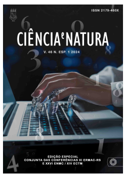
UFSM

CIÊNCIA e NATURA



ISSN 2179-460X
OPEN ACCESS

Ci. e Nat., Santa Maria, v. 46, n. spe. 1, e87224, 2024 • https://doi.org/10.5902/2179460X87224
Submitted: 03/22/2024 • Approved: 06/25/2024 • Published: 11/07/2024

ERMAC e ENMC

A look at the mathematics of radar-based nowcasting

Um olhar sobre na matemática da previsão a curto prazo baseada em dados de radar

Tiago Martinuzzi Buriol [I], **Leonardo Calvetti** [II], **Kerollyn Andrzejewski** [II], **Cesar Augustus Assis Beneti** [III],

[I] Universidade Federal de Santa Maria, Santa Maria, RS, Brazil
[II] Universidade Federal de Pelotas, Pelotas, RS, Brazil
[III] Sistema de Tecnologia e Monitoramento Ambiental do Paraná, Curitiba, PR, Brazil

ABSTRACT

AShort-term weather prediction systems that rely on radar information, known as nowcasting, involve multiple stages of numerical computation and visualization. The algorithms utilized encompass geometric transformations, statistical analysis, image processing, scalar and vector field calculations, and numerical computations based on various mathematical models. Current computer software used for nowcasting, which depends on data and images, requires a deep understanding of fundamental geometry, calculus, algebra, and various mathematical principles from both users and developers. This paper aims to provide a concise and straightforward overview of the process of handling weather radar data for visualization and nowcasting, while also delving into the mathematical principles underpinning these techniques. As a result, the application of mathematics topics covered in undergraduate courses was presented, in the context of their practical use in precipitation and severe events nowcasting systems.

Keywords: Mathematics; Image processing; Nowcasting; Radar data

RESUMO

Sistemas de previsão meteorológica de curto prazo baseados em dados de radares (*nowcasting*) requerem diversas etapas de computação numérica e visualização. Os algoritmos utilizados envolvem transformações geométricas, análises estatísticas, processamento de imagens, cálculo de campos escalares e vetoriais e computação numérica envolvendo diferentes modelos matemáticos. Os modernos pacotes computacionais para *nowcasting* com base em dados e imagens demandam de seus usuários e desenvolvedores conhecimento em geometria, cálculo, álgebra e outros tópicos da matemática. Este trabalho tem como objetivo descrever de forma breve e simplificada o fluxo básico do processamento de dados de radares meteorológico para a visualização e *nowcasting* e, a partir disso, apresentar um pouco da matemática existente por trás desses métodos. Como resultado foi apresentada a aplicação de tópicos de matemática abordados em cursos de graduação, no contexto do seu uso prático em sistemas de previsão de chuva e eventos severos a curto prazo.

Palavras-chave: Matemática; Processamento de imagem; Nowcasting; Dados de radar

1 INTRODUCTION

Nowcasting is short-term weather forecast with a local scale of up to 3-6 hours ahead, including a detailed description of the current state of the atmosphere (Wang et al., 2017). Obtaining reliable forecasts is a challenge since individual storms last, on average, less than an hour, move at speeds of up to 60 km/h, and can join and separate during their short life cycle. To this day, operational numerical models are not capable of making forecasts with sufficient accuracy on this scale, requiring extrapolation methods and artificial intelligence methods (Cui et al., 2023; Guo et al., 2023; Liu et al., 2022).

The primary observational system for identifying and tracking precipitating storms is weather radar. Radars emit an electromagnetic signal and measure the reflected amount at short intervals, providing a spatial resolution of 1 km and multiple vertical levels. This capability allows for a 3D visualization of weather events. The images from radar data enable the identification, tracking, and prediction of a storm's future behavior. Despite recent advances in numerical forecasting and machine learning methods, the predominant approach in operational environments continues to be radars data extrapolation-based forecasting (Pulkkinen et al., 2019).

Precipitation is highly variable in space and time, making it difficult to predict in a deterministic manner due to the chaotic nature of the atmosphere. Therefore, in recent decades, the problem has been addressed probabilistically. One approach involves introducing random perturbations to a set of initial conditions, creating an ensemble that reflects the uncertainties inherent in the prediction. Recently, a weather radar community developed and maintained a modular and open-source Python framework called PySTEPS (Pulkkinen et al., 2019). It provides a range of routines and algorithms for optical flow, storm detection and tracking, and ensemble generation. Projects like PySTEPS, alongside others such as WradLib (Heistermann et al., 2013) and PyART (Helmus and Collis, 2016), have proven to be very useful for radar data users worldwide.

At SIMEPAR (Parana's Technology and Environmental Monitoring System), we have tested and utilized the PyART, WradLib, and PySTEPS libraries for research and the development of nowcasting projects. Additionally, they are used to generate visualization products based on meteorological radar data. The algorithms in these systems encompass several mathematical models, calculations, and numerical methods, requiring proficiency in geometry, calculus, algebra, and various mathematical disciplines from users and developers. This paper aims to provide a brief description of the basic data processing flow for visualizing weather radar data and nowcasting usage. From there, we will present some of the mathematical techniques used in these methods. This text can provide a comprehensive explanation of a significant real-world application for the mathematics concepts taught in many courses.

2 METHODOLOGY

The methodology used in this study involved testing and evaluating certain algorithms from the PySTEPS nowcasting library. These algorithms were tested using weather radar data provided by SIMEPAR. The functionalities, modules, and subroutines of the library were then evaluated to better understand specific aspects of computational mathematics. The algorithmic execution involved establishing intermediate numerical processing methods, such as cascade decomposition, derivation of the advection field, and identifying features within the image. A brief exposition of the underlying mathematics was provided for the basic processes, covering methods that evoke topics of calculus, algebra, geometry, and numerical methods.

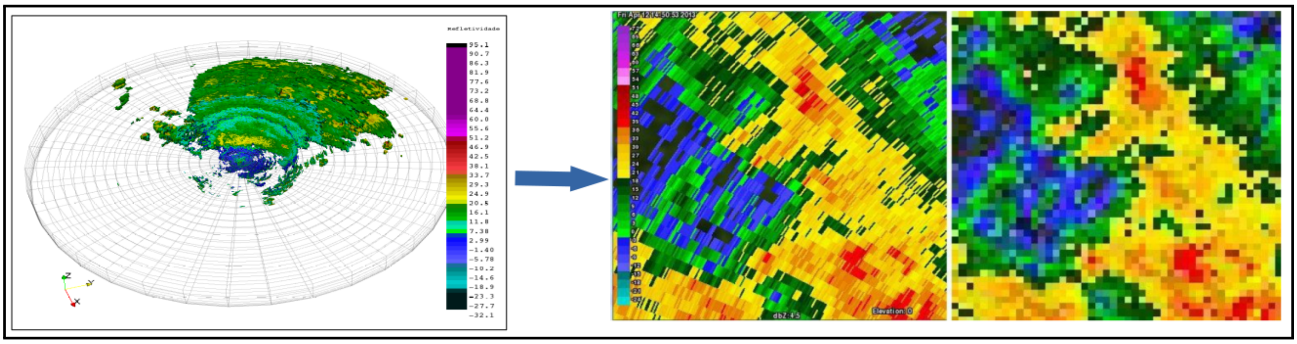
2.1 Radar data

Radar data collection involves the emission of beams of electromagnetic waves in various directions through the rotation of an antenna around a vertical axis. This process results in the creation of a regular grid of points defined by their spherical coordinates r, θ, and ϕ, which respectively represent *range*, *elevation*, and *azimuth*. Repeated scans at different elevations generate what is commonly referred to as a radar data volume.

SIMEPAR operates a radar that generates data every 1 degree through azimuthal scans and completes a full 360-degree rotation for each antenna elevation. At the end of one full cycle, 360 azimuthal scans are completed with 14 different antenna elevations, resulting in 800 evenly spaced points along each 200 km beam. This generates a grid in spherical coordinates with a resolution of 360x14x800, amounting to a total of 4,032,000 points. Each of these points may contain dozens of associated variables. On average, a data volume is generated every 5 minutes.

The reflectivity factor variable (Z) represents the ratio between the irradiance emitted by the radar and that received after being backscattered by hydrometeors (raindrops) present in the atmosphere. The unit used is dBZ, which is a logarithmic scale of reflectivity. Values range typically from -24 dBZ (minimum) to zero and up to 63 dBZ (maximum). Larger values indicate larger drop diameters (greater than 3 mm) within the measured volume, indicating higher precipitation intensity (Fig. 1). From this data, ”products” are obtained, which are 2D images representing one or more variables, which can be the projection of an elevation (PPI - Plan Position Indicator) or in a horizontal section of the volume (CAPPI - Constant Altitude Plan Position Indicator).

Figure 1 – Illustrative example of the process of converting 3D data into 2D products



Source: The authors (2024)

The projection of data onto the Earth’s surface is derived from the spatial location of the geometric center of the cone swept by the emitted energy. This process is mapped through geometric transformations from spherical coordinates to cartesian coordinates and geographical coordinates.

2.2 Nowcasting with PySTEPS

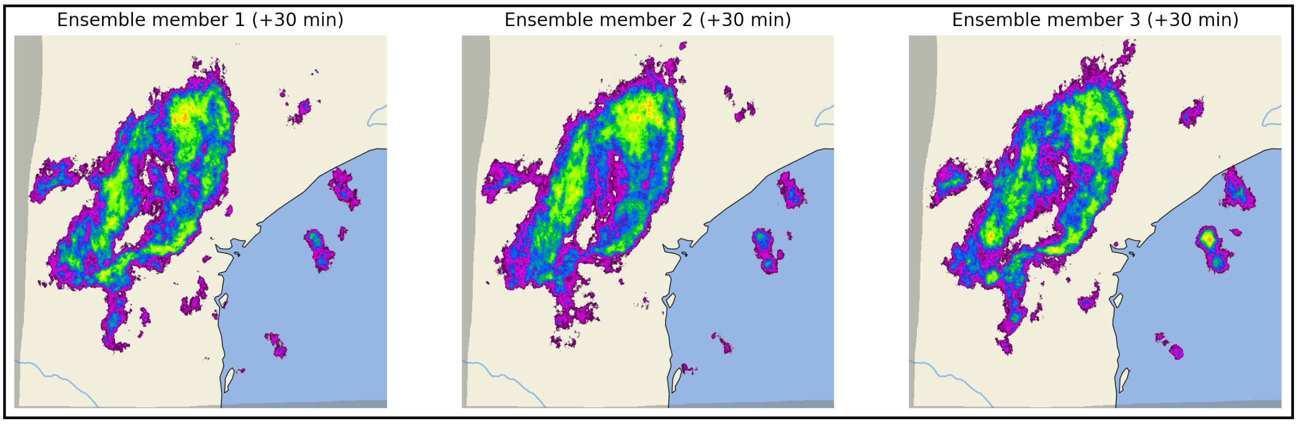
PySTEPS is a Python framework designed to be user-friendly, modular, free, and open source for developing ensemble-based nowcasting systems and research (Imhoff et al., 2023; Pulkkinen et al., 2019). The focus lies on the probabilistic forecasting of precipitation fields, although it is not limited to this application, allowing deterministic forecasts through extrapolation, storm detection, tracking, and various other applications. Extrapolation algorithms generate deterministic forecasts using methods that preserve variance or minimize error. Probabilistic methods are based on quantiles or ensemble.

Among the nowcasting models implemented in PySTEPS, one of the most complete is the Lagrangian Integro-Differential Equation with Autoregression (LINDA) model, which combines several other models and can produce deterministic and probabilistic forecasts. It was designed specifically for forecasting localized heavy rainfall and has the following components: (1) feature detection to identify rain cells; (2) advection-based extrapolation; (3) integrated autoregressive process for rainfall growth and decay; (4) convolution to account for loss of predictability; and (5) stochastic disturbances to simulate forecast errors.

An ensemble can be generated by introducing random perturbations to the precipitation field or advection field to produce a numerous set of predictions, thus simulating the chaotic nature of the phenomenon. The ensemble-based algorithm produces a time series with a set of predictions having multiple members for each time step. The average of many members can be comparable to deterministic methods. Figure 2 illustrates some members of an ensemble generated by the PySTEPS model, the mean, and a probabilistic forecast for a precipitation threshold, calculated based on the ensemble.

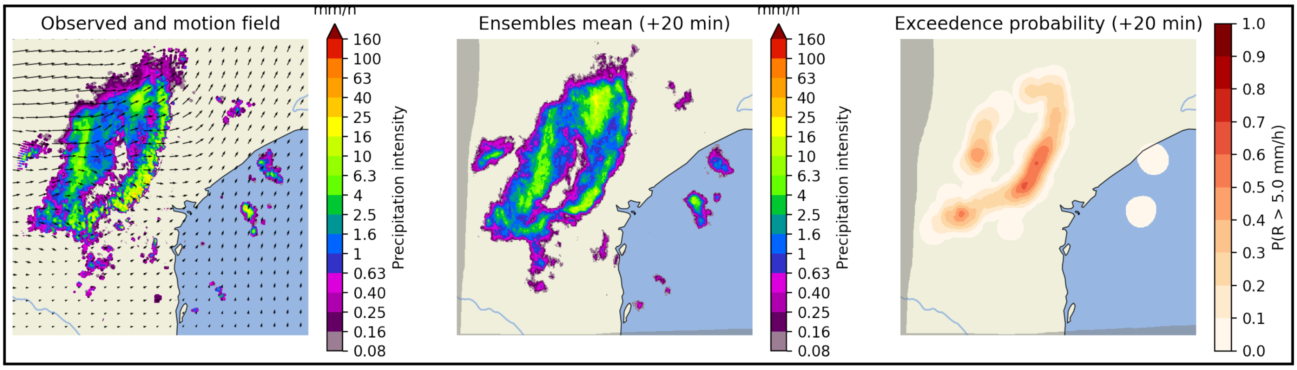
An essential element of nowcasting models is optical flow, consisting of a two-dimensional vector field describing motion for the advection field. Optical flow is the distribution of apparent velocity of motion of intensity patterns in an image, i.e., it describes apparent motion in a sequence of digital images. Two algorithms used to determine optical flow are the Lucas-Kanade (Lucas and Kanade, 1981) and Proesman (Proesmans et al., 1994).

Figure 2 – Members of a ensemble


Ensemble member 1 (+30 min)
Ensemble member 2 (+30 min)
Ensemble member 3 (+30 min)

Source: The authors, 2024

Figure 3 – Members of a ensemble



Source: The authors (2024)

The algorithm takes as input an ordered set of data in the form of a two-dimensional field of precipitation rates (mm h^{-1}) and the vector field of advection with velocities representing a time step between two input data points. Other parameters to be defined include the number of time steps to be forecasted, the number of ensemble members, the number of cascade levels to be used, the extrapolation method, cascade decomposition method, bandpass filter, noise generation method, autoregression order, among others.

3 RESULTS AND COMMENTS

The results presented in this section correspond to a small selection of numerical methods and computer vision algorithms involving topics of calculus and algebra that are employed in radar data processing-based nowcasting systems. par The selection of parameters and methods used in the models requires familiarity with mathematical topics commonly covered in undergraduate courses, underscoring their significance in such applications. Therefore, this section will describe some typical subroutines of these nowcasting systems and provide an overview of the mathematics involved.

3.1 Precipitation field

The precipitation field is obtained from radar reflectivity data and mapping geographical coordinates. The distance r from a point in space to the radar antenna corresponds to the distance s of this point projected onto the Earth's surface up to the radar base. The coordinates are mapped from the spherical system to polar and Cartesian systems and then to the cartographic coordinate system through equations such as those proposed by (Heymsfield et al., 1983). In these equations, R' is equal to 3/4 of the Earth's radius, and (x, y) are the Cartesian coordinates centered at the radar antenna.

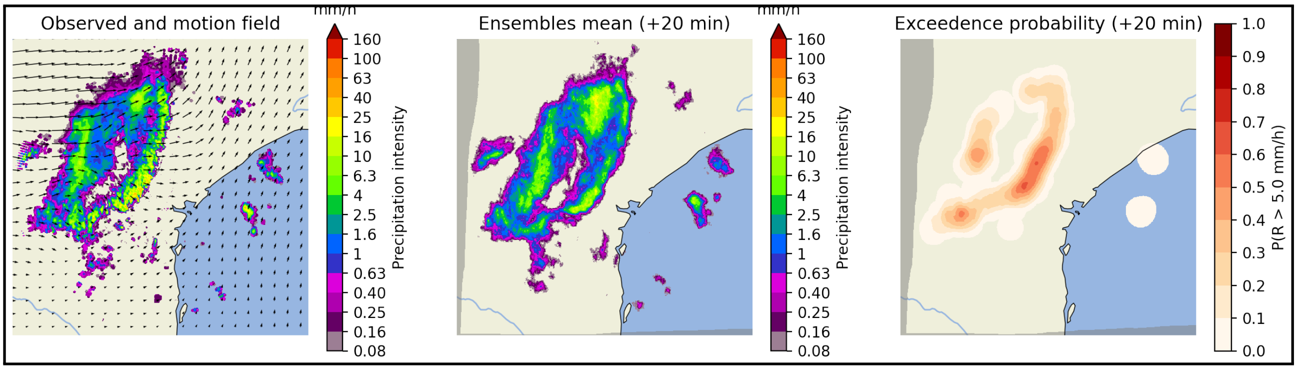
$$(x, y) = (s\, sen\theta, s\, cos\theta) \quad \text{with} \quad s = R' tan^{-1} \frac{r\, cos\phi}{R' + r\, sen\phi} \tag{1}$$

The precipitation field is a matrix with these values distributed on a regular grid of Cartesian coordinates, where each coordinate point (x, y) is associated with a precipitation rate value $R(x, y)$. Thus, precipitation is a function of two variables, and its visualization can be achieved through color maps or contour plots. However, the precipitation rate at each point must be calculated based on a logarithmic transformation from the reflectivity values provided by the radar. Some typical equations for these transformations are shown below, where a and b are experimentally obtained constants, and Z is the reflectivity.

$$Z = aR^b \quad \text{or} \quad R = \left(\frac{Z}{a}\right)^{\frac{1}{b}} = \left(\frac{10^{\frac{dBZ}{10}}}{a}\right)^{\frac{1}{b}} \tag{2}$$

The transformed fields are used to determine the motion fields with one of the optical flow methods (Pulkkinen et al., 2019). The Fig. 4 show a standard color mapped visualization of a reflectivity field (left), a precipitation field visualization obtained whith PySteps and a forecast precipitation 30 min ahead.

Figure 4 – Example of reflectivity field, transformed precipitation field and a forecast precipitation 30 min ahead.



Source: The authors (2024)

Understanding the mathematics involved in the basic process of converting the reflectivity field into its corresponding precipitation field requires background in topics such as trigonometric functions and their inverses, logarithmic and exponential functions, as well as two- or three-variable functions. Topics typically covered in Calculus I and II courses.

3.2 Motion field

In this text, the motion field refers to a two-dimensional vector field that describes the distribution of apparent velocity of motion of intensity patterns in an image. One of the methods implemented in PySTEPS is the Lucas-Kanade method (Lucas and Kanade, 1981), also known as the method of differences. It is a non-iterative method that assumes a constant optical flow at the point. A semi-Lagrangian backward-in-time scheme is then applied to obtain the advection field from the motion field.

This method assumes that the flow is essentially constant in a local neighborhood of the pixel under consideration and solves the basic optical flow equations for all pixels in that neighborhood using the least-squares criterion. The region is divided into windows of size $n \times n$, each with $p = n^2$ pixels. The local motion constraint used allows constructing an overdetermined system with p equations and only 2 unknowns. Thus, the optical flow equation can be considered valid for all pixels within a window centered at a point P. In other words, the local flow vector (V_x, V_y) must satisfy $I_x(p_i)V_x + I_y(p_i)V_y = -I_t(p_i)$ with $i = 1, 2, ..., n$. Where $p_1, ..., p_n$ are the pixels within the window, and I_x, I_y, and I_t are the partial derivatives of the image. These equations can be arranged in matrix form $Av = b$, where A is the matrix composed of the rows $[I_x(q_i), I_y(q_i)]$, $v = [V_x, V_y]^T$ is the unknown vector and $b = -[I_t(q_1), ..., I_t(q_n)]^T$. This system is generally overdetermined, and a solution can be obtained by a least-squares approximation.

$$\hat{v} = \arg\min_{v} \|Av - b\|^2 \quad \text{given by} \quad v = (A^T A)^{-1} A^T b \tag{3}$$

An advection vector is then added to each pixel of the image. The image on the left in Figure 3 shows an advection field obtained using this approach.

3.3 Image feature detection

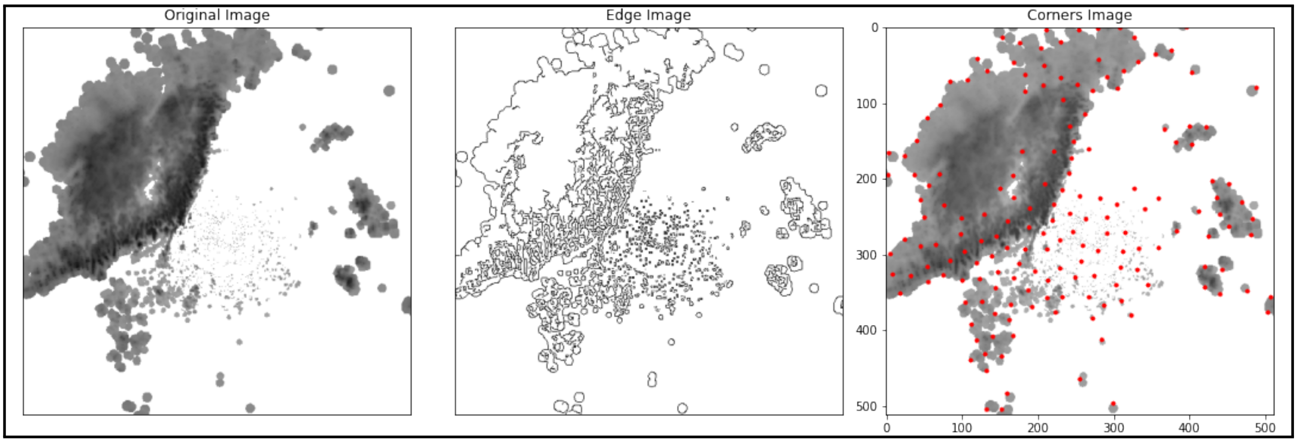
Image features are points of interest that provide important and useful information for characterizing the image based on edge maps, corners, and blobs. These points of interest are invariant under rotation, translation and changes in scale or intensity. In nowcasting, features are employed to detect and track storms and other events, and they are also used to feed machine learning-based algorithms. Corners in an image are defined as the intersection of two edges, which are sudden changes in brightness occurring in a single direction. Corners can be detected by searching for significant variations while shifting a small window of $n \times n$ pixels in all directions in the image. Blobs are regions of homogeneous area with roughly equal intensity values compared to the surrounding.

Corners can be found by looking for regions in the image with large intensity variations in all directions. The intensity difference $I(x, y)$ when shifting the pixel position by (u, v) is given by $I(x + u, y + v) - I(x, y)$. In corners, this intensity is high, so we seek the maxima of the function

$$E(u,v) = \sum_{x,y} \underbrace{w(x,y)}_{\text{window function}} \, [\underbrace{I(x+u, y+v)}_{\text{shifted intensity}} - \underbrace{I(x,y)}_{\text{intensity}}]^2 \tag{4}$$

by differentiating with respect to the X and Y axes. Then, using a Taylor series expansion in two variables, we can write $I(x+u, y+v) \approx I + uI_x + vI_y$ where I_x and I_y are the partial derivatives, providing us with the approximation.

Figure 5 – Example of result in edge and corner detection



Source: The authors (2024)

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} \text{ where } M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix} \tag{5}$$

After calculating the matrices with intensity variations, it is sufficient to select those containing corners using the eigenvalues of the matrix. A score R is assigned such that

$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 \tag{6}$$

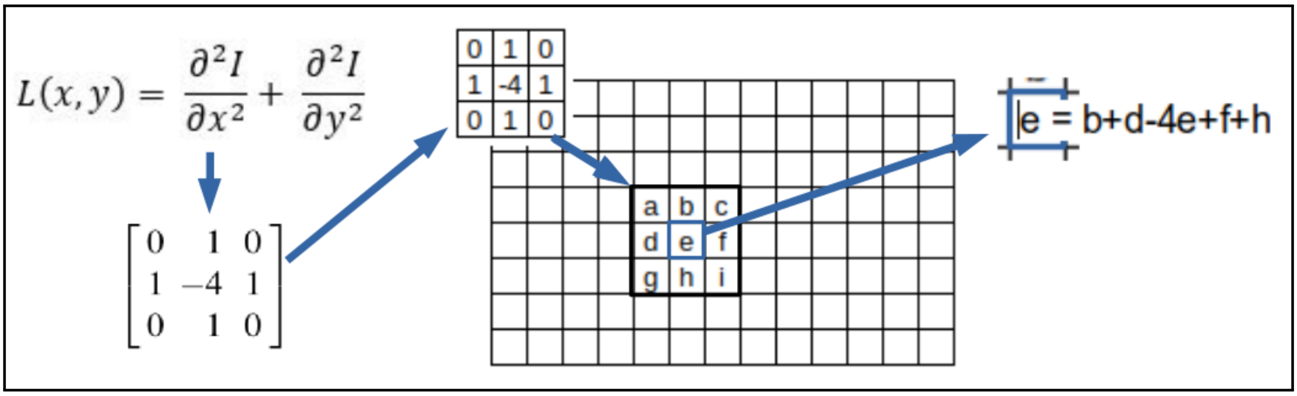
where λ_1 and λ_2 are the eigenvalues of M. In the end, a high value of R indicates a corner, while a negative value indicates an edge.

Other features are the so-called blobs, which are regions of an image with a transition of values, such as from light to dark, used to detect edges. Algorithms can use different methods such as: (1) Laplacian of Gaussian (LoG); (2) Difference of Gaussians (DoG); and (3) Determinant of the Hessian (DoH).

Laplacian is a second-order derivative operator that detects the zero-crossing in image intensity. The Laplace filter represents a discrete approximation to the mathematical Laplace operator. A Laplacian of an image $L(x,y)$ with intensity I per pixel is defined as the equation

$$L(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{7}$$

Figure 6 – Illustration of Laplacian filtering


$L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$
$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$
0 1 0
1 -4 1
0 1 0
a b c
d e f
g h i
e = b+d-4e+f+h

Source: The authors (2024)

where $\frac{\partial^2 f}{\partial x^2}$ e $\frac{\partial^2 f}{\partial y^2}$ are second-order derivatives with respect to x and y. As an image is a set of discrete pixels, we use second-order centered finite-difference approximations through convolution.

Convolution is a process performed on each pixel of the original image by summing the values of the surrounding pixels, multiplied by a defined value in a matrix known as the convolution kernel. The Laplacian filter sharpening process is essentially a convolution process.

Mathematically, convolution can be interpreted as the integral of the product of two functions and also as an inner product. The application of the Laplacian filter to an image through convolution is illustrated in the Figure 5. In image processing, a zero crossing denotes a point where the sign of a mathematical function changes in the function's graph.

The Laplacian accentuates abrupt changes in intensity within an image, effectively enhancing features marked by sharp discontinuities. Serving as a second-order filter in image processing, it is used primarily for edge detection and feature extraction. Unlike first-order derivative filters, which require distinct filters for vertical and horizontal edges, the Laplacian filter adeptly identifies edges in all directions simultaneously.

Using the Laplacian filter for edge detection, potential edge points are identified at zero crossings in the graph. While effective for detecting edges in various directions, this method may perform poorly in the presence of image noise. To address this, a common practice involves smoothing the image with a Gaussian filter before applying the Laplacian filter, creating a Laplacian of Gaussian (LoG) filter. In two dimensions, the

Gaussian filter can be described in mathematical terms as

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{8}$$

where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution.

The combined LoG filter involves the integration of Gaussian and Laplacian operations, as mathematically represented as follows

$$LoG(x,y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2+y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{9}$$

Difference of Gaussian (DoG) is calculated as the difference between two smoothed versions of an image obtained by applying two Gaussian kernels of different standard deviations (sigma) on that image. DoG generally serves as an edge enhancement algorithm that delineates the high-frequency content of the image free from noise Misra and Wu (2020).

Furthermore, the Determinant of Hessian (DoH) can also be utilized as an alternative approach for blob detection. This technique involves calculating the determinant of the Hessian matrix for each pixel in the image. The Hessian matrix represents the local curvature of the image intensity and the determinant highlight regions with significant curvature changes. By identifying local maxima in the determinant of the Hessian matrix, we can detect and localize blobs effectively.

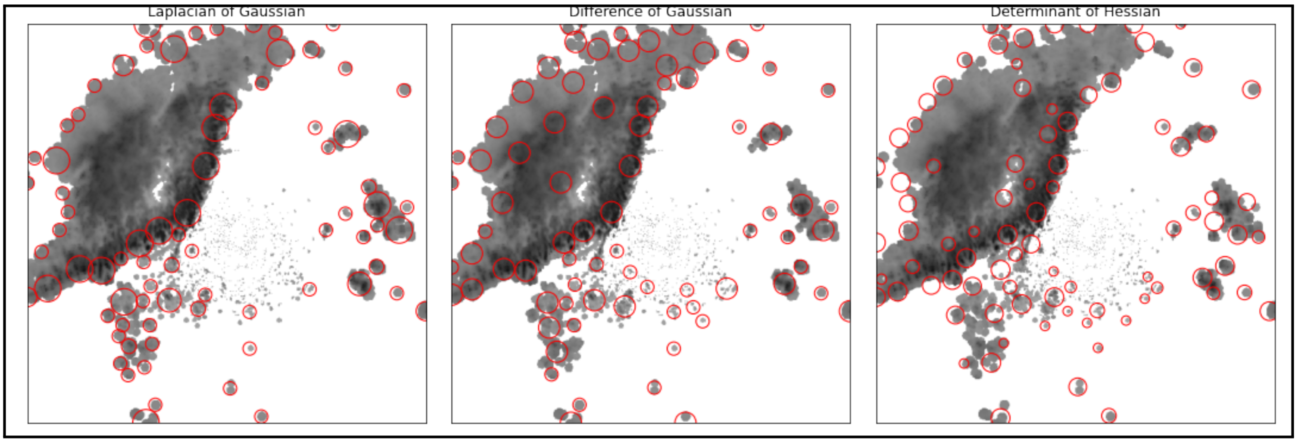
An image is scaled to a size defined by the scale parameter σ. The matrix H_σ is the Hessian matrix at a specific image location in level σ.

We can use the normalized determinant response of the Hessian

$$\sigma^4 \cdot \det(H_\sigma) = \sigma^4 \cdot \left(\partial_{xx} \cdot \partial_{yy} - \partial_{xy}^2\right) \quad \text{with} \quad H_\sigma = \begin{pmatrix} \partial_{xx} & \partial_{xy} \\ \partial_{yx} & \partial_{yy} \end{pmatrix}. \tag{10}$$

to detect blobs by searching for maxima in each image location across scale. Here, the $\partial_{xx} = \frac{\partial^2 I}{\partial^2 x^2}$ denoting the second-order derivative of the image I along the x-axis. The Fig. 7 illustrates the detection of blobs using the skimage feature module (Van der Walt et al., 2014).

Figure 7 – Illustration of LoG, DoG and DoH blobs detection



Source: The authors (2024)

3.4 Cascade decomposition and Fourier transform

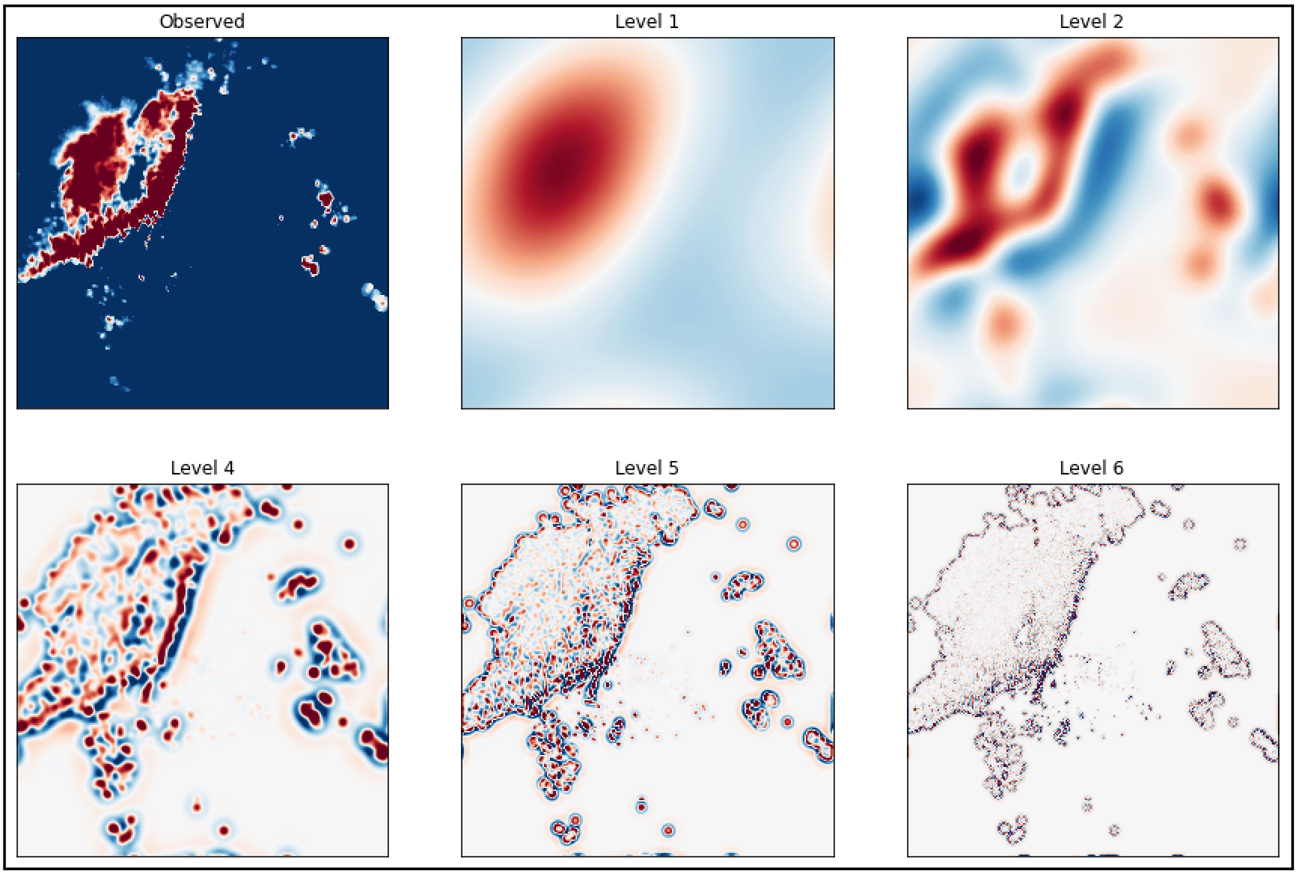
The lifetime of precipitation is related to its spatial scale, so it is useful to decompose the precipitation field into a multiplicative cascade. The cascade levels represent different spatial scales and treat them separately in the short-term forecast model. In PySTEPS, the scale specification is performed by applying a Fourier transform to the precipitation field as input (an example is shown in Fig. 8).

The Fourier transform is an integral transform that allows expressing a function in terms of sinusoidal trigonometric functions. This enables the decomposition of functions into a sum of sine and cosine functions, facilitating the analysis of their characteristics, such as measuring frequency, amplitude, and phase more easily.

A Fourier Series, with period T, is an infinite sum of sinusoidal functions (cosine and sine), each with a frequency that is an integer multiple of 1/T (the inverse of the fundamental period). The Fourier Series also includes a constant, and hence can be written as:

$$f(t) = a_0 + \sum_{m=1}^{\infty} a_m \cos\left(\frac{2\pi m t}{T}\right) + \sum_{n=1}^{\infty} b_n \sin\left(\frac{2\pi n t}{T}\right) \tag{11}$$

Figure 8 – Cascade decomposition



Source: The authors (2024)

The constants a_m, b_n are the coefficients of the Fourier Series. These determine the relative weights for each of the sinusoids. The optimal coefficients are given by

$$a_0 = \frac{1}{T}\int_0^T f(t)dt\,,\quad a_m = \frac{2}{T}\int_0^T f(t)\cos\left(\frac{2\pi mt}{T}\right)dt \quad \text{and} \quad b_n = \frac{2}{T}\int_0^T f(t)\sin\left(\frac{2\pi nt}{T}\right)dt \tag{12}$$

In many applied fields, using complex numbers makes things easier to understand and more mathematically elegant. The complex form of Fourier Series is obtained using the Euler identity $e^{it} = \cos t + i\sin t$ where $i = \sqrt{-1}$ and is given by

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{i\frac{2\pi nt}{T}} \quad \text{where} \quad c_n = \frac{1}{T}\int_0^T f(t)e^{-i\frac{2\pi nt}{T}}dt \tag{13}$$

The Fourier Transform shows that any waveform can be re-written as the sum of sinusoidals. The continuous Fourier transform for a single-variable function is expressed in both exponential and trigonometric forms, and these are shown below.

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(w)e^{iwt}dw \quad \text{or} \quad f(t) = \frac{1}{\pi}\int_{0}^{\infty}(A(w)\cos(wt) + B(w)\sin(wt))dw \tag{14}$$

The primary objective of the Fourier transform is to convert a signal from its original representation in the time or spatial domain into a different representation in the frequency domain and vice versa. Using this transform it is possible to express any periodic signal as a combination of sinusoidal waves featuring different amplitudes, frequencies, and phases. It allows us to dissect complex signals into simpler components. Since radar data are discrete sampling points, a discrete transform is necessary.

When the function is given by a discrete data set, the discrete transform is used. The discrete Fourier transform (DFT) converts a finite sequence of equally spaced samples of a function into a same-length sequence of equally-spaced samples of the discrete transform, which is a complex-valued function of frequency. The DFT of a discrete time signal $f(n)$ is defined as:

$$F_k = \sum_{n=0}^{N-1} f_n e^{-i2\pi kn/N} \quad k = 0, \ldots, N-1 \tag{15}$$

Where N is the number of samples, k represents the index in the frequency domain, and f_n is the signal value in the nth sample. The Fast Fourier Transform (FFT) is an optimized algorithm for the implementation of the DFT. Equation 15 represents the discrete analogue of equations 14 in complex form. In PySTEPS, Gaussian weight functions are used to separate the Fourier spectrum into a set of radial bands. After FFT and Gaussian filtering, each frequency band is transformed back into the spatial domain, resulting in a cascade with n levels, each representing a different scale. With a discrete Fourier transform, the precipitation field is decomposed in such an additive cascade by applying Gaussian filtering with weight functions and also by back-transforming this to the grid space. This results in k cascade levels representing different spatial scales.

4 CONCLUSIONS

This study showed how computational mathematics methods contribute to meteorological forecasting through the analysis of functionalities implemented in PySTEPS algorithms. Short-term forecasting (nowcasting) requires data processing including interpolations, storm identification, tracking, and advection field methods that operate rapidly and efficiently on a computer, providing operational results within a few minutes. PySTEPS is currently undergoing testing in Brazil at SIMEPAR, UFSM, and UFPEL through an innovation cooperation program (MAI/DAI CNPq 68 2022). Considering the challenge of monitoring and predicting severe storms, the methodology is intended to serve as a stimulus for teaching and research at the undergraduate and graduate levels in computational mathematical methods. The practical implementation of a short-term forecasting system is valuable for the monitoring of smart cities, contributing not only to the forecasting process, but also feeding into artificial intelligence systems.

REFERENCES

Cui, X., Chen, M., Qin, R., Li, C., and Han, L. (2023). The roles of surface convergence line and upper-level forcing on convection initiation ahead of a gust front: A case study. *Journal of Geophysical Research: Atmospheres*, 128(3):e2022JD036921.

Guo, S., Sun, N., Pei, Y., and Li, Q. (2023). 3d-unet-lstm: A deep learning-based radar echo extrapolation model for convective nowcasting. *Remote Sensing*, 15(6):1529.

Heistermann, M., Jacobi, S., and Pfaff, T. (2013). An open source library for processing weather radar data (wradlib). *Hydrology and Earth System Sciences*, 17(2):863–871.

Helmus, J. and Collis, S. (2016). The python arm radar toolkit (py-art), a library for working with weather radar data in the python programming language, j. open res. softw., 4, e25.

Heymsfield, G. M., Ghosh, K. K., and Chen, L. C. (1983). An interactive system for compositing digital radar and satellite data. *Journal of Applied Meteorology and Climatology*, 22(5):705–713.

Imhoff, R. O., De Cruz, L., Dewettinck, W., Brauer, C. C., Uijlenhoet, R., van Heeringen, K.-J., Velasco-Forero, C., Nerini, D., Van Ginderachter, M., and Weerts, A. H. (2023). Scale-dependent blending of ensemble rainfall nowcasts and numerical weather prediction in the open-source pysteps library. *Quarterly Journal of the Royal Meteorological Society*.

Liu, P., Yang, Z., Wang, X., Qiu, X., and Yang, Y. (2022). Assimilation of the pseudo-water vapor derived from extrapolated radar reflectivity to improve the forecasts of convective events. *Atmospheric Research*, 279:106386.

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *IJCAI'81: 7th international joint conference on Artificial intelligence*, volume 2, pages 674–679.

Misra, S. and Wu, Y. (2020). Chapter 10 - machine learning assisted segmentation of scanning electron microscopy images of organic-rich shales with feature extraction and feature ranking. In Misra, S., Li, H., and He, J., editors, *Machine Learning for Subsurface Characterization*, pages 289–314. Gulf Professional Publishing.

Proesmans, M., Van Gool, L., Pauwels, E., and Oosterlinck, A. (1994). Determination of optical flow and its discontinuities using non-linear diffusion. In *Computer Vision—ECCV'94: Third European Conference on Computer Vision Stockholm, Sweden, May 2–6 1994 Proceedings, Volume II 3*, pages 294–304. Springer.

Pulkkinen, S., Nerini, D., Pérez Hortal, A. A., Velasco-Forero, C., Seed, A., Germann, U., and Foresti, L. (2019). Pysteps: An open-source python library for probabilistic precipitation nowcasting (v1. 0). *Geoscientific Model Development*, 12(10):4185–4219.

Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., and Yu, T. (2014). scikit-image: image processing in python. *PeerJ*, 2:e453.

Wang, Y., Coning, E., Harou, A., Jacobs, W., Joe, P., Nikitina, L., Roberts, R., Wang, J., Wilson, J., Atencia, A., Bica, B., Brown, B., Goodmann, S., Kann, A., Li, P. W., Monterio, I., Schmid, F., Seed, A., and Sun, J. (2017). *Guidelines for Nowcasting Techniques*.

Author contributions

1 – Tiago Martinuzzi Buriol
PhD in Numerical Methods in Engineering from Universidade Federal de Santa Maria. Assistant Professor in the Department of Mathematics at UFSM
https://orcid.org/0000-0002-3054-4434 • tiago.buriol@ufsm.br
Contribution: Conceptualization – Data curation – Formal Analysis – Investigation – Methodology – Software – Validation – Visualization – Writing – original draft – Writing – Review & Editing

2 – Leonardo Calvetti
PhD in Meteorology from the University of São Paulo. Professor at the School of Meteorology at Universidade Federal de Pelotas
https://orcid.org/0000-0002-0620-5504 • lcalvetti@gmail.com
Contribution: Data curation – Funding acquisition – Formal Analysis – Investigation – Project administration – Supervision – Validation – Visualization – Writing – Review & Editing

3 – Kerollyn Andrzejewski
Degree in Metereology from the Universidade Federal de Pelotas. Master's student in Metereology at UFPel
https://orcid.org/0009-0001-7632-9692 • kekerollynoli@gmail.com
Contribution: Data curation – Formal Analysis – Investigation – Visualization – Writing – Review & Editing

4 – Cesar Augustus Assis Beneti
PhD in Meteorology from the University of São Paulo. Executive Director and Coordinator of Monitoring and Forecasting at Sistema de Tecnologia e Monitoramento Ambiental do Paraná (SIMEPAR)
https://orcid.org/0000-0002-0635-4710 • beneti@simepar.br
Contribution: Data curation – Funding acquisition – Project administration – Supervision – Visualization – Writing – Review & Editing

How to cite this article

Buriol, T. M., Calvetti, L. Andrzejewski, K., & Beneti, C. A. A. (2024). A look at the mathematics of radar-based nowcasting. *Ciência e Natura*, Santa Maria, v. 46, spe. 1, e87224. https://doi.org/10.5902/2179460X87224