# Optimization and Control in Shell Models of Turbulence

## Forcing optimization of the GOY model

Pedro Marçal[1]

[1]Departmento de Física
Faculdade de Ciências da Universidade Porto

(Tutor: Prof. Sílvio Gama, Departamento de Matemática, FCUP)

Novos Talentos da Matemática, Fundação Calouste Gulbenkian, 2020

# Content

# Outline

# Fluid Mechanics

- Conservation of Mass
- Conservation of Momentum
- Energy equation

# Turbulence and Laminar Flow

- What is Turbulence and Laminar Flow
- Reynolds number
  - $Re = \frac{\rho u L}{\mu}$
- Main characteristics

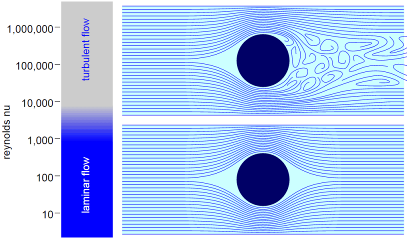



Figure:

# Examples of turbulent and fluid dynamics

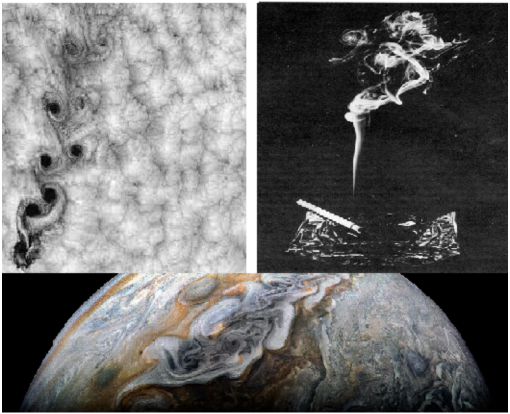


Figure: Atmosferic flow over Selkirr Island; Laminar to Turbulent flow in cigarette smoke; Jupiter turbulent gases

# Outline

# Navier-Stokes equations

## History and context

- Contribuitions made by Claude Navier & George Stokes
- Use and examples of Navier-Stokes equations
- Complexity and possible solutions

# The equations

- Continuity Equation
- Momentum Equations

**Three Dimensional components:**

$$\nabla \cdot \vec{V} = 0$$

$$\rho \frac{D\vec{V}}{Dt} = -\nabla p + \rho \vec{g} + \mu \nabla^2 \vec{V}$$

$+$ boundary conditions $+$ inicial conditions

# Shell Models & Energy

- Structure
- Reasoning behind Shell Models

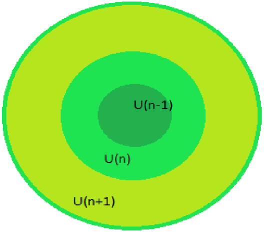



Figure:

# Shell Models & NSE

- Kolmogorov 1941 energy spectrum
- Invariants preserved to mimic the Navier-Stokes equations
- Energy cascade spectrum & energy transfer in shell models

# Outline

# Turbulence in Shell Models & NVE

- Fluid Mechanics and Turbulence basics
- Energity Density and Spectral energy Flux
- The four-fifth law
- Parameter Space for the GOY model
- 2D and 3D models

# Kolmogorov Theory

## Velocity in the K41

The state of the flow is characterized by the mean energy dissipation per unit mass: $\bar{\varepsilon}$

Velocity difference: $\delta u(l) \equiv | u(r+l) - u(r) |$, eddy of size $l \ll L$;

$\delta u(l) = f(\tilde{l}, \bar{\varepsilon})$ by dimensional analysis:

$[\delta u] = ms^{-1}$; $[l] = m$; $[\bar{\varepsilon}] = m^2 s^{-3}$

$[\delta u] = [l]^{\alpha}[\bar{\varepsilon}]^{\beta} \implies \alpha = \beta = 1/3$

Resulting in: $\lambda \delta u(l) = f[\lambda(\bar{\varepsilon} l)^{1/3}]$

# Kolmogorov Theory

## Scale and Dissipation

With the previous result and the NSE we can estimate the rate of change of the energy per unit volume due to dissipation at scale $\eta$, when it becomes important in contrast to L.

$\bar{\varepsilon} \sim \nu u_i \partial_{jj} u_i \sim \nu \delta u(\eta)^2/\eta^2$

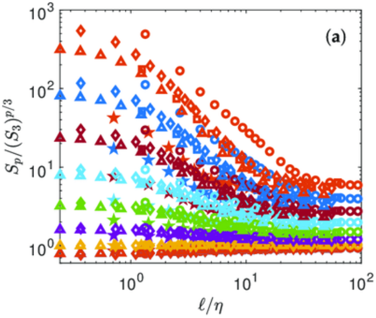
So the Kolmogorov scale:

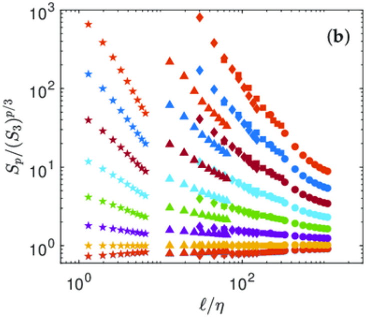
$\eta \sim (\bar{\varepsilon}/\eta^3)^{-1/4}$ depends on the Reynolds numbers as $\eta \sim Re^{-3/4}$

# Numeric simulation

The energy cascade can be visualized by setting diferent base parameters.

Figure:



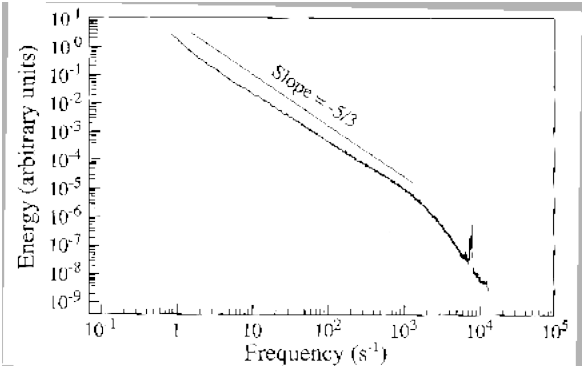
# Spectral energy density

- The mean of the square of the velocity difference is called second order structure function:
- $S_2(l) \equiv \prec \delta u(l)^2 \succ \sim (\bar{\varepsilon} l)^{2/3}$ is related to the spectral energy density through a Fourier transform.
- We can also express the energy density of the flux:
  - $E = \frac{1}{2} \int u(x)^2 dx = \frac{1}{2}(2\pi)^3 \int_0^\infty u_i(k) u_i(k)^* dk = \frac{1}{2}(2\pi)^3 4\pi \int_0^\infty k^2 \mid u(k) \mid^2 dk \equiv \int_0^\infty E(k) dk$
- Using the Parseval's identity in fourier transforms of spectral velocities:
  - $\frac{1}{(2\pi)^3} \int u_i(x) u_i(x) dx = \int u_i(-k') u_i(k') dk' = \int u_i(k') u_i(k')^* dk'$ with $u_i(x) \in \mathbb{R} \implies u_i(-k') = u_i(k')^*$

# Spectral energy density

- Integration was performed over the sphere and absorbing the spacial variable into dx, results in:
  - $E(k) = (2\pi)^4 k^2 \mid u(k) \mid^2$and $S_2(l) = \int [u(l+x) - u(x)]^2 dx$
  - Combining both e quations and after several complex integration processes we obtain:
  - $E(k) = \frac{1}{2\pi k} \int_0^\infty x sin(x) S_2(x/k) dx \sim \bar{\varepsilon}^{-2/3} k^{-5/3}$

# Spectral energy density

Figure: 

# The four-fifth law

- A special case occurs for a third order struture function, associated with energy conservation. Again considering a homogeneus and isotropic turbulent fluid left alone after intense stirring. As the kinetic change is 0, the mean energy dissipation is:
  - $\bar{\varepsilon} = -\frac{1}{2}\partial_t [u]$

## Scalling relation

After several integrations and calculation regarding isotropy and the NSE, remains a simple equation regarding the third order structure function:

$S_3 = -\frac{4}{5}\bar{\varepsilon} l$

This law is a cornerstone of fluid dynamics. Not only is it in agreement with the phenomenology described in the K41 as it is exact and among few exact results derived from NSE.

# Outline

# GOY Model

- Gledzer & Obukhov

## Model and given inicial conditions

- $\mathcal{L}u = f_n$
- $\mathcal{L}u = \frac{du_n}{dt} - ik_n \left( u_{n+1}u_{n+2} - \frac{\varepsilon}{q} u_{n+1}u_{n-1} + \frac{\varepsilon - 1}{q} u_{n-2}u_{n-1} \right)^* + \nu k_n^2 u_n$

- Constants used
- Force and solution for the GOY model

# Outline

# Parameter Space for the GOY model

- Number of Shells
- Kinematic viscosity
- Initial force
- Free Parameters

## Values used

$N = 24; \nu = 10^{-7}; f_n = (1 + i) \times 5 \times 10^{-3}$

# 2D and 3D models

- 2D/3D selector
- Complex non linear coeficients

# Outline

# Constant manipulation and restraints

- Constants changed

## GOY model constant substituiton

- $a_n = k_n = k_0 q^n$
- $b_n = -\frac{1}{2} k_{n-1}$
- $c_n = -\frac{1}{2} k_{n-2}$

- All components with shell index, $n \leqslant 0$ are taken as zero
- Resulting in:

$$\left(\frac{d}{dt} + vk_n^2\right) u_n = i\left(a_n u_{n+1} u_{n+2} + b_n u_{n+1} u_{n-1} + c_n u_{n-1} u_{n-2}\right)^* + f_n$$

# Separation

- Splitting complex variables $u_n = x_n + iy_n$:

## Real Part

$$\left(\frac{d}{dt} + vk_n^2\right) x_n = a_n \left(x_{n+1} y_{n+2} + x_{n+2} y_{n+1}\right) + b_n \left(x_{n-1} y_{n+1} + x_{n+1} y_{n-1}\right) + c_n \left(x_{n-2} y_{n-1} + x_{n-1} y_{n-2}\right) + f_x$$

## Imaginary Part

$$\left(\frac{d}{dt} + vk_n^2\right) y_n = ia_n \left(x_{n+1} x_{n+2} - y_{n+2} y_{n+1}\right) + ib_n \left(x_{n-1} x_{n+1} - y_{n+1} y_{n-1}\right) + ic_n \left(x_{n-2} x_{n-1} - y_{n-1} y_{n-2}\right) + if_y$$

# Verification

- Jupyter Notebook, Python, algorithm used to verify the separated variables equations.

## Algorithm

- Load a complex variable with pseudo-random numbers ($u = x + iy$);
- Write the complex GOY model in simple for loops, all the N=24 shells;
- Write the separated GOY equations as for loops;
- Write a function that retrieves the diference between the 3 functions;
- Return the biggest difference(0 meaning these equations were well determined).

# Outline

# Process

Linearized the system of real and imaginary equations around set solutions.

## Simple example

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} \tilde{x}_n \\ \tilde{y}_n \end{bmatrix} + \varepsilon \begin{bmatrix} x_n \\ y_n \end{bmatrix}$$

The term of order $\varepsilon^2$will not be considered ($\varepsilon \ll 1$)

# Linear Scheme

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \\ y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = -\nu \begin{bmatrix} k_1^2 x_1 \\ k_2^2 x_2 \\ k_3^2 x_3 \\ \vdots \\ k_n^2 x_n \\ k_1^2 y_1 \\ k_2^2 y_2 \\ k_3^2 y_3 \\ \vdots \\ k_n^2 y_n \end{bmatrix} + \begin{bmatrix} C_1 & C_2 \\ C_3 & C_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \\ y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} + ,$$

where the block matrices $C_1$ , $C_2$ , $C_3$ , $C_4$ ,are given by:

# Coeficient Part Matrix

$$C1 = \begin{bmatrix} 0 & a_1\tilde{y}_3 & a_1\tilde{y}_2 & & & & & \\ b_2\tilde{y}_3 & 0 & a_2\tilde{y}_4 + b_2\tilde{y}_1 & a_2\tilde{y}_3 & & & & \\ c_3\tilde{y}_2 & b_3\tilde{y}_4 + c_3\tilde{y}_1 & 0 & a_3\tilde{y}_5 + b_3\tilde{y}_2 & a_3\tilde{y}_4 & & & \\ & c_3\tilde{y}_2 & b_3\tilde{y}_4 + c_3\tilde{y}_1 & 0 & a_3\tilde{y}_5 + b_3\tilde{y}_2 & a_3\tilde{y}_4 & & \\ & & c_3\tilde{y}_2 & b_3\tilde{y}_4 + c_3\tilde{y}_1 & 0 & a_3\tilde{y}_5 + b_3\tilde{y}_2 & a_3\tilde{y}_4 & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & & & & c_n\tilde{y}_n & c_n\tilde{y}_{n-2} & 0 \end{bmatrix}$$

# Outline

# Adjoint operator

- Cost functional in the separated variables equations
- Determining the adjoint operator and transpose matrix

# Outline

# Classic fourth order RK

## Generic expression

$u^{n+1} = u^n + \Delta t \sum b_i k_i$

## Inicial Value Problem

$y' = f(t, y); y(t_0) = y_0$
$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$
$t_{n+1} = t_n + h$

# Classic fourth order RK & GOY model

## RK4

$$k_1 = f(t_n, y_n)$$
$$k_2 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1)$$
$$k_3 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2)$$
$$k_4 = f(t_n + h, y_n + hk_3)$$
$$h = 10^{-4}; n \approx 2.5 \times 10^8$$

# Computing the problem and force optimization

## Optimization Process

- Choosing an initial guess $f^{(0)}$; $n = 0$
- Solve the GOY model equations, with $f = f^{(0)}$
- Solve the adjoint operator from matrix C1...
- Obtain the cost functional gradient
- Find a parameter...
- Update the variable and repeat.

# Finding the particular solution

## Process

$$\left(\frac{d}{dt} + \nu k_n^2\right) u_n = i\left(a_n u_{n+1} u_{n+2} + b_n u_{n+1} u_{n-1} + c_n u_{n-1} u_{n-2}\right)^* + f_n$$

- A solution is calculated for every time step.
- A solution consists of 24 pairs of real and imaginary numbers.

# Finding other solutions and changing the inicial force.

Plugging the solutions $x_n y_n$and using the Runge Kutta method for integration, we find solutions for the values $\widetilde{x_n}$ & $\widetilde{y_n}$.
Python algorithm was used to find these solutions.

## Algorithm

Using the GOY separated equations $x_n; y_n$are found for a certain $\delta t$. Using RK4

Using the linearized equations, and matrices C1, C2, C3, C4 for each time step:

$\widetilde{x_n}$ and $\widetilde{y_n}$ are calculated

Repeat the process

Expected iterations around $2 \times 10^8$

# Conclusions

The force optimization process and the verification of convergence for the solutions was not fully completed, mostly due to lack of time, but also to some limitations in Python.
It would have been better to do the long computing algorithms in a simpler language such as Fortran. The objective of the long numeric runs was to find the optimal force or aproaching it, optimal in a way that would recreate and verify the energy cascade found in the GOY model and in the K41.

# Bibliography and References

Peter D.Ditlevsen.
*Turbulence and Shell Models*.
Cambridge University Press, 2011.

Nagwa Arafa; SÃlvio Gama; Fernando Lobo Pereira.
Optimization and control theory in shell models of turbulence.
*PHYSCON*, 2017.

D. Pisarenko, L. Biferale, D. Courvoisier, U. Frisch, and M. Vergassola.
Further results on multifractality in shell models.
*Physics of Fluids A: Fluid Dynamics*, 5(10):2533–2538, oct 1993.