Jupiter horizontal wind velocities at cloud level from Cassini

Produced by the Geophysical and Planetary Fluid Dynamics group,
Atmospheric, Oceanic and Planetary Physics,
Department of Physics, University of Oxford, UK

Roland M. B. Young, Peter L. Read, David Armstrong, Andrew J. Lancaster

10 June 2017 v1.0

This work was supported by the UK Science and Technology Facilities Council
via grants ST/I001948/1 and ST/K00106X/1.

1 Overview

This dataset contains horizontal wind velocity vectors at the top of Jupiter's main cloud deck, covering four rotation periods during December 2000. These wind measurements are based on a series of visible camera images taken by NASA's Cassini spacecraft. They were analysed using a cloud tracking procedure based on a Correlation Imaging Velocimetry (CIV) method developed to analyse fluid dynamics experiments. The dataset contains 1 123 505 horizontal wind velocity vectors covering 360° in longitude and ±50° in planetocentric latitude.

The procedure used to calculate the wind vectors was fully described by *Galperin et al.* (2014) (hereafter G14), and the user is referred to that paper for full details. That paper included, as Supplementary Material, gridded velocity fields in plaintext format, as a function of planetocentric latitude and east longitude, assuming a spherical planet with Jupiter's mean radius of 69 911 km (*Seidelmann et al.*, 2007).

Instead of gridded data, this dataset contains the raw, irregularly spaced, velocity vectors computed by the CIV procedure. This avoids errors introduced by the mosaicing procedure, and hence is a more accurate dataset. It is also more versatile as each velocity vector has its own time associated with it, and users may also wish to combine the vectors in their own way.

Like G14, these data assume spherical geometry, but we have provided a Python script **g14s.py** to convert to oblate spheroidal geometry. The equations and procedure to do this are described in Section 6 below. This script also allows the user to estimate the observational uncertainties in the data.

2 Data acquisition

During late 2000 NASA's Cassini spacecraft flew past the planet Jupiter, taking images and other scientific measurements over a several month period up to and after closest approach in early December 2000. The Imaging Science Subsystem (ISS) narrow angle camera (*Porco et al.*, 2004) took data in several filters during this time, covering the whole planet in longitude and latitude. A small subset of these images near closest approach (around 140 Jupiter radii) have been projected onto a System III west longitude / planetocentric latitude grid and been made available to the public via the NASA Planetary Data System (*Vasavada et al.*, 2008, **http://pds-atmospheres.nmsu.edu/Jupiter/CassiniMaps.txt**).

The first of these images was taken at 13:32:01 UTC on 11 December 2000, and the last was taken at 04:37:06 UTC on 13 December 2000. We have based our calculations on the CB2 near-infrared continuum band filter, which captures detail at the top of the main cloud deck and has the highest native resolution, $0.05°\text{pixel}^{-1}$. This corresponds to a typical image resolution of $60\,\text{km pixel}^{-1}$ at the equator.

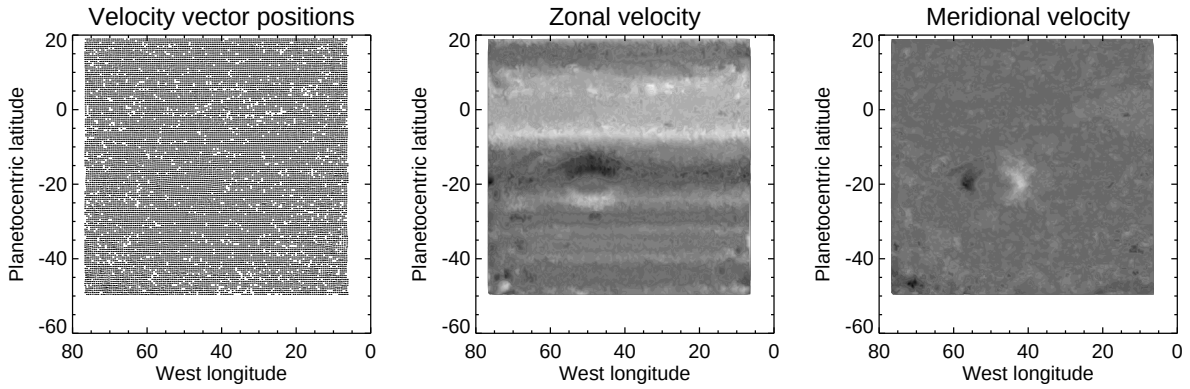

Figure 1: Example data from a single image pair. This example shows pair 41, one of the image pairs mainly covering the southern hemisphere, and this one contains the Great Red Spot. White is eastward and northward, and black is westward and southward.

3 Data processing

The procedure used to calculate the wind vectors was described in full by G14, and the user is referred to that paper for full details. The following is a summary.

The CIV method (*Fincham and Spedding*, 1997; *Fincham and Delerce*, 2000) takes rectangular patches of pixels in one image (the "correlation box") and compares them with a second image of the same location some time later. In the first stage, CIV1, the patch of pixels is moved around systematically within the second image, within a user-specified area around the initial point (the "search box"), and at each position the 2D correlation coefficient between the patterns of pixel brightnesses (cloud brightness, in this case) in the first and second images is calculated. The wind velocity is then defined by the displacement that maximises the correlation coefficient, at the mid-point of that displacement, subject to a number of conditions. In the second stage, CIV2, this is then repeated at sub-pixel resolution, with the patch also being deformed by shearing and rotation to obtain a more accurate match. The resulting displacement vector reflects the motion of the clouds between the two images, assuming the clouds retain their general shape over that time.

This procedure was done for all image pairs in the dataset that used the CB2 filter and were separated by approximately one hour. Image pairs were aligned according to their central longitude and only overlapping pixels were retained. Only part of each image could be used, due to low contrast near the edges of each image because of the curvature of the planet and the lack of light reaching the planet's high latitudes. Some other regions had to be removed due to the presence of the moon Io and its shadow. Typically each image yielded a 70° longitude by 60° latitude rectangle with velocity vectors on an approximate latitude-longitude grid with 0.5° separation. Of the original 304 images, we were able to obtain wind velocities from 70 pairs. This resulted in 1 123 505 velocity vectors arranged in 70 approximately square grids, one for each image pair analysed. Half of the image pairs are aimed at the northern hemisphere, and half at the southern hemisphere. An example field is shown in Fig. 1.

At this point in G14 we interpolated these vectors onto a global grid and based our analysis on those global maps. This combined all image pairs falling within one rotation period, combining velocity vectors at different times and using various methods to combine the instantaneous velocity fields. In this dataset we omit this step, and instead present the data simply as a list of (latitude, longitude, eastward velocity, northward velocity) vectors. In G14 these vectors are the 1 123 505 velocity vectors described on p.299 as the "filtered vectors", just before the mosaicing procedure.

This dataset has some advantages over the gridded data included with G14. First, the mosaicing procedure introduces some error and smoothing as a result of averaging in space. Second, there is significant error due to combining data at the same location from image pairs separated by up to a full Jovian rotation period, i.e. just under 10 h, as the flow will have evolved considerably during that time. This dataset avoids these problems. Third, the velocity vectors in each image pair are defined at exactly

the same time, which is useful (or necessary) for some purposes — there are typically 16 000 wind vectors in each image pair. Finally, users may want to re-combine the vectors into global maps in their own way, and this dataset allows this.

4 Dataset contents

The dataset consists of a netCDF file **g14s.nc** and a Python module **g14s.py**.

4.1 g14s.nc

The netCDF file contains the following quantities:

Summary data for each image pair:
npair = 70 values in each one:

- **pair_number**: Image pair number. This is a look up index for the other summary data, using the values in **pair** for each vector.
- **day**: Day number.
- **nvectors**: Number of vectors extracted from this pair.
- **time**: Time of first image in pair (seconds since 2000-Dec-11 00:00).
- **image1**, **image2**: First and second raw images that make up the pair. Both of these are from the list at http://pds-atmospheres.nmsu.edu/Jupiter/CassiniMaps.txt.
- **separation**: Separation time between the two images in the pair (s).
- **lat_min**, **lat_max**: Minimum / maximum planetocentric latitude of vectors in this image pair (degrees).
- **lon_min**, **lon_max**: Minimum / maximum west longitude of vectors in this image pair (degrees).
- **overlap**: Zero longitude overlap flag. =1 means this pair straddles the zero longitude line.

Data for each velocity vector
nvec = 1 123 505 values in each one:

- **pair**: Image pair the vector is from. Use this to look up values in the summary data.
- **longitude**: West longitude (degrees).
- **latitude**: Planetocentric latitude (degrees).
- **u**: Zonal (eastward) velocity ($\mathrm{m\,s^{-1}}$).
- **v**: Meridional (northward) velocity ($\mathrm{m\,s^{-1}}$).

4.2 g14s.py

The Python module contains a few utilities. It is loaded in Python with the command `import g14s`, and the functions assume the array inputs are NumPy arrays.

- **estimate_random_error(latitude, pair, separation, radius_mean)**
  Estimate the random error in the velocity measurements.
- **convert_latc_to_latg(latitude, radius_ratio)**
  Convert planetocentric to planetographic latitude.
- **convert_uv_to_oblate(latitude, u, v, radius_equator, radius_mean, radius_pole)**
  Convert velocities and (optionally) velocity errors from spherical to oblate spheroidal geometry.

5 Observational uncertainty

An estimate of the random error in the velocity measurements, along with a discussion of other systematic errors, is described in full detail in G14 Appendix A. We know of three potential sources of random error in these measurements:

- The position of each image is accurate to 0.2 pixel, based on NASA's limb-fitting algorithm (reported by *Salyk et al.* (2006), p.431). This corresponds to a u velocity error of about $4.6\cos\phi_c$ $\mathrm{m\,s^{-1}}$, where ϕ_c is planetocentric latitude, and a v velocity error of about $4.6\,\mathrm{m\,s^{-1}}$.
- *Choi et al.* (2007) showed that the CIV procedure introduces an error depending on the position of the feature being tracked within the correlation box. They estimated this error to be half the correlation box size times the local horizontal velocity shear. In this case this corresponds to a u velocity error of about $5.8\cos\phi_c$ $\mathrm{m\,s^{-1}}$ and a v velocity error of about $5.8\,\mathrm{m\,s^{-1}}$.
- The formal tracking error in the CIV method itself, based on the error in the position of the start and end points, gives an additional u error of of about $0.8\cos\phi_c$ $\mathrm{m\,s^{-1}}$ and a v velocity error of about $0.8\,\mathrm{m\,s^{-1}}$.

Combining these in quadrature we estimate the random error in the velocities to be about $7.4\cos\phi_c$ $\mathrm{m\,s^{-1}}$ in u and about $7.4\,\mathrm{m\,s^{-1}}$ in v. We considered including these errors as variables, but as it is a simple calculation to produce them, instead we have included a routine **estimate_random_error()** in the Python module, which calculates them for each velocity vector. This converts displacement errors in pixels Δx and Δy, which are combined errors from the three sources above, to velocity errors:

$$\Delta u \approx \frac{\Delta x}{\Delta t} p \frac{\pi}{180} r_m \cos\phi_c \quad (1)$$

$$\Delta v \approx \frac{\Delta y}{\Delta t} p \frac{\pi}{180} r_m \quad (2)$$

Δt is the image **separation**, $p = 0.05^\circ\mathrm{pixel}^{-1}$ is the image resolution, and $\Delta x = \Delta y = 0.4585$ pixels.

6 Conversion from spherical to oblate spheroidal geometry

The Python module routine **convert_uv_to_oblate()** can be used to convert velocities and velocity errors from spherical to oblate spheroidal geometry.

G14 assumed spherical geometry when calculating their velocities, but Jupiter has an oblateness around 0.065, and velocities accounting for its oblate spheroidal geometry, while similar, are measurably different. Oblate spheroidal polars require the radius of curvature in the zonal and meridional directions rather than just the mean radius of the planet. The radii of curvature are given by *Dowling and Ingersoll* (1988). For the zonal and meridional directions respectively (their Eqs 3–4), the radii of curvature are

$$r(\phi_g) = r_e(1 + \epsilon^{-2}\tan^2\phi_g)^{-1/2} \quad (3)$$

$$R(\phi_g) = \frac{r_e}{\epsilon^2}\left(\frac{r(\phi_g)}{r_e\cos\phi_g}\right)^3 \quad (4)$$

where

$$\epsilon = r_e/r_p \quad (5)$$

with $r_e = 71\,492\,\mathrm{km}$ the equatorial radius, $r_p = 66\,854\,\mathrm{km}$ the polar radius (*Seidelmann et al.*, 2007), and ϕ_g the planetographic latitude.

This dataset uses planetocentric latitude, so using the conversion from *Irwin* (2009), Eq. 2.23,

$$\tan\phi_g = \epsilon^2\tan\phi_c \quad (6)$$

where ϕ_c is planetocentric latitude, this gives

$$r(\phi_c) = r_e(1+\epsilon^2\tan^2\phi_c)^{-1/2} \quad (7)$$

$$R(\phi_c) = \frac{r_e}{\epsilon^2}\left(\frac{1+\epsilon^4\tan^2\phi_c}{1+\epsilon^2\tan^2\phi_c}\right)^{3/2} \quad (8)$$

To convert from u and v velocities (and their errors) in spherical polars as used in G14 to the equivalent in oblate spheroidal polars the conversions are

$$u_{obl} = u_{sph}\frac{r(\phi_c)}{r_m\cos\phi_c} \quad (9)$$

$$v_{obl} = v_{sph}\frac{R(\phi_c)}{r_m} \quad (10)$$

where $r_m = 69\,911$ km is Jupiter's mean radius.

This correction typically changes zonal velocities by between -1.8% and $+2.3\%$ (most positive change at the equator), and meridional velocities by between -10.6% and $+1.5\%$ (most negative change at the equator).

References

Choi, D. S., D. Banfield, P. Gierasch, and A. P. Showman (2007), Velocity and vorticity measurements of Jupiter's Great Red Spot using automated cloud feature tracking, *Icarus*, *188*, 35–46, doi:10.1016/j.icarus.2006.10.037.

Dowling, T. E., and A. P. Ingersoll (1988), Potential vorticity and layer thickness variations in the flow around Jupiter's Great Red Spot and White Oval BC, *J. Atmos. Sci.*, *45*, 1380–1396, doi:10.1175/1520-0469(1988)045<1380:PVALTV>2.0.CO;2.

Fincham, A., and G. Delerce (2000), Advanced optimization of correlation imaging velocimetry algorithms, *Exp. Fluids*, *29*, S13–S22, doi:10.1007/s003480070003.

Fincham, A. M., and G. R. Spedding (1997), Low cost, high resolution DPIV for measurement of turbulent fluid flow, *Exp. Fluids*, *23*, 449–462, doi:10.1007/s003480050135.

Galperin, B., R. M. B. Young, S. Sukoriansky, N. Dikovskaya, P. L. Read, A. J. Lancaster, and D. Armstrong (2014), Cassini observations reveal a regime of zonostrophic macroturbulence on Jupiter, *Icarus*, *229*, 295–320, doi:10.1016/j.icarus.2013.08.030.

Irwin, P. (2009), *Giant Planets of our Solar System*, 2nd ed., 403 pp., Springer/Praxis.

Porco, C. C., R. A. West, S. Squyres, A. McEwen, P. Thomas, C. D. Murray, A. Del Genio, A. P. Ingersoll, T. V. Johnson, G. Neukum, J. Veverka, L. Dones, A. Brahic, J. A. Burns, V. Haemmerle, B. Knowles, D. Dawson, T. Roatsch, K. Beurle, and W. Owen (2004), Cassini Imaging Science: Instrument characteristics and anticipated scientific investigations at Saturn, *Space Sci. Rev.*, *115*, 363–497, doi:10.1007/s11214-004-1456-7.

Salyk, C., A. P. Ingersoll, J. Lorre, A. Vasavada, and A. D. Del Genio (2006), Interaction between eddies and mean flow in Jupiter's atmosphere: Analysis of Cassini imaging data, *Icarus*, *185*, 430–442, doi:10.1016/j.icarus.2006.08.007.

Seidelmann, P. K., B. A. Archinal, M. F. A'hearn, A. Conrad, G. J. Consolmagno, D. Hestroffer, J. L. Hilton, G. A. Krasinsky, G. Neumann, J. Oberst, P. Stooke, E. F. Tedesco, D. J. Tholen, P. C. Thomas, and I. P. Williams (2007), Report of the IAU/IAG Working Group on cartographic coordinates and rotational elements: 2006, *Celest. Mech. Dyn. Astron.*, *98*, 155–180, doi:10.1007/s10569-007-9072-y.

Vasavada, A. R., C. C. Porco, and The Cassini Imaging Science Team (2008), NASA Planetary Data System: Cassini Cylindrical-Projection Maps near Jupiter Closest Approach. http://pds-atmospheres.nmsu.edu/Jupiter/CassiniMaps.txt.