

Balltracking: An highly efficient method for tracking flow fields

H. E. Potts, R. K. Barrett, and D. A. Diver

University of Glasgow, G12 8QQ Glasgow, UK
e-mail: hugh@astro.gla.ac.uk

Received 17 December 2003 / Accepted 16 April 2004

Abstract. We present a method for tracking solar photospheric flows that is highly efficient, and demonstrate it using high resolution MDI continuum images. The method involves making a surface from the photospheric granulation data, and allowing many small floating tracers or balls to be moved around by the evolving granulation pattern. The results are tested against synthesised granulation with known flow fields and compared to the results produced by Local Correlation tracking (LCT). The results from this new method have similar accuracy to those produced by LCT. We also investigate the maximum spatial and temporal resolution of the velocity field that it is possible to extract, based on the statistical properties of the granulation data. We conclude that both methods produce results that are close to the maximum resolution possible from granulation data. The code runs very significantly faster than our similarly optimised LCT code, making real time applications on large data sets possible. The tracking method is not limited to photospheric flows, and will also work on any velocity field where there are visible moving features of known scale length.

Key words. methods: numerical – Sun: photosphere – Sun: granulation

1. Introduction

With the advent of satellite missions producing constant streams of high resolution, high cadence solar data, the need for highly efficient processing tools becomes more urgent. Future missions such as Solar-B will produce an order of magnitude more data, and to avoid drowning in the torrent of information, new numerical methods are required. This paper presents a highly efficient method of tracking photospheric flows from data that shows the movement of granules. The data used as an example is high resolution MDI continuum data, although the method will work perfectly well on other high resolution solar images, such as TRACE filtergrams and ground based telescope images.

There are several methods that are commonly used to measure solar flow fields. Most of the methods rely on measuring the motion of granules, to infer the underlying flow field that is advecting them. Numerical models by Rieutard et al. (2001) indicate that for spatial scales larger than 2.5 Mm and time scales larger than 0.5 h, that granules do make good flow tracers, although they may tend to underestimate the flow speed.

The most important method is Local Correlation Tracking (LCT). This method involves calculating the rigid translation of small image elements between consecutive frames of data. The motion is calculated by the shift that gives the highest correlation between the corresponding subimages. November & Simon (1988) first demonstrated that this method could be used to measure the proper-motion of solar granulation, using high resolution images from the Sacramento Peak Vacuum Tower Telescope. At a similar time Title et al. (1989)

also carried out a detailed study of the statistical properties of granulation using LCT as a tool with data from the SOUP instrument on Spacelab-2. A description of the analysis of high resolution MDI data using LCT may be found in Shine et al. (2000), and an evaluation of the accuracy of this method, and problems associated with using it may be found in Potts et al. (2003). LCT has been used successfully on many different types of data ranging from the highest resolution ground based observations, to low resolution (2 arcsec) MDI dopplergrams, where granulation is not resolved, but patterns formed by granulation can be seen (Lisle et al. 2000). The strength of conventional LCT lies in its simplicity and flexibility, but it suffers from being computationally inefficient and is therefore very slow for analysing the largest data sets.

Another tracking method is the Coherent Structure Tracking (CST) method first used by Strous (1994) and further developed by Roudier et al. (1999). This method takes high resolution images, where individual granules can be clearly resolved, and labels and tracks individual granulation cells. This technique has the potential for the highest resolution, although care must be taken to avoid selection effects as the largest, longest lived granules are easiest to track. This method does not work well for the currently available satellite data, as the resolution is not high enough to clearly track individual granules, but will probably work well with Solar-B data. It is fairly complex to implement in comparison to LCT, and again computationally intensive.

Time-distance heliosiesmology is another tracking technique that works in a very different way (Duvall et al. 1996).

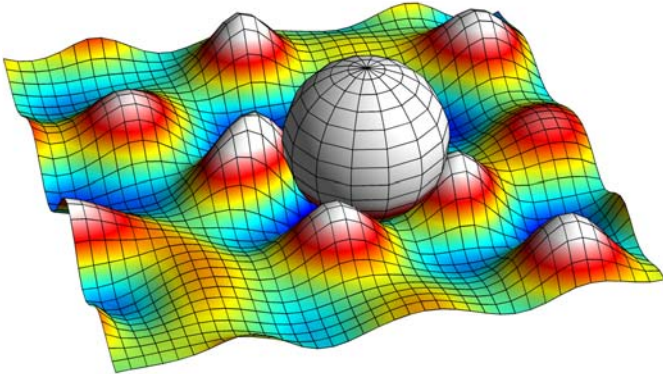


Fig. 1. One of the tracers or balls, floating on a simulated granulation surface. As the peaks that represent the granulation cells move and evolve, they push the ball around.

This method uses the measurement of difference in speed of acoustic waves propagating in opposite directions to work out the velocity of the bulk fluid. The spatial resolution of this method is substantially lower than the other methods described here (De Rosa et al. 2000).

In this paper we discuss a new tracking technique that produces highly accurate velocity fields, in a fraction of the time taken by any other method. We first describe the method and its implementation, and discuss the accuracy that can be expected from any velocity tracking method. We then describe a method for producing simulated granulation flow fields and demonstrate this new tracking method and LCT on the simulated data. We finally present results for flow fields from real MDI data, and show the evolution of a supergranular flow field at high resolution.

2. Balltracking method

2.1. Overview

Granulation cells move about in the photosphere with a directed random walk motion as they interact with each other and are advected by supergranular flow fields. Imagine turning an image of the granules into a moving surface, with granulation cells represented by bumps or ripples, that form and move with time. Now imagine this surface as the surface of a fluid, and drop a small floating ball on to it. The ball will first settle to a local minimum between a few granulation cells, with part of its volume submerged. Figure 1 shows a ball on a simulated granulation data surface, moving between several granulation peaks. The ball will then be nudged by each bump that comes towards it, and get caught in local minima between several moving ripples. On average the ball will be pushed in the same direction as the mean motion of the ripples. At all times the ball is moving under the influence of several granulation cells, which are constantly forming, moving and disappearing. The balls have mass and momentum, such that if all the granulation cells pushing a ball disappear, the ball will continue in the same direction. This helps the method cope with missing and noisy data. The momentum of the ball also makes it “predict” where to go at the next time step, making this method highly computationally efficient.

In order to track photospheric flows, thousands of these trackers are released onto the moving surface and their motions recorded and smoothed.

2.2. Generating tracking surfaces

The raw data we are using in this paper are SOHO MDI high resolution continuum images (Scherrer et al. 1995). These were chosen because of their freedom from seeing problems and availability of long continuous runs. The data set we are using is from a 64 h continuous run from 15–18th January 1997. The run consists of high resolution continuum, magnetogram and dopplergram images. The spatial resolution is 0.6 arcmin/pixel with a 1 min cadence. We use a 384×1024 pixel (3.8×10.2 arcmin), 36 h subset of this, (starting at 0930 on 15th Jan.) moving with the sun’s rotation that contains only quiet sun data. Within this there are three blocks of 8 consecutive missing frames, and 34 individual missing frames, or frames with missing data. It is impossible to track across the 8 min gaps, as they are longer than the granulation lifetime, so the tracking was stopped and restarted after the gaps. Individual missing frames were replaced by taking an average of the two neighbouring frames.

The data first needed to be processed to remove the solar rotation and the solar oscillations. These two operations were both done in one step, using a Fourier filter, and Fourier interpolation. To do this each frame of data is first 2D Fourier transformed. The transformed data is then masked to reject all wavelengths except those that are required to display the granulation. In this work we kept Fourier components with wavelengths between 1.5–3.5 arcmin. The short wavelength limit of 1.5 arcmin is close to the maximum resolution of the instrument (1.2 arcmin), and serves to reduce random single pixel noise from the CCD. Note that this is also close to the typical size of a granulation cell, meaning that granules are only barely resolved, and the smallest granules are not visible. The phase part of the remaining Fourier components is then modified to apply the rigid interpolation to derotate the data, and the result inverse transformed. Note that this rigid derotation does not attempt to correct for the geometric projection distortion. This reaches a maximum of about 10% for the extreme northern edge of the data set, and is less than 5% for 75% of the data. For smaller areas the distortion is not important, for example a 256×256 pixel region tracked over 12 h has a maximum distortion of about 1%. This simple derotation method is chosen as it avoids any phase errors caused by interpolation. It is also far more computationally efficient to remap and rescale the velocity field obtained from the tracking (a 32 MB data set) rather than the cumbersome raw data (around 6 GB). A full discussion of the interpolation method may be found in Potts et al. (2003). Note also that the filters used here are 2D Fourier filters; there is no use of time domain filtering, as this can cause spurious correlation between frames and distort the output data. The derotated, filtered data is then normalised to give it a mean of zero, and an rms value of one. A comparison of some raw and some filtered MDI data can be seen in Fig. 2.

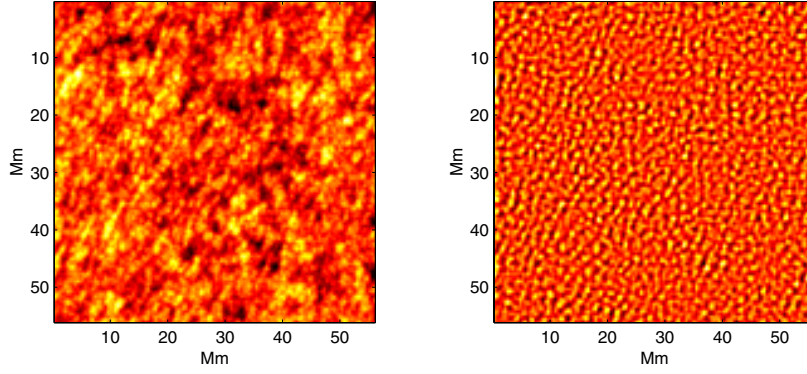


Fig. 2. Comparison of MDI data before and after being Fourier filtered. The granulation cells are easily visible in the filtered data. An animated version, that clearly shows the granulation motion may be found in `balltrack_filtering.html` which is available with the electronic version of the article at <http://www.edpsciences.org/aa>.

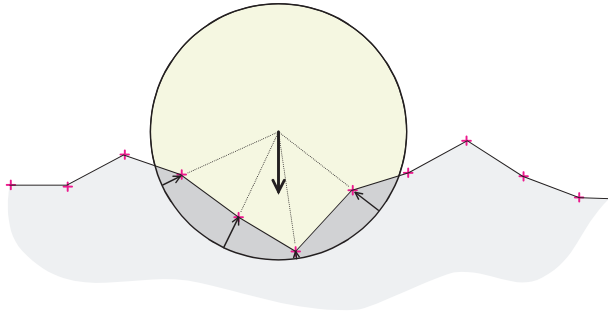


Fig. 3. Cross section of a ball on a granulation surface. The ball penetrates the surface, with the force on the ball from each surface point proportional to the penetration distance.

2.3. Ball motion

Figure 3 shows a cross section of a ball and the surface supporting it. The surface consists of the individual data points, uniformly spaced in the xy plane. With barely resolved data such as MDI data each ball typically rests on 10–16 data points. Each data point exerts a force \mathbf{f}_i on the ball directed towards the ball centre. The ball also experiences a downwards, gravity-like force $m\mathbf{g}$, proportional to its mass. The motion of the ball in response to these forces may be written as:

$$m\dot{\mathbf{v}} = \sum_i \mathbf{f}_i + m\mathbf{g} - \alpha\mathbf{v}. \quad (1)$$

The term, $\alpha\mathbf{v}$, is a damping force, proportional to and directed against the instantaneous velocity of the ball. This term is needed to damp out oscillations about the local minimum the ball is resting in. Each force component from the surface depends on how far the surface point has penetrated into the sphere, increasing linearly from zero when it is at the surface of the sphere:

$$\text{for } r_i < R_s, \mathbf{f}_i = k(1 - d_i)\hat{\mathbf{r}} \quad (2)$$

$$\text{for } r_i > R_s, \mathbf{f}_i = 0 \quad (3)$$

where d_i is the fractional penetration of each surface point, $d_i = (R_s - r_i)/R_s$, where R_s is the sphere radius and r_i is the penetration distance. The constant k determines how high the ball floats on the surface, and is a function of the average penetration distance.

2.4. Input parameters

To fully describe the motion of the ball, four non-dimensional parameters are used. The parameters are scaled by dividing by the characteristic length and time scales d_c and t_c , which are the computational grid dimension and time interval for measurements, and the characteristic velocity $v_c = d_c/t_c$. These are defined as:

$$\widetilde{R}_s = \frac{R_s}{d_c} \quad \text{Sphere Radius}$$

$$\widetilde{D}_p = \frac{D_p}{R_s} \quad \text{Average surface penetration}$$

$$\widetilde{T}_d = \frac{T_d}{t_c} \quad \text{Damping characteristic time}$$

$$\widetilde{A}_m = \frac{\delta(v/v_c)}{\delta(T/t_c)} \quad \text{Max velocity change per timestep.}$$

Sphere radius \widetilde{R}_s

The sphere radius is chosen to match the typical size of the features that we want to track. A sphere size much smaller than the features will be too sensitive to short wavelength noise, a sphere too large will not be sensitive to the features. A typical value for \widetilde{R}_s that works well is around half the mean granule spacing, which for high resolution MDI data is about $\widetilde{R}_s = 2$.

Average surface penetration \widetilde{D}_p

The value of the mean penetration depth is a compromise between accurate tracking and short wavelength noise rejection. A large penetration depth is less sensitive to noise, but reacts more slowly to features. A value of 0.2 is a good compromise for filtered MDI data.

Damping characteristic time \widetilde{T}_d

In the absence of any driving force the velocity of the ball will reduce as $\mathbf{v} = \mathbf{v}_0 \exp(-t/\widetilde{T}_d)$. This term sets the length of the ball's “memory” of its past movements, and also the

maximum speed it can attain. The damping force is necessary for numerical stability. A value of $\widetilde{T}_d = 1\text{--}2$ gives good stability, while still being able to cope with single missing frames and the track the fastest features. Tracking speed and stability can be increased by having a larger damping force in the vertical direction than the horizontal direction. Increasing the vertical damping to 10 times the horizontal damping works well.

Maximum ball acceleration \widetilde{A}_m

The maximum acceleration, which is roughly equivalent to the “gravitational” force needs to be large enough to allow the balls to track the fastest accelerating features, but no larger. If it is too large, the damping term needs to be increased to an excessive value to maintain stability. An optimum value for MDI granulation data is $A_m = 0.5$.

The equation of motion can be expressed in terms of these parameters.

$$\delta \mathbf{v} = \delta t \left(\frac{2A_m}{\widetilde{D}_p \pi \widetilde{R}_s^2} \sum_i \mathbf{d}_i - \widetilde{A}_m \hat{\mathbf{g}} - \widetilde{T}_d \mathbf{v} \right) \quad (4)$$

where $\hat{\mathbf{g}}$ is the unit vector in the downwards or “gravity” direction. A full derivation may be found in Appendix A.1.

2.5. Tracking procedure

To run the tracking procedure we proceed as follows

1. Choose number of balls to use. This is a compromise between trying to track every possible feature, and avoiding multiple balls tracking the same feature.
2. Divide the data surface into a grid, such that one ball will be in each grid cell. Randomly position a ball in each grid cell, just above the surface of the data plane.
3. Let each ball settle under gravity to the nearest local minimum on the data plane.
4. Update the surface to the next time step. Depending on the time sampling rate it can help stability to insert an intermediate frame that is the average of its neighbours between each data frame. For high resolution MDI data it helps to reduce the noise considerably, although it doubles the computational time.
5. Move the balls, as described by Eq. (4), and store the new position.
6. Remove any balls that have become too close to each other, and so are tracking the same feature. This is important, otherwise the smoothing statistics are badly damaged. It also prevents balls collecting at regions of flow convergence. Also remove any that have fallen off the edge of the grid or sunk below the surface. All balls are replaced in the region that has the lowest ball density. This allows the flow pattern in regions of strong divergence to be measured, preventing them from being swept clean of balls.
7. Repeat from Step 4 for all data.

The output from this procedure is many individual ball tracks. A small subset of them from tracking synthetic data can be seen in Fig. 4, along with the underlying flow field.

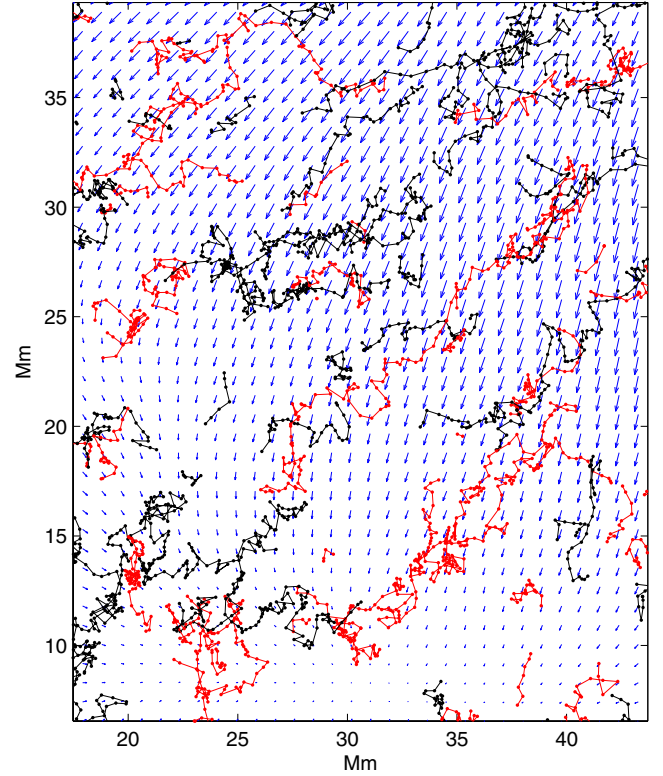


Fig. 4. A small subset of ball tracks from synthetic data, with underlying velocity field as arrows. The red and black colouring is to help distinguish different tracks.

3. Smoothing and rescaling the output data

The result of the balltracking procedure is a large number of unevenly spatially distributed velocity samples of a very noisy velocity field. To extract the underlying velocity field the data needs to be smoothed and resampled to a regular grid. Both these operations are done at the same time. First we allocate a Gaussian spatial weighting function to each ball velocity sample for each grid point (x_i, y_i) :

$$w_{n,t} = \frac{\exp \frac{-r_{n,t}^2}{2\sigma^2}}{\sum_{n=1}^N \exp \frac{-r_{n,t}^2}{2\sigma^2}} \quad (5)$$

where N is the total number of balls and ball n at time t has velocity $\mathbf{v}_{n,t}$ and is at position $(x_{n,t}, y_{n,t})$. $r_{n,t}$ is the distance between the grid point and the ball position, given by $r_{n,t} = \sqrt{(x_{n,t} - x_i)^2 + (y_{n,t} - y_i)^2}$. The value of the smoothed velocity at the grid point (x_i, y_i) and time t is then given by:

$$\mathbf{V}(x_i, y_i, t) = \sum_{t-\Delta t/2}^{t+\Delta t/2} \sum_{n=1}^N w_{n,t} \mathbf{v}_{n,t}. \quad (6)$$

This method uses a Gaussian weighted spatial “sphere of influence” to each ball velocity sample. The smoothing radius used, σ , equivalent to the maximum spatial resolution recoverable, is a trade off between spatial and time resolution. If a high spatial resolution is chosen the time resolution is poor. Note that

although this is a property of the smoothing method, it is also a property of the underlying data, and even with perfect tracking of every granule, the same compromise has to be made. This is covered in more detail in the next section.

3.1. Smoothing resolution

The smoothing process is a compromise between spatial and temporal resolution. There is a finite resolution limit beyond which no motion tracking method that relies upon granular motions can improve, no matter how perfectly the motions are tracked. Consider one component of the velocity of granules in a given region to be normally distributed about some local mean \bar{v}_u (the underlying velocity field), with variance σ_u^2 . The smoothing process takes the mean of a sample of these velocity values. If we are smoothing in time and space, the number of actual samples is given by:

$$n = \pi \left(\frac{R_s}{\Delta r} \right)^2 \frac{T_s}{\Delta t} \quad (7)$$

where R_s and T_s are the sample region radius and timelength, and Δr and Δt are the time and space sampling intervals. If all the samples are independent of each other then the expected standard deviation σ_v of a velocity measure made from the samples is simply:

$$\sigma_v = \frac{\sigma_u}{\sqrt{n}}. \quad (8)$$

If, however, the samples are not completely independent of each other, the effective number of samples is reduced. If we assume that the time samples have a Gaussian autocorrelation function, of width σ_t , the effective number of time sample points n_t^* in time T is given by:

$$n_t^* = \frac{T}{\Delta t} \operatorname{erf} \left(\frac{\Delta t}{2\sqrt{2}\sigma_t} \right) \quad (9)$$

where erf is the error function of a normal distribution. Similarly for a characteristic spatial autocorrelation length σ_r , the effective number of spatial samples in an area πR_s^2 is:

$$n_r^* = \pi \left[\frac{R_s}{\Delta r} \operatorname{erf} \left(\frac{\Delta r}{2\sqrt{2}\sigma_r} \right) \right]^2. \quad (10)$$

For samples taken at intervals significantly larger than the correlation scale, the error function term tends rapidly to one and may be ignored. For the case of very frequent samples, the effective sampling frequency becomes only a function of the autocorrelation scale, and defines the maximum recoverable resolution. For the time samples:

$$n_t^* \rightarrow \frac{T_s}{\sqrt{2\pi}\sigma_t} \quad \text{as} \quad \frac{\Delta t}{\sigma_t} \rightarrow 0, \quad (11)$$

and for space samples:

$$n_r^* \rightarrow \pi \left[\frac{R_s}{\sqrt{2\pi}\sigma_r} \right]^2 = \frac{1}{2} \left(\frac{R_s}{\sigma_r} \right)^2 \quad \text{as} \quad \frac{\Delta r}{\sigma_r} \rightarrow 0. \quad (12)$$

The minimum standard deviation that may be expected for the velocity field, derived from the granular motions is therefore

limited by the autocorrelation functions of the granular motions and is given by:

$$\sigma_v^* = \frac{\sigma_u}{\sqrt{n_r^* n_t^*}} = \sigma_u \left[2\sqrt{2\pi} \left(\frac{\sigma_r}{R_s} \right)^2 \frac{\sigma_t}{T_s} \right]^{\frac{1}{2}}. \quad (13)$$

For real granulation, the observed distribution of physical parameters is strongly dependent on the measuring scheme and resolution of measurements. The distribution of a single x or y component of translational velocities may be approximated by a Gaussian distribution with $\sigma_u = 800 \text{ m s}^{-1}$ (Roudier et al. 1999).

The temporal autocorrelation function for the granule velocity can be estimated by considering the number of events that change its velocity through its lifetime. The lifetime is poorly defined, ranging from values of around 5 min found by Title et al. (1989) from SOUP data, to only 140 s from higher resolution Pic du Midi images by Roudier et al. (1999). As the images we are using are low resolution, we will not see the smallest, shortest lived granules so we will assume a lifetime of 5 min. The main events that change the granule velocity are caused by nearest neighbour granules appearing and disappearing. If we assume that the granule has 6 nearest neighbours (hexagonal packing), then we can say that the granule changes direction, on average 6 times in its lifetime, giving an average autocorrelation time of around $\sigma_t \approx 50 \text{ s}$.

The spatial autocorrelation function for granular velocities is also hard to define. Individual granules interact strongly with their nearest neighbours, but not in such a way as to correlate their motions significantly. Consequently we are choosing a correlation length equal to the mean granule spacing of around 1 Mm, derived from high resolution ground based observations (Berrelli et al. 2002).

Figure 5 shows the minimum random error obtainable on any velocity field measured from the granulation motion, assuming a measuring cadence of 1 minute for a range of spatial and temporal smoothing parameters. This result assumes perfect tracking of all granules. For MDI data the smallest granules are not resolved, resulting in a lower spatial density of samples, so the errors will be slightly larger than this.

Due to the uncertainties in the correlation scales this result may have considerable error. As an example, for a four hour average, smoothed with a radius of 4 MDI pixels, the minimum uncertainty expected in the velocity would be around $65^{+30}_{-15} \text{ m s}^{-1}$.

3.2. Speed calibration

One problem of this method is that the balls do not track the speed of the underlying flow, they always move on average slower than it. There are two reasons for this. The first is that the balls are not always tracking features, they spend a proportion of their time searching for local minima to track. If the balls spend a fraction of the time α tracking features, then the maximum average velocity that can be obtained is $v = \alpha v_0$. The second, and most important reason, is that the balls are moving under the influence of two forces. There is a stochastic driving term, A that on average pushes the balls in the direction of the flow, and a damping or drag term B that tries to stop

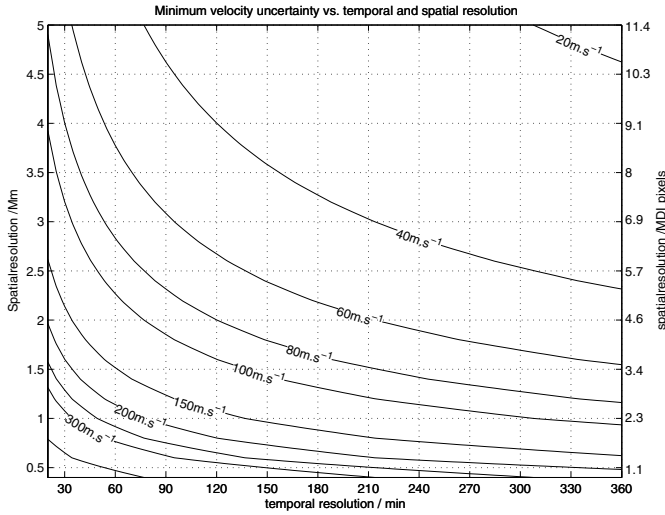


Fig. 5. The maximum velocity accuracy that can be obtained from granulation for various temporal and spatial smoothing windows, assuming a sampling cadence of 1 min.

them. The equation of motion for the balls with respect to the velocity of the driving field V_0 can therefore be expressed as:

$$\frac{dv}{dt} = A(\alpha v_0 - v) - Bv. \quad (14)$$

The terminal velocity of a ball in the field, when $dv/dt = 0$ is therefore:

$$v = v_0 \left(\frac{\alpha A}{A + B} \right). \quad (15)$$

As α is always less than one, the velocity obtained from the balltrack method will always be a constant scale factor less than the true underlying velocity. This scale factor is related to the damping rate, the statistical properties of the data, and the size of the driving term. Consequently it will be a constant, for any statistically similar data sets. Calculating the constant directly from the balltrack data is hard.

A simple and quick way to calibrate the data is to use local correlation tracking (LCT) on a small subset of data. From Balltrack, the spatial distribution of the velocity field is known. A small subset of this known velocity field, that contains regions of high x and y velocity is chosen. LCT, using Fourier interpolation to avoid systematic errors, as described in Potts et al. (2003), is performed on this area. Not very many sample areas are required, around 25 will do, depending on the accuracy required. A correlation plot between the velocity components obtained from the two methods is then done, and the gradient gives the scaling factor. This scaling factor will be correct for any data set of the same type, providing that the balltrack parameters are kept constant. This method is simple to implement and fast.

If only the spatial shape of the velocity field is required, for example for studying the evolution of supergranular flow fields, the uncalibrated raw balltrack data can be used. The velocities obtained in this case are typically 70–85% of the true velocities.

4. Generating test data

In order to evaluate the tracking procedures it is necessary to generate a fake granulation field with a known underlying flow pattern. To do this we make a simple model that simulates the essential features of a real granulation field. The model works by considering the motion of granulation cell centres, and their interaction with each other.

We start with a spatial distribution of cell centres, randomly arranged, but with uniform density. Each granulation cell behaves as an automaton that is repelled by all the other cells around it. The cells have a range of lifetimes, distributed normally about a mean value. The size of a cell varies throughout its life, starting at zero, rising to a maximum, then falling back to zero. When it is large it repels other cells more strongly, when it is small it is more easily repelled by other cells.

The motion of the cell centre has two parts. First there is the motion due to the repulsive interaction between it and surrounding cells. Superimposed on that is an advective velocity field that can be specified externally. If the specified velocity field has divergence then the source term for new cells needs to be a function of the divergence in order to keep the cell density constant. Applying the continuity equation to the density of granulation cells n :

$$\frac{\partial n}{\partial t} + \nabla \cdot (n\mathbf{v}) = S - L \quad (16)$$

where n is the density of the granules, S is the source term and L is the loss term.

We want the average density of granulation cells to be constant, n_0 . The loss term is just a function of the mean lifetime τ of the cells and the mean density, and is hence also a constant:

$$L = \frac{n_0}{\tau} \quad (17)$$

where n_0 is the mean density and τ is the mean cell lifetime. Rearranging gives the mean source term required:

$$S = n_0 \left[\nabla \cdot \mathbf{v} + \frac{1}{\tau} \right]. \quad (18)$$

In order to maintain the distribution of granule lifetimes the source term must always be positive. This limits the divergence of the velocity field to $\nabla \cdot \mathbf{v} \geq -1/\tau$. The velocity field can have any other, divergence free field added to it. Once the cell centre motions are calculated, the images produced are processed to produce a realistic looking image. First, each cell is represented by a Gaussian, with a size and amplitude a function of its instantaneous size. The image produced is then filtered with a Fourier filter in the same way as the raw granulation data. In Fig. 6 the three stages of image production are shown. A movie showing the simulated granulation in comparison to some real MDI data may be found in `balltrack_simdata.html` which is available with the electronic version of the article at <http://www.edpsciences.org/aa>.

5. Results from test data

In Fig. 7 the velocity recovered using the Balltrack method and LCT are compared with the true underlying velocity field.

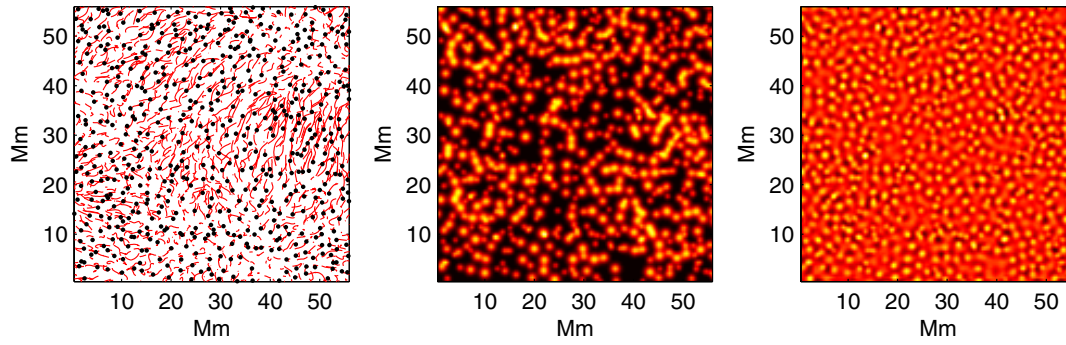


Fig. 6. The three stages of producing simulated granulation data. From left to right: the granulation cell centres, with tracks showing their motion; modified with Gaussians; filtered with Fourier filter.

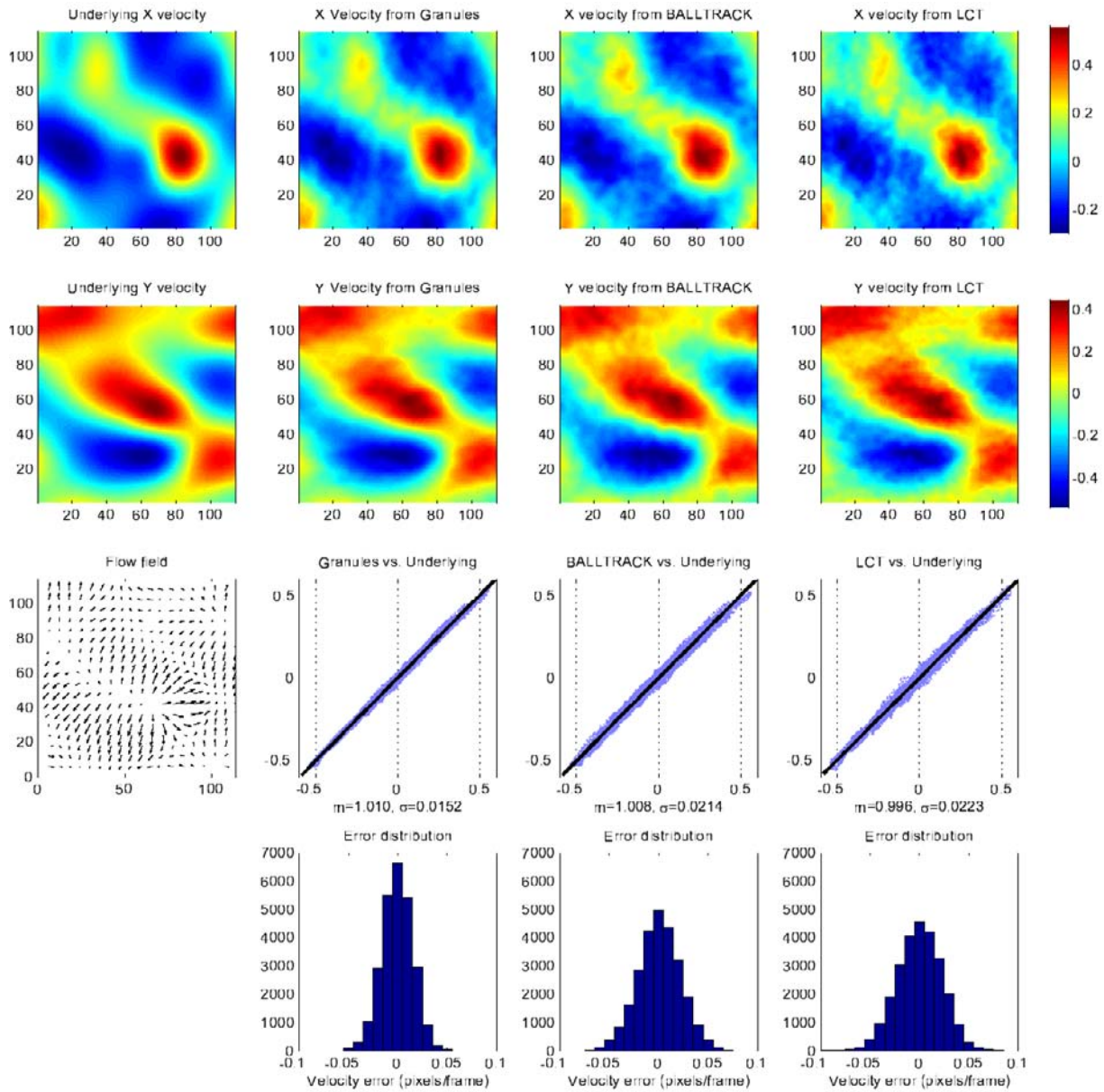


Fig. 7. Comparison of Balltrack and LCT recovered velocity fields. The first column is the underlying velocity field, the second the velocity field recovered from the exact granule positions. The third and fourth columns are the velocity field obtained using balltrack and LCT respectively. The graphs at the bottom show the correlation and error distribution between the recovered velocity fields and the underlying field. The units in all graphs are pixels and pixels/timestep.

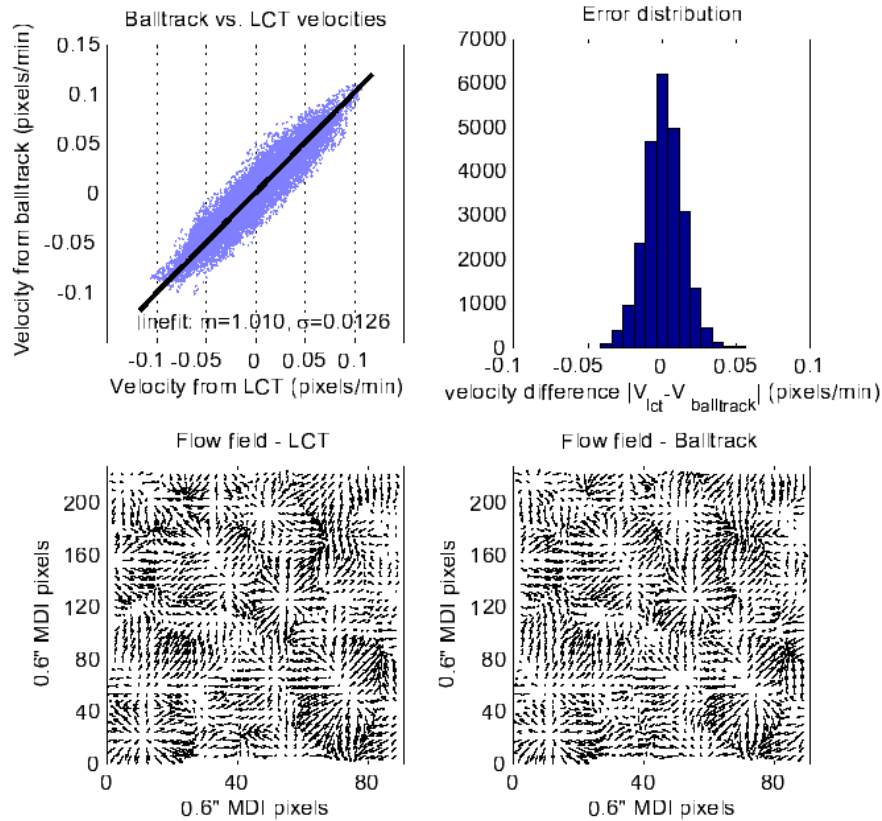


Fig. 8. Comparison of Balltrack and LCT recovered velocity fields from real MDI data. The original data set consisted of $240\,256 \times 256$ pixel (approx. $150 \times 150''$) frames taken at one minute intervals.

The velocity field obtained by smoothing the actual granule motions is also displayed. In all cases the data was smoothed to have 4 pixel spatial resolution and a smoothing time of 360 samples (equivalent to 6 h of MDI data). The test data was made with a higher velocity range than true granulation data, with typical bulk flow velocities reaching 0.2 pixels/s (equivalent to 1500 m s^{-1} on MDI data) to test that the balltracker gave a linear response over the full velocity range. The granule velocity was also higher, with an rms value of around 2000 m s^{-1} .

As can be seen from Fig. 7, the results from LCT and balltrack are very similar to each other, with almost identical variance from the original data. Note that the contribution to the variance of the result from the actual granule motion is larger than that from the tracking methods, showing that both these methods come close to the best possible result. The expected standard deviation of the velocity error on the data derived directly from the simulated granule motions, of about 0.015 pixels/timestep agrees well with the expected value of the uncertainty, calculated as described in Sect. 3.1 of 0.017 pixels/timestep. Note also the linear velocity response from both methods over the full range.

6. Application to real data

In order to evaluate the performance of the balltracking method we took a 240 min sample of high resolution MDI data, and compared the results from LCT and the balltracker. The data was first derotated, filtered and normalised as described in

Sect. 2.2. LCT was then run on it as described in Potts et al. 2003, using 9×9 pixel subimages, separated by 4 pixels in each direction, with the correlation weighted with a 2D Gaussian of $\sigma = 5$ pixels, centred on the cell. The balltracking was then performed with the following parameters: $\tilde{R}_s = 2$, $\tilde{D}_p = 0.2$, $\tilde{A}_m = 0.6$, $\tilde{T}_d = 1$. The ball spacing was $3\tilde{R}_s$ and the z damping was increased by a factor of 10 with respect to the xy damping. The results were smoothed with a smoothing radius of $\sigma = 4$ pixels. The Balltrack results were then calibrated by using a sample of $25\,9 \times 9$ pixel LCT subimages, chosen from regions with high flow.

A comparison of the results is shown in Fig. 8. The first graph shows the correlation between the x and y components of the recovered velocity field. As can be seen the correlation is good with $R^2 = 0.94$, and the scaling is correct to about 1%. The absolute differences in each velocity measurement are shown in the histogram in the top right. These have a standard deviation of 0.013 pixels/timestep, which is equivalent to a velocity uncertainty of 92 m s^{-1} , close to the maximum resolution possible.

We then measured the velocity field of the large 6GB, 36 h data set described in Sect. 2.2 using balltrack with the same parameters. Figure 9 shows the supergranular boundaries obtained from by using a cell finding algorithm on the velocity field from 3 h of MDI data. The dark lines are the points where the flow converges to make an intergranular lane, with the darkness of the lane proportional to the strength of convergence. An animation showing the

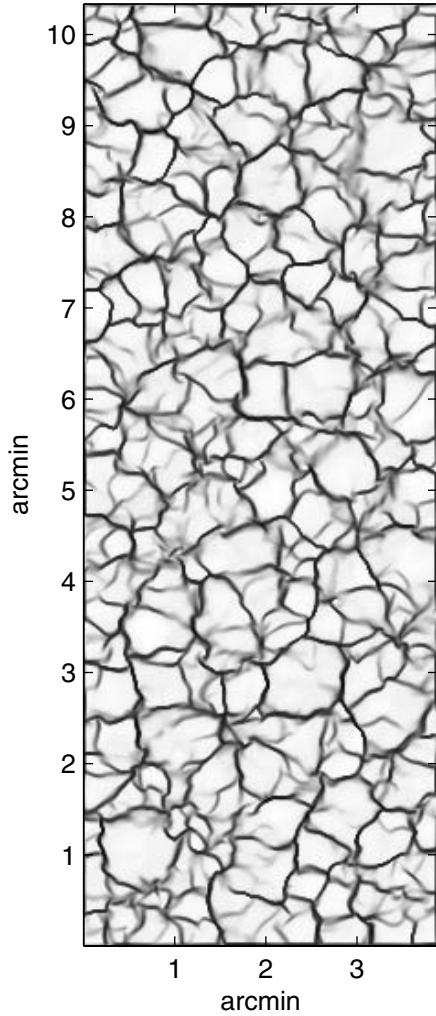


Fig. 9. Supergranular boundaries obtained from a 5 h, high resolution MDI data set. The image is 384×1024 pixels (3.8×10.2 arcmin). A movie showing the evolution of these lanes over a 36 h period may be found in `balltrack_realdata2.html` which is available with the electronic version of the article at <http://www.edpsciences.org/aa>.

time evolution of the supergranules over the full 36 h sample may be found in `balltrack_realdata2.html` which is available with the electronic version of the article at <http://www.edpsciences.org/aa>. In the movie several supergranules can be seen growing from a small, high divergence feature to full sized cell which then start to break up. This large data set took about 8 hours to calculate on a moderate desktop PC (1GHz Pentium 3), including all filtering and derotation. New data sets may be found on the web at <http://www.astro.gla.ac.uk/users/hugh/balltrack>.

7. Conclusions

We have developed a motion tracking technique, that produces results with the same accuracy as LCT, but is significantly more efficient. This increase in speed gives the potential for real time processing of large fields of data. This gives the capability to recognise dynamic flow features, and, for example change observing instrument parameters in reaction to them. When the

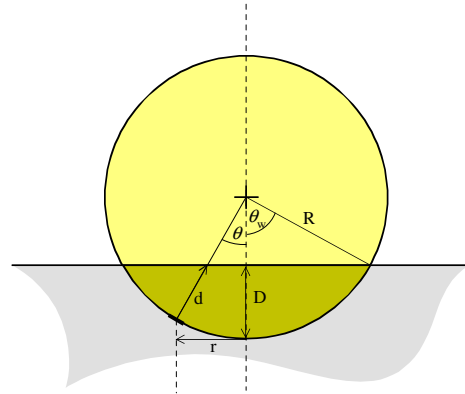


Fig. A.1. Ball on flat surface.

high cadence Solar-B data comes on line this type of algorithm will be essential, allowing rapid searching of data sets for velocity features.

Another advantage of the high efficiency of the code is that it can be written in a high level programming language, greatly reducing programming and debugging time. The code used here was written in MATLAB, and in total is only around 250 lines long. If you are interested in using the code please contact the authors. More information is available on the web from the Astronomy and Astrophysics website, <http://www.edpsciences.org/aa>, and from Potts (2003a).

The key to the high efficiency of the method is the implicit predictive data searching provided by the momentum of the trackers. This minimises the proportion of data used. For example, for MDI data the optimum spacing of the balls is around 3 times their radius. This means that the proportion of data that is used by the tracking algorithm is around $\pi/9 \sim 35\%$, as only the data directly under the trackers is used. The data is also only ever used once (twice if an intermediate frame is used). This is in contrast to any correlation method that by its nature uses all the data many times; for example, a single interpolation cycle involves each data point 4 times, and many such cycles are required.

The tracking accuracy obtained is close to the maximum possible, with the limitation being largely a function of the granule motion, rather than the tracking algorithm.

Although this method was designed for tracking photospheric flows, it will work in any situation where there is a characteristic spatial scale to features that need to be tracked. Other examples where it may work well would include tracking the motion of dense crowds of people from live video, tracking surface waves on the sea, clouds in the sky, or perhaps in industrial control systems tracking the movement of granular materials.

Acknowledgements. This work was funded by PPARC rolling grant number PPA/G/0/2001/00472.

Appendix A: Ball dynamics

A.1. Floating depth

In order to calculate the equations of motion for a floating ball we consider a ball floating on a flat surface as shown in

Fig. A.1. The ball has radius R and it penetrates the surface by a maximum depth D .

To set the depth that the ball floats at we need to set the force scaling factor k . To do this we first work out the force on an annular element of the surface of width $Rd\theta$ that makes an angle θ with the a vertical line through the centre of the sphere. The radial penetration distance of this element through the surface is d_i . Writing this in terms of the maximum penetration depth D_p gives:

$$d_i = R - \frac{R - D}{\cos \theta}. \quad (\text{A.1})$$

For simplicity we scale the penetration depth so that it is expressed as a fraction of the ball radius R , such that $\tilde{d}_i = d_i/R$ and $\tilde{D}_p = D/R$. Substituting these in gives:

$$\tilde{d}_i = 1 - \frac{1 - \tilde{D}_p}{\cos \theta}. \quad (\text{A.2})$$

We allocate a pressure P_i to this surface element proportional to the penetration distance, and acting radially to the centre of the sphere $P_i = k\tilde{d}_i$. Note that this is not the pressure that would be seen in the case of a real object floating in hydrostatic equilibrium, but is a convenient approximation which works well for dynamic surfaces. The force from this annulus is just the vertical component of the product of the pressure and its area A_i .

$$A_i = 2\pi R^2 \sin \theta d\theta \quad (\text{A.3})$$

$$\begin{aligned} f_i &= P_i \cos \theta A_i \\ &= k \left(1 - \frac{1 - \tilde{D}_p}{\cos \theta} \right) \cos \theta \cdot 2\pi R^2 \sin \theta d\theta. \end{aligned} \quad (\text{A.4})$$

The total vertical force from all elements is obtained by integrating around the submerged part of the ball up to the surface piercing point at θ_w . This force must balance the gravity force $-mg$:

$$\begin{aligned} -mg &= \int_0^{\theta_w} f_i d\theta \\ &= 2\pi k R^2 \left[\frac{1}{2} \sin^2 \theta + (1 + \tilde{D}_p) \cos \theta \right]_0^{\theta_w}. \end{aligned} \quad (\text{A.5})$$

Substituting in $\cos \theta_w = (R - D)/R = (1 - \tilde{D}_p)$ and $\sin^2 \theta = 1 - \cos^2 \theta$ and reducing gives:

$$mg = \pi k R^2 D_p^2. \quad (\text{A.6})$$

This gives a value for the constant k of:

$$k = \frac{mg}{\pi R^2 \tilde{D}_p^2}. \quad (\text{A.7})$$

For the case where the sphere is supported by discrete data points with separation in the xy plane of d_c , and scaling all quantities to this spatial scale then $\tilde{R}_s = R/d_c$ and each data point represents unit area. The total vector force on the sphere from the surface is then:

$$\mathbf{f}_b = \sum_i k d_i \quad (\text{A.8})$$

$$= \sum_i \frac{mg}{\pi \tilde{R}_s^2 \tilde{D}_p^2} \tilde{d}_i. \quad (\text{A.9})$$

A.2. Ball motion

The motion of the balls is calculated entirely in velocity space. In addition to the buoyancy force calculated above there is a dissipative drag force $-\alpha \mathbf{v}$, proportional to the velocity. In each time step δt the momentum \mathbf{p} of a ball changes by the impulse on it:

$$\delta \mathbf{P} = \delta t (\mathbf{f}_b - m\mathbf{g} - \alpha \mathbf{v}) \quad (\text{A.10})$$

or, in terms of velocity:

$$\delta \mathbf{v} = \frac{\delta t}{m} (\mathbf{f}_b - m\mathbf{g} - \alpha \mathbf{v}). \quad (\text{A.11})$$

If a ball is slowing down on a flat plane so the only force is the frictional force $-\alpha \mathbf{v}$, its velocity will go as:

$$m \frac{d\mathbf{v}}{dt} = -\alpha \mathbf{v} \quad \Rightarrow \quad \mathbf{v} = \mathbf{v}_0 \exp \frac{-\alpha t}{m}. \quad (\text{A.12})$$

Therefore the characteristic damping time is just:

$$\tilde{T}_d = \frac{m}{\alpha}. \quad (\text{A.13})$$

The maximum sustainable velocity change that can occur in a timestep \tilde{A}_m is equal to gravitational acceleration \mathbf{g} .

Substituting these values in to Eq. (A.11) gives the full equation of motion in terms of the parameters defined in Sect. 2.4:

$$\delta \mathbf{v} = \delta t \left(\frac{\tilde{A}_m}{\pi \tilde{D}_p^2 \tilde{R}_s^2} \sum_i \tilde{d}_i - \tilde{A}_m \hat{\mathbf{g}} - \frac{\mathbf{v}}{\tilde{T}_d} \right) \quad (\text{A.14})$$

where $\hat{\mathbf{g}}$ is the unit vector in the downwards or “gravity” direction.

References

- Berrilli, F., Consolini, G., Pietropaolo, E., et al. 2002, A&A, 381, 253
 De Rosa, M., Duvall, T. L., & Toomre, J. 2000, Sol. Phys., 192, 351
 Duvall, T. L., Kosovichev, A. G., Scherrer, P. H., et al. 1996, Sol. Phys., 170, 63
 EDP Sciences, Balltrack web pages 2004, <http://www.edpsciences.org/aa>
 Lisle, J., De Rosa, M., & Toomre, J. 2000, Sol. Phys., 197, 21
 November, L. J., & Simon, G. W. 1988, ApJ, 333, 427
 Rieutord, M., Roudier, T., Ludwig, H. G., Nordlund, A., & Stein, R. 2001, A&A, 377, L14
 Roudier, T., Rieutord, M., Malherbe, J. M., & Vigneau, J. 1999, A&A, 349, 301
 Scherrer, P. H., Bogart, R. S., Bush, R. I., et al. 1995, Sol. Phys., 162, 129
 Shine, R. A., Simon, G. W., & Hurlburt, N. E. 2000, Sol. Phys., 193, 313
 Potts, H. E., Barrett, R. K., & Diver, D. A. 2003, Sol. Phys., 217, 69
 Potts, H. E. 2003, Balltrack web pages <http://www.astro.gla.ac.uk/users/hugh/balltrack>
 Strous, L. 1994, Ph.D. Thesis, University of Utrecht
 Title, A. M., Tarbell, T. D., Topka, K. P., Fergusson, S. H., & Shine, R. A. 1989, ApJ, 336, 475