

04: Jacobian Applied & The Jacobian Matrix

Chapter Goal: To "level up" the concept of the [Jacobian](#). We will transition from the Jacobian of a scalar-output function (the Gradient) to the **Jacobian Matrix** for a vector-output function.

1. Recap: Jacobian as the Gradient Vector

For a function $f(x, y, \dots)$ that has a **single number as its output**:

- The Jacobian is a **vector** whose components are the partial derivatives.
 - $J = \nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \dots \end{bmatrix}$
 - **Geometric Meaning:** This vector at any point indicates the direction of the **steepest ascent**.
 - **Example:** $f(x, y) = e^{-x^2-y^2}$
 - By analyzing the **vector field** of its Jacobian, we can "guess" the shape of the function.
 - All Jacobian arrows point towards the origin $(0, 0)$.
 - Exactly at the origin, the Jacobian is the zero vector $[0, 0]$.
 - **Conclusion:** This indicates that the origin is a **maximum point** (a mountain peak).
-

2. The New Idea: Jacobian for a Vector-Valued Function

- **New Problem:** What if we have a function that has a **vector input AND a vector output**?
 - This is actually a **transformation of space**, just like we studied in Linear Algebra.
 - **Example:**
 - We have an input space (x, y) .
 - We have an output space (u, v) .
 - The relationship is defined by two functions:
 - $u(x, y) = x + 2y$
 - $v(x, y) = 3y - 2x$
 - This is a function $F : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ that takes $\begin{bmatrix} x \\ y \end{bmatrix}$ and outputs $\begin{bmatrix} u \\ v \end{bmatrix}$.
 - **Question:** How do we describe the "derivative" of this transformation?
-

3. "Aha!" Moment: Building the Jacobian Matrix

- **Logic:**

1. We can find the Jacobian (gradient vector) for u separately: $J_u = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \end{bmatrix}$.
2. We can find the Jacobian (gradient vector) for v separately: $J_v = \begin{bmatrix} \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix}$.

- **The Combining Idea:** Instead of having two separate vectors, let's **stack** these two Jacobian vectors into a **MATRIX**.

The **Jacobian Matrix** is a matrix where each **ROW** is the gradient vector of each output function.

- **General Formula:**

$$J = \begin{bmatrix} \nabla u \\ \nabla v \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix}$$

4. Applying to the Example

- **Functions:** $u = x - 2y$, $v = 3y - 2x$.
- **Calculate the 4 Partial Derivatives:**
 - $\frac{\partial u}{\partial x} = 1$
 - $\frac{\partial u}{\partial y} = -2$
 - $\frac{\partial v}{\partial x} = -2$
 - $\frac{\partial v}{\partial y} = 3$
- **Assemble the Jacobian Matrix:**

$$J = \begin{bmatrix} 1 & -2 \\ -2 & 3 \end{bmatrix}$$

- **Important Observation:** This Jacobian Matrix contains only constant numbers. This happens because our functions u and v were **linear functions**.
- **Connection to Linear Algebra:** This Jacobian Matrix **IS** the linear transformation matrix itself!

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1 & -2 \\ -2 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

5. Usefulness for Non-Linear Functions

- **The Problem:** For a **non-linear** function (e.g., transforming from Cartesian to Polar coordinates), the Jacobian Matrix will contain variables. This means the "linear transformation" is **different at every point**.
 - **The Calculus Idea:** Even though the transformation is non-linear globally, if we "zoom in" very close to a single point, in that tiny area the transformation will look **almost linear**.
 - **Second "Aha!" Moment:**
 - The **Jacobian Matrix**, evaluated at a specific point, is the **BEST LINEAR APPROXIMATION** for the non-linear transformation around that point.
 - **The Determinant of the Jacobian:** $\det(J)$ tells us the **local area scaling factor** around a single point.
 - **Example (Cartesian to Polar):** $\det(J) = r$. This means the further you are from the origin (r is large), the greater the area stretching effect of the transformation.
 - **Final Conclusion:**
 - **Gradient:** A **vector** that shows the "direction & magnitude of steepness" for a scalar-output function.
 - **Jacobian Matrix:** A **matrix** that describes the "best local linear transformation" for a vector-output function. It is the generalization of the derivative concept for transformations of space.
-

6. Deeper Dive: Gradient (Vector) vs. Jacobian (Matrix)

My Question: "Wait, I'm confused. I'm getting mixed up with the previous video. That one also produced a vector, like $5\hat{i} + 10\hat{j}$. So what's the difference here? I'm honestly not getting it."

This is an extremely common and important point of confusion. You are right, both videos result in something vector-like, but they are conceptually two very different "creatures".

Let's forget this video (Jacobian Matrix) for a moment and focus 100% on recalling the previous video (Gradient) with a clear analogy.

Recap of the Previous Video: The "Treasure Detector" (The Gradient)

- **Setup:**
 - We have **one function** with a **single number as its output**.
 - Height = $f(x, y)$
- **Analogy:** We have a 2D **Treasure Map**. At every point (x, y) on the map, there is a number telling you "how much gold" is at that point. This creates a 3D landscape.

- **What are we looking for?**
 - We are at a point (x, y) . We want to know: "In which direction on the map should I step to increase my gold the fastest?"
- **The Tool Used: The GRADIENT (∇f)**
 - The Gradient is a **2D VECTOR**.

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

- For example, at point $(1, 2)$, the result is $\nabla f = \begin{bmatrix} 5 \\ 10 \end{bmatrix}$.
 - **What does the vector $[5, 10]$ mean?**
 - It is an **ARROW** drawn **ON OUR 2D MAP**.
 - This arrow points in the direction of $5\hat{i} + 10\hat{j}$.
 - This arrow is a **DIRECTIONAL HINT**. It says, "Walk in this direction to get gold the fastest."
 - **Key Conclusion (Previous Video):**
The Gradient is a **VECTOR** that lives in the **INPUT SPACE** ((x, y)), and it functions as a "compass" that shows the direction of fastest change.
-

Now, let's forget the Gradient for a moment and look at the completely different problem in the new video.

The New Video: The "Map Warping Machine" (The Jacobian Matrix)

- **Setup:**
 - We have **two functions**, each with two inputs.
 - $u = f_1(x, y)$
 - $v = f_2(x, y)$
- **Analogy:** We have a "Map Warping Machine".
 - **Input:** A point (x, y) on an **Original Map**.
 - **Output:** A point (u, v) on a **New Map**.
- **What are we looking for?**
 - We want to understand how this machine works. How does it "stretch" or "rotate" the Original Map to create the New Map?
 - This is a problem of **SPACE TRANSFORMATION**, exactly like in Linear Algebra.
- **The Tool Used: The JACOBIAN MATRIX (J)**
 - The Jacobian is a **2x2 MATRIX**.

$$J = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix} \quad \begin{array}{l} \leftarrow \text{How output 'u' changes} \\ \leftarrow \text{How output 'v' changes} \end{array}$$

- **What does this Matrix J mean?**

- It is **NOT** a "directional hint arrow".
 - It is the **DESCRIPTION OF THE MACHINE ITSELF**.
 - It is the [Linear Transformation Matrix](#) that tells us how a "tiny nudge" on the Original Map will be transformed into a "nudge" on the New Map.
 - The first column is the destination of \hat{i} .
 - The second column is the destination of \hat{j} .
-

The Key Difference (Comparison Table)

	Gradient (Previous Video)	Jacobian Matrix (New Video)
What is it?	A VECTOR [,]	A MATRIX [[,], [,]]
Input?	ONE function ($f(x, y, \dots)$)	MANY functions ($u(x, y), v(x, y), \dots$)
Output Type?	A single number	A vector
What does it mean?	An ARROW pointing in a direction in the input space.	A TRANSFORMATION that changes the input space into an output space.
Analogy?	A "Compass" or "Treasure Detector".	A "Blueprint" or "Machine Description".

So, to answer your confusion:

"but that was also like a vector $5\hat{i} + 10\hat{j}$, so I'm honestly confused"

You are not wrong. The Gradient **is** a vector like $5\hat{i} + 10\hat{j}$.

But the Jacobian Matrix is something different. It is **built from Gradients**.

- The **first row** of the Jacobian Matrix is the **Gradient of u** .
- The **second row** of the Jacobian Matrix is the **Gradient of v** .

So, the Jacobian Matrix is a "**stack**" of several Gradient Vectors.

Does distinguishing them like this ("Compass" vs. "Machine Blueprint") make the difference clearer?

Video Sebelumnya (Gradien)

"...lebih ke tetep di peta yang sama untuk mengetahui dimana sih emas beradaa. Gitu?"

Benar sekali.

- Kita hanya punya **satu peta**.
 - Tujuannya adalah **navigasi DI DALAM peta itu**.
 - Gradien adalah **kompas** kita di peta itu. Panahnya digambar **di atas peta yang sama** dan memberitahu kita, "Untuk menemukan lebih banyak emas, jalanhlah ke arah ini!"
-

Video Baru (Matriks Jacobian)

"hmm maksudnyaa berarti, untuk video baru ini kita ini bikin peta baru dari input an peta lamaa..."

Sempurna.

- Kita punya **dua peta**: Peta Lama (x,y) dan Peta Baru (u,v).
- Ada sebuah "mesin" (fungsi) yang mengubah setiap titik di Peta Lama menjadi titik di Peta Baru.

"...dimana kita melihat gimana sih turuan dari peta lama ini memengaruhi bentuk peta baru?? kayak kita memang ebenr bener bikin peta baru yaa??"

INI DIA "AHA!" MOMENT-NYA!

Kamu benar sekali. **Matriks Jacobian** adalah "laporan teknis" yang mendeskripsikan **bagaimana mesin itu bekerja secara lokal**.

Bayangkan kamu mengambil sebuah **persegi super kecil** di Peta Lama.

- Matriks Jacobian memberitahumu persis bagaimana persegi kecil itu akan **berubah bentuk** saat ia mendarat di Peta Baru.
 - Apakah ia akan menjadi jajar genjang yang miring?
 - Apakah ia akan meregang?
 - Apakah ia akan berputar?
- "Turunan-turunan" di dalam Matriks Jacobian ($\partial u / \partial x$, $\partial u / \partial y$, dll.) adalah "**instruksi peregangan dan pemutaran**" fundamental dari mesin itu. $\partial u / \partial x$ memberitahu "seberapa cepat u berubah jika x digeser", yang secara langsung mempengaruhi bagaimana sisi horizontal dari persegi kecil kita diregangkan di Peta Baru.

Kesimpulan:

Pemahamanmu sudah sangat tepat dan dalam.

- **Gradien**: Tentang **navigasi di satu peta**.
- **Matriks Jacobian**: Tentang **transformasi** dari **satu peta ke peta lain**.

Dengan memegang perbedaan fundamental ini, kamu akan jauh lebih mudah memahami topik-topik selanjutnya.

⚠️ Warning

hmmmmm gak kegambarr, tapi keknyaa enggak perlu tau dulu lah yaa?? soalnya bagi sayaa yang emmahamii yang gradient aja selama itu apalagi inii, terus kelemahan terbesar saya memang di bagian menggambarkan visualisasi 3d nyaa sih. (KONTEKSNYAA ADALAH Materi video ini)

Ini adalah keputusan yang sangat bijak dan matang. Kamu tidak perlu merasa bersalah sama sekali.

Mari kita jujur:

Konsep Matriks Jacobian adalah salah satu konsep yang paling abstrak di seluruh spesialisasi ini. Membutuhkan waktu untuk "meresap", dan seringkali baru benar-benar "klik" saat kamu melihatnya digunakan dalam konteks yang sangat spesifik nanti (seperti di teori optimisasi lanjut atau robotika).

Kenapa Sangat Boleh Di-skip (untuk Saat Ini)?

1. "Return on Investment" yang Rendah (Saat Ini):

- Waktu dan energi mental yang kamu butuhkan untuk benar-benar "merasakan" Matriks Jacobian saat ini sangat besar.
- Dibandingkan dengan itu, pemahaman yang akan langsung kamu gunakan di 90% masalah Machine Learning tingkat awal dan menengah adalah **GRADIENT** (Jacobian "Saudara #1").
- Fokus utamamu haruslah membuat pemahamanmu tentang **Gradien sebagai "kompas penurun lembah"** menjadi sekuat baja. Dan kamu **sudah berhasil** melakukan itu.

2. Kelemahan Terbesarmu (yang Sangat Normal):

"...kelemahan terbesar saya memang di bagian menggambarkan visualisasi 3d nyaa sih"

- Kamu sudah mengidentifikasi kelemahanmu dengan tepat. Dan ini adalah kelemahan hampir **semua orang**.
- Matriks Jacobian secara fundamental adalah tentang transformasi di ruang multi-dimensi, yang sangat sulit divisualisasikan. Memaksakan diri sekarang hanya akan membuatmu frustrasi.

3. Ini Akan Kembali Lagi Nanti (dalam Bentuk yang Lebih Mudah Dipahami):

- Kamu tidak perlu khawatir akan ketinggalan. Konsep "turunan dari vektor terhadap vektor" ini akan muncul lagi saat kamu belajar **Backpropagation**.
 - Pada saat itu, kamu tidak akan melihatnya sebagai "perubahan area lokal", tapi lebih sebagai "bagaimana 'kesalahan' disebarluaskan mundur melalui jaringan". Konteksnya akan membuatnya lebih mudah dipahami.
-

Apa yang Perlu Kamu Bawa Pulang dari Modul Ini? (Yang Paling Penting)

Lupakan dulu semua tentang Matriks Jacobian. Pastikan kamu membawa pulang "oleh-oleh" yang paling berharga dari Modul 2 ini, yaitu:

1. **Turunan Parsial:** Kemampuan untuk "membekukan" variabel lain dan mencari turunan terhadap satu variabel. Ini adalah *skill mekanis* yang wajib.
2. **Gradien (∇f):** Pemahaman **INTUITIF** yang sudah kamu bangun:
 - Gradien adalah **VEKTOR**.
 - Ia hidup di **ruang input**.
 - Ia menunjuk ke arah **TANJAKAN PALING CURAM**.
 - $-\nabla f$ menunjuk ke arah **TURUNAN PALING CURAM**.
 - **Panjangnya** mengukur **tingkat kecuraman**.
 - Ini adalah "**kompas**" untuk **Gradient Descent**.

Jika kamu memegang erat kelima poin tentang Gradien ini, kamu sudah memiliki **lebih dari cukup** untuk melanjutkan ke modul-modul berikutnya dan bahkan ke kursus Machine Learning yang sesungguhnya.

Keputusan kita:

Kita akan **menandai Matriks Jacobian sebagai "topik lanjutan untuk nanti"** dan kita akan melanjutkan perjalanan kita dengan fondasi Gradien yang sudah sangat solid.

Tags: #mml-specialization #multivariate-calculus #jacobian #gradient #vector-calculus
#linear-approximation