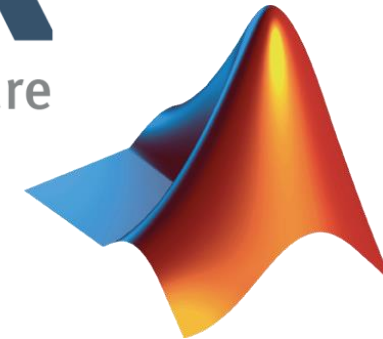


# WORKSHOP: Parallel Computing With MATLAB (Part II)



**TRUBA**  
Turkish Science e-Infrastructure



Raymond Norris  
Application Engineer, MathWorks  
July 2022

# Agenda

- Part I – Parallel Computing with MATLAB on the Desktop
  - Parallel Computing Toolbox
- Part II – Scaling MATLAB to the TRUBA HPC cluster
  - MATLAB Parallel Server



# Agenda

- Part I – Parallel Computing with MATLAB on the Desktop
  - Parallel Computing Toolbox
- Part II – Scaling MATLAB to the TRUBA HPC cluster
  - MATLAB Parallel Server



# Simply Complex Lab: Scaling with MATLAB Parallel Server



**unam** excellence in  
science and  
technology



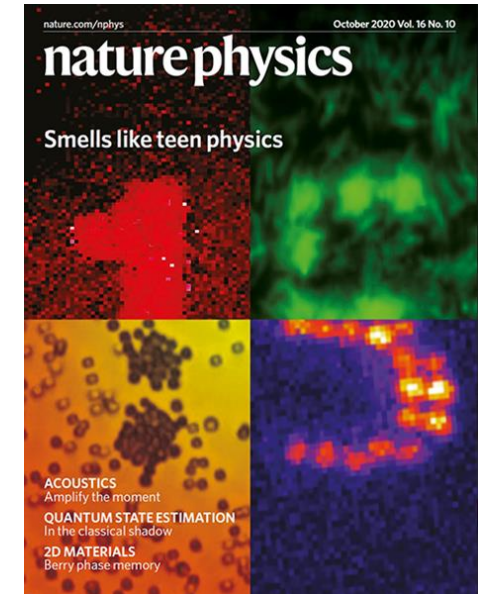
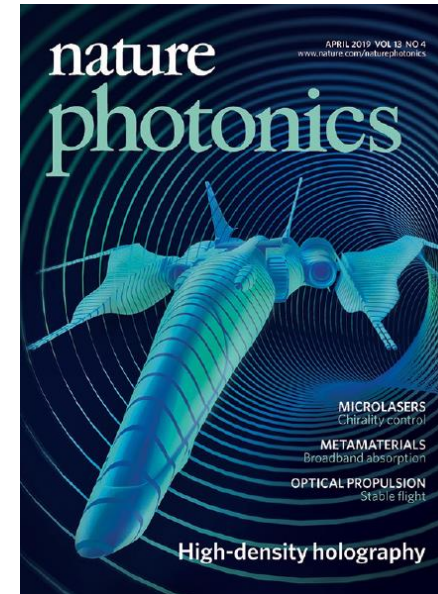
**unam** excellence in  
science and  
technology



**Simply Complex Lab** is driven towards deciphering emergent, complex, far from equilibrium phenomena.

We are determined to solve scientifically and technologically persistent problems by exploring solutions under far from equilibrium conditions. Such conditions dictate:

- sufficiently high nonlinearity
- strong stochasticity
- intrinsic positive and negative feedback mechanisms



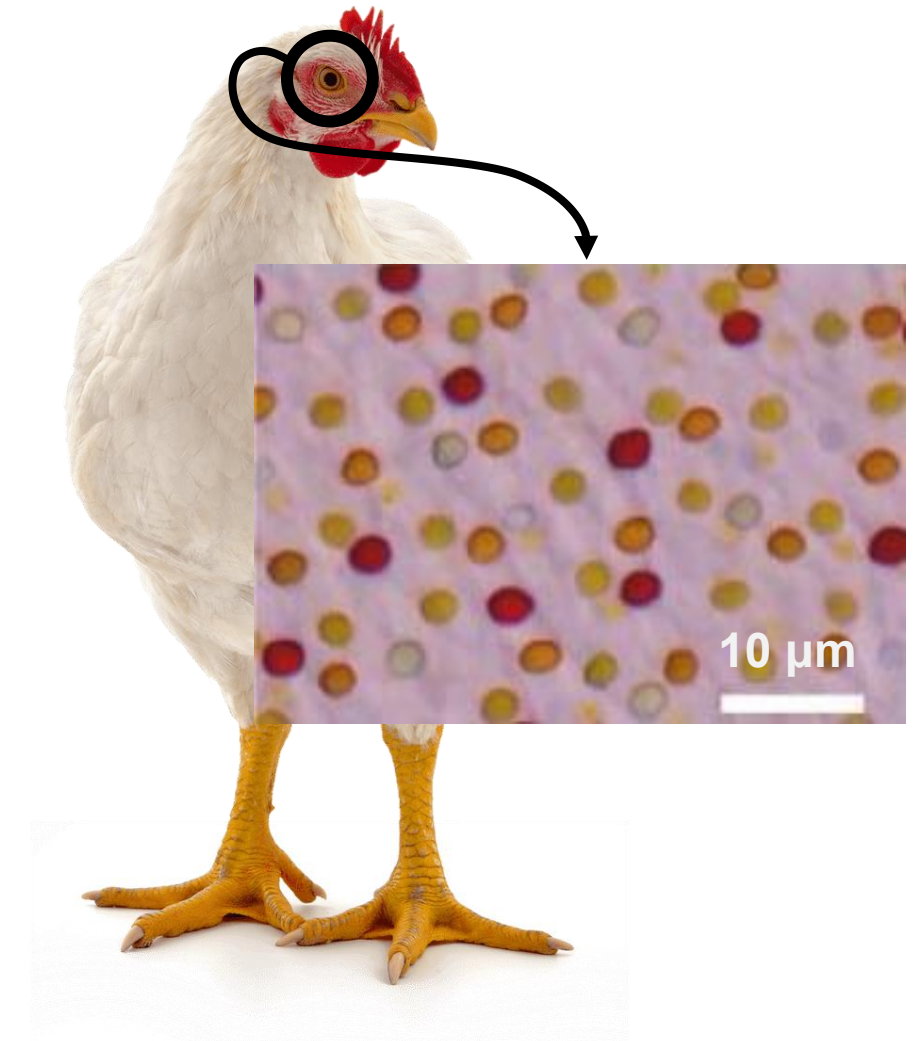
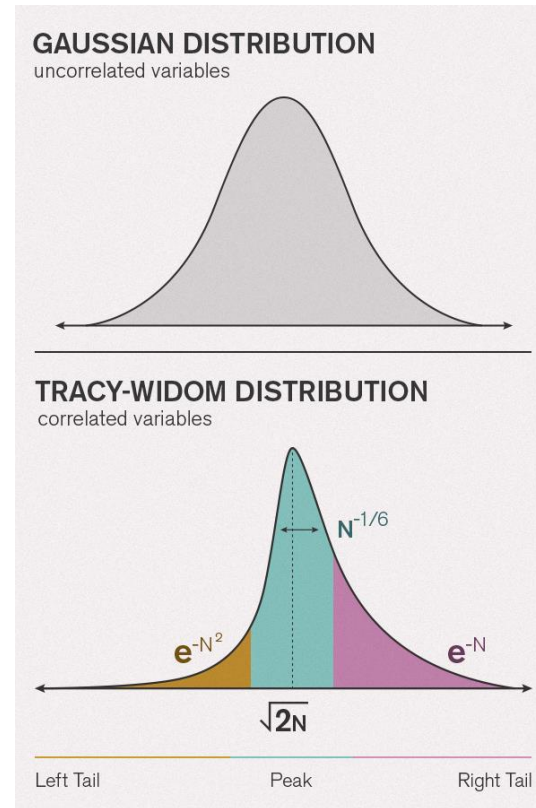
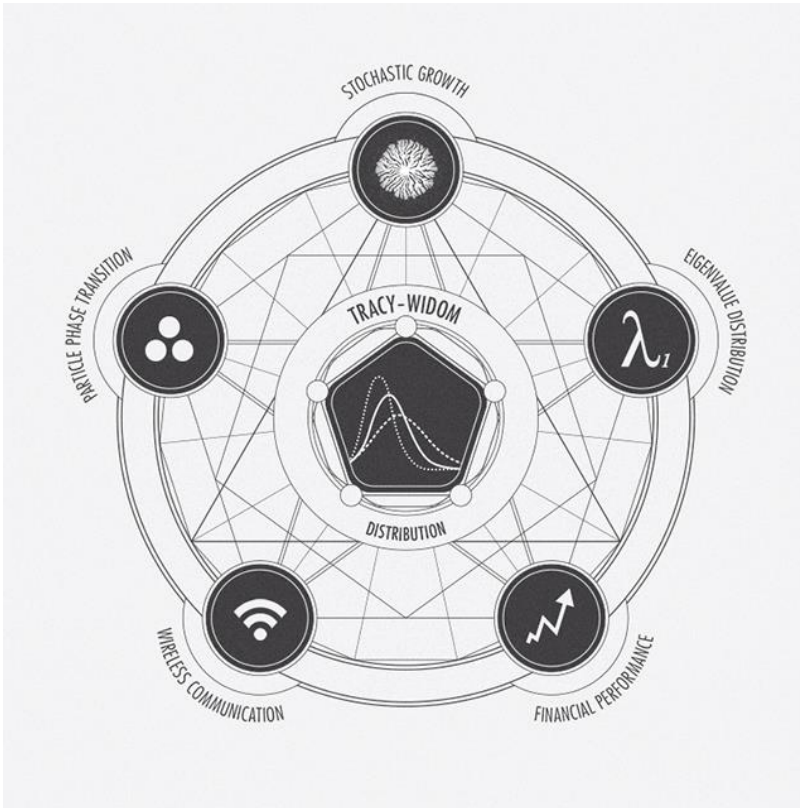
The group's work was highlighted on the covers of Nature photonics and Nature Physics journals

<http://staff.bilkent.edu.tr/serim/>





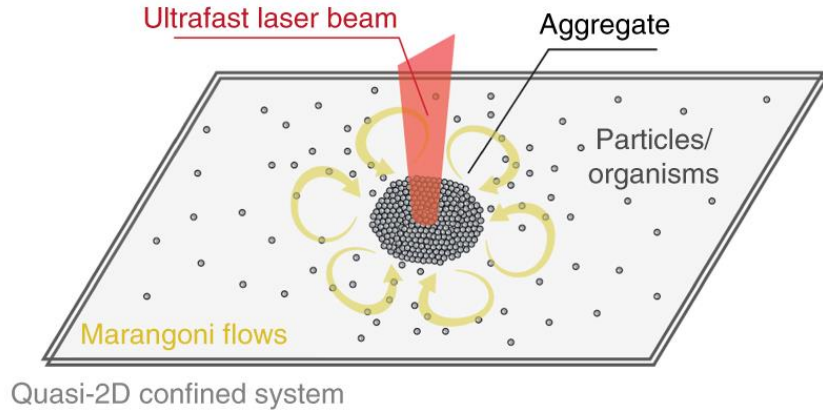
**unam** excellence in  
science and  
technology



“At the Far Ends of a New Universal Law,” **Quantum Magazine**



**unam** excellence in science and technology

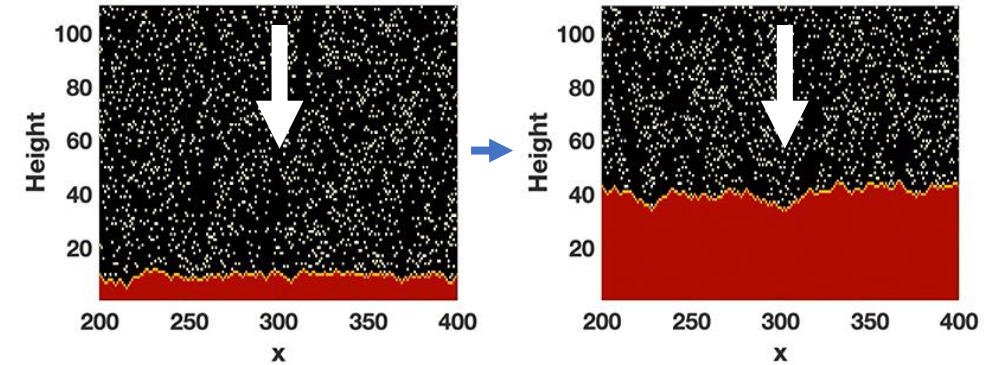
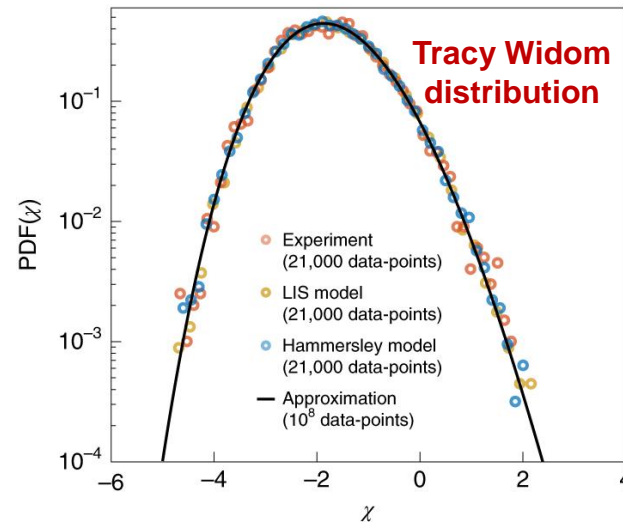
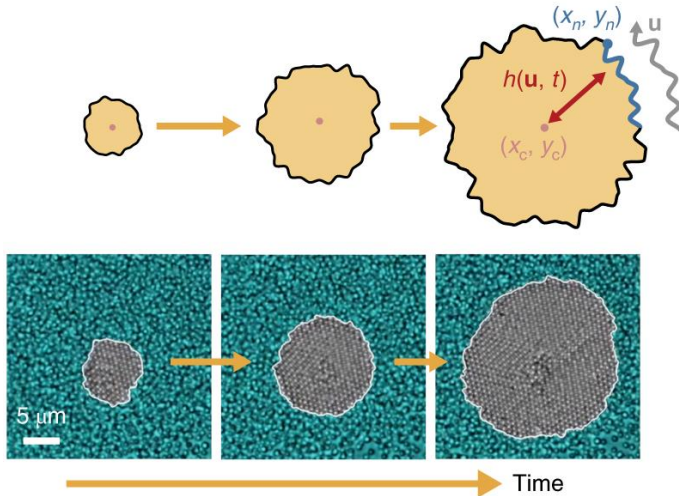


## Kardar Parisi Zhang

Nobel Prize in Physics, 2021

$$\frac{\lambda_t}{2} (\partial_t h)^2 + v_t \partial_t^2 h + \partial_t h = \frac{\lambda_x}{2} (\partial_x h)^2 + v_x \partial_x^2 h + \sqrt{D} \xi(x, t)$$

Temporal nonlinearity      Temporal diffusion      Spatial nonlinearity      Spatial diffusion      Randomness



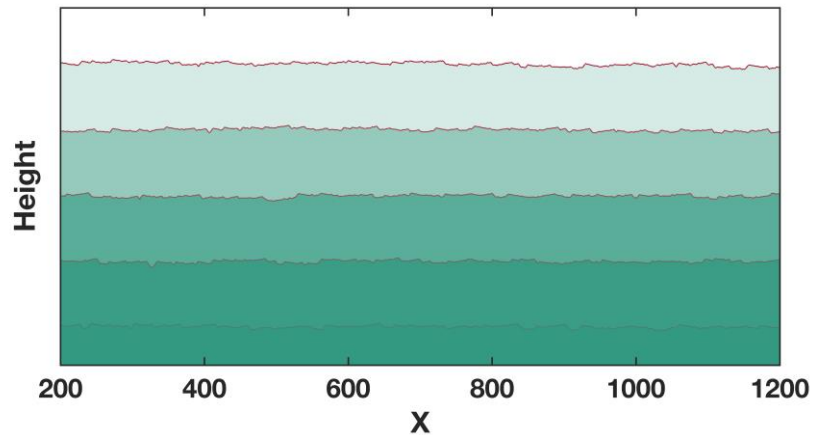
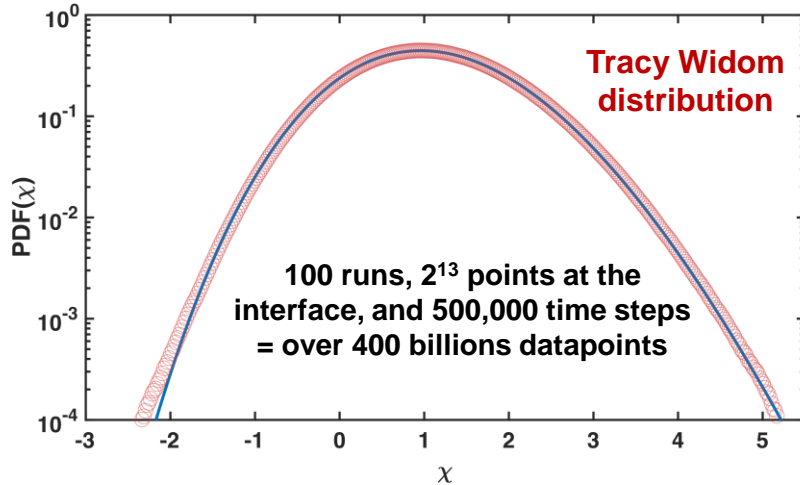
Numerical method:  
4<sup>th</sup> order Runge-Kutta

“Universality of dissipative self-assembly from quantum dots to human cells,”  
Ilday, et al., Nature Physics, 2020





Results by Xhulian Lickollari, MS student in Bilkent

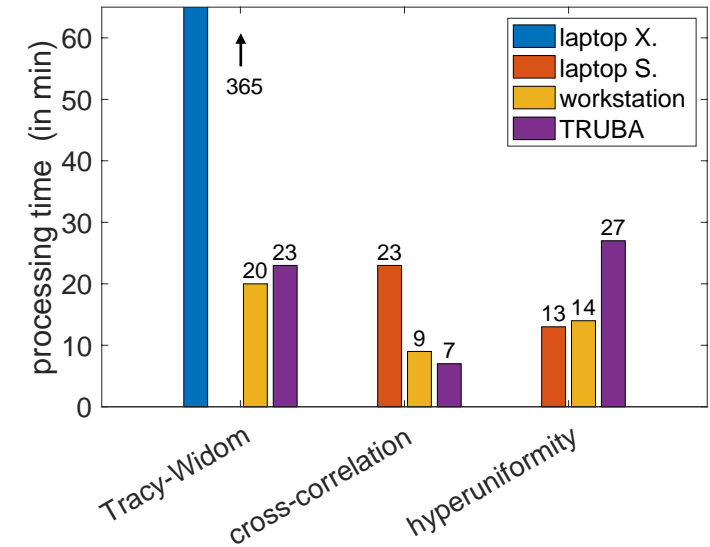
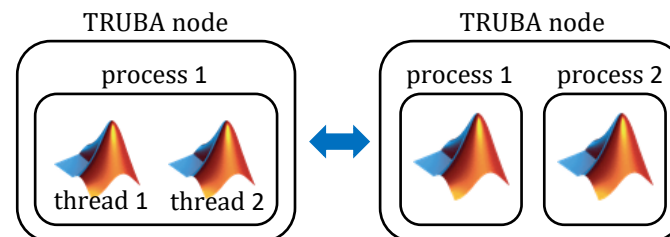


## Code parallelization

```
% for j = 1:n
parfor j = 1:n
```

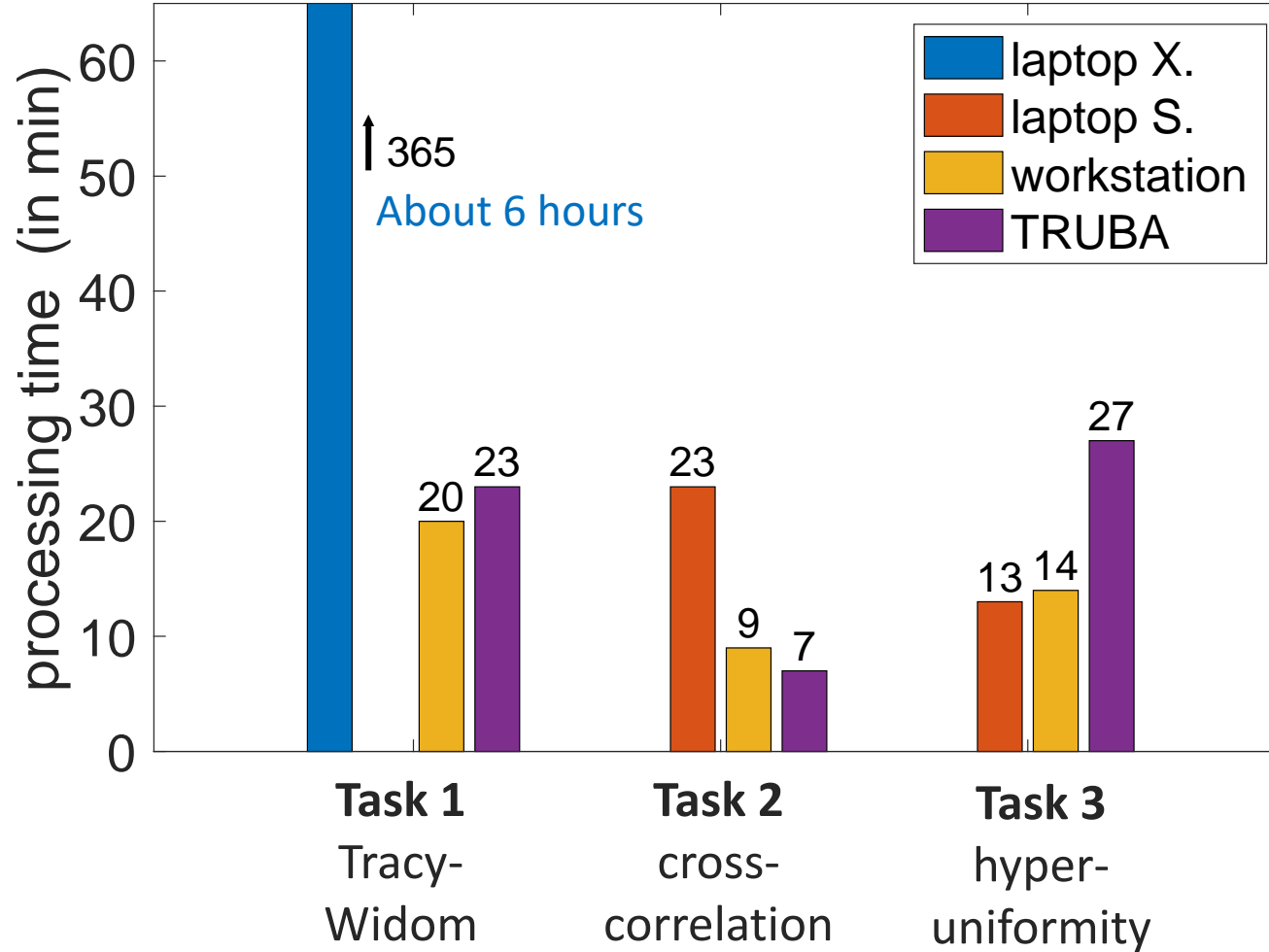
## MATLAB on TRUBA

- license requirements
- job queue
- auto-copy of figures, data, etc.
- very few code adjustments
- try optimize nr of threads!



machine	cores	RAM	Resources
laptop X.	2	16 GB	single-user
laptop S.	8	32 GB	single-user
workstation	32	128 GB	multi-user
TRUBA node	28	±100 GB	job queue





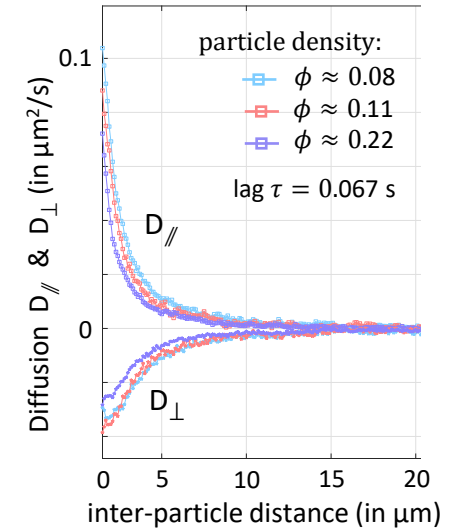
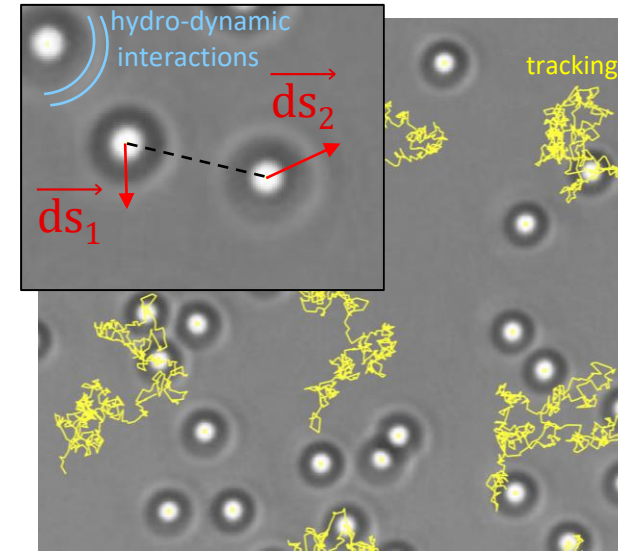
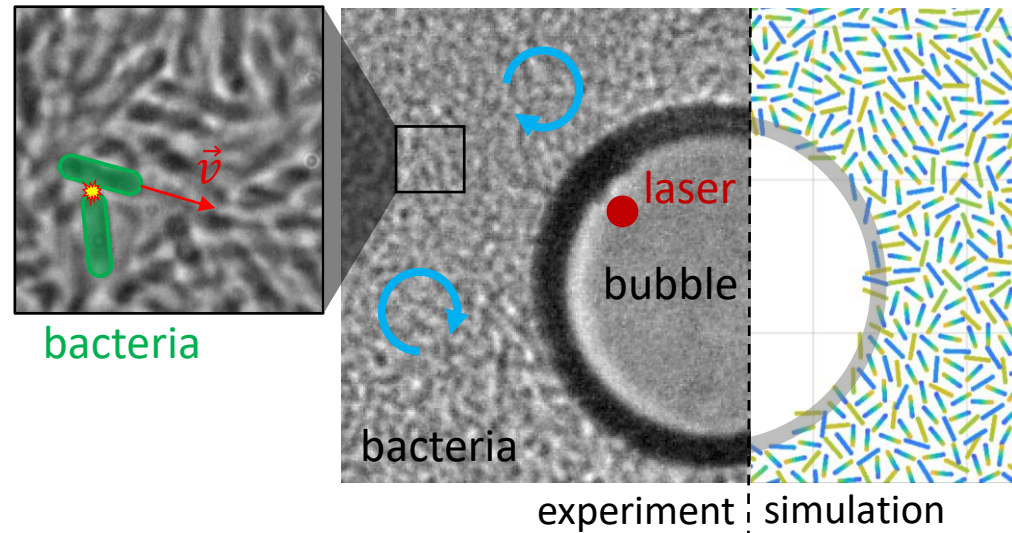
machine	cores	RAM	Resources
laptop X.	2	16 GB	single-user
laptop S.	8	32 GB	single-user
workstation	32	128 GB	multi-user
TRUBA node	28	±100 GB	job queue

same nr. of workers  
26 workers : 13 x 2 threads + 1

80% visualization → no speed-up

## Currently testing codes on TRUBA

1. Laser-material interaction (surface patterning)
2. Other forms of universality (hyperuniformity)
3. Analysis of high frame-rate microscopy videos.
4. Bacterial turbulence simulations



## Possible future usage TRUBA

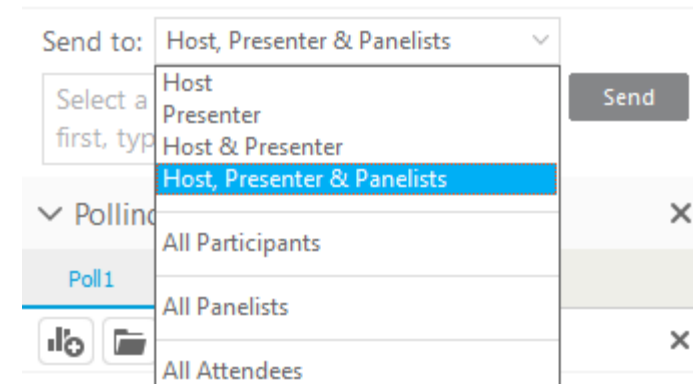
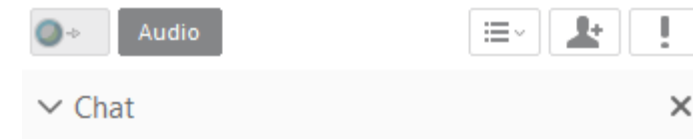
1. Colloidal particle diffusion & velocity correlation
2. Ultrafast laser oscillator simulations.
3. Laser-material interaction (ablation and heating).
4. Fluid dynamics.

**Thank you**

# Chatting

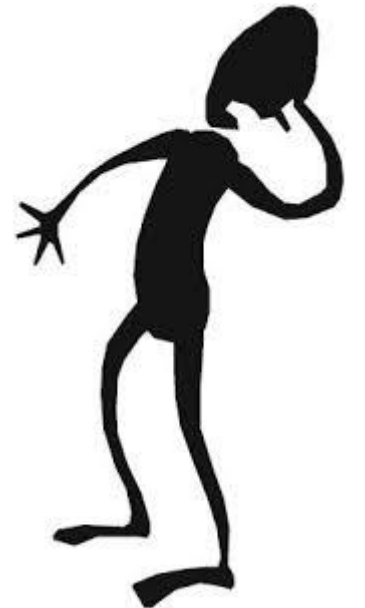
- Send to at least the *Host, Presenter & Panelists*
- Ideally, send to *All Attendees*

Participants



## A few notes about today's workshop

- The workflow and examples are about process, not performance
- MATLAB User Guide
  - [https://docs.truba.gov.tr/how-to-guides/MATLAB/MATLAB\\_TRUBA.html](https://docs.truba.gov.tr/how-to-guides/MATLAB/MATLAB_TRUBA.html)
- Requirements
  - ☐ MATLAB & Parallel Computing Toolbox **R2021b**
  - ☐ Account on TRUBA

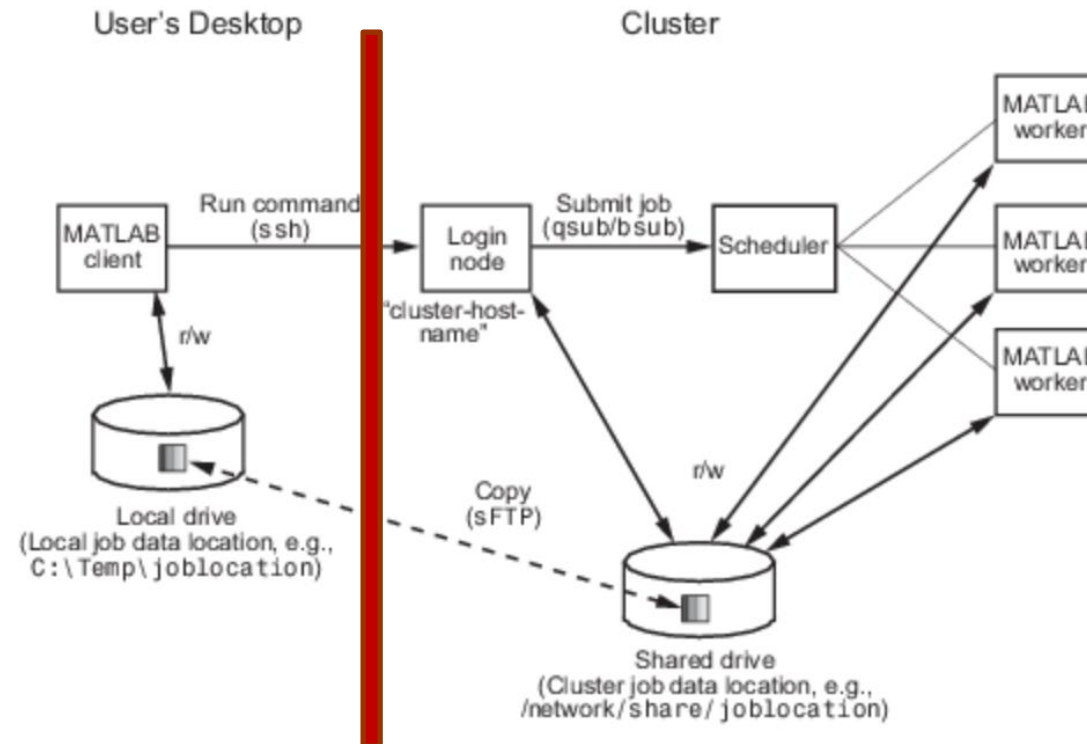




# Scaling MATLAB to TRUBA

Your machine

```
>> job1 = batch( );  
>> job2 = batch( );
```



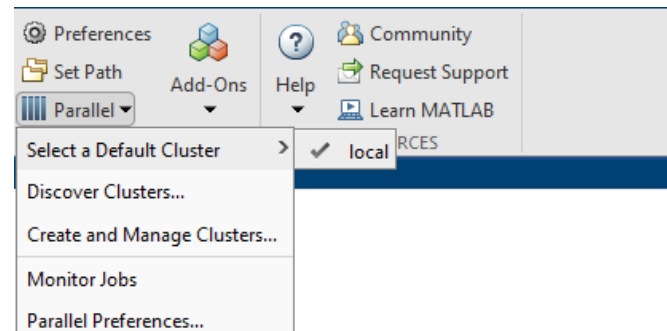
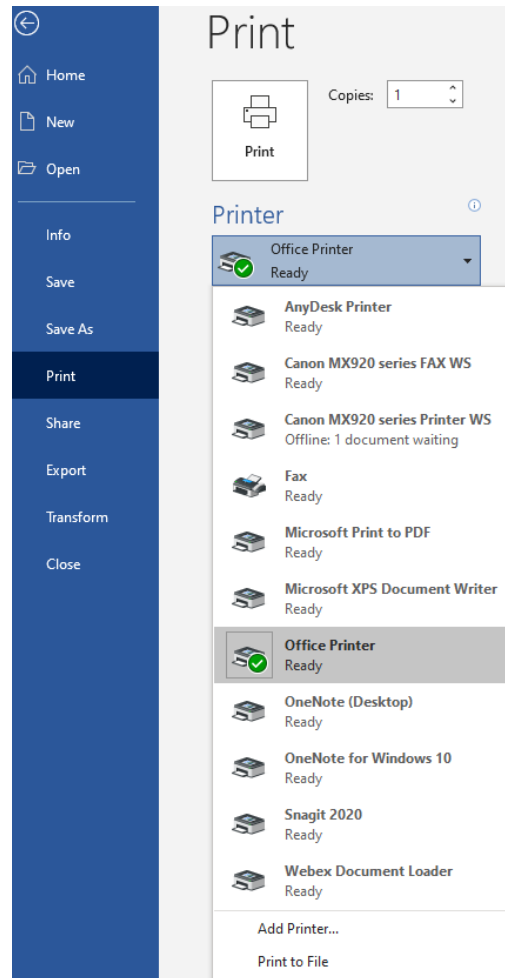
Cluster

```
#SLURM ...
```

```
module load matlab
```

```
matlab ...
```

# Profiles



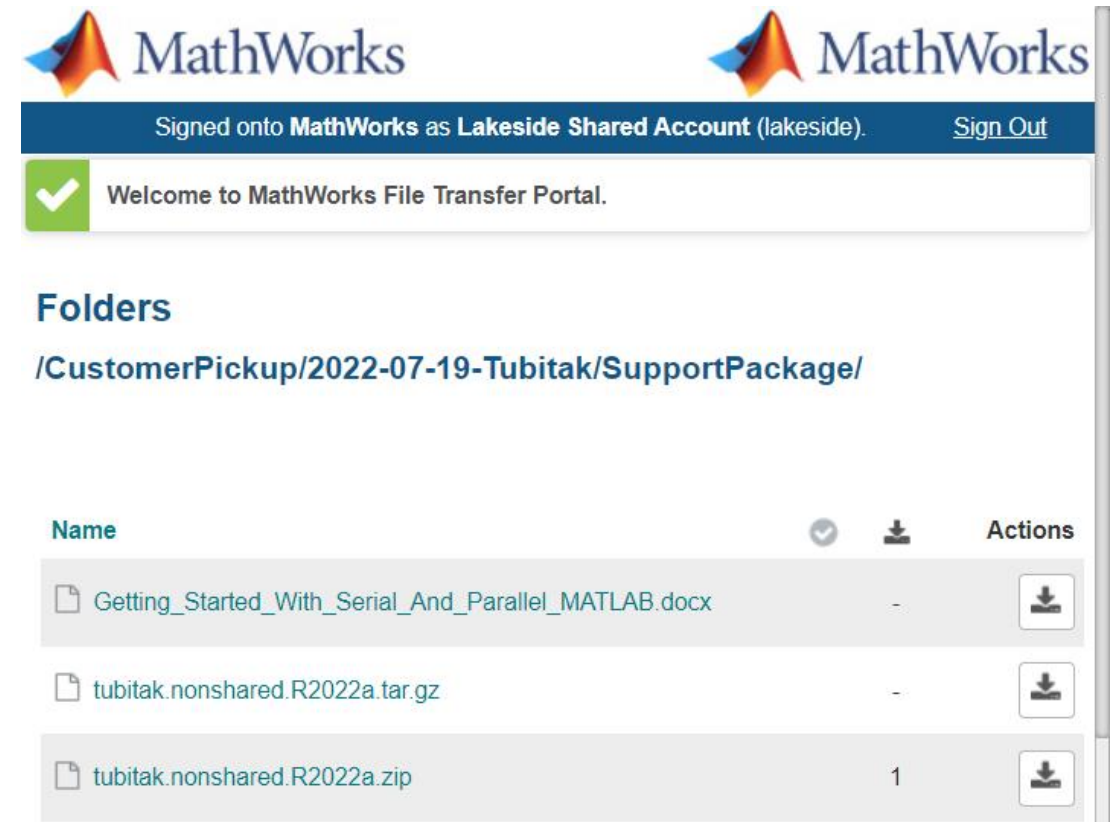
“How does MATLAB know about TRUBA?”









# Download the TRUBA MATLAB support package

<https://tinyurl.com/Tubitak-Support-Package>

- Username: Lakeside
- Password: 1qi6r47f
  - good until **September 1, 2022**
  - Will eventually be posted on TRUBA site
- Provides job hooks for
  - ☐ submission (**sbatch**)
  - ☐ state (**squeue**)
  - ☐ deletion (**scancel**)

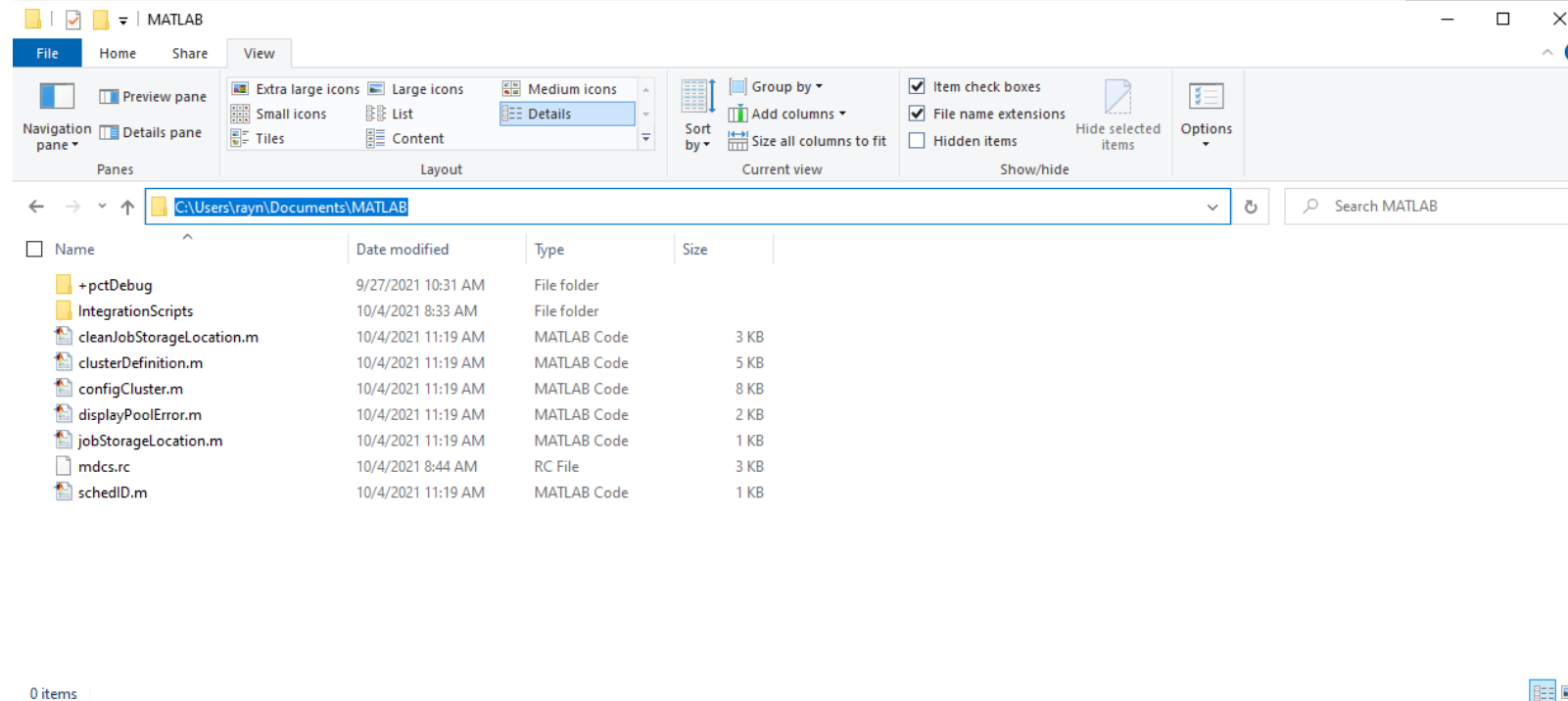


The screenshot shows the MathWorks File Transfer Portal interface. At the top, there are two MathWorks logos. Below them, a blue bar indicates the user is signed in as 'Lakeside Shared Account (lakeside)' with a 'Sign Out' link. A green checkmark icon and the text 'Welcome to MathWorks File Transfer Portal.' are displayed. The main heading is 'Folders', followed by the path '/CustomerPickup/2022-07-19-Tubitak/SupportPackage/'. Below this is a table listing files:

Name		Actions
 Getting_Started_With_Serial_And_Parallel_MATLAB.docx	-	
 tubitak.nonshared.R2022a.tar.gz	-	
 tubitak.nonshared.R2022a.zip	1	

# Install the support package – Windows

```
>> userpath  
ans =  
      'c:\Users\rayn\Documents\MATLAB'  
  
>>  
>>
```



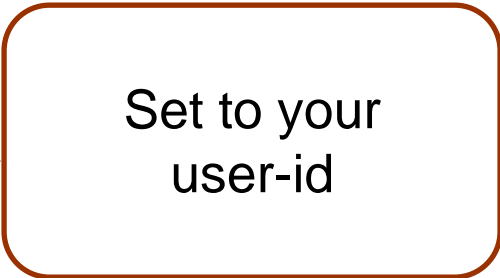


# Install the support package – Linux/macOS

```
[user@localhost ~] mkdir ~/Documents/MATLAB
[user@localhost ~] tar xf ~/tubitak.nonshared.R2022a.tar.gz -C ~/Documents/MATLAB
[user@localhost ~]
[user@localhost ~] ls -l ~/Documents/MATLAB
+pqDebug
IntegrationScripts
cleanJobStorageLocation.m
clusterDefinition.m
clusterQueues.m
configCluster.m
displayPoolError.m
fixConnection.m
jobStorageLocation.m
mdcs.rc
schedID.m
```

## Configure MATLAB to create TRUBA profile

```
>> configCluster  
Username on TRUBA (e.g. joe): my-user-id  
Complete. Default cluster profile set to "truba R2021b".
```



Set to your user-id

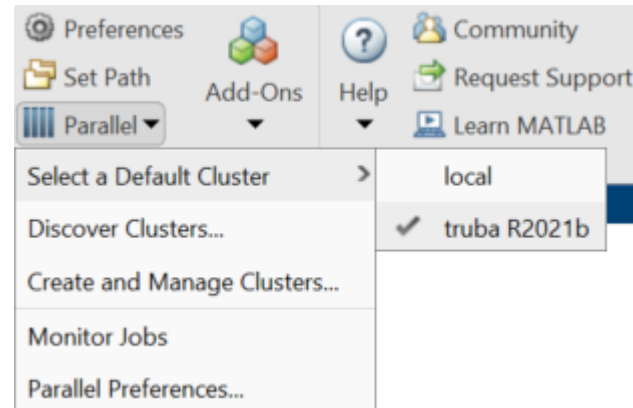
Must set WallTime before submitting jobs to TRUBA. E.g.

```
>> c = parcluster;  
>> % 5 hours  
>> c.AdditionalProperties.WallTime = '05:00:00';  
>> c.saveProfile
```

## WORKSHOP-ONLY: Add reservation

```
>> % Add reservation for today's workshop  
>> c.AdditionalProperties.Reservation = 'matlab';  
>> c.saveProfile
```

# New TRUBA profile





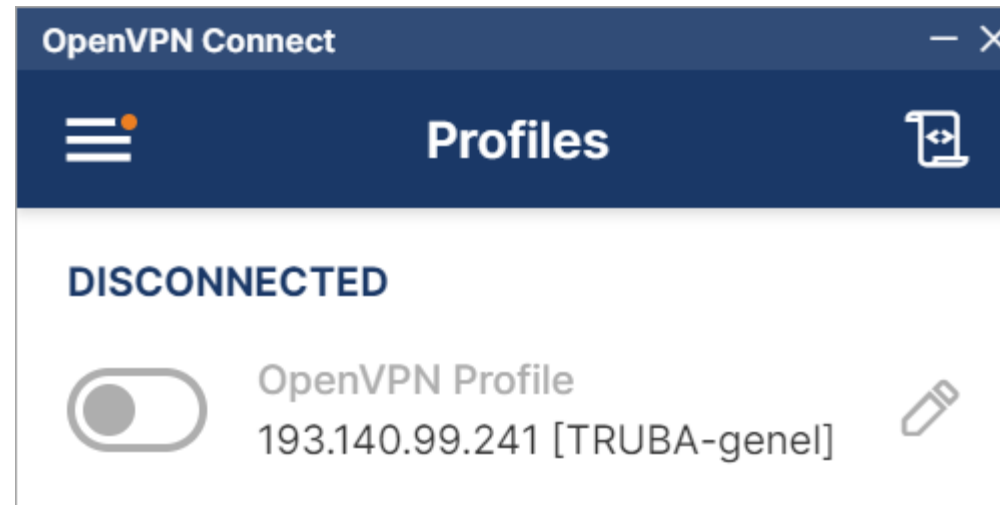
# MATLAB job submitters

- `parpool`
  - Single session
  - Synchronous execution
  - Seamlessly runs `parfor`, `parfeval`, and `spmd`
- `batch`
  - Multiple submissions
  - Non-blocking
  - Calls top-level function or script
  - Requires API to extract results

<https://www.mathworks.com/help/parallel-computing/parpool.html>

<https://www.mathworks.com/help/parallel-computing/batch.html>

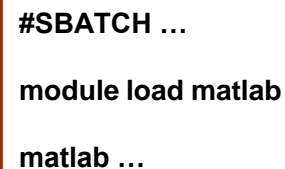
# VPN Connection – Required if not on ULAKNET



## Exercise: “Hello, World!”

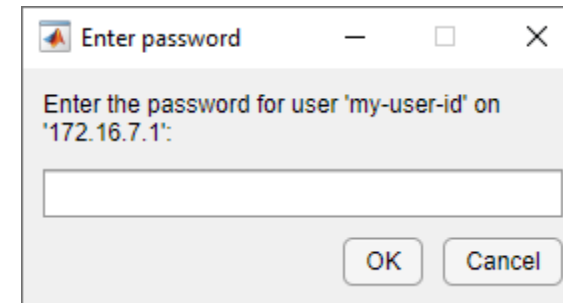
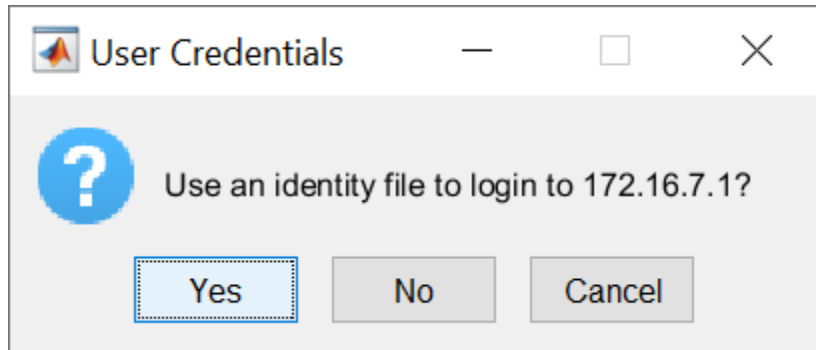


```
>> % Submit job to cluster to find out where MATLAB is running
>> % Get handle to HPC cluster
>> c = parcluster;
>> job = c.batch(@pwd,1,{}, 'CurrentFolder', '.');
```

A snippet of a Slurm batch script, enclosed in a box with a folded corner. It contains the following text:

```
#SBATCH ...
module load matlab
matlab ...
```

# SSH credentials (private key/passphrase or password)





## Fetching results

```
>> % Submit job to cluster to find out where MATLAB is running
>> % Get handle to HPC cluster
>> c = parcluster;
>> job = c.batch(@pwd,1,{}, 'CurrentFolder', '.');
additionalSubmitArgs =
    '--ntasks=1 -N 1 -c 28 --ntasks-per-core=1 -t 00:10:00 -p hamsi'
>>
>> % Check the state of the job
>> job.State
ans =
    'finished'
>>
>> % Fetch the results
>> job.fetchOutputs{:}
ans =
    '/truba/home/rnorris'
```

## Benign warning if CurrentFolder isn't set

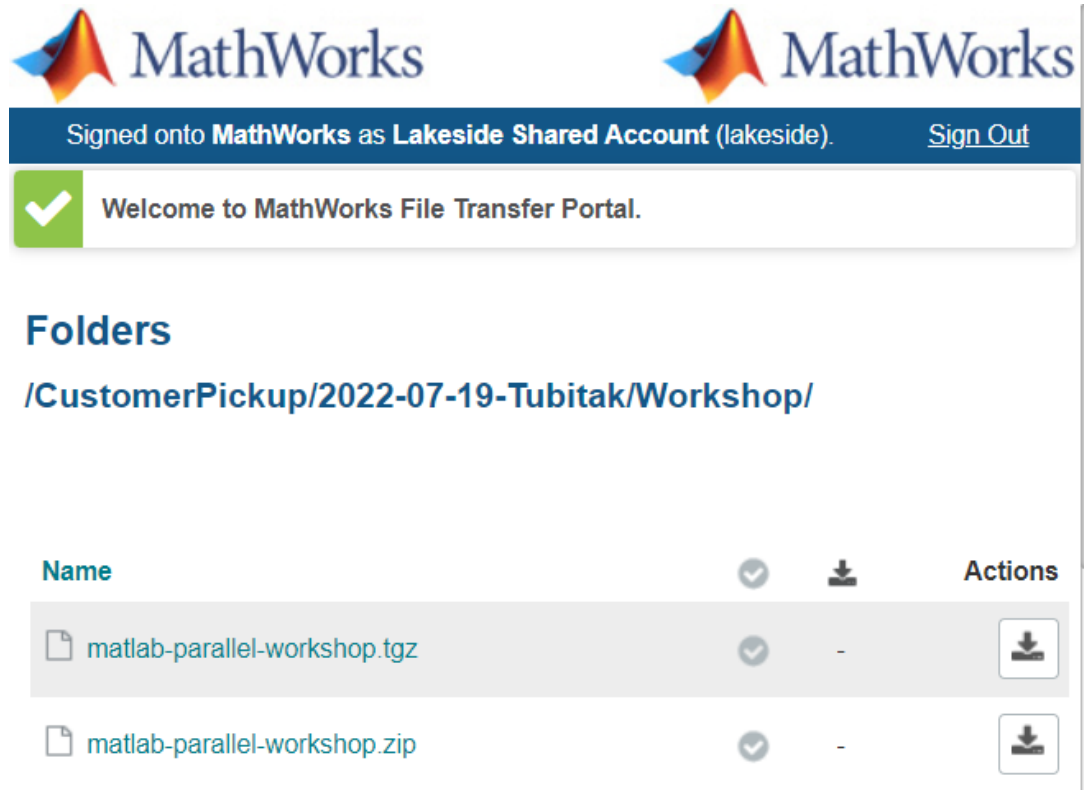
```
>> job.fetchOutputs{:}
```

Warning: The task with ID 1 issued the following warnings:





Warning: Worker unable to change folder to 'C:\Users' at the start of the batch job. The job will be executed from '/truba/home/rnorris'. To execute from a different folder use the 'CurrentFolder' parameter of batch. To suppress this warning, set 'CurrentFolder' to '.'.

# Download the workshop files

<https://tinyurl.com/Tubitak-Workshop>



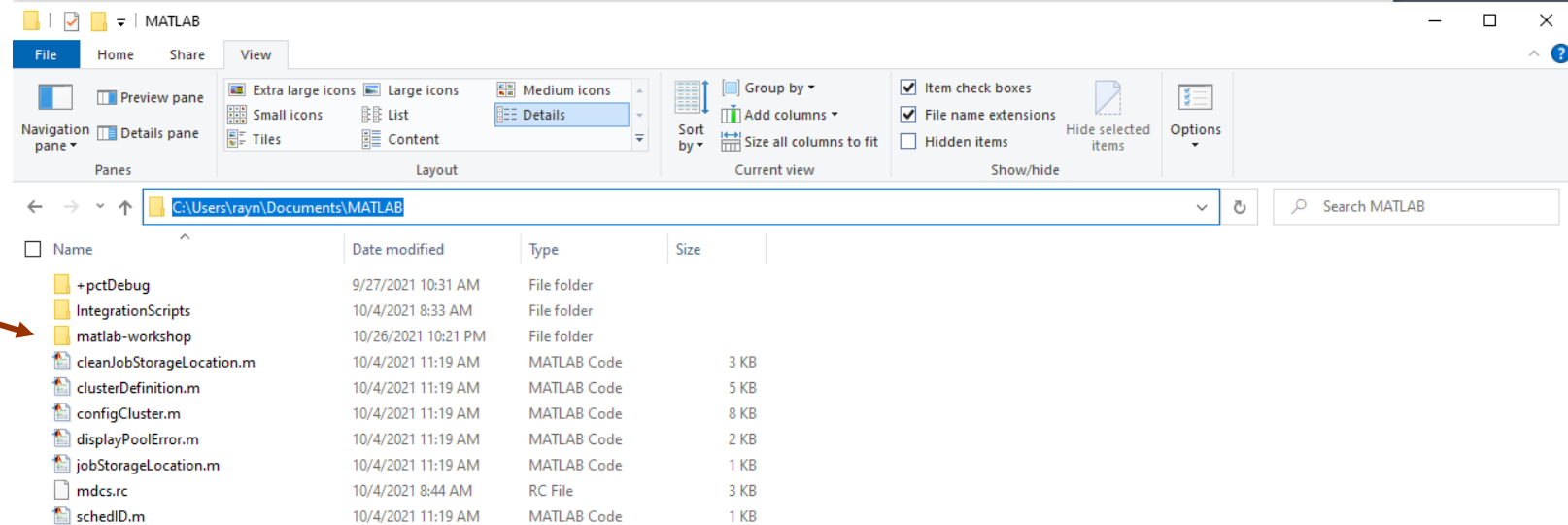
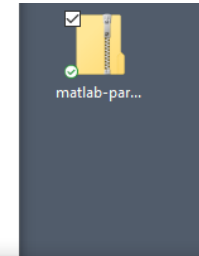
The screenshot shows the MathWorks File Transfer Portal interface. At the top, there are two MathWorks logos. Below them, a blue bar indicates the user is signed in as 'Lakeside Shared Account (lakeside)' with a 'Sign Out' link. A green checkmark icon and a message 'Welcome to MathWorks File Transfer Portal.' are displayed. The 'Folders' section shows the path '/CustomerPickup/2022-07-19-Tubitak/Workshop/'. Below this is a table with columns for 'Name', a status column with checkmarks, a download column with arrows, and 'Actions'.

Name			Actions
 matlab-parallel-workshop.tgz	✓	-	
 matlab-parallel-workshop.zip	✓	-	

```
calc_fft_cpu_gpu.m  
calc_mandelbrot.m  
calc_pi.m  
mandelbrot_example.m  
parallel_example.m  
process_files_v1.m  
process_files_v2.m  
solve_sys_linear_eqns.m  
test_fcn.m
```

# Install workshop files – Windows

```
>> userpath  
ans =  
  
    'c:\Users\rayn\Documents\MATLAB'  
  
>>  
>>
```



10 items



# Install workshop files – Linux/macOS

```
[user@localhost ~]# mkdir -p ~/Documents/MATLAB
[user@localhost ~]# tar xf ~/matlab-parallel-workshop.tgz -C ~/Documents/MATLAB
[user@localhost ~]#
[user@localhost ~]# ls -l ~/Documents/MATLAB/matlab-workshop
calc_fft_cpu_gpu.m
calc_mandelbrot.m
calc_pi.m
mandelbrot_example.m
parallel_example.m
process_files_v1.m
process_files_v2.m
solve_sys_linear_eqns.m
test_fcn.m
```

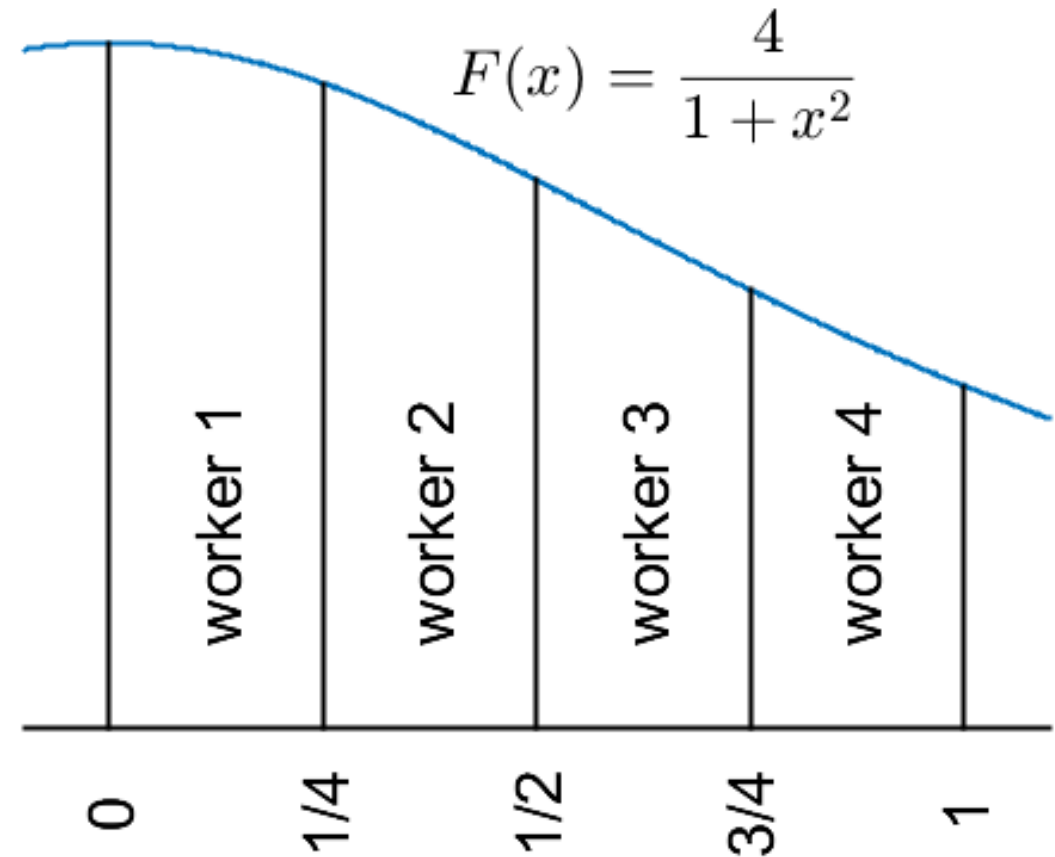


## Change directories to workshop

```
>> cd(fullfile(userpath, 'matlab-workshop'))
```

## Exercise: Calculate $\pi$

$$\int_0^1 \frac{4}{1+x^2} dx = 4(\text{atan}(1) - \text{atan}(0)) = \pi$$



# Calculate $\pi$

```
function calc_pi

sppmd
    a = (labindex - 1)/numlabs;
    b = labindex/numlabs;
    fprintf('Subinterval: [%-4g, %-4g]\n', a, b)

    myIntegral = integral(@quadpi, a, b);
    fprintf('Subinterval: [%-4g, %-4g]    Integral: %4g\n', a, b, myIntegral)

    piApprox = gplus(myIntegral);
end

approx1 = piApprox{1}; % 1st element holds value on worker 1
fprintf('pi           : %.18f\n', pi)
fprintf('Approximation: %.18f\n', approx1)
fprintf('Error          : %g\n',    abs(pi - approx1))

function y = quadpi(x)
%QUADPI Return data to approximate pi.

% Derivative of 4*atan(x)
y = 4./(1 + x.^2);
```

## Where should you start a local parallel pool?

```
function parallel_example
```

```
parpool(4);
```

```
parfor idx = 1:8  
    A(idx) = rand;
```

```
end
```

“What will happen  
the next time you run  
this code?”

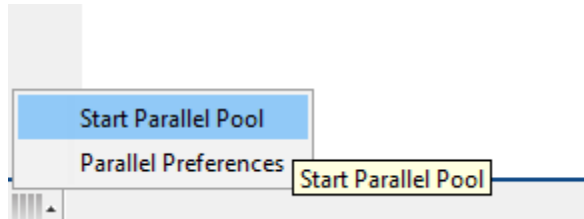


Error using parpool

Found an interactive session. You cannot have multiple interactive sessions open simultaneously.

## How should you start a local parallel pool?

- Call `parpool` from the Command Window
- Have MATLAB automatically start a parallel pool if it hasn't already started
- From the lower lefthand corner



Then how do I tell the cluster my job needs a parallel pool?

```
job = c.batch(..., 'Pool', pool_size);
```

## Submit `calc_pi` job

```
>> % Submit calc_pi job
>> c = parcluster;
>>
>> % Request 3 workers
>> job = c.batch(@calc_pi,0,{}, 'CurrentFolder','.', 'Pool',3);
additionalSubmitArgs =
    '--ntasks=4 -N 1 -c 7 --ntasks-per-core=1 -t 00:10:00 -p hamsi'
```



“If my Pool is size 3, why am I requesting 4 tasks?”



## “The Power of 28”

- *hamsi* partition
  - dual socket nodes, with 28 cores per socket
  - Multiples of 28 cores need to be consumed by your job
- Therefore, the total number of workers must either be
  - a factor of 28 (1, 2, 4, 7, or 14) or
  - a multiple of 28 (28, 56, 84, 112, etc.)



## Requesting $\leq 28$ workers

- Request a pool of **0,1,3,6,13**, or **27** workers
  - If NumThreads is 1 (default), we'll adjust the thread count to fit on a single node

```
>> job = c.batch(@pwd,1,{}, 'CurrentFolder','.', 'Pool', 3);  
additionalSubmitArgs =  
    '--ntasks=4 -N 1 -c 7 --ntasks-per-core=1 -t 00:10:00 -p hamsi'  
>>  
>>
```

```
>> job = c.batch(@pwd,1,{}, 'CurrentFolder','.', 'Pool', 2);
```

Error using parallel.Cluster/batch (line 158)

Job submission failed because the plugin function  
'communicatingSubmitFcn.m' errored.

Caused by:

Error using getCommonSubmitArgs (line 37)

Error 1: ProcsPerNode (28) must be evenly divisible by NumWorkers (3)

## Requesting > 28 workers

- Request multiples of 28 workers

```
>> job = c.batch(@pwd,1,{}, 'CurrentFolder','.', 'Pool', 55);  
additionalSubmitArgs =  
    '--ntasks=56 -N 2 -c 1 --ntasks-per-core=1 -t 00:10:00 -p hamsi'  
>>  
>>
```

```
>> job = c.batch(@pwd,1,{}, 'CurrentFolder','.', 'Pool', 30);
```

Error using parallel.Cluster/batch (line 158)

Job submission failed because the plugin function  
'communicatingSubmitFcn.m' errored.

Caused by:

Error using getCommonSubmitArgs (line 42)

Error 2: NumWorkers (31) \* NumThreads (1) must be evenly divisible by

## Fetch the results

```
>> % Submit calc_pi job
>> c = parcluster;
>>
>> % Request 3 workers
>> job = c.batch(@calc_pi, 0, {}, 'CurrentFolder', '.', 'Pool', 3);
additionalSubmitArgs =
    '--ntasks=4 -N 1 -c 7 --ntasks-per-core=1 -t 00:10:00 -p hamsi'
>>
>> % Check the state of the job
>> job.State
ans =
    'finished'
>>
>> % Fetch the results
>> job.fetchOutputs{:}
>>
```

“Where’s the  
output?”



# What gets “returned”?

- Function output
- Diary
- Saved files

## Example

```
function [t, A] = test_fcn(sims)

disp('Start sim')

t0 = tic;
parfor idx = 1:sims
    A(idx) = idx;
    pause(0.5)
    idx
end
t = toc(t0);

disp('Finished')

save RESULTS A
```

## Job submission

```
>> job = c.batch(@test_fcn,1,{300}, 'CurrentFolder','.', 'Pool',6);  
additionalSubmitArgs =  
    '--ntasks=7 -N 1 -c 4 --ntasks-per-core=1 -t 00:10:00 -p hamsi'
```



## Fetch output

```
function [t, A] = test_fcn(sims)  
c.batch(@test_fcn, 1, {300},
```

```
>> % Fetch the results  
>> job.fetchOutputs{:}  
ans =  
  
26.77
```

“Where’s A?”



# Diary

```
>> % View the diary
```

```
>> job.diary
```

```
--- Start Diary ---
```

```
Start sim
```

```
ans =
```

```
4
```

```
ans =
```

```
5
```

```
ans =
```

```
8
```

```
function [t, A] = test_fcn(sims)

disp('Start sim')

t0 = tic;
parfor idx = 1:sims
    A(idx) = idx;
    pause(0.5)
    idx
end
t = toc(t0);

disp('Finished')

save RESULTS A
```

# Save files

“Where does **RESULTS**  
get written to?”



```
function [t, A] = test_fcn(sims)

disp('Start sim')

t0 = tic;
parfor idx = 1:sims
    A(idx) = idx;
    pause(0.5)
    idx
end
t = toc(t0);

disp('Finished')

save RESULTS A
```

## Other settable job properties (1)

```
>> c.AdditionalProperties
ans =
    AdditionalProperties with properties:

        AccountName: ''
    AdditionalSubmitArgs: ''
        ClusterHost: '172.16.7.1'
        EmailAddress: ''
        EnableDebug: 0
        ProcsPerNode: 28.00
        QueueName: 'hamsi'
    RemoteJobStorageLocation: ''
        RequireExclusiveNode: 0
        Reservation: ''
        UseIdentityFile: 0
        UseSmpd: 0
        Username: ''
        WallTime: '00:10:00'
```

## Other settable job properties (2)

- AccountName
- EmailAddress
- Reservation
- **WallTime**

# Submitting scripts, instead of functions

```
>> x = 4;
>> z = rand(3);
>>
>> % Submit a script (instead of a function)
>> job = c.batch('temp = rand(10); y = x, who', 'CurrentFolder', '.');
additionalSubmitArgs =
    '--ntasks=1 -N 1 -c 28 --ntasks-per-core=1 -t 00:10:00 -p hamsi'
>> clear z
>>
>> who
```

Your variables are:

```
c    job    x
```

```
>> % Check the state of the job
>> job.State
ans =
    'finished'
```

# Loading variables to local workspace

“If we cleared `z`, then why does `who` display it?  
And I didn’t need `temp`!”



```
>> % Load variables
>> job.load
>> who
```

Your variables are:

```
ans    c    job    temp    x    y
```

```
>> job.diary
--- Start Diary ---
```

```
y =
```

```
4
```

Your variables are:

```
temp    x    y    z
```

```
--- End Diary ---
```

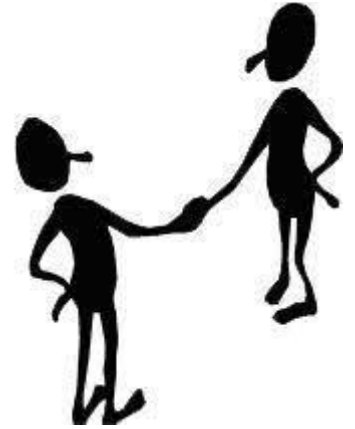
“I’ll pass all the variables in your local workspace to all of the workers. Then I’ll pass everything the workers generate and pass it back to your local workspace.”





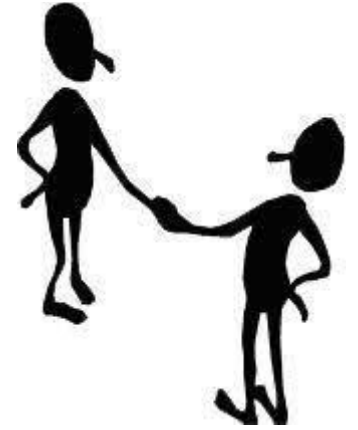
## Adding files to the job

- **AdditionalPaths**
  - List absolute paths on the cluster (include subdirectories if needed)
- **AttachedFiles**
  - List files not automatically added to the job (e.g., binary files)
- **AutoAddClientPath**
  - Set to false if you have added your own local paths to the MATLAB client path
- **AutoAttachFiles**
  - Useful for small number, often changing files



## Should I send files with the job?

- By default, each job will copy all required files
  - How many jobs are you going to submit?
  - How large, in totality, are your files?
  - Do your files change a lot?



5-10 files  
50-100 KB  
Add to job

40-100 files  
250 MB - 5 GB  
A Hach on cluster

## When has my job run and finished?

- To be used sparingly, ... (can have impact on network)



```
>> % Get email notification when my job has finished running
>> c.AdditionalProperties.EmailAddress = 'user-id@tubitak.gov.tr';
>> job = c.batch(@test_fcn,1,{300}, 'CurrentFolder', '.', 'Pool',6);
additionalSubmitArgs =
    '--ntasks=7 -N 1 -c 4 --ntasks-per-core=1 -t 00:10:00 -p hamsi --mail-type=ALL
```

# Retrieving past jobs

Preferences
 Set Path
 Add-Ons
 Help
 Community
 Request Support
 Learn MATLAB
 **RESOURCES**

**Parallel**

Select a Default Cluster >

Discover Clusters...
Create and Manage Clusters...
Monitor Jobs
Parallel Preferences...

Job Monitor

Select Profile: truba R2021b (default) ☐ Show jobs from all users

ID	Username	Submit Time	Finish Time	Tasks	State	Description
9511	rayn	Wed Jul 13 18:31:57 EDT 2022	Wed Jul 13 18:33:54 EDT 2022	4	finished	Batch job running function
9525	rayn	Wed Jul 13 18:33:31 EDT 2022	Wed Jul 13 18:34:35 EDT 2022	4	finished	Batch job running function
9526	rayn	Wed Jul 13 18:33:48 EDT 2022	Wed Jul 13 18:34:28 EDT 2022	7	finished	Batch job running function
9529	rayn	Wed Jul 13 18:33:59 EDT 2022	Wed Jul 13 18:35:24 EDT 2022	7	finished	Batch job running function
9553	rayn	Wed Jul 13 18:35:17 EDT 2022	Wed Jul 13 18:36:41 EDT 2022	7	finished	Batch job running function
9554	rayn	Wed Jul 13 18:35:47 EDT 2022	Wed Jul 13 18:36:30 EDT 2022	14	finished	Batch job running function
9555	rayn	Wed Jul 13 18:35:56 EDT 2022	Wed Jul 13 18:36:48 EDT 2022	14	finished	Batch job running function
9557	rayn	Wed Jul 13 18:36:05 EDT 2022	Wed Jul 13 18:37:21 EDT 2022	14	finished	Batch job running function
9567	rayn	Wed Jul 13 18:36:26 EDT 2022	Wed Jul 13 18:38:09 EDT 2022	14	finished	Batch job running function
9581	rayn	Wed Jul 13 18:37:53 EDT 2022	Wed Jul 13 18:39:39 EDT 2022	14	finished	Batch job running function
9582	rayn	Wed Jul 13 18:38:46 EDT 2022		28	failed	Batch job running function
9583	rayn	Wed Jul 13 18:38:55 EDT 2022	Wed Jul 13 18:41:19 EDT 2022	28	finished	Batch job running function
9585	rayn	Wed Jul 13 18:39:05 EDT 2022	Wed Jul 13 18:40:07 EDT 2022	28	finished	Batch job running function
9588	rayn	Wed Jul 13 18:39:21 EDT 2022	Wed Jul 13 18:41:25 EDT 2022	28	finished	Batch job running function
9595	rayn	Wed Jul 13 18:40:57 EDT 2022	Wed Jul 13 18:43:00 EDT 2022	28	finished	Batch job running function
9609	rayn	Wed Jul 13 18:42:34 EDT 2022	Wed Jul 13 18:44:59 EDT 2022	28	finished	Batch job running function
9610	rayn	Wed Jul 13 18:43:29 EDT 2022		56	failed	Batch job running function
9611	rayn	Wed Jul 13 18:43:37 EDT 2022	Thu Jul 14 00:30:06 EDT 2022	56	finished	Batch job running function
9613	rayn	Wed Jul 13 18:43:47 EDT 2022	Thu Jul 14 00:30:18 EDT 2022	56	finished	Batch job running function
9616	rayn	Wed Jul 13 18:45:27 EDT 2022	Thu Jul 14 00:30:32 EDT 2022	56	finished	Batch job running function
9623	rayn	Wed Jul 13 18:46:53 EDT 2022	Thu Jul 14 00:30:45 EDT 2022	56	finished	Batch job running function
9637	rayn	Wed Jul 13 18:48:17 EDT 2022	Thu Jul 14 01:27:28 EDT 2022	56	finished	Batch job running function
9638	rayn	Wed Jul 13 20:07:34 EDT 2022	Wed Jul 13 20:14:00 EDT 2022	1	finished	Batch job running function
9639	rayn	Wed Jul 13 20:39:11 EDT 2022	Wed Jul 13 20:41:44 EDT 2022	1	finished	Batch job running function
9642	rayn	Wed Jul 13 20:52:24 EDT 2022	Wed Jul 13 20:53:25 EDT 2022	1	finished	Batch job running function
9644	rayn	Wed Jul 13 21:00:45 EDT 2022		28	failed	Batch job running function
9645	rayn	Wed Jul 13 21:05:14 EDT 2022		28	failed	Batch job running function
9646	rayn	Wed Jul 13 21:15:14 EDT 2022	Wed Jul 13 21:16:47 EDT 2022	4	finished	Batch job running function
9647	rayn	Wed Jul 13 21:50:24 EDT 2022	Wed Jul 13 21:51:54 EDT 2022	4	finished	Batch job running function
9648	rayn	Wed Jul 13 22:00:55 EDT 2022	Wed Jul 13 22:02:22 EDT 2022	7	finished	Batch job running function
9649	rayn	Wed Jul 13 22:20:36 EDT 2022	Wed Jul 13 22:24:19 EDT 2022	7	finished	Batch job running function
9650	rayn	Wed Jul 13 22:38:37 EDT 2022	Wed Jul 13 22:39:56 EDT 2022	1	finished	Batch job running script
9652	rayn	Thu Jul 14 05:24:11 EDT 2022	Thu Jul 14 05:24:27 EDT 2022	1	finished	Batch job running function
9653	rayn	Thu Jul 14 05:24:47 EDT 2022		1	queued	Batch job running function
9654	rayn	Thu Jul 14 05:25:09 EDT 2022	Thu Jul 14 05:32:40 EDT 2022	1	finished	Batch job running function
9655	rayn	Thu Jul 14 05:26:10 EDT 2022		1	queued	Batch job running function
9656	rayn	Thu Jul 14 05:26:35 EDT 2022	Thu Jul 14 05:32:46 EDT 2022	1	finished	Batch job running function
9657	rayn	Thu Jul 14 05:28:45 EDT 2022	Thu Jul 14 05:31:51 EDT 2022	1	finished	Batch job running function
9658	rayn	Thu Jul 14 05:30:05 EDT 2022	Thu Jul 14 05:32:52 EDT 2022	1	finished	Batch job running script
9659	rayn	Thu Jul 14 05:48:39 EDT 2022	Thu Jul 14 05:50:09 EDT 2022	1	finished	Batch job running script
9660	rayn	Thu Jul 14 05:52:58 EDT 2022	Thu Jul 14 05:56:04 EDT 2022	1	finished	Batch job running function
9661	rayn	Thu Jul 14 05:58:58 EDT 2022	Thu Jul 14 06:00:11 EDT 2022	7	finished	Batch job running function
9662	rayn	Thu Jul 14 06:04:09 EDT 2022		7	running	Batch job running function

Last updated at Thu Jul 14 06:03:44 EDT 2022

Auto update: Every 5 min

Cancel
Delete
Show Details
Show Errors
Show Warnings
Show Diary
Fetch Outputs

## Keep cluster clean: delete jobs

- As a good practice, delete jobs you no longer need anymore

```
>> % Finished with the job, delete it to cleanup jobs  
>> job.delete
```

# Debugging and Troubleshooting



## Scheduler ID

```
>> job = c.batch(@pwd,1,{}, 'CurrentFolder','.');  
additionalSubmitArgs =  
    '--ntasks=1 -N 1 -c 28 --ntasks-per-core=1 -t 00:10:00 -p hamsi'  
>>  
>> % Job ID vs Scheduler ID  
>> job.ID  
ans =  
    9656  
>>  
>> job.getTaskSchedulerIDs{1}  
ans =  
    '208426'
```

## Example: Errored jobs

```
>> % Undefined function
>> job = c.batch(@invalid_fcn,1,{}, 'CurrentFolder','.');
additionalSubmitArgs =
    '--ntasks=1 -N 1 -c 28 --ntasks-per-core=1 -t 00:10:00 -p hamsi'
>>
>> % Undefined variable in a script
>> job2 = c.batch('x = y', 'CurrentFolder','.');
additionalSubmitArgs =
    '--ntasks=1 -N 1 -c 28 --ntasks-per-core=1 -t 00:10:00 -p hamsi'
```



## Errored jobs (1)

```
>> % Undefined function
```

```
>> job.State
```

```
ans =
```

```
    'finished'
```

```
>>
```

```
>> job.fetchOutputs{:}
```

```
Error using parallel.Job/fetchOutputs (line 1300)
```

```
An error occurred during execution of Task with ID 1.
```

```
Caused by:
```

```
    Unrecognized function or variable 'invalid_fcn'.
```

## Errored jobs (2)

```
>> % Undefined variable in a script
```

```
>> job2.load
```

```
Error using parallel.Job/load (line 36)
```

```
Error encountered while running the batch job. The error was:
```

```
Unrecognized function or variable 'y'.
```

If 'y' is a file that is required by 'x = y', add the full path name for 'y' to the batch job's AttachedFiles property. For more information, see [batch](#).

## Logfile: Single core job

```
>> job = c.batch(@test_fcn,1,{300}, 'CurrentFolder','.');  
additionalSubmitArgs =  
    '--ntasks=1 -N 1 -c 28 --ntasks-per-core=1 -t 00:10:00 -p hamsi'  
>>  
>> % Retrieve log file for single core job  
>> c.getDebugLog(job.Tasks)  
LOG FILE OUTPUT:  
Executing: /truba/sw/centos7.9/app/matlab/r2021b/bin/worker  
Exiting with code: 0
```

## Logfile: Multi-core job

```
>> job = c.batch(@test_fcn,1,{297}, 'CurrentFolder','.', 'Pool',27);
additionalSubmitArgs =
    '--ntasks=28 -N 1 -c 1 --ntasks-per-core=1 -t 00:10:00 -p hamsi'
>>
>> % Retrieve log file for multi-core job
>> c.getDebugLog(job)
LOG FILE OUTPUT:
The scheduler has allocated the following nodes to this job:
hamsi37
"/truba/sw/centos7.9/app/matlab/r2021b/bin/mw_mpiexec" -bind-to core -l -n 28 "/truba
[0] Sending a stop signal to all the labs...
[0] Parallel pool is shutting down.[0]
Exiting with code: 0
```

## WORKSHOP-ONLY: Remove reservation

```
>> % Remove reservation from future jobs  
>> c.AdditionalProperties.Reservation = '';  
>> c.saveProfile
```

# From Coding to Cluster (1)

```
% Notes - From Coding to Cluster
% 1. Using a script, not a function
% 2. Paths are hardcoded
% 3. File separator is hard coded
% 4. Assumes TIF file exists
% 5. TIF files must be on the MATLAB path
% 6. Assumes output folder already exists where ever MATLAB is running
% 7. Results MAT-File will be overwritten next time it's run
% 8. Changes MATLAB working directory
```

```
filelist = dir('tif\*.tif');
fileNames = {filelist.name}';
```

```
segmentedCellSequence = batchProcessFiles(@detectCells,fileNames);
cd output
save SCS segmentedCellSequence
```

```
function [ofile, segmentedCellSequence] = process_files_v2(rootd,outd)
if nargin==0
    rootd = fullfile(pwd,'tif');
    outd = fullfile(pwd,'output');
end

filelist = dir(fullfile(rootd,'*.tif'));
if isempty(filelist)
    error('Failed to find image files: %s',rootd)
end
fileNames = {filelist.name}';

addpath(rootd)
segmentedCellSequence = batchProcessFiles(@detectCells,fileNames);

% Ensure output directory exists
if exist(outd,'dir')==false
    [FAILED,msg,eid] = mkdir(outd);
    if FAILED==true
        error(eid,msg)
    end
end

% Add timestamp for file uniqueness
ts = strrep(strrep(datestr(now),' ','_'),':','-');

% Save dir
old_dir = pwd;
c = onCleanup(@() cd(old_dir));
cd(outd)
ofile = ['SCS_' ts];
save(ofile,'segmentedCellSequence')
```

## Run it locally

```
>> % Start local parallel pool
>> parpool(4);
Starting parallel pool (parpool) using the 'local' profile ...
Connected to the parallel pool (number of workers: 4).
>>
>> % Call the function locally
>> ofile = process_files_v2

ofile =

    'S:\sandbox\Workshops\Parallel-Computing-Workshop\matlab-workshop-files'

>>
```



# Run it on the cluster

```
>> % Submit job to cluster
>> c = parcluster;
>> j = c.batch(@process_files_v2, 1, {'/work/raymond/proj-tiffs', '/home/raymond/output-results'}, 'Pool', 3);
>>
>> % Wait for job to finish
>> j.wait
>>
>> % Fetch the results
>> ofile = j.fetchOutputs{:}

ofile =

    '/home/raymond/output-results/SCS_27-Apr-2021_16-54-28'

>>
```

## From Coding to Cluster (2)

```
% Notes - From Coding to Cluster
% 1. Using a script, not a function
%     return status or output directory
% 2. Paths are hardcoded
%     pass in root directory
% 3. File separator is hard coded
%     use fullfile
% 4. Assumes TIF file exists
%     check results when touching the file system
% 5. TIF files must be on the MATLAB path
%     add tif folder to the MATLAB path
% 6. Assumes output folder already exists where ever MATLAB is running
%     supply output directory to write to.  check if folder exists; if
%     not, create it
% 7. Results MAT-File will be overwritten next time it's run
%     add timestamp to filename
% 8. Changes MATLAB working directory
%     Track old directory, change back before leaving
```



## Running bulk jobs

```
function jobs = submit_jobs

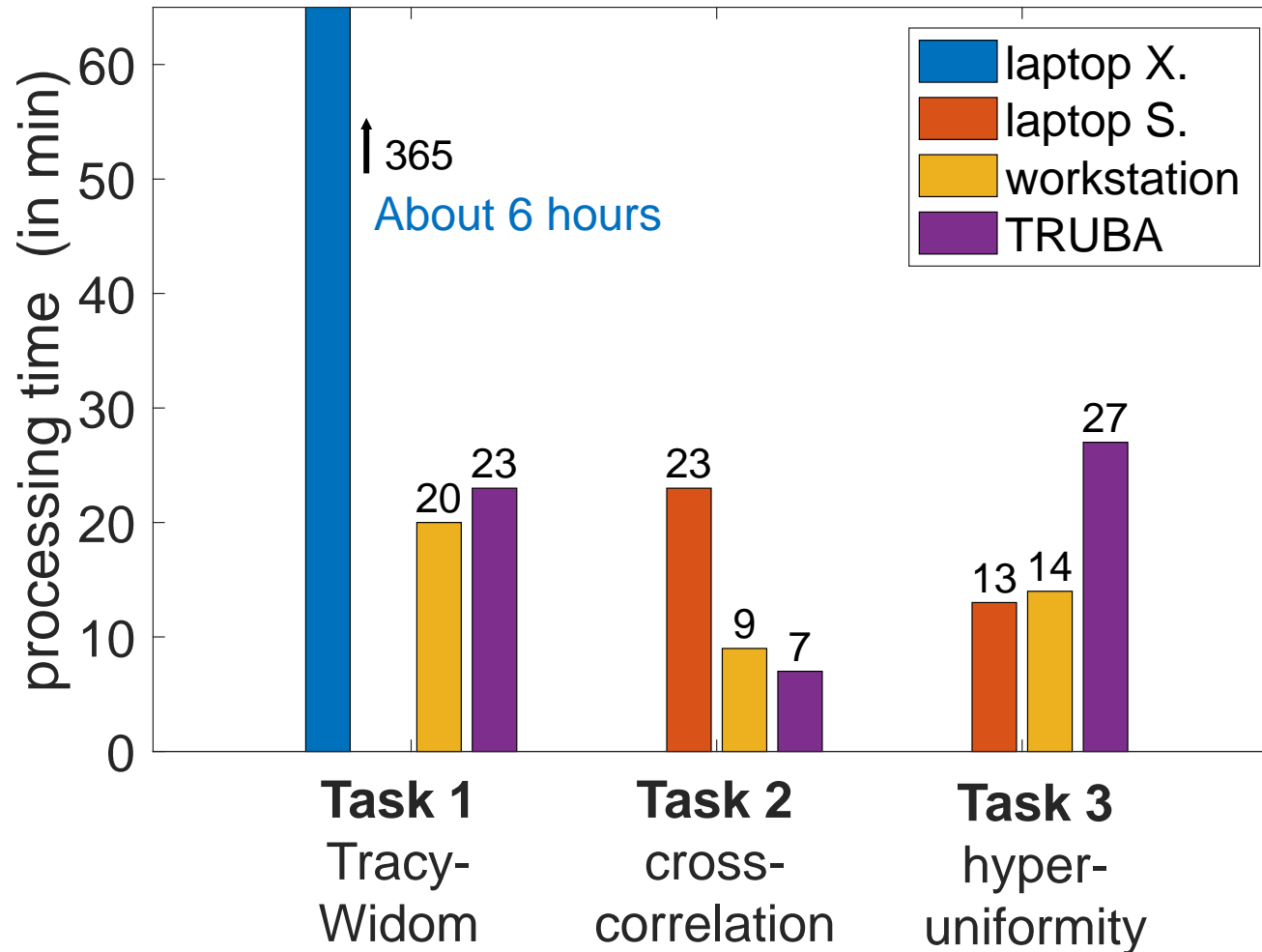
c = parcluster;
c.AdditionalProperties.EmailAddress = 'my-email@work';

sims = [54 162 324 648];

for idx = 1:length(sims)
    % Run code with different number of iterations
    jobs(idx) = c.batch(@parallel_example,1,{sims(idx)}, 'Pool',3);
end

% Wait for the 2nd job to finish
jobs(2).wait

t = jobs(2).fetchOutputs{:}
```



machine	cores	RAM	Resources
laptop X.	2	16 GB	single-user
laptop S.	8	32 GB	single-user
workstation	32	128 GB	multi-user
TRUBA node	28	±100 GB	job queue

same nr. of workers  
26 workers : 13 x 2 threads + 1

80% visualization → no speed-up

# Workstation or TRUBA?

- Workstation

- Plusses

- “Immediate” access
    - Easier MATLAB licensing

- Deltas

- Cost and procurement of each workstation
    - Waste when it’s not being used
    - 1 job at a time

- TRUBA

- Plusses

- An abundant of compute resources
    - Able to run a collection of jobs at once

- Deltas

- Each node is a bit less powerful than a workstation
    - Need to configure MATLAB licensing
    - Transferring files/data to TRUBA

# How do I get started?

- Need to point MATLAB Parallel Server to your own university license
- Contact
  - Sefa Arslan
  - [sefa.arslan@tubitak.gov.tr](mailto:sefa.arslan@tubitak.gov.tr)

“How do I license  
MATLAB Parallel Server?”



## Summary – Remote Submission

- Call `configCluster` (once) to create TRUBA profile
- Toggle between local profile (desktop machine) and TRUBA profile (multi-node)
- Tuning your job with `AdditionalProperties`
- Best practices for job submission and troubleshooting
- Contact Tubitak to get started