

DATA
TERRA
•••



Offres de services du pôle (Infrastructures HPC & VRE)

Gwenaël CAËR, Cyril GERMINEAUD

Les infrastructures HPC

Le pôle ODATIS mets à disposition l'accès aux infrastructures de calcul du CNES (*TREX*) et de l'Ifremer (*DATARMOR*).



Cluster du CNES (TREX)



Cluster de l'Ifremer (DATARMOR)

Les infrastructures HPC

Comment y accéder ? Une simple commande SSH, et vous voilà connecté au cluster.

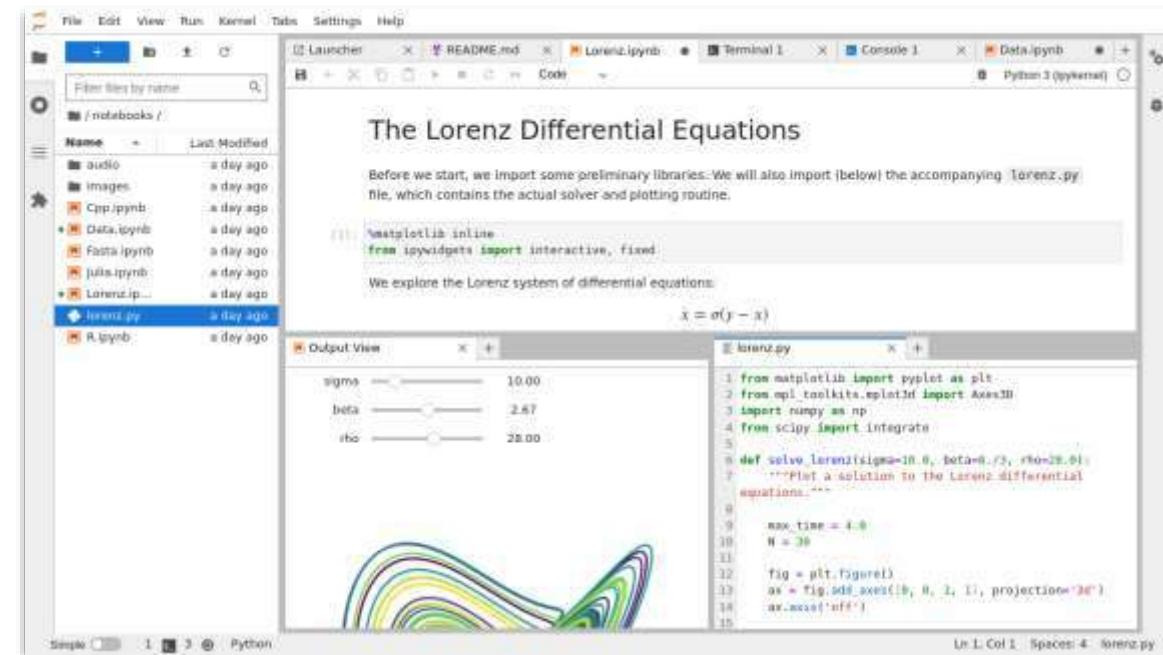
Les infrastructures HPC

Comment y accéder ? Une simple commande SSH, et vous voilà connecté au cluster.



VRE (Environnement Virtuel de Recherche)

Un VRE pour Environnement Virtuel de Recherche, correspond à une plateforme interactive, permettant de faciliter l'accès aux données et aux ressources des infrastructures (*calculs, logiciels, documentations* ...).



VRE (Concrètement, comment ça fonctionne ?)

Travailler à distance ... comme si vous étiez en local.

1 Accéder à l'URL
(et se connecter)

<https://jupyterhub.cnes.fr> (ou ifremer.fr)

connexion

2 Sélectionner les
ressources (CPU,
RAM, Durée, ...)

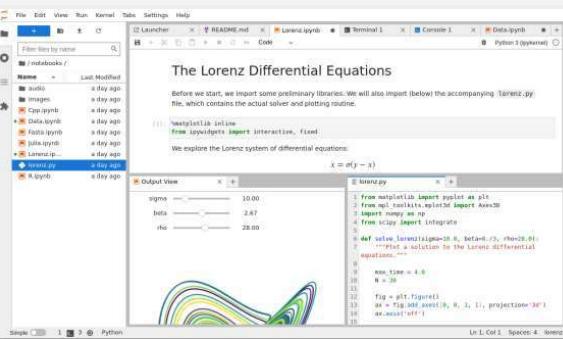
Sélection des ressources

1 cœur, 4gb, 8 heures

8 coeurs, 32gb, 8 heures

28 coeurs, 115gb, 8 heures

3 Commencer à
l'utiliser la VRE
(Données, Analyse,
Visualisation, ...)



```
The Lorenz Differential Equations

Before we start, we import some preliminary libraries. We will also import (below) the accompanying lorenz3.py file, which contains the actual solver and plotting routine.

In [1]: #analytical inline
In [2]: from ipyvolume import interactive, Fixed
In [3]: We explore the Lorenz system of differential equations:
In [4]: x = (y - z)
In [5]: sigma = 10.00
In [6]: beta = 2.67
In [7]: rho = 28.00
In [8]: fig = plt.figure()
In [9]: ax = fig.add_subplot(111, projection='3d')
In [10]: ax.set_axis('off')
```

VRE (Concrètement, comment ça fonctionne ?)

Travailler à distance ... comme si vous étiez en local.

1 Accéder à l'URL (et se connecter)

<https://jupyterhub.cnes.fr> (ou ifremer.fr)

connexion

2 Sélectionner les ressources (CPU, RAM, Durée, ...)

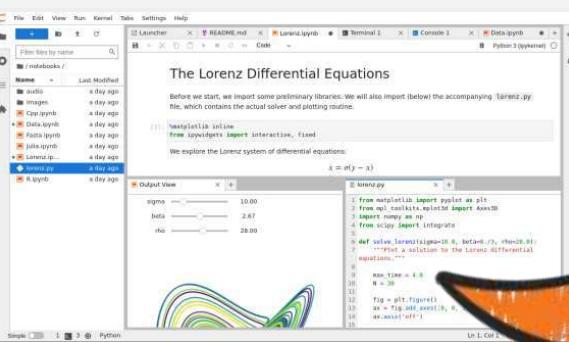
Sélection des ressources

1 cœur, 4gb, 8 heures

8 coeurs, 32gb, 8 heures

28 coeurs, 115gb, 8 heures

3 Commencer à l'utiliser la VRE (Données, Analyse, Visualisation, ...)



```
The Lorenz Differential Equations
Before we start, we import some preliminary libraries. We will also import (below) the accompanying lorenz3.py file, which contains the actual solver and plotting routine.

In [1]: analytical = False
In [2]: from lorenz import *
In [3]: import numpy as np
In [4]: from scipy import integrate
In [5]: def solve_lorenz(tspan=(0, 10), beta=10, rho=28.0)
In [6]:     """Solve the Lorenz differential equations.
In [7]:     """
In [8]:     t0, t1 = tspan
In [9]:     N = 1000
In [10]:     dt = (t1 - t0) / N
In [11]:     t = np.arange(t0, t1, dt)
In [12]:     x0 = np.array([10, 10, 10])
In [13]:     sol = integrate.odeint(lorenz3, x0, t)
In [14]:     return sol
In [15]: l = solve_lorenz()
In [16]: l
Out[16]: array([[[ 10.0, 10.0, 10.0],
In [17]:        [ 10.0, 10.0, 10.0],
In [18]:        [ 10.0, 10.0, 10.0],
In [19]:        [ 10.0, 10.0, 10.0],
In [20]:        [ 10.0, 10.0, 10.0],
In [21]:        [ 10.0, 10.0, 10.0],
In [22]:        [ 10.0, 10.0, 10.0],
In [23]:        [ 10.0, 10.0, 10.0],
In [24]:        [ 10.0, 10.0, 10.0],
In [25]:        [ 10.0, 10.0, 10.0],
In [26]:        [ 10.0, 10.0, 10.0],
In [27]:        [ 10.0, 10.0, 10.0],
In [28]:        [ 10.0, 10.0, 10.0],
In [29]:        [ 10.0, 10.0, 10.0],
In [30]:        [ 10.0, 10.0, 10.0],
In [31]:        [ 10.0, 10.0, 10.0],
In [32]:        [ 10.0, 10.0, 10.0],
In [33]:        [ 10.0, 10.0, 10.0],
In [34]:        [ 10.0, 10.0, 10.0],
In [35]:        [ 10.0, 10.0, 10.0],
In [36]:        [ 10.0, 10.0, 10.0],
In [37]:        [ 10.0, 10.0, 10.0],
In [38]:        [ 10.0, 10.0, 10.0],
In [39]:        [ 10.0, 10.0, 10.0],
In [40]:        [ 10.0, 10.0, 10.0],
In [41]:        [ 10.0, 10.0, 10.0],
In [42]:        [ 10.0, 10.0, 10.0],
In [43]:        [ 10.0, 10.0, 10.0],
In [44]:        [ 10.0, 10.0, 10.0],
In [45]:        [ 10.0, 10.0, 10.0],
In [46]:        [ 10.0, 10.0, 10.0],
In [47]:        [ 10.0, 10.0, 10.0],
In [48]:        [ 10.0, 10.0, 10.0],
In [49]:        [ 10.0, 10.0, 10.0],
In [50]:        [ 10.0, 10.0, 10.0],
In [51]:        [ 10.0, 10.0, 10.0],
In [52]:        [ 10.0, 10.0, 10.0],
In [53]:        [ 10.0, 10.0, 10.0],
In [54]:        [ 10.0, 10.0, 10.0],
In [55]:        [ 10.0, 10.0, 10.0],
In [56]:        [ 10.0, 10.0, 10.0],
In [57]:        [ 10.0, 10.0, 10.0],
In [58]:        [ 10.0, 10.0, 10.0],
In [59]:        [ 10.0, 10.0, 10.0],
In [60]:        [ 10.0, 10.0, 10.0],
In [61]:        [ 10.0, 10.0, 10.0],
In [62]:        [ 10.0, 10.0, 10.0],
In [63]:        [ 10.0, 10.0, 10.0],
In [64]:        [ 10.0, 10.0, 10.0],
In [65]:        [ 10.0, 10.0, 10.0],
In [66]:        [ 10.0, 10.0, 10.0],
In [67]:        [ 10.0, 10.0, 10.0],
In [68]:        [ 10.0, 10.0, 10.0],
In [69]:        [ 10.0, 10.0, 10.0],
In [70]:        [ 10.0, 10.0, 10.0],
In [71]:        [ 10.0, 10.0, 10.0],
In [72]:        [ 10.0, 10.0, 10.0],
In [73]:        [ 10.0, 10.0, 10.0],
In [74]:        [ 10.0, 10.0, 10.0],
In [75]:        [ 10.0, 10.0, 10.0],
In [76]:        [ 10.0, 10.0, 10.0],
In [77]:        [ 10.0, 10.0, 10.0],
In [78]:        [ 10.0, 10.0, 10.0],
In [79]:        [ 10.0, 10.0, 10.0],
In [80]:        [ 10.0, 10.0, 10.0],
In [81]:        [ 10.0, 10.0, 10.0],
In [82]:        [ 10.0, 10.0, 10.0],
In [83]:        [ 10.0, 10.0, 10.0],
In [84]:        [ 10.0, 10.0, 10.0],
In [85]:        [ 10.0, 10.0, 10.0],
In [86]:        [ 10.0, 10.0, 10.0],
In [87]:        [ 10.0, 10.0, 10.0],
In [88]:        [ 10.0, 10.0, 10.0],
In [89]:        [ 10.0, 10.0, 10.0],
In [90]:        [ 10.0, 10.0, 10.0],
In [91]:        [ 10.0, 10.0, 10.0],
In [92]:        [ 10.0, 10.0, 10.0],
In [93]:        [ 10.0, 10.0, 10.0],
In [94]:        [ 10.0, 10.0, 10.0],
In [95]:        [ 10.0, 10.0, 10.0],
In [96]:        [ 10.0, 10.0, 10.0],
In [97]:        [ 10.0, 10.0, 10.0],
In [98]:        [ 10.0, 10.0, 10.0],
In [99]:        [ 10.0, 10.0, 10.0],
In [100]:        [ 10.0, 10.0, 10.0],
In [101]:        [ 10.0, 10.0, 10.0],
In [102]:        [ 10.0, 10.0, 10.0],
In [103]:        [ 10.0, 10.0, 10.0],
In [104]:        [ 10.0, 10.0, 10.0],
In [105]:        [ 10.0, 10.0, 10.0],
In [106]:        [ 10.0, 10.0, 10.0],
In [107]:        [ 10.0, 10.0, 10.0],
In [108]:        [ 10.0, 10.0, 10.0],
In [109]:        [ 10.0, 10.0, 10.0],
In [110]:        [ 10.0, 10.0, 10.0],
In [111]:        [ 10.0, 10.0, 10.0],
In [112]:        [ 10.0, 10.0, 10.0],
In [113]:        [ 10.0, 10.0, 10.0],
In [114]:        [ 10.0, 10.0, 10.0],
In [115]:        [ 10.0, 10.0, 10.0],
In [116]:        [ 10.0, 10.0, 10.0],
In [117]:        [ 10.0, 10.0, 10.0],
In [118]:        [ 10.0, 10.0, 10.0],
In [119]:        [ 10.0, 10.0, 10.0],
In [120]:        [ 10.0, 10.0, 10.0],
In [121]:        [ 10.0, 10.0, 10.0],
In [122]:        [ 10.0, 10.0, 10.0],
In [123]:        [ 10.0, 10.0, 10.0],
In [124]:        [ 10.0, 10.0, 10.0],
In [125]:        [ 10.0, 10.0, 10.0],
In [126]:        [ 10.0, 10.0, 10.0],
In [127]:        [ 10.0, 10.0, 10.0],
In [128]:        [ 10.0, 10.0, 10.0],
In [129]:        [ 10.0, 10.0, 10.0],
In [130]:        [ 10.0, 10.0, 10.0],
In [131]:        [ 10.0, 10.0, 10.0],
In [132]:        [ 10.0, 10.0, 10.0],
In [133]:        [ 10.0, 10.0, 10.0],
In [134]:        [ 10.0, 10.0, 10.0],
In [135]:        [ 10.0, 10.0, 10.0],
In [136]:        [ 10.0, 10.0, 10.0],
In [137]:        [ 10.0, 10.0, 10.0],
In [138]:        [ 10.0, 10.0, 10.0],
In [139]:        [ 10.0, 10.0, 10.0],
In [140]:        [ 10.0, 10.0, 10.0],
In [141]:        [ 10.0, 10.0, 10.0],
In [142]:        [ 10.0, 10.0, 10.0],
In [143]:        [ 10.0, 10.0, 10.0],
In [144]:        [ 10.0, 10.0, 10.0],
In [145]:        [ 10.0, 10.0, 10.0],
In [146]:        [ 10.0, 10.0, 10.0],
In [147]:        [ 10.0, 10.0, 10.0],
In [148]:        [ 10.0, 10.0, 10.0],
In [149]:        [ 10.0, 10.0, 10.0],
In [150]:        [ 10.0, 10.0, 10.0],
In [151]:        [ 10.0, 10.0, 10.0],
In [152]:        [ 10.0, 10.0, 10.0],
In [153]:        [ 10.0, 10.0, 10.0],
In [154]:        [ 10.0, 10.0, 10.0],
In [155]:        [ 10.0, 10.0, 10.0],
In [156]:        [ 10.0, 10.0, 10.0],
In [157]:        [ 10.0, 10.0, 10.0],
In [158]:        [ 10.0, 10.0, 10.0],
In [159]:        [ 10.0, 10.0, 10.0],
In [160]:        [ 10.0, 10.0, 10.0],
In [161]:        [ 10.0, 10.0, 10.0],
In [162]:        [ 10.0, 10.0, 10.0],
In [163]:        [ 10.0, 10.0, 10.0],
In [164]:        [ 10.0, 10.0, 10.0],
In [165]:        [ 10.0, 10.0, 10.0],
In [166]:        [ 10.0, 10.0, 10.0],
In [167]:        [ 10.0, 10.0, 10.0],
In [168]:        [ 10.0, 10.0, 10.0],
In [169]:        [ 10.0, 10.0, 10.0],
In [170]:        [ 10.0, 10.0, 10.0],
In [171]:        [ 10.0, 10.0, 10.0],
In [172]:        [ 10.0, 10.0, 10.0],
In [173]:        [ 10.0, 10.0, 10.0],
In [174]:        [ 10.0, 10.0, 10.0],
In [175]:        [ 10.0, 10.0, 10.0],
In [176]:        [ 10.0, 10.0, 10.0],
In [177]:        [ 10.0, 10.0, 10.0],
In [178]:        [ 10.0, 10.0, 10.0],
In [179]:        [ 10.0, 10.0, 10.0],
In [180]:        [ 10.0, 10.0, 10.0],
In [181]:        [ 10.0, 10.0, 10.0],
In [182]:        [ 10.0, 10.0, 10.0],
In [183]:        [ 10.0, 10.0, 10.0],
In [184]:        [ 10.0, 10.0, 10.0],
In [185]:        [ 10.0, 10.0, 10.0],
In [186]:        [ 10.0, 10.0, 10.0],
In [187]:        [ 10.0, 10.0, 10.0],
In [188]:        [ 10.0, 10.0, 10.0],
In [189]:        [ 10.0, 10.0, 10.0],
In [190]:        [ 10.0, 10.0, 10.0],
In [191]:        [ 10.0, 10.0, 10.0],
In [192]:        [ 10.0, 10.0, 10.0],
In [193]:        [ 10.0, 10.0, 10.0],
In [194]:        [ 10.0, 10.0, 10.0],
In [195]:        [ 10.0, 10.0, 10.0],
In [196]:        [ 10.0, 10.0, 10.0],
In [197]:        [ 10.0, 10.0, 10.0],
In [198]:        [ 10.0, 10.0, 10.0],
In [199]:        [ 10.0, 10.0, 10.0],
In [200]:        [ 10.0, 10.0, 10.0],
In [201]:        [ 10.0, 10.0, 10.0],
In [202]:        [ 10.0, 10.0, 10.0],
In [203]:        [ 10.0, 10.0, 10.0],
In [204]:        [ 10.0, 10.0, 10.0],
In [205]:        [ 10.0, 10.0, 10.0],
In [206]:        [ 10.0, 10.0, 10.0],
In [207]:        [ 10.0, 10.0, 10.0],
In [208]:        [ 10.0, 10.0, 10.0],
In [209]:        [ 10.0, 10.0, 10.0],
In [210]:        [ 10.0, 10.0, 10.0],
In [211]:        [ 10.0, 10.0, 10.0],
In [212]:        [ 10.0, 10.0, 10.0],
In [213]:        [ 10.0, 10.0, 10.0],
In [214]:        [ 10.0, 10.0, 10.0],
In [215]:        [ 10.0, 10.0, 10.0],
In [216]:        [ 10.0, 10.0, 10.0],
In [217]:        [ 10.0, 10.0, 10.0],
In [218]:        [ 10.0, 10.0, 10.0],
In [219]:        [ 10.0, 10.0, 10.0],
In [220]:        [ 10.0, 10.0, 10.0],
In [221]:        [ 10.0, 10.0, 10.0],
In [222]:        [ 10.0, 10.0, 10.0],
In [223]:        [ 10.0, 10.0, 10.0],
In [224]:        [ 10.0, 10.0, 10.0],
In [225]:        [ 10.0, 10.0, 10.0],
In [226]:        [ 10.0, 10.0, 10.0],
In [227]:        [ 10.0, 10.0, 10.0],
In [228]:        [ 10.0, 10.0, 10.0],
In [229]:        [ 10.0, 10.0, 10.0],
In [230]:        [ 10.0, 10.0, 10.0],
In [231]:        [ 10.0, 10.0, 10.0],
In [232]:        [ 10.0, 10.0, 10.0],
In [233]:        [ 10.0, 10.0, 10.0],
In [234]:        [ 10.0, 10.0, 10.0],
In [235]:        [ 10.0, 10.0, 10.0],
In [236]:        [ 10.0, 10.0, 10.0],
In [237]:        [ 10.0, 10.0, 10.0],
In [238]:        [ 10.0, 10.0, 10.0],
In [239]:        [ 10.0, 10.0, 10.0],
In [240]:        [ 10.0, 10.0, 10.0],
In [241]:        [ 10.0, 10.0, 10.0],
In [242]:        [ 10.0, 10.0, 10.0],
In [243]:        [ 10.0, 10.0, 10.0],
In [244]:        [ 10.0, 10.0, 10.0],
In [245]:        [ 10.0, 10.0, 10.0],
In [246]:        [ 10.0, 10.0, 10.0],
In [247]:        [ 10.0, 10.0, 10.0],
In [248]:        [ 10.0, 10.0, 10.0],
In [249]:        [ 10.0, 10.0, 10.0],
In [250]:        [ 10.0, 10.0, 10.0],
In [251]:        [ 10.0, 10.0, 10.0],
In [252]:        [ 10.0, 10.0, 10.0],
In [253]:        [ 10.0, 10.0, 10.0],
In [254]:        [ 10.0, 10.0, 10.0],
In [255]:        [ 10.0, 10.0, 10.0],
In [256]:        [ 10.0, 10.0, 10.0],
In [257]:        [ 10.0, 10.0, 10.0],
In [258]:        [ 10.0, 10.0, 10.0],
In [259]:        [ 10.0, 10.0, 10.0],
In [260]:        [ 10.0, 10.0, 10.0],
In [261]:        [ 10.0, 10.0, 10.0],
In [262]:        [ 10.0, 10.0, 10.0],
In [263]:        [ 10.0, 10.0, 10.0],
In [264]:        [ 10.0, 10.0, 10.0],
In [265]:        [ 10.0, 10.0, 10.0],
In [266]:        [ 10.0, 10.0, 10.0],
In [267]:        [ 10.0, 10.0, 10.0],
In [268]:        [ 10.0, 10.0, 10.0],
In [269]:        [ 10.0, 10.0, 10.0],
In [270]:        [ 10.0, 10.0, 10.0],
In [271]:        [ 10.0, 10.0, 10.0],
In [272]:        [ 10.0, 10.0, 10.0],
In [273]:        [ 10.0, 10.0, 10.0],
In [274]:        [ 10.0, 10.0, 10.0],
In [275]:        [ 10.0, 10.0, 10.0],
In [276]:        [ 10.0, 10.0, 10.0],
In [277]:        [ 10.0, 10.0, 10.0],
In [278]:        [ 10.0, 10.0, 10.0],
In [279]:        [ 10.0, 10.0, 10.0],
In [280]:        [ 10.0, 10.0, 10.0],
In [281]:        [ 10.0, 10.0, 10.0],
In [282]:        [ 10.0, 10.0, 10.0],
In [283]:        [ 10.0, 10.0, 10.0],
In [284]:        [ 10.0, 10.0, 10.0],
In [285]:        [ 10.0, 10.0, 10.0],
In [286]:        [ 10.0, 10.0, 10.0],
In [287]:        [ 10.0, 10.0, 10.0],
In [288]:        [ 10.0, 10.0, 10.0],
In [289]:        [ 10.0, 10.0, 10.0],
In [290]:        [ 10.0, 10.0, 10.0],
In [291]:        [ 10.0, 10.0, 10.0],
In [292]:        [ 10.0, 10.0, 10.0],
In [293]:        [ 10.0, 10.0, 10.0],
In [294]:        [ 10.0, 10.0, 10.0],
In [295]:        [ 10.0, 10.0, 10.0],
In [296]:        [ 10.0, 10.0, 10.0],
In [297]:        [ 10.0, 10.0, 10.0],
In [298]:        [ 10.0, 10.0, 10.0],
In [299]:        [ 10.0, 10.0, 10.0],
In [300]:        [ 10.0, 10.0, 10.0],
In [301]:        [ 10.0, 10.0, 10.0],
In [302]:        [ 10.0, 10.0, 10.0],
In [303]:        [ 10.0, 10.0, 10.0],
In [304]:        [ 10.0, 10.0, 10.0],
In [305]:        [ 10.0, 10.0, 10.0],
In [306]:        [ 10.0, 10.0, 10.0],
In [307]:        [ 10.0, 10.0, 10.0],
In [308]:        [ 10.0, 10.0, 10.0],
In [309]:        [ 10.0, 10.0, 10.0],
In [310]:        [ 10.0, 10.0, 10.0],
In [311]:        [ 10.0, 10.0, 10.0],
In [312]:        [ 10.0, 10.0, 10.0],
In [313]:        [ 10.0, 10.0, 10.0],
In [314]:        [ 10.0, 10.0, 10.0],
In [315]:        [ 10.0, 10.0, 10.0],
In [316]:        [ 10.0, 10.0, 10.0],
In [317]:        [ 10.0, 10.0, 10.0],
In [318]:        [ 10.0, 10.0, 10.0],
In [319]:        [ 10.0, 10.0, 10.0],
In [320]:        [ 10.0, 10.0, 10.0],
In [321]:        [ 10.0, 10.0, 10.0],
In [322]:        [ 10.0, 10.0, 10.0],
In [323]:        [ 10.0, 10.0, 10.0],
In [324]:        [ 10.0, 10.0, 10.0],
In [325]:        [ 10.0, 10.0, 10.0],
In [326]:        [ 10.0, 10.0, 10.0],
In [327]:        [ 10.0, 10.0, 10.0],
In [328]:        [ 10.0, 10.0, 10.0],
In [329]:        [ 10.0, 10.0, 10.0],
In [330]:        [ 10.0, 10.0, 10.0],
In [331]:        [ 10.0, 10.0, 10.0],
In [332]:        [ 10.0, 10.0, 10.0],
In [333]:        [ 10.0, 10.0, 10.0],
In [334]:        [ 10.0, 10.0, 10.0],
In [335]:        [ 10.0, 10.0, 10.0],
In [336]:        [ 10.0, 10.0, 10.0],
In [337]:        [ 10.0, 10.0, 10.0],
In [338]: 
```

VRE (Concrètement, comment ça fonctionne ?)

Travailler à distance ... comme si vous étiez en local.

1 Accéder à l'URL (et se connecter)

<https://jupyterhub.cnes.fr> (ou ifremer.fr)

connexion

2 Sélectionner les ressources (CPU, RAM, Durée, ...)

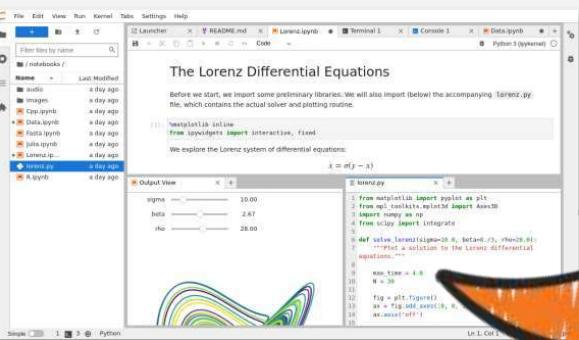
Sélection des ressources

1 cœur, 4gb, 8 heures

8 coeurs, 32gb, 8 heures

28 coeurs, 115gb, 8 heures

3 Commencer à l'utiliser la VRE (Données, Analyse, Visualisation, ...)



```
The Lorenz Differential Equations
Before we start, we import some preliminary libraries. We will also import (below) the accompanying lorenz3.py file, which contains the actual solver and plotting routine.

In [1]: import numpy as np
In [2]: from scipy.integrate import solve_ivp
In [3]: from Lorenz import lorenz3
In [4]: from matplotlib import pyplot as plt
In [5]: from mpl_toolkits.mplot3d import Axes3D
In [6]: import numpy as np
In [7]: from scipy import integrate
In [8]: def solve_lorenz(tspan=0.0, tfinal=4.0, rho=28.0, sigma=10.0, beta=8/3.0):
In [9]:     """Solve the Lorenz differential equations.
In [10]:     """
In [11]:     rho = 28.0
In [12]:     sigma = 10.0
In [13]:     beta = 8/3.0
In [14]:     t0, tf = tspan
In [15]:     Nt = 100000
In [16]:     Ns = 10000
In [17]:     h = (tf - t0) / Nt
In [18]:     dt = h / 10.0
In [19]:     t = np.empty(Nt + 1)
In [20]:     t[0] = t0
In [21]:     for i in range(1, Nt + 1):
In [22]:         t[i] = t[i - 1] + h
In [23]:     y0 = np.array([1.0, 1.0, 1.0])
In [24]:     sol = solve_ivp(lorenz3, (t0, tf), y0, t_eval=t, method='RK45', rtol=1e-05, atol=1e-08)
In [25]:     y = sol.y
In [26]:     x = y[0, :Ns]
In [27]:     y = y[1, :Ns]
In [28]:     z = y[2, :Ns]
In [29]:     fig = plt.figure()
In [30]:     ax = fig.add_subplot(111, projection='3d')
In [31]:     ax.plot(x, y, z)
In [32]:     ax.set_xlabel('x')
In [33]:     ax.set_ylabel('y')
In [34]:     ax.set_zlabel('z')
In [35]:     plt.show()
```

Les données

- Accès au catalogue 
- Accès distant (via S3, HTTPS, OPeNDAP, FTP, ...)

Les ressources

- Calcul (CPU/GPU, RAM, ...)
- Langages (Python, R, Julia)
- Environnements (de base (pangeo) ou personnalisés)
- Outils (QGIS, Git, ...)

VRE (Concrètement, comment ça fonctionne ?)

Travailler à distance ... comme si vous étiez en local.

1 Accéder à l'URL (et se connecter)

<https://jupyterhub.cnes.fr> (ou ifremer.fr)

connexion

2 Sélectionner les ressources (CPU, RAM, Durée, ...)

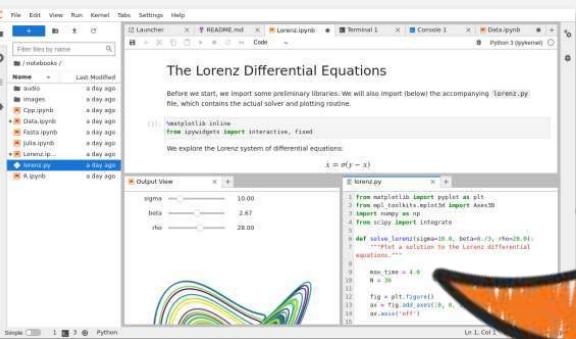
Sélection des ressources

1 cœur, 4gb, 8 heures

8 coeurs, 32gb, 8 heures

28 coeurs, 115gb, 8 heures

3 Commencer à l'utiliser la VRE (Données, Analyse, Visualisation, ...)



Les données

- Accès au catalogue 
- Accès distant (via S3, HTTPS, OPeNDAP, FTP, ...)

Les ressources

- Calcul (CPU/GPU, RAM, ...)
- Langages (Python, R, Julia)
- Environnements (de base (pangeo) ou personnalisés)
- Outils (QGIS, Git, ...)

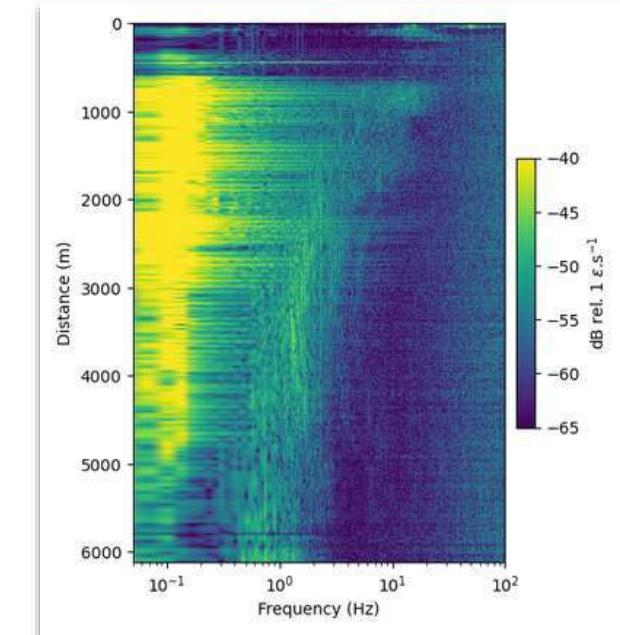
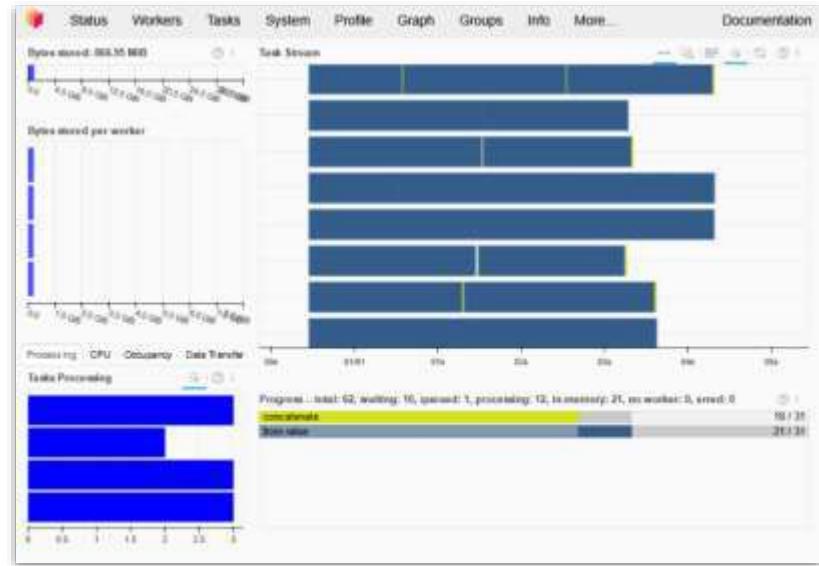
Les services (Helpdesk)

- Documentation (Premiers pas, FAQ, Tutoriels ...) <https://odatis-public.gitlab-pages.ifremer.fr/vre/documentation>
- Cas d'usages (Illustration de l'utilisation de la VRE pour traiter des problématiques scientifiques.)

Cas d'usages

Traitement des données de fibre optique (DAS)

Dans le cadre d'une campagne menée en Sicile par une équipe de [Geo-Ocean](#), un volume très important de données a été acquis (environ 20To). L'idée de ce cas d'usages est d'étudier comment optimiser la lecture, le traitement et la visualisation d'un tel volume de données ?



| 10

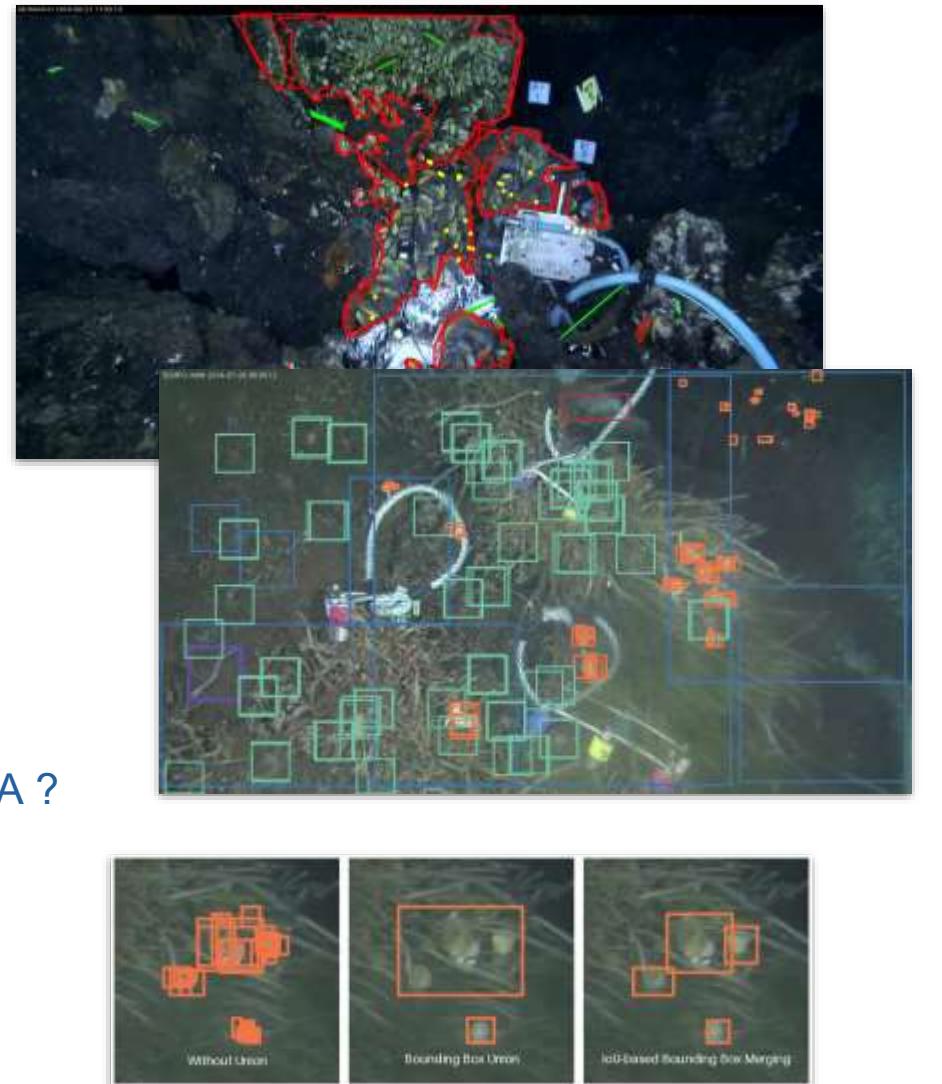
Cas d'usages

Traitement des données d'imagerie benthique

Dans le cadre du projet [DeepSeaSpy](#) mené par le LEP, un grand nombre d'annotation d'espèce provenant de la science participative sont disponibles. L'idée de ce cas d'usage est d'étudier comment utiliser des données citoyennes pour entraîner un modèle d'IA, afin d'annoter de futures images de manière automatique ?

➤ Les notions abordées:

- Comment accéder aux données sur le HPC ?
- Comment nettoyer les données ?
- Comment utiliser les ressources GPU pour entraîner un modèle d'IA ?
- Comment utiliser le modèle pour annoter de nouvelles images ?

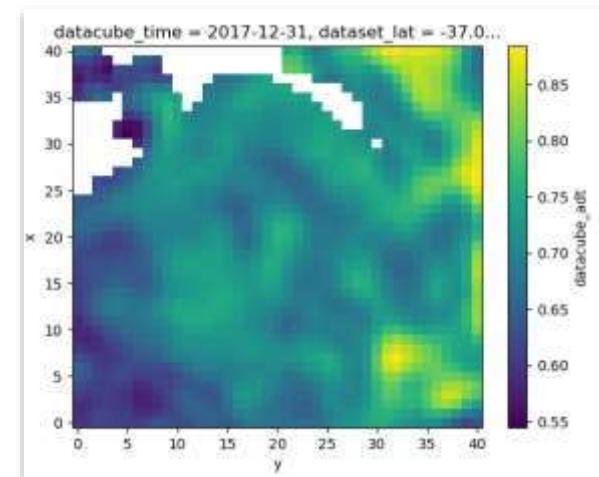
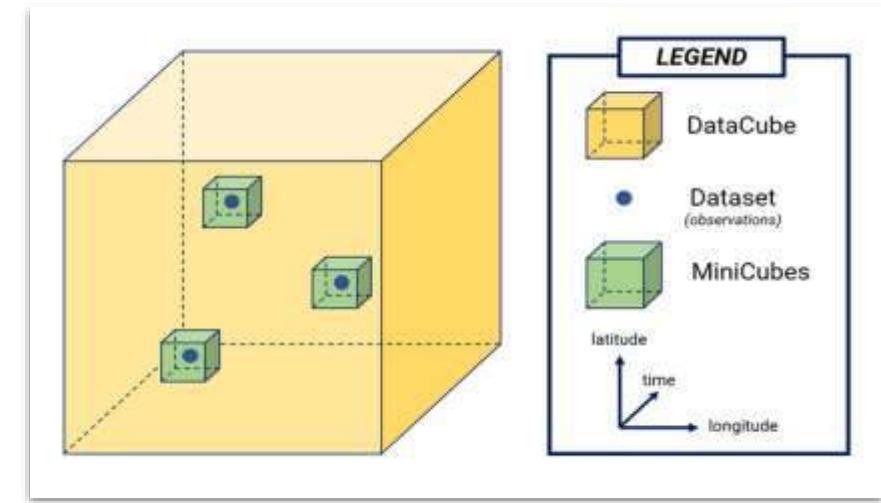


Cas d'usages

Colocalisation des données in situ et satellitaires

Dans le cadre de nombreux travaux de recherche, une étape importante est la colocalisation des données. L'idée de ce cas d'usage est d'étudier comment utiliser des outils communs aux géosciences ([Xarray](#), [Kerchunk](#), [Dask](#)), pour optimiser la colocalisation d'un grand nombre de données ?

- Les notions abordées:
 - Comment utiliser Kerchunk pour émuler un fichier zarr à partir de milliers de fichiers unitaire ?
 - Comment utiliser Xarray pour ouvrir un datacube ?
 - Comment utiliser et optimiser dask pour la lecture et l'extraction des données ?
- Développement d'une librairie de colocalisation : [PytCube](#)



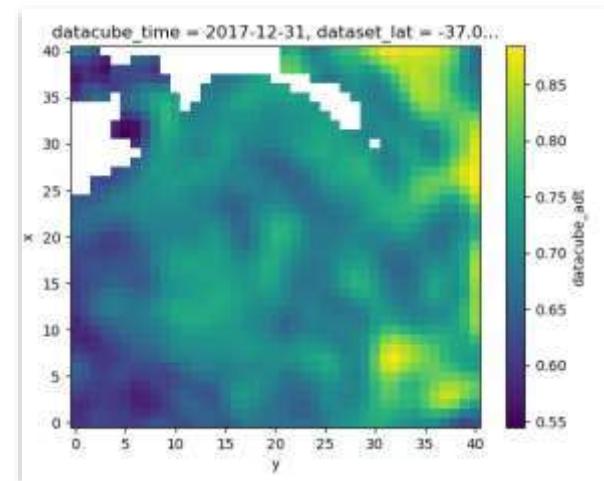
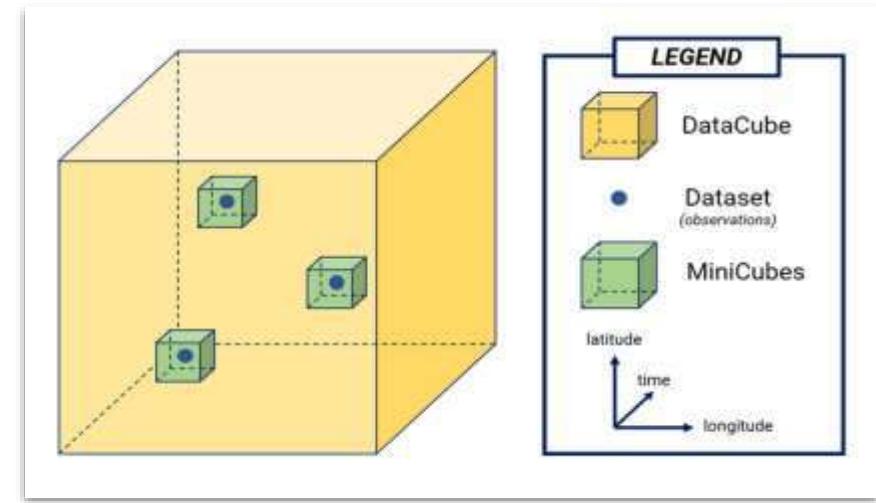
 PytCube

Cas d'usages

Colocalisation des données in situ et satellitaires

Dans le cadre de nombreux travaux de recherche, une étape importante est la colocalisation des données. L'idée de ce cas d'usage est d'étudier comment utiliser des outils communs aux géosciences ([Xarray](#), [Kerchunk](#), [Dask](#)), pour optimiser la colocalisation d'un grand nombre de données ?

- Les notions abordées:
 - Comment utiliser Kerchunk pour émuler un fichier zarr à partir de milliers de fichiers unitaire ?
 - Comment utiliser Xarray pour ouvrir un datacube ?
 - Comment utiliser et optimiser dask pour la lecture et l'extraction des données ?
- Développement d'une librairie de colocalisation : [PytCube](#)



Vous avez une idée de cas d'usages ? N'hésitez pas à nous contacter.

 **PytCube**

Perspectives

L'offre de service VRE du pôle ODATIS se concentre pour l'instant sur la solution JupyterLab, mais elle a vocation à s'élargir à l'avenir en incluant d'autres solutions, telles que [Galaxy](#).

