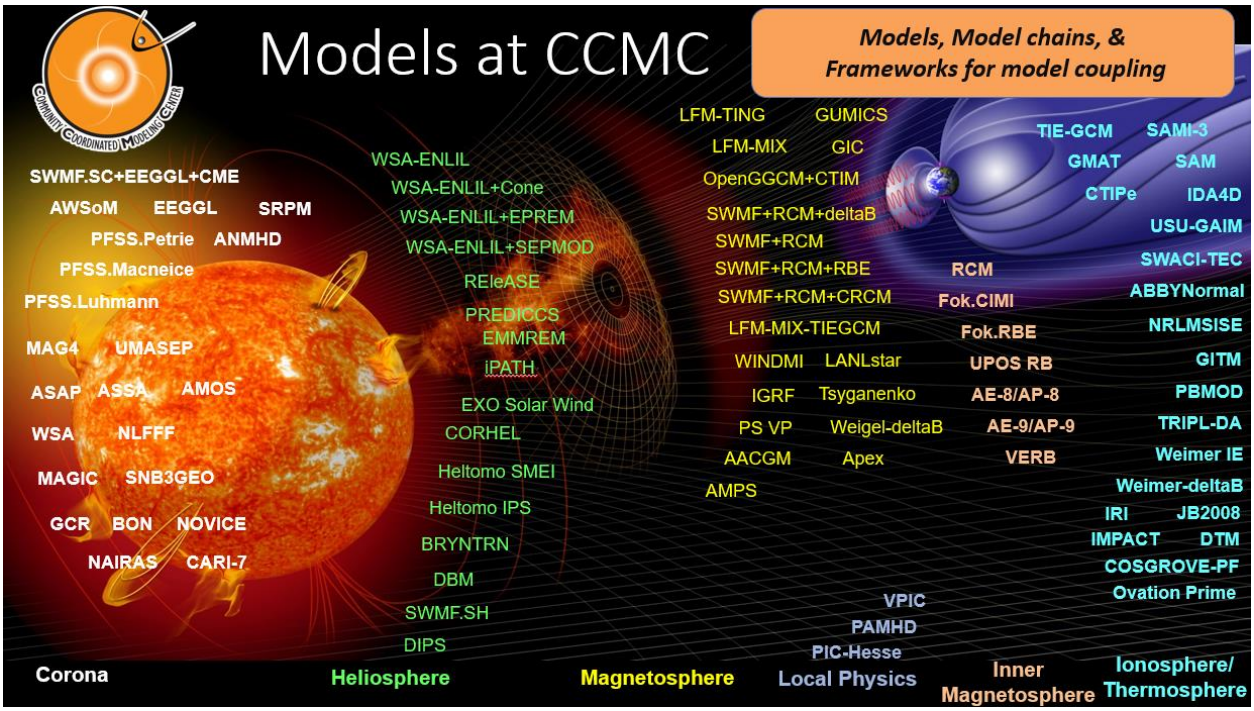


图1：CCMC 托管的模型、模型链和模型框架。为特定领域提供模型数据的模型名称（全称或缩写）使用与图底部领域名称相同颜色的字体。

目前，多个领域的模型读取器正在开发中。在电离层领域，我们正通过合作为 SAMI3 模型开发模型读取器（Sami3：电离层模型，Huba 和 Krall，2013）。最近，我们扩展了对磁层模型输出的支持，新增了 SWMF 模型的磁层部分

输出。该模型读取器首次纳入了从另一种语言调用的自定义插值器脚本，并将作为多个不同领域合作项目的示例（详情请参阅第3.3节和第6节）。

Kamodo 中也包含用于读取观测数据的读取器，例如来自 CDAWeb（坐标数据分析网站[11]）的数据集，以及来自 SSCWeb（卫星态势中心网站[12]）的卫星轨迹。这些读取器有助于进行数据与模型的比较，但不在本文讨论范围之内。通过开发模型读取器示例库及相关文档，我们正在为协作者构建资源，以便他们为 Kamodo 添加更多模型。此外，我们还开发了实用的分析工具，供这些新的模型读取器接入，以此推动此类合作。Kamodo 软件采用开源许可并始终保持开源，极大地简化了此类合作；这也是 CCMC 最初开发该软件时优先考虑的要求之一。

第*3*节：模型读取器

我们将 Kamodo 中的模型读取器定义为用户与所选模型数据之间直接的、模型无关的接口。按照标准，所有这些模型读取器都通过简单统一的命令，将所选模型数据作为函数化变量提供给用户访问（图2）。这一简单命令将所有特定于模型的细节对用户隐藏起来，尤其是复杂网格上的数据插值细节，以及处理各种文件格式时遇到的问题。由于这些模型读取器基于 Kamodo 构建，用户随后便可在分析所选模型数据时使用 Kamodo 的各种功能，例如自动交互式绘图生成（图9）、函数复合和单位转换。模型读取器通过以下方式进行验证：将每个模型读取器的样本输出与 CCMC 在线可视化工具当前版本中先前已验证的相应输出进行比较，或与模型开发者合作，使用其提供的样本输出进行验证。有关 CCMC 的 Kamodo 当前支持的模型输出列表，请参见图3。

第*3.1*节：使用模型读取器

用于将所选数据和变量功能化的模型无关命令显示在图2中灰色框的第三行。通常，调用模型读取器时唯一需要提供的信息是模型数据的完整文件路径。此命令所需的前置步骤见前两行，包括导入模型目录（以及其他功能）的语句，以及获取所选模型读取器的语句（第二行）。第三行会将所选文件目录（`文件_目录`）中的所有变量数据功能化——本例中共有19个变量。通常，模型输出数据的文件路径取决于用户机器或在线环境中的数据结构，

[11] https://cdaweb.gsfc.nasa.gov/index.html/[12] https://sscweb.gsfc.nasa.gov/

因此，模型读取器本身无法提供此信息，必须由用户提供。第二行使用的字符串值（‘GITM’）可从图3所示命令的输出中选择。

从 GitHub代码库 安装 Kamodo 后（见上方脚注中的第一个链接），有两种方法可以获取用于检索所选模型读取器所需的一组命令。用户可以直接访问*/kamodo_ccmc/readers/*文件目录，获取模型读取器的名称；也可以使用*/kamodo_ccmc/flythrough/* 文件目录中的模型_封装脚本所提供的功能，如图2所示。例如，图2中的前两行代码可以替换为图4中的前两行代码。替换后仍会像之前一样，返回用于 GITM模型输出的同一个模型读取器，但还需额外查找正确的模型读取器名称。为简便起见，我们推荐使用图2所示的方法。

```
import kamodo_ccmc.flythrough.model_wrapper as MW
reader = MW.Model_Reader('GITM')
kamodo_object = reader(file_dir)
kamodo_object
```

$$\rho_n(\vec{r}_{GDZsph4D})[\frac{kg}{m^3}] = \lambda(\vec{r}_{GDZsph4D})$$

$$\rho_{nijk}(time[hr], lon[deg], lat[deg], height[km])[\frac{kg}{m^3}] = \lambda(time, lon, lat, height)$$

$$\mathrm{T_n}\,(\vec{r}_{GDZsph4D})[K] = \lambda(\vec{r}_{GDZsph4D})$$

$$\mathrm{T_{nijk}}(time[hr], lon[deg], lat[deg], height[km])[K] = \lambda(time, lon, lat, height)$$

图2：展示模型读取器命令简洁性的示例。前两行代码会为所选模型获取模型读取器实例，第三行则会将指定文件目录文件_目录中的模型数据所包含的所有变量函数化。在最后一行代码中调用“kamodo_对象”变量，会打印出表示给定模型数据中每个函数化变量的 LaTeX 输出，如代码下方的输出框所示（展示了 19 个函数化变量中的 2 个）。请注意，默认情况下，每个变量都包含两个函数化版本（见文档[13]）。

对于许多科学问题，用户需要了解数据的时间和坐标范围，以及给定文件目录中包含哪些变量。上文提到的 model_wrapper 脚本中有几个简单的函数，可以轻松提供这些信息。用户只需提供模型名称以及存储所需模型输出的文件路径，然后执行图5中的代码行，即可查看这些文件的起始和结束时间（协调世界时）。该函数还会以 datetime[14] objects 的形式返回这两个时间，供其他用途使用。若要获取所选数据中包含的变量列表，用户可以执行图6中的一行代码。这项功能很重要，因为给定数据集所包含的变量通常只是某个模型可访问变量的子集。

[13] https://github.com/nasa/Kamodo[14] https://docs.python.org/3/library/datetime.html

输出中包含每个变量的名称，后面跟着一系列元数据，包括描述、整数、坐标系缩写、坐标系类型、坐标名称列表以及单位。即使没有任何模型数据，也可以使用同一函数在模型库中搜索指定变量（例如温度）。在图2（或图4）中创建所需的 *kamodo_对象* 后，可以执行图 7中的单条命令，获取所需变量各自坐标的最小值和最大值。这些函数及各读取器中其余关键字的功能，在美国国家航空航天局 GitHub代码库的 notebook 目录中的模型读取器教程笔记本里有演示。

```
1 import kamodo_ccmc.flythrough.model_wrapper as MW
2 MW.Choose_Model()
```

```
Possible models are:
ADELPHI: AMPERE-Derived ELectrodynamic Properties of the High-latitude Ionosphere https://doi.org/10.
1029/2020SW002677
AMGeO: Assimilative Mapping of Geospace Observations https://doi.org/10.5281/zenodo.3564914
CTIPe: Coupled Thermosphere Ionosphere Plasmasphere Electrodynamics Model https://doi.org/10.1029/200
7SW000364
DTM: The Drag Temperature Model https://doi.org/10.1051/swsc/2015001
GAMERA_GM: Grid Agnostic MHD for Extended Research Applications - Global Magnetosphere outputs http
s://doi.org/10.3847/1538-4365/ab3a4c (coming soon)
GITM: Global Ionosphere Thermosphere Model https://doi.org/10.1016/j.jastp.2006.01.008
IRI: International Reference Ionosphere Model https://doi.org/10.5194/ars-16-1-2018
OpenGGCM_GM: The Open Geospace General Circulation Model - Global Magnetosphere outputs only https://
doi.org/10.1023/A:1014228230714
SuperDARN_uni: SuperDARN uniform grid output https://doi.org/10.1029/2010JA016017
SuperDARN_equ: SuperDARN equal area grid output https://doi.org/10.1029/2010JA016017
SWMF_IE: Space Weather Modeling Framework - Ionosphere and Electrodynamics outputs https://doi.org/1
0.1029/2006SW000272
SWMF_GM: Space Weather Modeling Framework - Global Magnetosphere outputs https://doi.org/10.1029/2006
SW000272
TIEGCM: Thermosphere Ionosphere Electrodynamics General Circulation Model https://doi.org/10.1029/201
2GM001297
WACCMX: Whole Atmosphere Community Climate Model With Thermosphere and Ionosphere Extension https://d
oi.org/10.1002/2017MS001232
WAMIPE: The coupled Whole Atmosphere Model - Ionosphere Plasmasphere Model https://doi.org/10.1002/20
15GL067312 and https://doi.org/10.1029/2022SW003193
Weimer: Weimer Ionosphere model https://doi.org/10.1029/2005JA011270
```

图3： CCMC 的 Kamodo 当前支持的模型库。导入语句之后，可以使用选择_模型 函数，显示 CCMC 的 Kamodo 当前支持的模型库。每行开头的字符串是获取相应模型的模型读取器脚本所需的字符串（用作上一幅图第 2 行的输入）。该字符串后面是模型的全名及其主要 DOI。

创建模型读取器对象后，用户便可通过 Kamodo核心软件包提供的功能，以同样简洁的语法访问每个函数化变量的插值器。图 8 展示了调用函数化变量插值器的各种方法。第一条和第三条 Python 命令展示了获取某个示例函数化变量的 LaTeX表示的两种等效方法：使用属性表示法（第一行）或字典表示法（第三行）。在本例中，这两个函数化

变量是同一变量的两个版本：一个非网格插值器（上方两组输入/输出对）和一个网格插值器（下方两组输入/输出对）。非网格插值器要求输入位置列表的列表（用于指定一维排列中的位置），如第二行所示，其中给出了两个位置。网格插值器接受一个或多个坐标位置，以提取低维切片或超平板；第四行展示了针对同一位置的示例。请注意，输入坐标——时间（以文件目录中第一个数据文件午夜起算的小时数表示）、经度（单位为度）、纬度（单位为度）和高度（单位为千米）——以及各自得到的值均完全相同。更多示例见 Kamodo 核心文档。图7和图8中表示变量名称的字符串因模型而异，可通过多种方法获取；这些方法在美国国家航空航天局 Github 仓库（见脚注 8）的各类笔记本以及图6中均有展示，其中包括执行命令 *kamodo_对象.detail()*。

```
from kamodo_ccmc.readers.gitm_4Dcdf import MODEL
reader = MODEL()
kamodo_object = reader(file_dir)
kamodo_object
```

$$\rho_n\left(\vec{r}_{GDZsph4D}\right)[\frac{kg}{m^3}] = \lambda\left(\vec{r}_{GDZsph4D}\right)$$

$$\rho_{nijk}(time[hr], lon[deg], lat[deg], height[km])[\frac{kg}{m^3}] = \lambda(time, lon, lat, height)$$

$$\mathrm{T_n}\left(\vec{r}_{GDZsph4D}\right)[K] = \lambda\left(\vec{r}_{GDZsph4D}\right)$$

$$\mathrm{T_{nijk}}(time[hr], lon[deg], lat[deg], height[km])[K] = \lambda(time, lon, lat, height)$$

图4： 加载模型读取器的另一种方法。第一行代码导入所需的脚本。第二行代码为所选模型加载模型读取器类实例。第三行和第四行代码以及下方的输出与图2相同。

```
MW.File_Times('GITM', file_dir)
```

```
UTC time ranges
----------------------------------------
Start Date: 2015-03-17  Time: 00:00:00
End Date: 2015-03-19  Time: 01:48:59

(datetime.datetime(2015, 3, 17, 0, 0, tzinfo=datetime.timezone.utc),
 datetime.datetime(2015, 3, 19, 1, 48, 59, 996338, tzinfo=datetime.timezone.utc))
```

图5：获取数据集的时间范围。所选数据集的协调世界时起止日期和时间会打印出来，该数据集位于文件_目录中。返回值是以 datetime对象表示的相同日期和时间。

通过 Kamodo，可以轻松为任意一对维度绘制标准二维热图。将多维变量通过 Kamodo 函数化后，用户只需一行代码即可生成交互式二维图，如下方图9所示。语法只需指定目标变量的名称（网格化版本）、坐标的名称，以及切片额外维度时所需的取值。可通过图6所示的方法获取变量名称。需要指出的是，

经度范围随 SpacePy 和 AstroPy 软件包所定义的坐标系而变化。有关更多详情，请参阅 轨迹_坐标_绘图笔记本中有关坐标转换飞越函数的文档（SpacePy：Morley 等，2011；AstroPy：Astropy Collaboration 2018）。第4节展示了其他交互式绘图功能。

```
MW.Variable_Search('', 'GITM', file_dir)
```

```
The file directory contains the following standardized variable names:
--------------------------------------------------------------------------------
rho_N2 : '['mass density of molecular nitrogen', 13, 'GDZ', 'sph', ['time', 'lon', 'lat', 'height'], 'kg/m**3']'
rho_N2plus : '['mass density of molecular nitrogen ion', 14, 'GDZ', 'sph', ['time', 'lon', 'lat', 'height'], 'kg/m**3']'
rho_NO : '['mass density of nitric oxide', 20, 'GDZ', 'sph', ['time', 'lon', 'lat', 'height'], 'kg/m**3']'
rho_NOplus : '['mass density of nitric oxide ion', 21, 'GDZ', 'sph', ['time', 'lon', 'lat', 'height'], 'kg/m**3']'
rho_O2 : '['mass density of molecular oxygen', 22, 'GDZ', 'sph', ['time', 'lon', 'lat', 'height'], 'kg/m**3']'
rho_O2plus : '['mass density of molecular oxygen ion', 24, 'GDZ', 'sph', ['time', 'lon', 'lat', 'height'], 'kg/m**3']'
rho_O3P : '['mass density of atomic oxygen (3P state)', 27, 'GDZ', 'sph', ['time', 'lon', 'lat', 'height'], 'kg/m**3']'
```

图6： 检索给定数据集中变量列表及其元数据。输出包含每个变量的名称，后跟一组元数据，包括描述、整数、坐标系缩写、坐标系类型、坐标名称列表和单位。更多详情请参阅正文和文档，尤其是 */docs/notebook/* 目录中的 START_HERE 笔记本。

```
MW.Coord_Range(kamodo_object, ['rho_N2'])
```

```
The minimum and maximum values for each variable and coordinate are:
rho_N2:
time: [0.0, 49.81666564941406, 'hr']
lon: [-180.0, 180.0, 'deg']
lat: [-90.0, 90.0, 'deg']
height: [96.62001037597656, 640.3912963867188, 'km']
```

图 7： 检索给定变量的每个坐标的 范围。此处的输入取自图 2 和图 6。列表中给出了给定变量所依赖的每个坐标的最小值和最大值，列在各坐标名称旁边，后跟该坐标的单位。对于 CCMC 开发的模型读取器，时间始终以小时为单位，从给定目录中数据的第一天午夜（协调世界时）起算。选定数据完成函数化后，执行命令 *kamodo_object.filedate*即可检索该日期和时间的 datetime 表示形式。有关给定数据集的开始和结束时间，请参阅图 5 中的 文件_时间命令。有关这些 函数的更多选项和信息，请参阅文档。

第 *3.2* 节：模型读取器组件

为了提供上述功能并兼容更高层级的工具，我们确定了模型读取器所需的基本组件，并在此介绍其概要及演示。不同类型的数据需要采用不同的方法，以确保执行时间较短，具体将在下文说明。更多详情请参阅 Kamodo 文档（https://github.com/nasa/Kamodo）。

```
kamodo_object.rho_n
```

$$\rho_n(\vec{r}_{GDZsph4D})[\frac{kg}{m^3}] = \lambda(\vec{r}_{GDZsph4D})$$

```
kamodo_object.rho_n([[5.5, 20., 55., 350.], [35.1, -45., -30., 200.]])
```

```
Time slice index 33 added from file.
Time slice index 34 added from file.
Time slice index 35 added from file.
Time slice index 211 added from file.
Time slice index 212 added from file.

array([5.38050868e-12, 2.95876514e-10])
```

```
kamodo_object['rho_n_ijk']
```

$$\rho_{nijk}(time[hr], lon[deg], lat[deg], height[km])[\frac{kg}{m^3}] = \lambda(time, lon, lat, height)$$

```
kamodo_object['rho_n_ijk'](5.5, 20., 55., 350.)
```

```
array(5.38050868e-12)
```

图 8：调用插值器以计算给定函数化变量的多种方法示例。'rho_n'是 GITM 模型读取器中的函数化中性质量密度。'rho_n_ijk'是同一函数的网格化版本。请注意，输入坐标和得到的插值结果完全相同。详情请参阅正文和文档。

模型读取器最重要的功能，是对模型数据所关联的整个时间范围内（包括不同数据文件之间）的每个变量的数据进行插值。对于不随时间或位置变化的坐标网格，我们提供了一个标准接口，该接口基于 SciPy 的 规则网格插值器[15] (Virtanen et al. 2020) 实现线性插值方案。SciPy 的插值器通常以 C语言编写，并通过各种 Python 命令调用，以确保执行快速且准确。与 SciPy 提供的类似选项相比，规则网格插值器具有多项优势，这也是我们选择它的原因，例如无需网格点等间距、能够对任意维度的数据进行线性插值，并避免计算量更大的方法（例如使用 Delauney 三角剖分）。

我们发现，这些优势尤其适合 Kamodo 目前支持的、托管于 CCMC 的模型数据的插值需求，也很可能同样适用于更广泛的模型数据（更复杂场景的讨论请参见第3.3节）。通常情况下，无论模型读取器为何种类型，该默认插值器都能在约半秒内，为一整天的模型数据计算出 40,000 个三维（时间+2D 空间）或四维（时间+3D 空间）位置的数值，但具体执行时间会因数据数组的大小而有所不同。此外，在假设所有必要的文件转换均已完成的情况下（见下文），为单个变量生成 Kamodo 对象所需的执行时间通常不到半秒。尽管

[15] https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.RegularGridInterpolator.html

这些执行和插值时间在大多数情况下略慢于 CCMC 在线可视化服务当前使用的脚本，但其应用范围要广泛得多，因此付出的努力非常值得。

```
kamodo_object.plot('rho_n_ijk', plot_partial={'rho_n_ijk': {'time': 25.7, 'height': 605.7}})
```

```
Time slice index 155 added from file.
Time slice index 156 added from file.
```

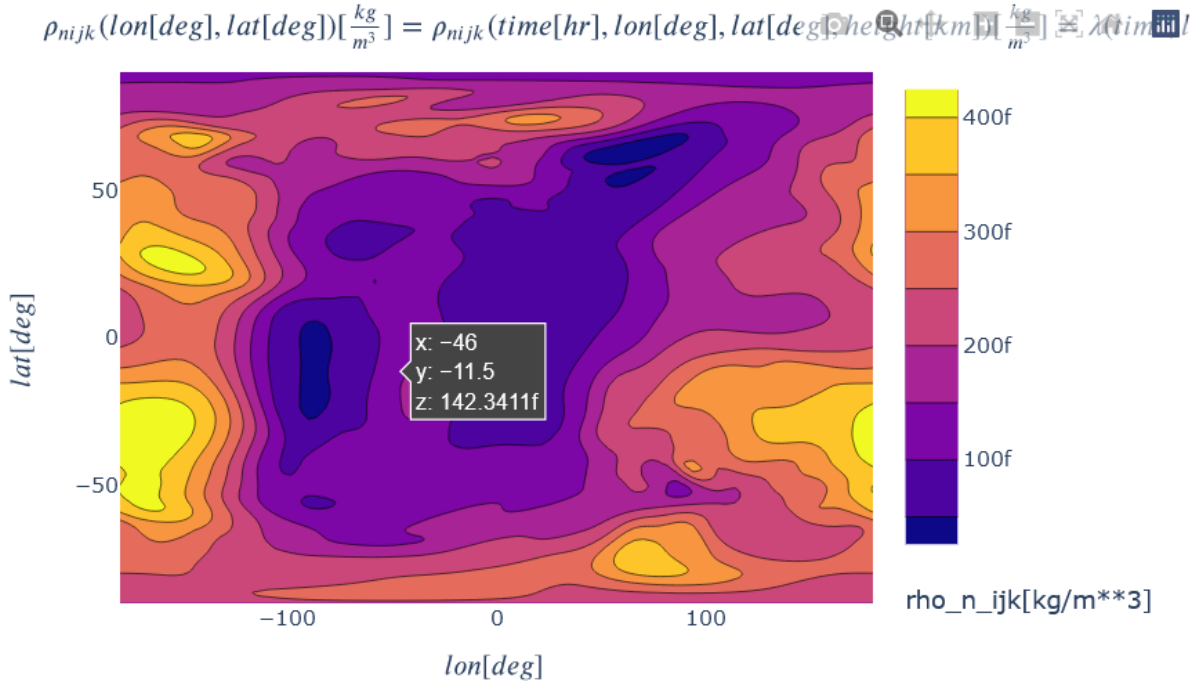


图9：Kamodo 提供的自动交互式绘图生成示例。顶部的灰色框中包含生成纬度-时间图的命令，该图显示由 GITM 模型读取器函数化的中性密度变量（'rho_n_ijk'）。鼠标指针所在位置的绘图区中的方框（此处为灰色背景）会显示用户鼠标位置对应的时间、纬度和中性温度值；单位见绘图顶部函数的 LaTeX 表示。请注意，绘图右上角的符号可启用 Plotly 的常见交互式绘图功能，例如缩放和保存。命令块下方的"time slice..."输出是惰性插值方案打印的消息，指明为生成绘图而加载了哪些时间切片。更多详情请参阅正文和文档。

每个模型读取器还提供针对各个模型数据格式的定制化访问方式。对于采用 netCDF4 文件等标准格式的模型输出数据，这种访问方式很容易编写代码实现；但对于其他格式，尤其是未压缩或压缩二进制文件，情况就明显复杂得多（例如，比较一下用于读取 netCDF4 模型输出文件的 iri_4D.py 模型读取器脚本有多简单，再看看读取未压缩二进制文件所需的 gitm_4D.py 模型读取器与 gitm_tocdf.py 文件转换脚本）。用于转换 OpenGGCM 全球磁层模型输出的文件转换脚本使用 Python 编写，并通过

f2py 接口调用基于 FORTRAN 的函数来处理压缩二进制文件（Peterson 2009）。该脚本将由 OpenGGCM 模型读取器调用，并可在 Kamodo 的 GitHub[16] 上公开获取。

将给定文件目录中存储的整个数据集函数化，会对插值要求产生重要影响。我们不能假设整个数据集都能一次性读入内存，除了一维时间序列数据（例如，仅依赖时间而不依赖位置的变量）这种情况。对于一维情形（插值_标志*=0*），所有相关文件中的整个数据集都会被读入内存，并分配给 *interp1d* SciPy 插值器[17]。对于不需要惰性插值的高维变量，也可以使用此选项；此时会分配标准的 规则网格插值器（参见文档）。对于依赖时间的变量，我们设计了一套惰性插值器，只将插值请求所需时间步对应的数据读入内存并函数化。这样，即使数据集大小超过典型计算机内存（~16 GB），我们也能对其进行插值。

要高效实现惰性插值，软件必须在打开文件之前知道哪些数据文件包含哪些时间值和变量。我们会在选定的模型读取器首次针对某个文件目录运行时，计算并存储时间值与文件名之间的映射，以缩短插值耗时。为计算此映射，脚本会搜索并打开目录中的每个数据文件，以获取各文件中存储的时间值。随后，这些从各文件中获取的时间值会以不同格式保存到两个文本文件中，并存放在该目录下。将映射拆分为两个文件，可以妥善处理稍复杂的情况：每个时间步包含多个文件，且数据按变量拆分（例如，不同于同一变量按处理器编号拆分）。由于这一步需要额外耗时，并会增加云环境中运行的成本（尤其是模型数据存储在 S3 存储桶中时），因此我们不会在每次执行模型读取器时重复这一步。请求的变量名称与相关数据文件之间的映射只需打开并查询每种类型的一个文件，因此无需执行这一额外步骤。在模型读取器执行期间，系统会将这两种映射载入内存，并用于初始化每个变量的插值，无论该变量采用何种插值方法。

根据数据文件的结构，有三种惰性插值选项可用。最简单的惰性插值方法（插值_标志*=1*）适用于模型输出按每个时间步一个或多个文件分开存储的情况。另两种惰性插值方法的逻辑相似，适用于每个数据文件包含多个时间步（也称为时间块）的模型数据输出。插值_标志*=2* 选项适用于能够轻松装入典型计算机内存（GB 随机存取存储器）的时间分块数据文件，

[16] https://github.com/nasa/Kamodo/tree/master/kamodo_ccmc/readers/OpenGGCM[17] https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.interp1d.html

而插值_标志=*3* 选项则适用于较大的时间分块数据文件。对于较小的文件（选项 1 或 2），会按需将指定的时间切片或数据块载入内存。对于较大的文件（选项 3），会将从时间块中选定的特定时间步的数据载入内存，以降低对计算机内存的需求。

每次调用插值时，软件都会确定执行指定插值所需的时间步或时间块，通常是目标值前后各一个时间步；对于选项 2，则是相应的时间块。选定模型时间值后，软件会利用这些时间值与文件之间的映射，确定哪些文件包含所需数据。随后，软件将这些信息发送给模型读取器，由其检索请求的数据、执行所需的数据整理，并将数据返回给插值程序。对于选项 2，系统会返回包含该时间值的时间块，并在条件允许时从下一个时间块中额外添加一个时间步，以便在时间块之间正确插值。最后，系统会为每个时间步或时间块创建标准的 SciPy 插值器，并为时间维度指定一个*interp1d*插值器，因此每次插值调用都会执行两层插值。我们发现，这种方法通过避免随着请求更多时间而反复创建插值器，减少了插值执行时间总量。对于请求的数据中已存在的时间值（例如绘图请求），我们还只使用该时间值对应的数据，从而进一步缩短计算时间。

所有插值方法的逻辑都假定，给定变量和时间步的全部数据都位于同一个文件中。对于模型数据，若不同变量在同一时间步的数据存储在不同文件中，也采用这种处理方式。在这些情况及类似情况下，变量与文件集合之间的映射由模型读取器的逻辑负责（例如 GITM 模型读取器）。然而，当前的结构不便于处理这样的模型输出：给定时间步和变量的数据分散在多个文件中（例如按处理器编号拆分）。针对这些复杂情况，三种惰性插值方案都内置了便捷功能，可用于集成自定义插值器（见第 3.3 节）或自定义读取例程。

此外，与 CDF、netCDF、HDF5 文件或类似格式相比，将模型数据存储在文本文件或二进制文件中时，执行速度也会成为一个问题。为解决这些问题，模型读取器会在首次调用时自动将目录中的每个数据文件转换为 netCDF 文件，但仅限于原始数据以非理想文件格式存储的情况。如果用户向同一目录添加数据文件，则需要删除模型读取器先前在该目录中创建的时间文件，并重新运行模型读取器，以纳入新文件。这样做是为了避免每次运行模型读取器时都执行目录列表命令，因为这既耗时，在云环境中也不划算。

文件转换过程还会移除模型开发者认为多余的数据（例如状态日志变量），并对所有剩余变量执行必要的数据重新格式化和处理，而不受所请求变量名称的影响。我们发现，这一做法显著缩短了后续调用时模型读取器的执行时间；在某些情况下，还能明显减小文件总大小，但不包括原始文件的大小（表1）。不过，目前我们选择保留所有原始模型输出文件，以便今后重新处理，并保留当前未纳入转换后文件中的数据。

表1比较了五个需要进行文件转换的模型读取器的执行时间。初始执行时间包括文件转换时间以及将单个变量函数化所需的时间（第二列），而最终执行时间仅包括对同一单个变量进行函数化的时间。所给出的不确定性表示在相同条件下使用五次执行时间计算得出的标准差。GITM（Ridley 等，2006）、Weimer、ADELPHI 和 SWMF 电离层电动力学模型输出所报告的数值对应一天的数据，而 OpenGGCM（全球磁层）（Raeder 等，2001）模型输出所报告的数值对应一小时的数据。请注意，所列出的所有模型读取器的最终执行时间均显著短于初始执行时间。

表1：所选模型读取器首次调用与后续调用的执行时间比较。

模型读取器	初始执行 时间（秒）	最终执行 时间（秒）	初始总计 文件大小（GB）	最终总计 文件大小（GB）
GITM	974. ± 19。	0.563 ± 0.033	20.3	8.37
SWMF（电离层）	200.7 ± 1.4	0.301 ± 0.029	3.89	1.97
ADELPHI	21.0 ± 1.0	0.309 ± 0.059	0.131	0.00783
Weimer	84.73 ± 0.73	0.296 ± 0.016	1.92	0.157
OpenGGCM (GM)	736 ± 59†	0.107 ± 0.022	4.38	15.6

†初始执行时间不包括 OpenGGCM (GM) 模型读取器的执行时间。最终文件大小较大，是因为对原始压缩二进制文件格式进行了解压。

许多电离层-热层-中间层（Ionosphere-Thermosphere-Mesosphere，ITM）模型在球坐标系中的输出包含一些依赖于气压层而非高度或半径的变量。每个模型输出中的气压层坐标网格定义各不相同，且会随空间和时间变化，这使得对给定卫星位置进行插值，或绘制指定变量在恒定高度处的图像，都变得复杂。不过，模型输出中的气压层坐标网格值被定义为固定参考点，因此可以采用常见的 规则网格插值器，尽管用法有所创新。由于这种关系存在于多个模型中，我们创建了一种模型无关方法

该方法基于标准插值器，对各模型中气压层与高度之间的关系进行反演[18]，且无需对底层数据作任何额外假设。

每个模型输出（包括依赖于气压层的变量）还会给出高度与时间、经度、纬度和气压层之间的函数关系。要返回给定时间、经度、纬度和高度对应的气压层，必须对这一关系求逆。我们通过几个简单步骤实现这一点。首先，将高度变量函数化并网格化，以便进行切片。对于每一组唯一的时间、经度和纬度值，我们通过高度函数对模型提供的气压层坐标网格值进行插值，从而得到一组高度值，分别对应于该时间和经纬度位置下的各个气压层网格值。然后，我们使用 SciPy 提供的*interp1d*插值器创建一个一维插值器，将插值得到的高度值作为“X”值，将模型提供的气压层值作为“Y”值。创建完成后，该一维插值器便可根据请求的高度返回气压层。这样，对于每个请求的时间、经度和纬度组合，都能建立高度与气压层之间的自定义逆向关系。这使模型读取器能够仅利用模型提供的数据，通过 Kamodo 的函数复合功能，自动将每个依赖于气压层的变量重新定义为依赖于高度的新变量。我们将这段代码推广到所有相关模型，从而为高效纳入自定义插值器奠定了框架（见第3.3节）。

所选的模型读取器结构使执行时间随着变量数量的增加而线性增长。图 10 展示了 CCMC Kamodo 模型库中各模型读取器的执行时间与函数化变量数量之间的关系。这些调用同时请求了每个变量的常规版本和网格化版本（*gridded_int =True*，详见文档），并且所有必要的文件转换都已提前完成。对于采用惰性插值方案的变量（维度不少于两个），模型读取器运行时不会将任何模型输出数据加载到内存中，坐标网格除外。因此，这些计时测试中函数化的变量至少具有两个维度。遗憾的是，这一限制使 SuperDARN 测试只能使用三个可能的变量（白色方块和三角形）。DTM 模型输出总共只有九个变量，因此缺少第十个变量的数据点（紫色三角形）。Weimer 模型输出未纳入测试，因为它只有一个或两个变量，具体取决于数据集。

除了绘制每个模型读取器对应的线条和符号外，还以黑色虚线绘出了平均执行时间，并用一个标准差的误差线表示。一个变量的平均执行时间为 0.50 ± 0.19 秒，如图 10 的插图所示；每增加一个变量，执行时间平均增加 0.4618 ±0.0046 秒。执行时间的标准差

[18] 请参阅 GitHub（https://github.com/nasa/Kamodo）上 *kamodo_ccmc/readers/* 目录中的 reader_utilities.py 脚本里的 *PLevelInterp* 函数。

也会随着变量数量的增加而增大：一个变量时为 0.19 秒，10 个变量时增至 0.94 秒；变量数量较多时，增大的离散程度主要由 WACCM-X 的执行时间导致。总体而言，各模型读取器的执行时间通常在每个变量平均执行时间的一个标准差范围内。这种紧密的吻合源于各读取器在多种属性上的差异，例如唯一坐标网格的数量、每个坐标网格的大小、是否存在需要额外初始化步骤的模型专用坐标（如气压层）、变量文件映射逻辑的复杂程度，以及其他类似因素。

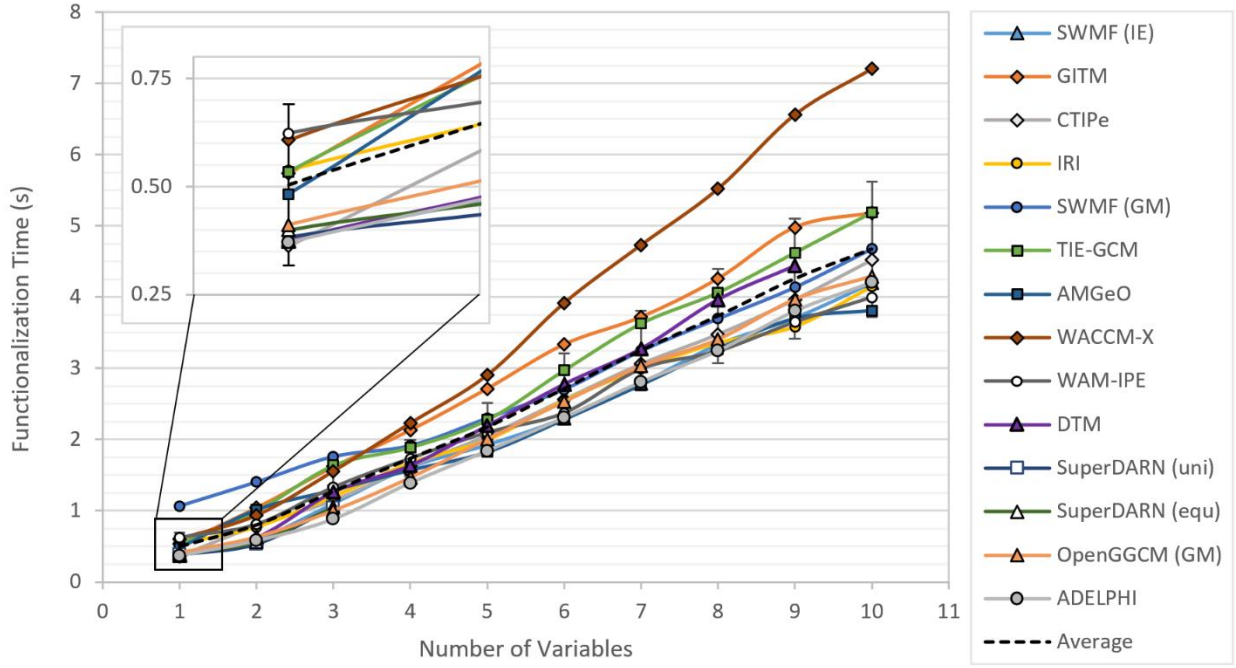


图 10：所选模型读取器的总执行时间与请求变量数量的关系。实线表示不同颜色和形状的模型读取器（见图例）。黑色虚线表示所采样模型读取器的平均执行时间。平均值曲线上的误差棒表示每个变量数量下的 1σ 标准差。除变量数量较少时 SWMF (GM) 的执行时间，以及变量数量较多时 WACCM-X 的执行时间外，各执行时间通常集中在较窄范围内。左上角插图显示了使用一个变量时的执行时间分布。更多详情请参阅正文。

不过，这一趋势有两个例外。变量数量较少时，SWMF (GM) 模型读取器执行时间（蓝色圆线）远高于误差线；变量数量达到四个或更多时，WACCM-X 模型读取器执行时间（棕色菱形线）远高于误差线。与其他代码相比，SWMF (GM) 的执行时间之所以如此，很可能是因为需要额外的初始化步骤，才能妥善准备自定义插值 C 代码以供执行。相较之下，变量数量较大时，WACCM-X 模型读取器代码的执行时间较长，可能是因为其坐标网格规模较大（与 OpenGGCM (GM) 大致相当），再加上气压层

对于所选变量集，变量数量较大时会自动将压力反转函数化。

模型读取器还包含其他功能，例如在模型数据输出文件之间自动进行时间线性插值；在极点处进行标量和矢量平均（对于输出位于不包含极点的球面网格上的模型，请参阅文档）；在通常晦涩难懂的模型变量名称与标准化变量名称之间进行映射；以及将模型数据与由 SpacePy 和 AstroPy 软件包定义的标准坐标系对齐（SpacePy：Morley 等，2011；AstroPy：Astropy Collaboration 2018）。虽然模型运行通常是在连续的时间范围内生成的，但输出文件会收集指定数量的时间步或指定时间范围内的所有数据，因此文件之间会产生一个人为的时间间隙，其长度对应于模型运行的时间分辨率。要正确表示模型运行生成的完整数据集，必须对这个时间间隙进行插值。最简单的方法是在文件之间的时间值上采用最近邻插值，即选择相邻文件中最接近请求时间值的时间值，并对空间分量执行常规的线性插值。然而，这种方法与其他时刻对所有时间和空间分量进行的线性插值并不一致。因此，我们选择在所有模型读取器中统一实现数据文件之间的时间线性插值。上述惰性插值方法已考虑文件间插值。有关详情，请参阅 NASA Kamodo Github 仓库中的 Kamodo 入门指南。

球面网格上的模型输出并不总是延伸至极点。在大多数情况下，这是因为模型报告的网格值位于包含极点的网格单元中心（例如，分辨率为 1 度、位于纬度 89.5 度的网格单元）。在这些情况下，我们实现了额外的逻辑，用于计算极点处变量数据的预期值。对于密度或笛卡尔分量等标量量，该逻辑只需对目标极点附近的一系列值取平均，并将该值填入与该极点纬度对应的数组向量中。每个时间值都分别进行此操作。对于球面分量，尤其是非径向分量，我们会对相同的值进行向量平均，同样针对每个时间值分别计算，并将相应的值填入极点处对应的位置。有关此逻辑的更多详情和示例，请参阅入门文档。

变量名称映射通常由模型开发者协同确定：将模型输出中的每个变量描述与其他模型中已实现的类似变量进行比较。此映射对于确定每个变量最合适的 LaTeX 表示至关重要，以便与文献中该变量的标准表示相对应（例如，总密度用 ‘rho’ 表示，中性密度用 ‘rho_n’ 表示）。每个模型读取器都会将变量映射作为键值对存储在 模型_变量名字典中

可通过上述方法访问（例如，在从所需读取器导入 模型_变量名字典后，使用 *print(key, value[0] for key, value inmodel_varnames.items()* ）。同样，使变量数据与 SpacePy 和 AstroPy 软件包定义的标准坐标系保持一致，通常也需要与模型开发者协同完成。

此处未介绍的其他功能可确保与更高级别的功能（例如飞越函数）兼容。有关更多详情和其他示例，请参阅文档、示例笔记本以及托管在 NASA Kamodo GitHub 仓库中的代码[19]。

第*3.3*节：集成自定义插值器

标准插值器 SciPy 的规则网格插值器，以及上一节所述的插值方案，仅适用于在时间和各个维度上均保持静态、彼此正交且不包含网格细化（即围绕对象嵌套分布、分辨率逐级提高的区域）的空间网格。对于使用非标准网格的模型，尤其是地球空间领域中的模型，需要采用自定义插值例程；为提高效率，这些例程通常使用 Python 以外的语言编写。本节以 SuperDARN（一种电离层热层中层模型）和 SWMF/BATS-R-US 磁层输出数据为例，介绍我们如何轻松支持这些插值器。

SuperDARN 模型输出采用两种网格分辨率：默认的均匀分辨率网格和等面积网格。使用前文介绍的方法即可轻松对默认网格进行插值，但等面积网格的经度网格分辨率会随纬度变化。因此，需要使用自定义插值器。这个自定义插值器函数很简单，可以使用 Python 和 SciPy 插值器 *interp1d* 编写；不过，还需要在插值方案中创建一个关键字选项，以接受自定义插值器（即 *func_*默认关键字，见下方图 11）。

将 SuperDARN 等面积输出函数化，可用于测试是否能够加入自定义插值器；但由于该自定义插值器是用 Python 编写的，因此不涉及跨语言桥接。接下来，我们选择将 SWMF/BATS-R-US 磁层输出数据函数化，因为其输出目前缺少适用于复杂网格的四维（空间+时间）插值器；它是磁层物理中常用的模型，而且需要初始化并调用以 C语言编写的自定义插值例程。此外，这项工作很容易迁移到其他采用八叉树块网格结构的模型输出，包括 GUMICS（大统一磁层-电离层耦合模拟；Janhunen 等，2012年）和 ARMS（自适应细化磁流体动力学求解器；Wyper 等，2018年）模型。更广泛地说，我们确定的将这项工作高效整合到 Kamodo 模型读取器中的方法将

[19] https://github.com/nasa/Kamodo

扩展到其他自定义插值器，包括我们计划通过合作开发的插值器（见第6节）。

SWMF 包含许多组件模型，包括块自适应树太阳风 Roe 迎风格式（BAT-S-RUS：Powell 等，1998）磁流体力学求解器。该求解器使用块自适应网格，在建模域内解析精细结构和激波（图 12）。每个块都是一个立方体，由固定数量（至少 4x4x4）的等尺寸单元组成。在最粗的细化级别上，各块排列起来以填充仿真域。通过将每个单元沿 3 个维度分别一分为二，可将块细化为 8 个“子块”。由此形成八叉树结构，可沿该结构查找仿真域中任意给定位置所属的叶块（即没有任何子块的块）。

A

```python
def func(i):
    '''i is the file number.'''
    # get data from file
    file = self.pattern_files[key][i]
    cdf_data = Dataset(file)
    data = array(cdf_data.variables[gvar])
    cdf_data.close()
    return data

# define and register the interpolators
self = RU.Functionalize_Dataset(self, coord_dict, varname,
                                self.variables[varname],
                                gridded_int, coord_str,
                                interp_flag=1, func=func)
```

B

```python
def func(i):  # i is the file/time number
    # get data from file(s)
    cdf_data = Dataset(self.pattern_files[key][i])
    lat = array(cdf_data.variables['lat'])
    lat_keys = [str(lat_val).replace('.', '_').replace('-', 'n')
                if lat_val < 0 else 'p'+str(lat_val).replace(
                        '.', '_') for lat_val in lat]
    lon_dict = {lat_val: array(cdf_data.groups[lat_key]['lon'])
                for lat_key, lat_val in zip(lat_keys, lat)}
    data = {lat_val: array(cdf_data.groups[lat_key][gvar])
            for lat_key, lat_val in zip(lat_keys, lat)}
    cdf_data.close()

    # assign custom interpolator
    interp = custom_interp(lon_dict, lat, data)
    return interp

# functionalize the variable dataset
tmp = self.variables[varname]
tmp['data'] = zeros((2, 2, 2))  # saves execution time
self = RU.Functionalize_Dataset(
    self, coord_dict, varname, tmp, gridded_int, coord_str,
    interp_flag=1, func=func, func_default='custom')
```

图 11： 简单惰性插值调用（A）与自定义惰性插值调用（B）的比较。A 面板来自 GITM 模型读取器，它只需从文件中读取所需时间步的数据并返回。B 面板来自 SuperDARN 等面积模型读取器；它同样从文件中读取该时间步的数据，但还必须遍历更复杂的数据结构（分组式 HDF5 文件），并调用专为该数据结构设计的外部自定义插值器例程。请注意，A 面板中函数 *func* 的 return 语句返回一个 NumPy[20] 数组，而 B 面板中的对应语句返回一个插值器。通过在 B 面板底部将 *func_default* 关键字设为“custom”，可将这一行为差异告知插值软件。更多详情请参阅文档和代码。

SpacePy 软件包包含多个 SWMF 组件模型的读取器，包括用于读取 BATS-R-US 输出的 pyBats 软件包。Kamodo 使用*SpacePy.PyBats.IdlFile*类和*read_idl_bin* 函数，读取经过 IDL 脚本处理、用于可视化的输出，其中包含单元中心处的数密度、压力、速度、电流密度和

[20] https://numpy.org/

磁场。不过，PyBats 并未提供能够利用 BATS-R-RUS 输出[21]的八叉树块结构的插值器。

为提供 CCMC 的标准可视化服务，我们一直使用一个由 C语言 编写的定制函数包，该函数包遍历八叉树网格结构，以确定给定位置所在的叶块，然后对该位置的磁流体变量执行三线性插值，类似于三维中的规则网格插值器。Python 是一种解释型语言，在对大型数据数组进行搜索和数值计算时，其速度远慢于由 C语言 源代码编译而成的库，因此我们采用了跨语言方法来调用 C 程序，并将 C 函数的共享库链接到 Kamodo 中。

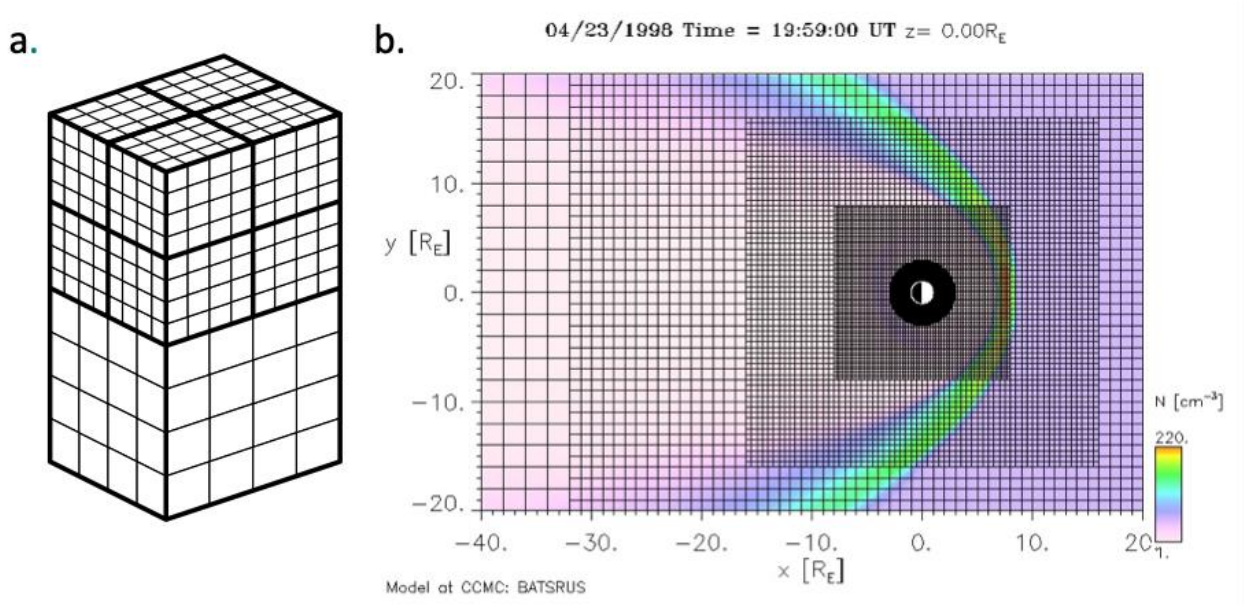


图 12： BATS-R-US 网格结构：(a) 构成八叉树网格的块（取自 Toth 等，2006 的图 1），该网格用于 SWMF BATS-R-US 模型的磁层（如图所示为笛卡尔坐标）、日冕和日球层（球坐标）。(b) CCMC 在线可视化服务对 CMCC 磁层模拟中等离子体（H+）数密度 N（颜色）和网格（黑色）的呈现，切片位于 Z = 0 R_E（地球半径）处。太阳风从右侧流来，在磁层（左侧较浅的颜色）和地球周围形成磁鞘（由绿色至红色表示）；地球位于原点，周围的黑色区域不属于仿真域。图中太阳位于右侧。

[21] 我们正与 SpacePy 开发者合作，为 PyBats.IdlFile() 脚本添加一个新关键字，防止二进制输出文件读取器自动按照物理位置重新排列彼此接近的网格位置，而是保留它们按块标识以及各块内逻辑单元位置 (i,j,k) 排列的顺序。我们已为 PyBats.IdlFile() 提交拉取请求，以添加关键字 *sort_unstructured_data*，使用户可以像以前使用二进制读取器时那样对数据排序，也可以保持数据不排序（更多信息请参见 https://github.com/spacepy/spacepy/pull/584）。

为兼容多个 SWMF/BATSRUS 版本，树结构仅利用原始格式化二进制输出文件中包含的位置进行重建，而不考虑在较新模型版本中被写入单独文件的树信息。该插值器在八叉树网格结构中执行搜索，然后在叶块内进行三线性插值，以获得所需的插值值。原则上，可以通过定义树搜索，并为每个量的每个块使用 SciPy 的规则网格插值器实例，在 Python 中重新实现这一过程。然而，对于大型网格，当块的数量超过 100,000 时，这将变得难以管理且效率低下，并且每个块都需要定义一个单独的插值器对象。

使用 Python 中的 cffi 包[22], ，可以轻松指定库中 Python 需要访问的 C 函数接口。现有源文件和构建脚本中定义的其他函数会打包到一个共享库中，然后像加载其他 Python 包一样加载它（例如下方脚注所引脚本的第 32 行）。此功能已在 Windows 和 Mac 计算机上成功测试，并已提供于当前的 CCMC Kamodo 版本[23], 中，正如 Kamodo 对函数化 B_z 变量的绘图所示（Rastaetter 等人，2022年和下方图13）。完成后，这项工作将扩展到具有类似网格的其他模型输出（例如 GUMICS 和 ARMS），并进一步推广到需要自定义插值例程的其他模型。此处介绍的用于纳入自定义插值脚本的方法，也可用于自定义文件读取器和转换例程（例如前文提到的 OpenGGCM 压缩二进制文件转换器）。

第*4*节：自定义可视化

通常，人们会先使用可视化方法来研究大型数据集，结果往往是每位用户针对每个数据集都编写复杂的脚本。Kamodo 只需一行代码，就能提供丰富的出版级交互式图形，大幅简化这一过程。虽然任何可视化软件包都可以集成到 Kamodo-core 软件包中，但在开发早期，由于 Plotly 当时提供的功能数量多、质量高，因此被选为默认的可视化软件包。Plotly 是一款免费的 Python 开源数据可视化工具包，可生成风格丰富且可自定义的图像。这些可视化内容具有高度交互性（可缩放、平移、探查和旋转），可以加入时间滑块，

[22] https://cffi.readthedocs.io/[23] 下方链接指向一个 Python 脚本，它在 Python 中使用 cffi 模块（可通过 pypi 获取）提供的外部函数接口来构建库。cdef() 方法用于定义 Python 需要了解的 C 数据类型和函数声明。set_source() 方法用于定义除 sources[] list 中列出的文件内代码之外所需的 C 源代码（以 // 开头的行是注释）。所需的外部库列在 libraries[] 中（此处：通过 -l 编译标志添加数学库 libm.so）。compile() 方法用于构建共享库文件（例如，在使用 Python 3.7 的 MacOS™ 上为_interpolate_amrdata.cpython-37m-darwin.so）。https://github.com/nasa/Kamodo/blob/master/kamodo_ccmc/readers/OCTREE_BLOCK_GRID/interpolate_amrdat a_extension_build.py

并且可以轻松嵌入网页或仪表板，同时保留其交互功能。

```
fig = kamodo_object.plot('B_z_ijk', plot_partial={'B_z_ijk': {'time': 1.2, 'Z': 15.7}})
fig.update_traces(colorscale="Viridis", ncontours=200, contours=dict(coloring="fill", showlines=False))
fig
```

```
Time slice index 1 added from file.
Time slice index 2 added from file.
```

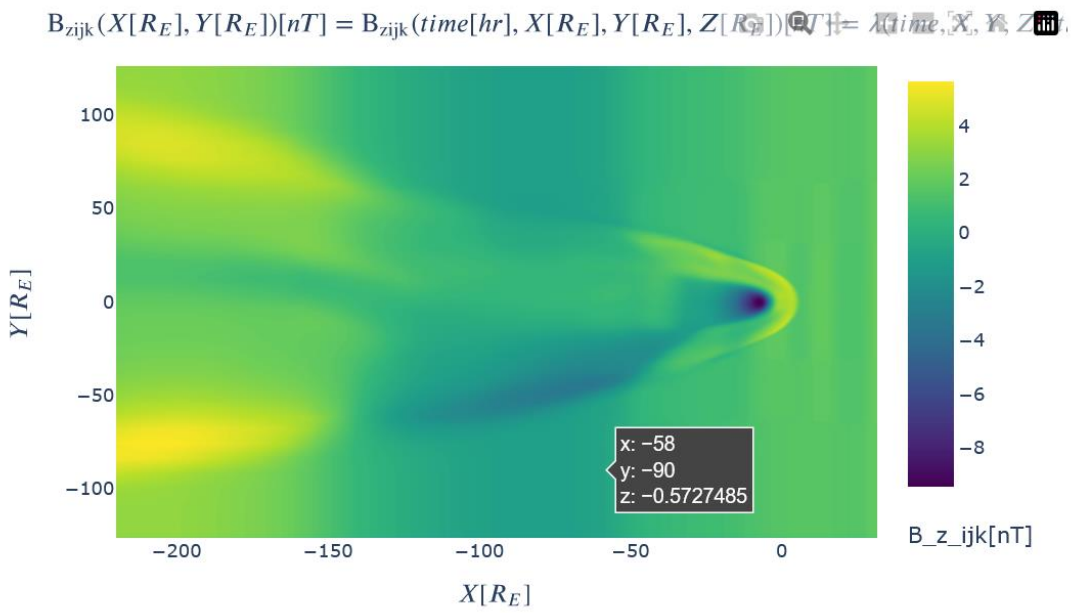


图13：Kamodo 对 SWMF 磁层输出中网格化 B_z 变量的可视化。B_z 变量的 LaTeX表示显示在顶部，右上方叠加了标准的 Plotly 按钮。所显示的交互式可视化展示了 2010年12月18日协调世界时午夜后 1.2 小时、Z 值为 15.7 R_E 时磁场的 z 分量，对应于*TreShunda_James_071322_1* 模型运行[24]。使用文中介绍的自定义插值器，对指定时间和 Z 位置处的显示值进行了插值。灰色框显示鼠标在绘图区域中的 X 和 Y 坐标，以及同一位置处的 B_z 值。坐标轴以 R_E 表示，颜色对应于右侧的颜色条。图右侧清晰显示了地球磁层对太阳风产生反应形成的弓激波。太阳风从右侧的太阳发出（未显示），而大致呈圆形的深色区域包含地球。该图使用与图9相同的命令生成，但增加了几行代码来更改颜色映射和绘图方法。“Time slice…”消息由底层的惰性插值逻辑生成。详情请参阅文档和正文。

Kamodo 基于 Plotly 的功能，为用户提供自动生成的绘图库。这个绘图函数内部注册表按照函数调用参数的形状和返回输出的形状进行索引。例如，如果请求绘制一个输入 N 个点并返回 N 个点的函数，就会生成一维线图（例如图14）。完整的默认绘图类型列表，请参阅

[24] 用于生成此图的数据可在 https://ccmc.gsfc.nasa.gov/RoR_WWW/output_files/KAMODO_DEMO/SWMF-GM/TreShunda_James_071322_1/ 获取。

Kamodo 可视化文档中的选项表及相关图示，其中列出了适用于各种函数签名的默认绘图类型[25]。

我们提供了一些示例，展示如何使用 Kamodo 将若干默认绘图类型应用于函数化科学数据。Kamodo 会注册数据，使用用户提供的插值将其函数化，并根据数据形状通过一条简单命令显示交互式图表。图14展示了一个简单示例：将包含 265 个时间戳的数组传递给一个函数，该函数会返回各个时刻的合成密度；示例中还包含了生成图表所需的命令。‘kamodo.plot’ 函数调用会返回一个对象，可对其进行显示、以多种方式导出，或进一步修改、组合以创建其他图形。作为组合 Plotly 对象的示例，图15展示了一个在 Kamodo 中返回的绘图对象（‘ror’ 变量），其数据来自模型输出（CCMC 按需运行产品[26]）；另一个绘图对象则来自数据源（GOES-12 数据，通过 HAPI（Weigel 等人，2021）接口从 CDAWeb 获取）（GOES：地球静止业务环境卫星，Menzel 等人，1994）。在 Jupyter 笔记本[27] （Kluyver 等人，2016）中，只需编写几行 Python 代码，即可组合这些图表、更新标签并显示生成的图形（另一种方法请参见图21和图22）。Plotly 文档[28]介绍了 Plotly 提供的其他选项。

如图9所示，借助 Kamodo 的 *partial_plot* 关键字，也可以使用默认的 Kamodo 绘图功能处理维度更高的数据集。只需一行代码，便可将四维数据集降至二维，并以交互式图表呈现，效果类似于文档中的图表。具体做法是：对未绘制的两个维度，按照给定的切片值对数据进行插值；对其余两个维度，则使用原始坐标网格（更多详情请参阅文档[29]）。然而，这些默认功能不足以帮助社区充分理解模型输出所呈现的数据，尤其是考虑到这些输出的维度很高。

虽然自动化绘图有助于不同技能水平的用户轻松生成有意义的图表，但 Kamodo 和 Plotly 的功能远不止于此。我们将通过一个案例研究，展示 Kamodo 如何沿合成卫星轨迹提取模拟输出中的数据，并借此介绍 Kamodo 中的几种自定义绘图选项。（有关此功能及相关功能的详细说明，请参阅 Ringuette 等人（2022）。）在本例中，我们使用合成卫星轨迹，从 CCMC 的 CTIPe 模型运行中提取数据。提取结果之一是沿卫星轨道的中性密度，称为“rho”。将其映射到我们的

[25] https://ensemblegovservices.github.io/kamodo-core/notebooks/Visualization/
[26] https://ccmc.gsfc.nasa.gov/tools/runs-on-request/
[27] https://docs.jupyter.org/en/latest/
[28] https://plotly.com/python-api-reference/
[29] https://ensemblegovservices.github.io/kamodo-core/API/#multi-function-plots

```python
from kamodo import Kamodo, kamodofy
import numpy as np
import pandas as pd
from plotly.offline import iplot
```

```python
t_N = pd.date_range('Nov 9, 2018', 'Nov 20, 2018', freq = 'H')

@kamodofy(units = 'kg/m**3')
def rho_N(t_N = t_N):
    dt_days = (t_N - t_N[0]).total_seconds()/(24*3600)
    return 1+np.sin(dt_days) + .1*np.random.random(len(dt_days))

kamodo = Kamodo(rho_N = rho_N, verbose = False)
```

```python
fig = kamodo.plot('rho_N')
iplot(fig)
```

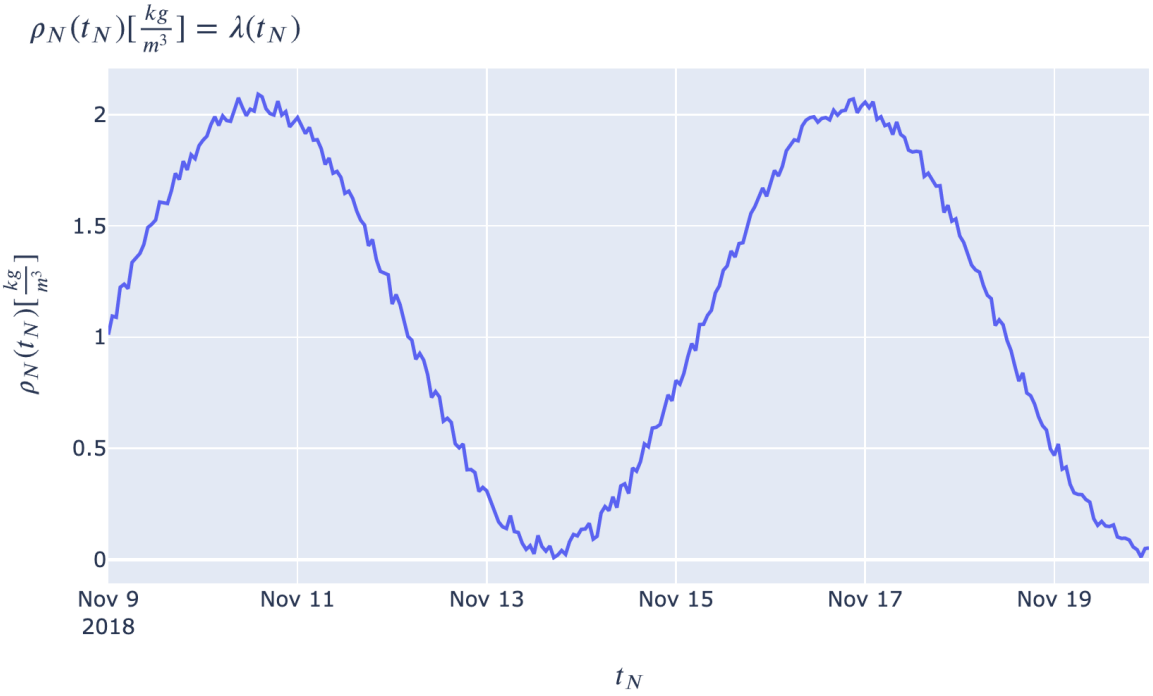


图14： 用于根据输入和输出形状 N 生成一维线图的 Python 代码。第一段代码展示导入语句，第二段演示如何将一个一维示例数据集函数化，第三段则展示了在底部生成图表的简单命令。请注意，图表左上方显示了函数化变量的 LaTeX 表示，坐标轴也已自动标注。该图表支持完整的交互功能，包括缩放、平移等（未显示）。

自动化绘图映射中，该函数需要一个 (N,3) 维输入（位置），并生成一个 (N) 维输出（密度），时间则作为额外信息。通常，这会生成一张显示三维热图的标准图像，但我们可以通过多种方式对其进行自定义。位置分量可以轻松拆分出来，并以多种一维图的形式呈现，不过由于

Kamodo 还可利用 SpacePy 和 AstroPy 的扩展功能进行坐标转换，从而创建更复杂的自定义绘图。如图 16 所示，同一组数据可以显示为二维等高线图，并将位置映射到球形地心赤道惯性坐标系中的经度和纬度。有关此处所用坐标系的说明，请参阅 SSCWeb[30]。

```
fig = ror.get_plot(type = "1Dvar", var = "B_z")
fig2 = hapi2.get_plot(type = "1Dvar", var = "B_GSE_c")
fig.add_trace(fig2.data[2])
fig.update_yaxes(title_text='nT')
fig['data'][0]['name'] = 'B_z (MODEL)'
fig['data'][1]['name'] = 'B_z (DATA)'
iplot(fig)
```

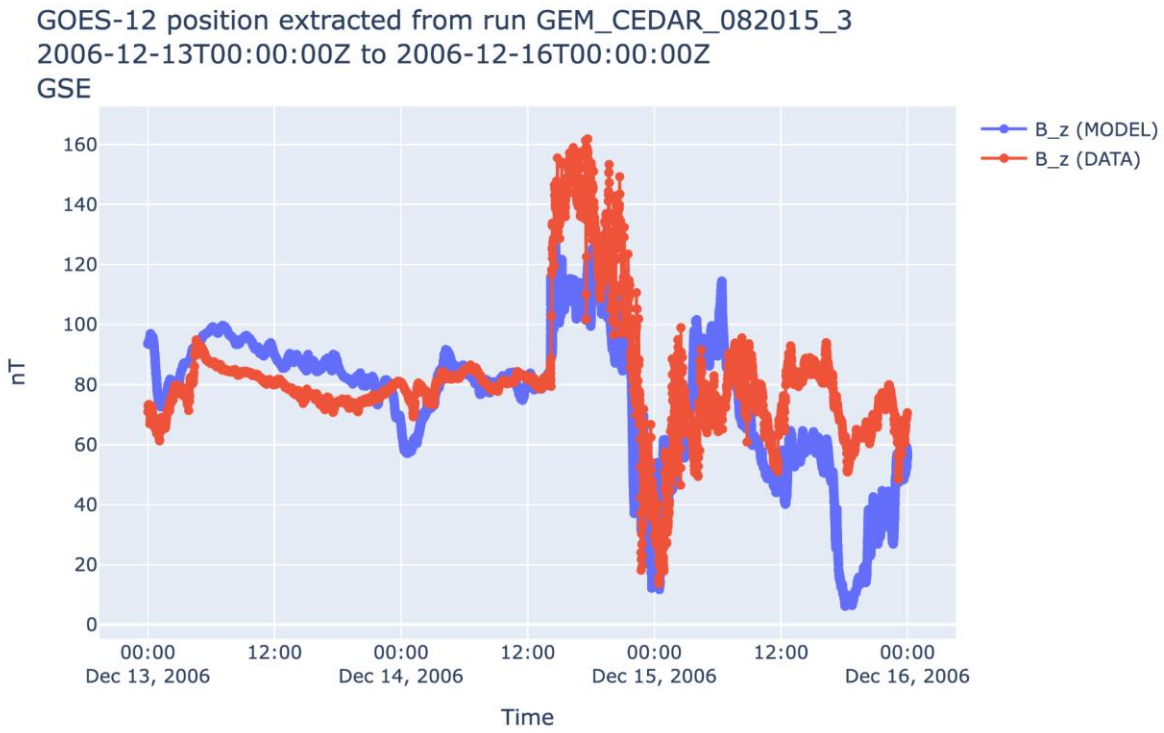


图15：用于组合两个 Kamodo 绘图对象（顶部）并显示更新后图形（底部）的 Python 代码。‘fig2.data[2]’ 从该绘图对象的 B 向量的三个分量中选取 B_z 值。用于生成该图的数据提取自 GEM_CEDAR_082015_3运行，其可通过 CCMC 的按需运行服务[31]获取。

图上方的 Python 代码展示了该绘图程序的语法。在显示的代码块中，执行了两条导入语句，随后使用 *SFcsv_reader* 函数将 CSV 文件中的数据读入嵌套字典，并使用 *SatPlot4D* 函数创建

[30] https://sscweb.gsfc.nasa.gov/users_guide/Appendix_C.html[31] ht

并显示完全交互式的绘图，这两项操作均只需一条命令即可完成。‘cdf_dict’变量的各个组成部分是一维时间序列数组，包含时间戳（‘utc_time’）、位置信息（‘c1’、‘c2’和‘c3’），以及带单位的变量计算结果（‘rho’）。命令中使用的字典元数据部分包含三个值：所用模型的名称和坐标系信息。用户可以从多种选项中进行选择，例如绘图坐标系、标签、轨迹切片方法（例如图16和图17中的‘all’，以及图18中的‘orbitE’），还可以选择将绘图保存为 HTML 文件，同时在文件中保留完整的交互功能。‘type’关键字选择‘二维’时，会生成图16所示的绘图。更改此关键字的值，则可生成同一数据的其他表示形式，例如图17（‘2DPN’）和图18（‘三维’）所示。更多信息请参阅文档。

```
from kamodo_ccmc.flythrough.wrapper_output import SFcsv_reader
from kamodo_ccmc.flythrough.plots import SatPlot4D
cdf_dict = SFcsv_reader('FakeFlightExample_CTIPe20s.csv')
var = 'rho' # variable name
SatPlot4D(var, cdf_dict['utc_time']['data'], cdf_dict['c1']['data'],
          cdf_dict['c2']['data'], cdf_dict['c3']['data'],
          cdf_dict[var]['data'], cdf_dict[var]['units'],
          'GDZ', 'sph', 'GEI', 'all', 'CTIPe', type = '2D',
          htmlfile = 'Figure15.html')
```

```
-saving full html file:  Figure15.html
```

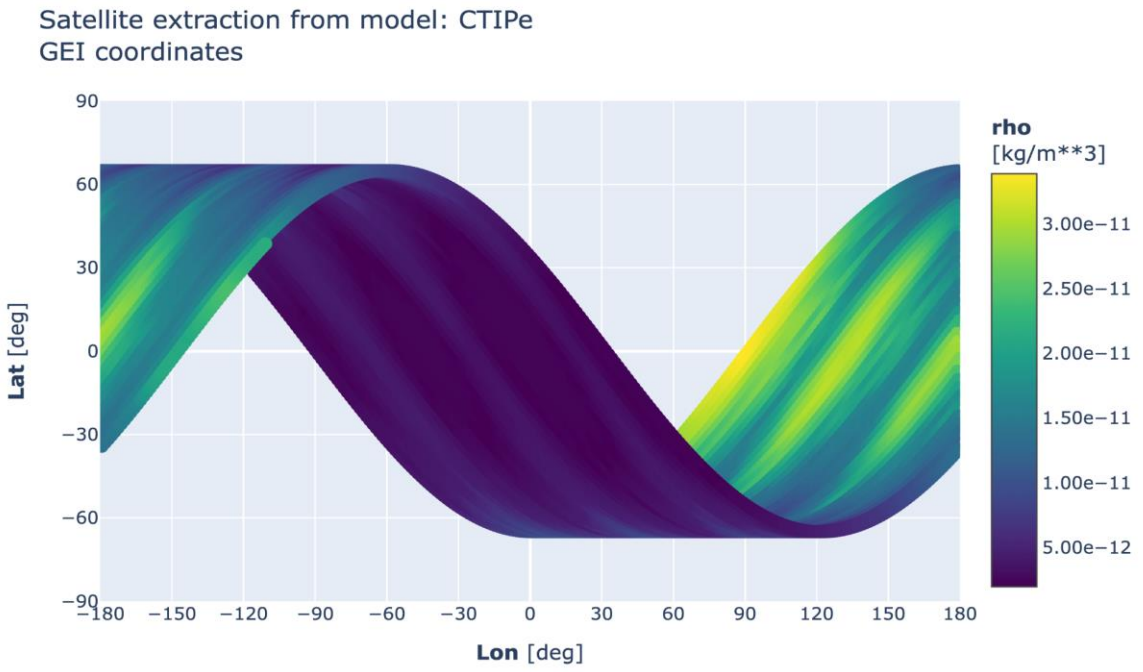


图16：将完整轨迹映射到纬度和经度位置，并按密度着色。绘图上方代码块中的代码展示了生成该图所需的语法。数据通过一条命令从文件中读入，然后传递给绘图命令，生成底部所示的完全交互式图表。卫星的位置以 GEI 球坐标表示，该坐标由大地坐标系球坐标自动转换而来。完全交互式图表会显示在笔记本中，同时保存为 HTML 文件“Figure15.html”，之后可供查看或放置在网页上。

科学家经常使用的另一种视角是从北极上方向下俯瞰。图17展示了地理坐标系（GEO）下北半球的密度值和卫星轨迹，并以地球大陆轮廓作为背景。图中还显示了一个信息框，悬浮在绘图中鼠标指针所在的位置。Plotly 图像具有交互性，可显示当前光标位置附近数据点的其他元数据。本例中显示了 GEO X、Y 和 Z 位置，大地坐标系（GDZ）下的经度、纬度和高度（lon、lat、alt）位置，密度值及其单位，以及协调世界时（UTC）时间戳。

请注意，虽然这里展示的是静态图像，但实际绘图可以通过缩放查看特定区域；对于三维图像，还可以旋转视角。这些图使用同一个 Kamodo 绘图工具函数创建，只对几个调用参数做了少量调整。虽然它们展示了数据可视化方式的多样性，也加入了背景图像以提供上下文，但仍未体现时间的动态变化。为了以更具交互性的方式纳入这一维度，我们使用 Plotly 功能直接支持的滑块，在时间轴上滚动浏览。还可以配置播放/暂停按钮来启动动画。图18展示了这种显示方式的一个示例，只需对图16和图17中使用的绘图命令稍作修改即可实现。

当然，即使是初学者，也能在相当程度上自定义这些可视化。在 Jupyter 笔记本或 Python 脚本中，Kamodo 可以返回一个对象，以便显示或在显示前进一步修改。在上方图14的示例中，返回了一个名为‘fig’的对象，随后将其显示出来。可以通过简单的 Python 调用修改此对象，相关用法详见内容全面的 Plotly 在线文档；也可以直接操作对象。

图19和图20给出了两个高级自定义示例。图19是在 Kamodo 默认图的基础上进行自定义后生成的，图20则是在 SatPlot4D 默认图中叠加了一些磁矢量高级自定义内容。图19展示了一幅高度自定义的图，其中 GRACE-1卫星[32] （Wahr 等，2004年）穿越了 CCMC 的一次 GITM 模拟运行（轨迹取自 SSCWeb）。选择这一模型和卫星作为一组，是因为二者的高度范围有所重叠。GITM 模拟地球周围约 90 千米至 600 千米高度范围内的电离层-热层系统，而 GRACE-1卫星的轨道高度约为 400 千米至 500 千米。从 GITM 输出中提取了卫星位置（取自 SSCWeb），并在实际卫星位置周围一个 +/- 100 千米的范围内额外提取了八组数据。最小值和最大值以数值包络的形式显示，用来表示位置误差条。图中还使用单独的 Y 轴刻度（右轴）显示卫星位置。将鼠标悬停在图上，即可显示指定时间的所有数值。

[32] https://www.nasa.gov/mission_pages/Grace/index.html

```
var = 'rho' # variable name
SatPlot4D(var, cdf_dict['utc_time']['data'], cdf_dict['c1']['data'],
          cdf_dict['c2']['data'], cdf_dict['c3']['data'],
          cdf_dict[var]['data'], cdf_dict[var]['units'],
          'GDZ', 'sph', 'GEO', 'all', 'CTIPe', type = '2DPN',
          body = 'lines', zoom = False,
          htmlfile = 'Figure16.html')
```

```
-saving full html file:  Figure16.html
```

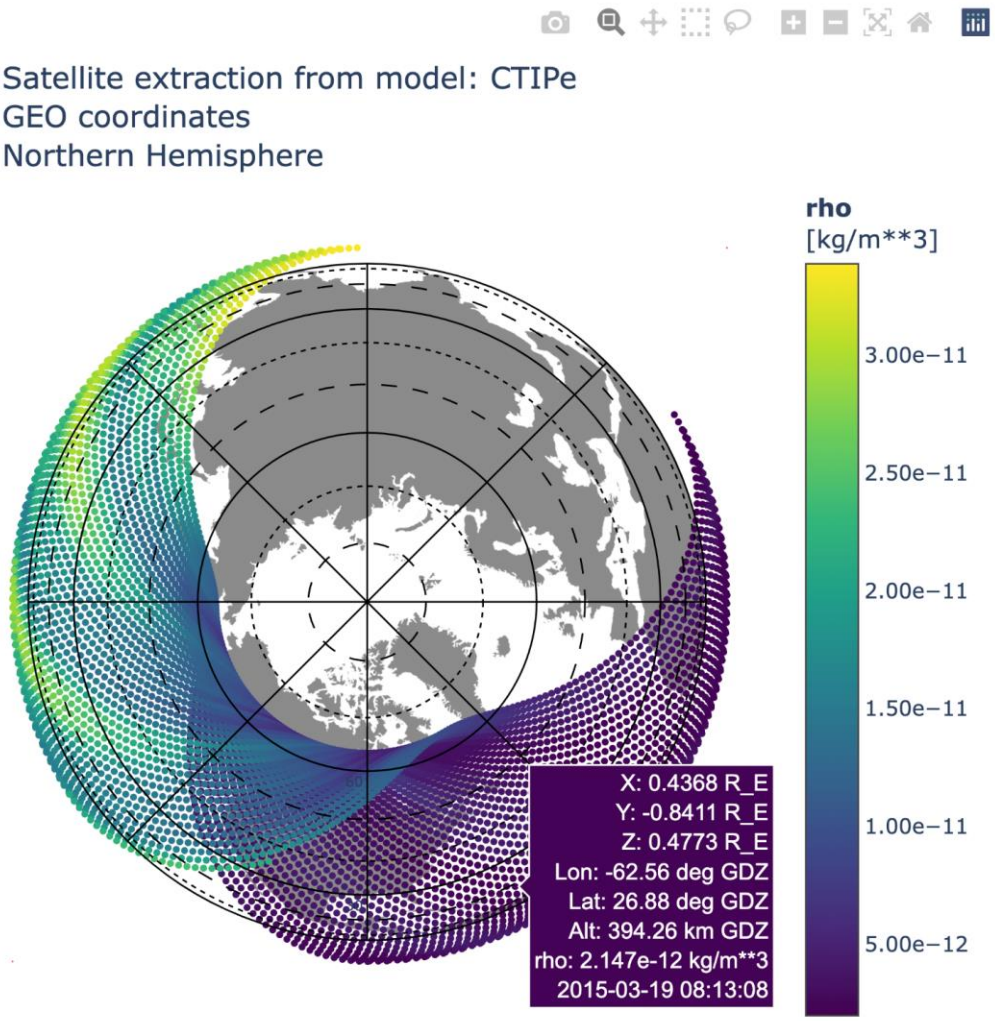


图17：从北极视图中显示的位置和密度值，以及其他悬停元数据。绘图上方显示的代码使用了与图16中相同的检索数据。绘图命令（使用 *SatPlot4D*）与图16中的几乎相同，仅更改了少数关键字值以生成所示绘图。卫星的位置以地球同步轨道坐标显示，以便在绘图下方添加陆地/水域图层，作为位置参考。

这些 Plotly 可视化还提供了另一项非常实用的功能：可以将其保存为 HTML 文件，从而在网页浏览器中将其用作动态绘图。生成的文件

包含复现该图形并在客户端与其交互所需的全部数据。若要查看这些图形的交互特性，可在线查看图16、图17和图18[33]。

```
var = 'rho' # variable name
SatPlot4D(var, cdf_dict['utc_time']['data'], cdf_dict['c1']['data'],
          cdf_dict['c2']['data'], cdf_dict['c3']['data'],
          cdf_dict[var]['data'], cdf_dict[var]['units'],
          'GDZ', 'sph', 'GSE', 'orbitE', 'CTIPe', type = '3D',
          body = 'none', htmlfile = 'Figure17.html')
```

```
-saving full html file:  Figure17.html
```

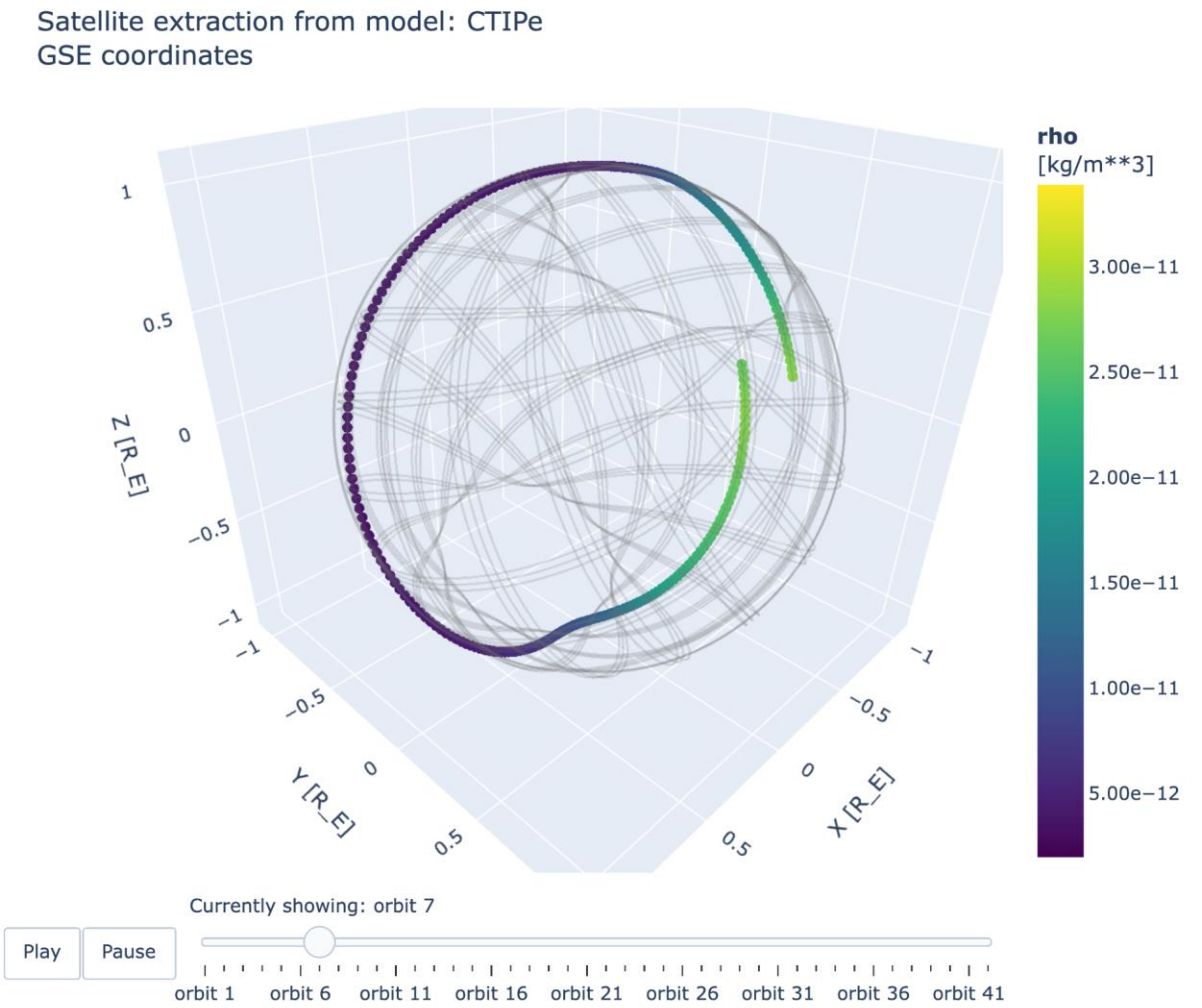


图18： 当卫星沿北向穿过GSE坐标中的赤道时，完整的卫星轨迹会被分割成多个轨道。可以播放动画，也可以拖动底部的轨道滑块来查看动画。请注意，图上方所包含的绘图命令使用了与图16中检索到的相同数据，其数值也几乎完全相同。

[33] https://ccmc.gsfc.nasa.gov/Kamodo/Figures/

还可以为磁层中的矢量制作更多自定义图形，并且很快也将支持其他领域。图20展示了根据GOES-13卫星（围绕地球运行的地球同步轨道）模型输出计算得到的磁场值（Hillger和Schmit，2009）。利用GOES-13卫星的位置，从CCMC的一次全球磁层运行中提取数值。随后可在Kamodo中以多种坐标系对这些提取结果进行可视化。还可以添加或移除磁场矢量，并调整其采样频率。

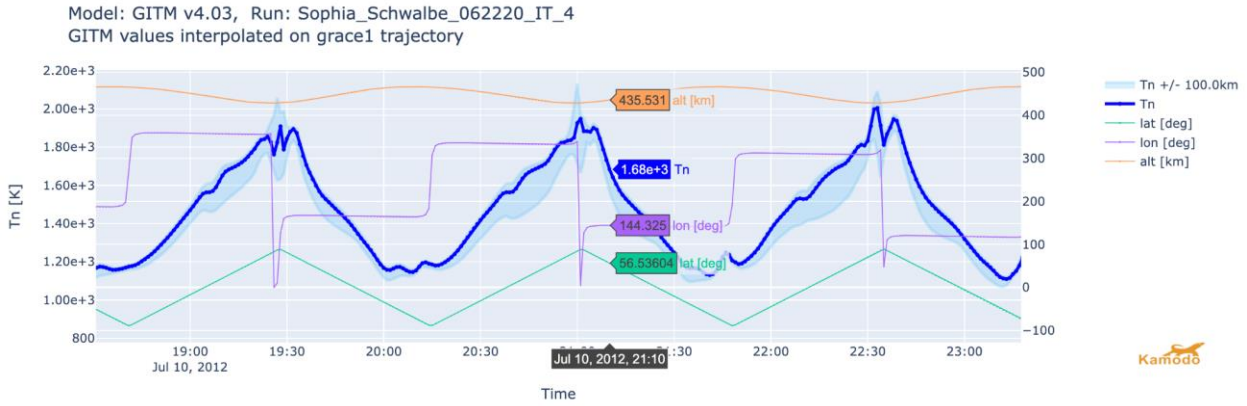


图19： 定制图表，叠加显示多个折线图，用于提取 GITM模型一次模型运行中沿 GRACE-1卫星轨迹的中性温度（Tn）数值。深蓝色线条表示中性温度（左侧坐标轴），浅蓝色阴影表示根据文中所述的额外提取结果确定的位置误差，其余三条线表示卫星在大地坐标中的位置（右侧坐标轴上的纬度、经度和高度）。彩色方框显示图中各变量的颜色编码数值，并会随用户鼠标移动而更新。此图所用数据可通过 CCMC 的按需运行服务获取[34]。

第5节：比较不同模型的数据

结合迄今讨论的各项功能，可以得到一种强大而简洁的方法，即使面对不同的文件格式、各异的坐标网格和定制的插值方法，也能直接比较不同模型的结果。图21以某个磁暴日的 CTIPe 和 GITM 模型所模拟的离子温度为例，展示了如何用极少的代码实现这一点。只需一条简单的导入语句，在前两个代码块中用几行代码即可将各模型的离子温度数据函数化，而且两个代码块的语法完全相同。随后，在第三个代码块中将这些函数汇集到一个 Kamodo 对象中。图 22 中使用该 Kamodo 对象，在相同的时间和高度切片上绘制两个数据集，以便直接进行视觉比较。生成这两幅图使用的代码与图9所示的单行代码相同，每幅图后还各有一行代码，用于展示一些简单的修改，从而获得更平滑的可视化效果。通过模型读取器，Kamodo 已纳入的任意模型中的任意变量，都可以采用这一示例工作流程进行绘图和比较。

[34] https://ccmc.gsfc.nasa.gov/results/viewrun.php?domain=IT&runnumber=Sophia_Schwalbe_062220_IT_4

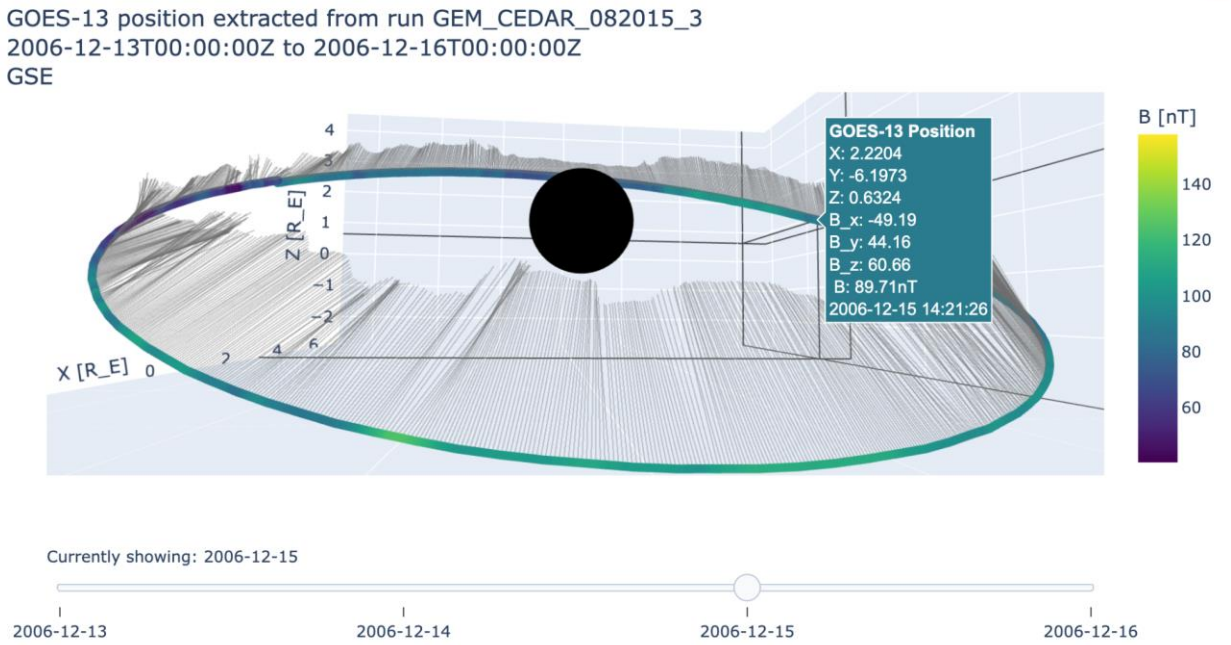


图20： 自定义图表，显示从 CCMC 全球磁层运行中提取的 GOES-13 卫星位置数据。图表按天分段，可使用滑块一次选择一天的数据。颜色表示该位置磁场的大小。黑色尖峰表示沿卫星轨道每隔 10 个点显示的磁场矢量。将鼠标悬停在图表上会显示该位置的精确数值。生成此图表所用的数据集与图15中的相同。

第*6*节：总结

上述模型读取器结构为日球物理学界带来了多项重要功能。对于编程经验不多的用户而言，如今他们首次拥有了一种简单、直观且与模型无关的方法，无需了解各种文件格式的复杂性，也无需处理其中必然涉及的插值，就能在研究中访问和利用多个基于物理的模型及经验模型的模型数据。基于 Kamodo 构建这些软件功能，进一步简化了通过该软件包提供的其他功能，对多个模型输出中的这些数据进行分析和比较的过程。使用 Kamodo 的另一项重要优势，是它简化了与其他 Python 软件包的互操作（两个示例见 Polson 等人（2022）和 Ringuette 等人（2022））。我们指出，这项工作很容易扩展到其他领域的观测数据和模型数据。感兴趣的读者可参阅 NASA Kamodo GitHub 仓库中的示例，尤其是《Kamodo 入门指南》文档，以便为自己的数据集实现同样的功能。用户可访问 CCMC 网站[35]，获取大量模型的日球物理学模型数据。

[35] https://ccmc.gsfc.nasa.gov/

A

```
import kamodo_ccmc.flythrough.model_wrapper as MW
reader = MW.Model_Reader('CTIPe')
kamodo_object_ctipe = reader(ctipe_file_dir, variables_requested=['T_i'])
kamodo_object_ctipe
```

$$\mathrm{T_i}\left(\vec{r}_{GDZsph4D}\right)[K] = \lambda\left(\vec{r}_{GDZsph4D}\right)$$
$$\mathrm{T_{ijk}}(time[hr], lon[deg], lat[deg], height[km])[K] = \lambda(time, lon, lat, height)$$

B

```
# Retrieve model reader and access model data for the GITM model.
# Note the identical syntax!
reader = MW.Model_Reader('GITM')
kamodo_object_gitm = reader(gitm_file_dir, variables_requested=['T_i'])
kamodo_object_gitm
```

$$\mathrm{T_i}\left(\vec{r}_{GDZsph4D}\right)[K] = \lambda\left(\vec{r}_{GDZsph4D}\right)$$
$$\mathrm{T_{ijk}}(time[hr], lon[deg], lat[deg], height[km])[K] = \lambda(time, lon, lat, height)$$

C

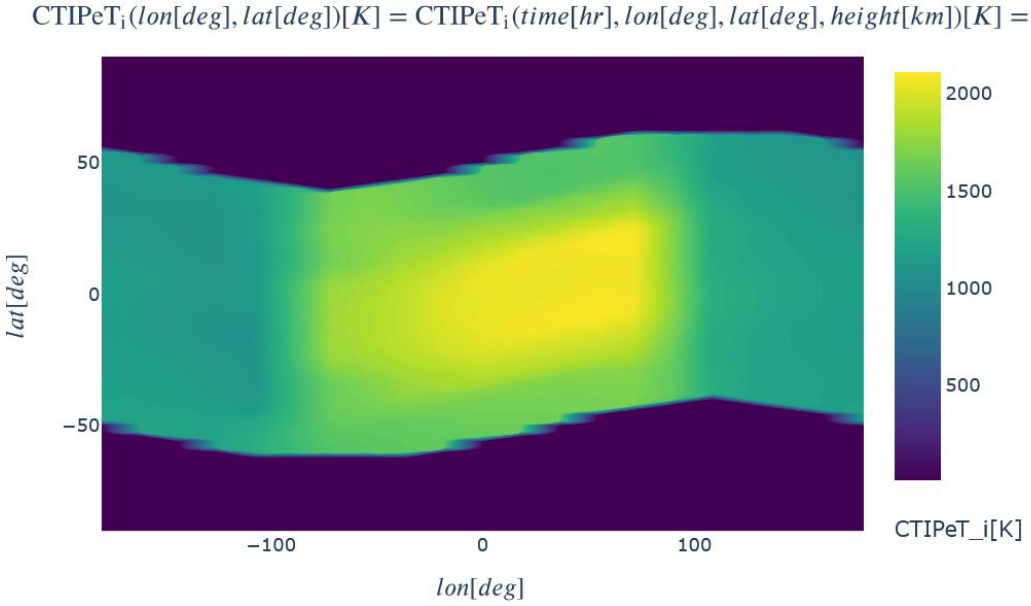
```
# Collect functions into a single Kamodo object.
# Note the names of the coordinates each variable depends on.
from kamodo import Kamodo
kamodo_object = Kamodo()
kamodo_object['CTIPeT_i[K]'] = kamodo_object_ctipe['T_i_ijk']
kamodo_object['GITMT_i[K]'] = kamodo_object_gitm['T_i_ijk']
kamodo_object
```

$$\mathrm{CTIPeT_i}(time[hr], lon[deg], lat[deg], height[km])[K] = \lambda(time, lon, lat, height)$$
$$\mathrm{GITMT_i}(time[hr], lon[deg], lat[deg], height[km])[K] = \lambda(time, lon, lat, height)$$

图21： 直接比较模拟结果的示例工作流程第一部分。该图展示了如何用函数形式表示来自两个不同模型的模拟数据。在本例中，CTIPe 和 GITM 模型的模拟离子温度数据采用相同的语法函数化（图 A 和 B），随后合并到同一个 *kamodo_*对象变量中（图 C）。两个数据集将在下一幅图中绘出。此工作流程可通过模型读取器进行调整，适用于 Kamodo 中包含的任意变量和模型。

目前，磁层领域的多个模型读取器正在开发中，其中包括尚待完成的 SWMF（全球磁层部分，Toth 等，2007）模型读取器，其他模型读取器也在独立开发。针对 SWMF/BATS-R-US 模型读取器设计的自定义插值器，将应用于正在开发的 GUMICS 和 ARMS 模型输出读取器，因为它们具有相似的模型数据网格结构。ARMS 模型读取器将是我们首次涉足太阳日冕领域。模型读取器结构和自定义惰性插值设计也将更广泛地应用于通过合作开发的模型，例如 GAMERA（面向扩展研究应用的网格无关磁流体力学，Zhang 等，2019）、VERB（多功能电子辐射带代码；Wang 等，2019）、MARBLE（磁层极光重联边界层探测器，Bard 和 Dorelli，2021）以及 RAM-SCB（具有自洽磁场的环电流—大气相互作用模型；Zaharia 等，2010）。CIMI（综合内磁层—电离层模型，Fok 等，2014）的初步开发也已启动，它是首个用于漂移动力学模型输出的模型。

```
fig = kamodo_object.plot('CTIPeT_i', plot_partial={'CTIPeT_i': {'time': 12., 'height': 500.}})
fig.update_traces(colorscale="Viridis", ncontours=200,
                  contours=dict(coloring="fill",showlines=False))
fig
```




```
fig = kamodo_object.plot('GITMT_i', plot_partial={'GITMT_i': {'time': 12., 'height': 500.}})
fig.update_traces(colorscale="Viridis", ncontours=200,
                  contours=dict(coloring="fill",showlines=False))
fig
```

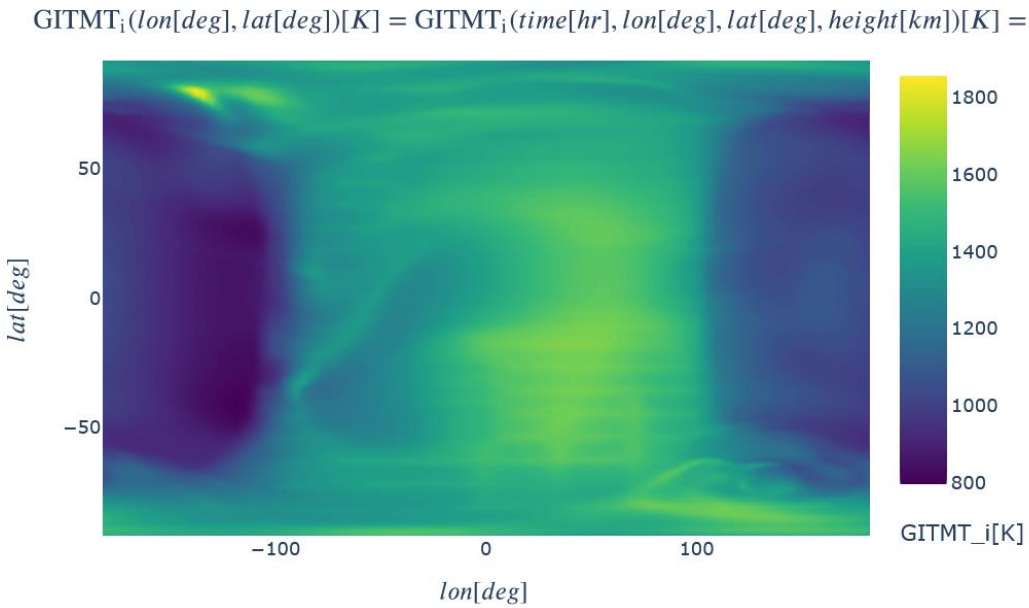


图 22：直接比较模拟结果的示例工作流程的第二部分。在这一部分中，我们使用相同的语法，为同一天的 CTIPe（上）和 GITM（下）模型生成模拟离子温度数据的绘图。惰性插值消息仅在加载给定时间切片时首次打印，因此此处未显示。通过模型读取器在 Kamodo 中包含的任何变量和任何模型，都可以采用此工作流程。

在我们努力扩展 Kamodo 所涵盖的模型和领域库的同时，也在致力于拓展 Kamodo 的实用性，使其同时服务于普通用户和 CCMC 应用。通过模型读取器实现的独特模型无关能力，目前正成为进一步开发其他功能的基础，例如一组同样简洁的穿越函数，可让给定轨迹"穿越"一组模型数据，从而大幅简化涉及观测数据与模型数据比较的分析；以及用于研究卫星星座轨迹和排列方式的工具（将在未来的工作中介绍）。针对各种科学应用的示例工作流正在陆续完成并添加到 NASA Kamodo Github，用户可以自由地根据自身需求进行修改（也可以贡献自己的工作流）。这些工作流不仅基于 Kamodo 的功能，还包含社区常用的其他 Python 软件包（例如 Polson 等人（2022））。

在 CCMC，针对部分模型，使用 Kamodo 为 CCMC即时运行界面提供支持的初步工作已经开始；同时，正在规划 CCMC 按需运行可视化的示例用法，其中将包括一组选定的派生变量。我们还计划开发适用于以不同语言编写的模型代码的单向和双向模型耦合功能，这将依赖于 Ensemble 即将推出的新型交互式跨语言Kamodo接口。这种全新的即插即用模型耦合方法有望大幅简化模型耦合问题，其简化程度将与本文展示的直接访问模型数据的简化效果相当。此外，我们正在研究如何在 Kamodo 的穿越探测能力之上添加 HAPI接口，并最终将其添加到模型读取器本身之上。我们的开发合作伙伴 Ensemble 正在开发一种系统，可自动将模型输出容器化，并通过亚马逊云科技云端，利用适当的 CloudFormation[36] 脚本实现容器的自动部署。从长远来看，我们计划向 CCMC 用户提供这些功能，以支持在云端对模型输出进行缩减和可视化，从而减少用户需要下载的数据量。

尽管受其源自美国国家航空航天局的背景所带来的限制，将这款软件以开源代码的形式维护至关重要，这样才能与日球物理学界的成员——包括商业、政府和学术界人士——开展合作，改进并扩展这一资源。虽然目前这项功能仅适用于少数模型，但我们正与社区合作，将更多模型纳入这一资源。若读者有意开展此类合作，欢迎联系作者。即使在当前，Kamodo 的应用范围和实用性也在不断拓展。我们正致力于应用这项技术，为日球物理学和空间天气领域打造新一代科学软件。

[36] https://aws.amazon.com/cloudformation/

致谢

作者感谢审稿人迄今为本文重大改进提供的指导，并感谢 C. Wiegand 对本文提出的反馈。社区协调建模中心（CCMC，https://ccmc.gsfc.nasa.gov/）使用了空间物理数据设施（SPDF，https://spdf.gsfc.nasa.gov/）的多项服务，包括文中提及的 SSCWeb 和 CDAWeb，以获取模型的输入数据，并与正在执行及已完成的空间科学任务开展模型数据比较。Kamodo-core 功能的开发工作获得了美国国家航空航天局小企业创新研究第二阶段：空间天气研究到运行／运行到研究技术开发资助的支持，该资助项目名为“Kamodo容器化空间天气模型”（合同编号 #80NSSC20C0290），授予 Ensemble Consultancy。

参考文献

AMGeO合作组织（2019）。面向地球空间社区的协作式数据科学平台：地球空间观测同化制图（AMGeO）v1.0.0。Zenodo。https://doi.org/10.5281/zenodo.3564914。

Astropy合作组织（2018）。Astropy项目：构建开放科学项目及 v2.0 核心软件包现状。《天文杂志》，**156**，*3*，123。https://doi.org/10.3847/1538-3881/aabc4f。Bard, C. 和 J. Dorelli（2021）。类地磁层中的离子尺度磁尾重联不对称性。*Ann. Geophys*。**39**，991–1003。https://doi.org/10.5194/angeo-39-991-2021。Barnum, J.、A. Masson、R. H. W. Friedel、A. Roberts 和 B. A. Thomas（2022）。日球物理学Python社区（PyHC）：现状与未来展望。《空间研究进展》。已接受。https://doi.org/10.1016/j.asr.2022.10.006。Bilitza, D.（2018）。IRI：国际电离层标准。*Adv. Radio Sci*，**16**，1-11，https://doi.org/10.5194/ars-16-1-2018。Bruinsma, S（2015）。DTM-2013热层模型。《空间天气与空间气候杂志》 **5**，A1。https://doi.org/10.1051/swsc/2015001。Codrescu, M. V.、Fuller-Rowell, T. J.、Munteanu, V.、Minter, C. F. 和 Millward, G. H.（2008）。耦合热层-电离层-等离子体层电动力学模型的验证：CTIPe与质谱仪非相干散射温度比较。《空间天气》，**6**，S09005，https://doi.org/10.1029/2007SW000364。Cousins, E. D. P. 和 S. G. Shepherd（2010）。基于SuperDARN等离子体漂移测量得出的高纬度对流动力学模型。《地球物理研究杂志》，**115**，A12329，https://doi.org/10.1029/2010JA016017。Dask开发团队（2016）。Dask：用于动态任务调度的[illegible]、Millward[illegible]气环境。《空间天气》 20，e2022SW[illegible] 2022SW003193。Fok, M.-C.、N. Y. Buzulukova、S.-H. Chen、A. Glocer、T. Nagai 等（2014）。综合内磁层-电离层模型。*J. Geophys. Res. Space Physics*，**119**，7522-7540，https://doi.org/10.1002/2014JA020239。Hillger, D. W. 和 T. L. Schmit（2009）。GOES-13科学测试。《美国气象学会公报》，**90**，5。https://www.jstor.org/stable/26220978。Hoyer, S. 和 Hamman, J.（2017）。xarray：Python中的N维带标签数组和数据集。《开放研究软件期刊》 **5**，*1*，10。DOI：https://doi.org/10.5334/jors.148。

Huba, J. 和 J. Krall（2013）。使用SAMI3对等离子体层建模。地球物理研究快报，**40**，1，6-10。https://doi.org/10.1029/2012GL054300。Janhunen, P.、M. Palmroth、T. Laitinen、I. Honkonen、L. [illegible] *hys.*，**80**，48，https://doi.org/10.1016/j.jastp.2012.05.006。Liu, H.-L.、C. G. Bardeen、B. T. Foster、P. Lauritzen 等（2018）。全大气社区气候模型热层和电离层扩展版（WACCM-X 2.0）的开发与验证。*Journal o* [illegible]。Kluyver, [illegible] Notebooks——一种用于可复现计算工作流的出版格式。《学术出版中的定位与权力：参与者、代理人与议程》，87-90，IOS Press电子书，美国弗吉尼亚州克利夫顿。http://dx.doi.org/10.3233/978-1-61499- 649-1-87。Maruyama, N.、Sun, Y.-Y.、Richards, P. G.、Middleoff, J.、Fang, T.-W.、Fuller-Rowell, T. J. 等（2016）。新的电离层-等离子体层模型揭示了中纬度电离层峰值密度结构的一种新来源。Geophys. Res. Lett. 43。https://doi.org/10.1002/2015GL067312。Menzel, W. P. 和 J. F. W. Purdom（1994）。介绍GOES-I：新一代地球静止业务环境卫星的首颗卫星，《美国气象学会公报》，**75**，5，757-782。https://doi.org/10.1175/1520-0477(1994)075<0757:IGITFO>2.0.CO;2。Meurer, A.、Smith, C.P.、Paprocki, M.、Čertík, O.、Kirpichev, S.B. 等（2017）SymPy：Python中的符号计算。*PeerJ*计算机科学，3:e103 https://doi.org/10.7717/peerj-cs.103。Morley, S. K.、Welling, D. T.、Koller, J.、Larsen, B. A.、Henderson, M. G. 等（2011）。SpacePy——基于Python的空间科学工具库。https://doi.org/10.25080/Majora-92bf1922-00c。Pembroke, A.、De Zeeuw, D.、Rastaetter, L.、Ringuette, R.、Gerland, O. 等（2022）。Kamodo：用于空间天气模型和数据的函数式API。*JOSS*，**7**，*75*，4053，https://doi.org/10.21105/joss.04053。Peterson, P.（2009）。F2PY：连接Fortran和Python程序的工具。《国际计算科学与工程杂志》，**4**，4，296–305。https://doi.org/10.1504/IJCSE.2009.029165。Plotly Technologies Inc.（2015）协作式数据科学。*Plotly Technologies Inc.*，魁北克省蒙特利尔。https://plot.ly。Polson, S.、Ringuette, R.、Zheng, Y.、Neihof, J.、Grimes, E. 和 Murphy, N.（2022）。与日球物理学Python社区合作开发可执行论文。地球与空间科学开放档案，https://doi.org/10.1002/essoar.10510006.1。Qian, L.、Burns, A.、Emery, B.、Foster, B.、Lu, G. 等（2013）。美国国家大气研究中心TIE-GCM：耦合热层/电离层系统的社区模型。地球物理专著系列 **201,** 73-83，https://doi.org/10.1029/2012GM001297。Raeder, J.、Wang, Y. L.、Fuller-Rowell, T. J.、Singer, H. J.（2001）。巴士底日风暴空间天气效应的全球模拟。*Solar Phys.*，**204**，325。https://doi.org/10.1023/A:1014228230714。Rastaetter, L.、R. Ringuette、D. De Zeeuw 和 O. Gerland（2022）。使用Kamodo在内磁层中进行磁映射。在2022年美国地球物理联合会秋季会议上报告，12月12日至16日，美国伊利诺伊州芝加哥。摘要编号：SH42E- 2338。https://doi.org/10.22541/essoar.167214301.16153548/v1。Ridley, A.、Deng, Y.、Tóth, G.（2006）。全球电离层-热层模型（GITM）。*Journal of Atmospheric and Solar-Terrestrial Physics*，**68**，839-864，https://doi.org/10.1016/j.jastp.2006.01.008。Ringuette, R.、D. De Zeeuw、L. Rastaetter、A. Pembroke、O. Gerland 和 K. Garcia-Sage（2022）。Kamodo的模型无关卫星穿越探测：降低日球物理模型输出的使用门槛。*Frontiers inAstronomy and Space Science: Space Physics*，已接受。https://doi.org/10.3389/fspas.2022.1005977。Robinson, R. M.、L. Zanetti、B. Anderson、S. Vines 和 J. Gjerloev （ 2021 ）。利用 AMPERE 沿磁力线电流测量确定极光电动力学参数。《空间天气》，**19**，

使用Pyt [illegible]

Stoneback, R. A., Burrell, A. G., Klenzing, J., & Depew, M. D. (2018). PYSAT：Python卫星数据分析工具包。《地球物理研究杂志：空间物理》, 123, 5271–5283. https://doi.org/10.1029/2018JA025297. Toth, G., De Zeeuw, D. L., Gombosi, T. I., Manchester, W. B., Ridley, A. J., et al. (2007). 使用空间天气建模框架（SWMF）对 2003 年 10 月 28—30 日风暴进行从太阳到热层的模拟。《空间天气》, **5**, S06003. https://doi.org/10.1029/2006SW000272. Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., et al. (2020) SciPy 1.0：Python科学计算的基础算法。《自然·方法》, **17**, *3*, 261-272. Wahr, J.,

31, L115 /. Y. Shprits, I. S. Zhelavskaya, O. V. Agapitov, A. Y. Drozdov, and N. A. Aseev (2019). 基于范艾伦探测器观测推导的解析合唱波模型。《地球物理研究杂志：空间物理》,**124**, 1063. https://doi.org/10.1029/2018JA026183. Weigel, R. S., J. Vandegriff, J. Faden, T. King, D. A. Roberts, et al. (2021). HAPI：访问日球物理时间序列数据的API标准。《地球物理研究杂志：空间物理》, **126**, 12. https://doi.org/10.1029/2021JA029534. Wyper, P. F., C. R. Devore, and S. K. Antiochos (2018). 一种带有细丝的太阳日冕喷流突破模型。《天体物理学杂志》,**852**, 2. https://doi.org/10.3847/1538-4357/aa9ffc. Zaharia, S., V. K. Jordanova, D. Welling, D., and G.

https:// a, J. G. Lyon, V Garretson, and M. Wiltberger (2019). GAMERA：一种用于非正交曲线坐标的三维有限体积磁流体力学求解器。《天体物理学杂志增刊》, **244**, *20*, https://doi.org/10.3847/1538-4365/ab3a4c。