Homework #2

- [x] Green’s function discretization.
- [x] Simpson’s rule trick
- [x] Binary output - easier to read in?
- [x] x-ticks when creating scaling plots
- [x] Verification - convergence plots
- [x] Error verses time plot
- [x] Put everything in one PDF

Numerical integration

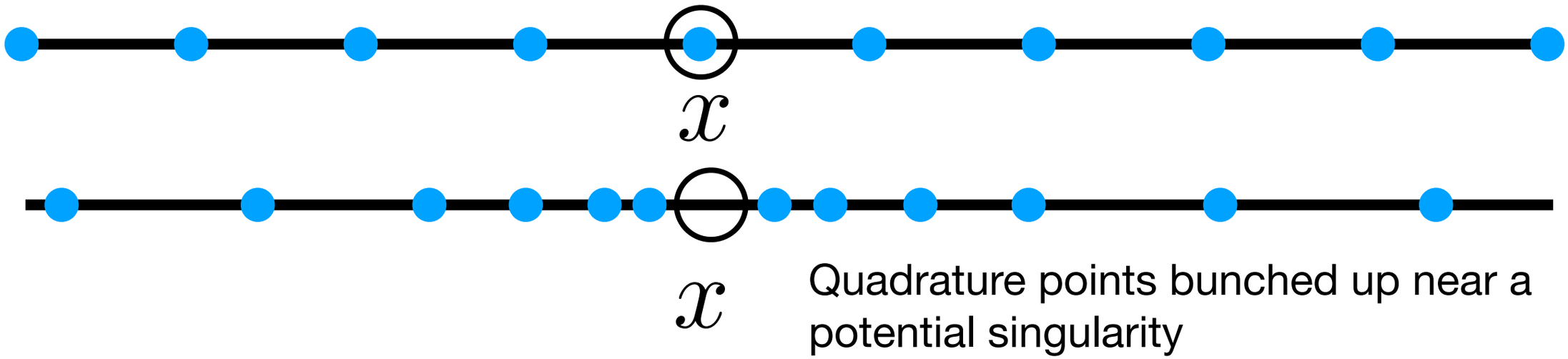
$$u(x) = a + (b-a)x + \int_0^1 G(x,t) f(t)\ dt$$

In 2d :

$$G(\mathbf{r}) = -\log(\mathbf{r})$$

This works in the present case because

- G(x,t) remains finite at t=x
- Trapezoidal rule and Simpson rule are on equally spaced meshes
- We are evaluating u at one of the points used in the quadrature rule.



Quadrature points bunched up near a potential singularity

Simpson's Rule trick

$$T(f) = \frac{f(a) + f(b)}{2} h$$ Trapezoidal Rule

$$M(f) = hf(c), \qquad c = \frac{a+b}{2}$$ Midpoint Rule

Only one code is needed!

Combine to get Simpsons' Rule :

$$S(f) = \frac{f(a) + 4f(c) + f(b)}{6} h = \frac{T(f) + 2M(f)}{3}$$

Compute T(f) on ● Compute M(f) on ○

Binary output

In C

```
/* Write out binary data for reading in Python */
if (my_rank == 0)
{
    FILE *fout = fopen("prob3.out","w");
    fwrite(&N,1,sizeof(int),fout);
    fwrite(err,3,sizeof(double),fout);
    fclose(fout);
}
```

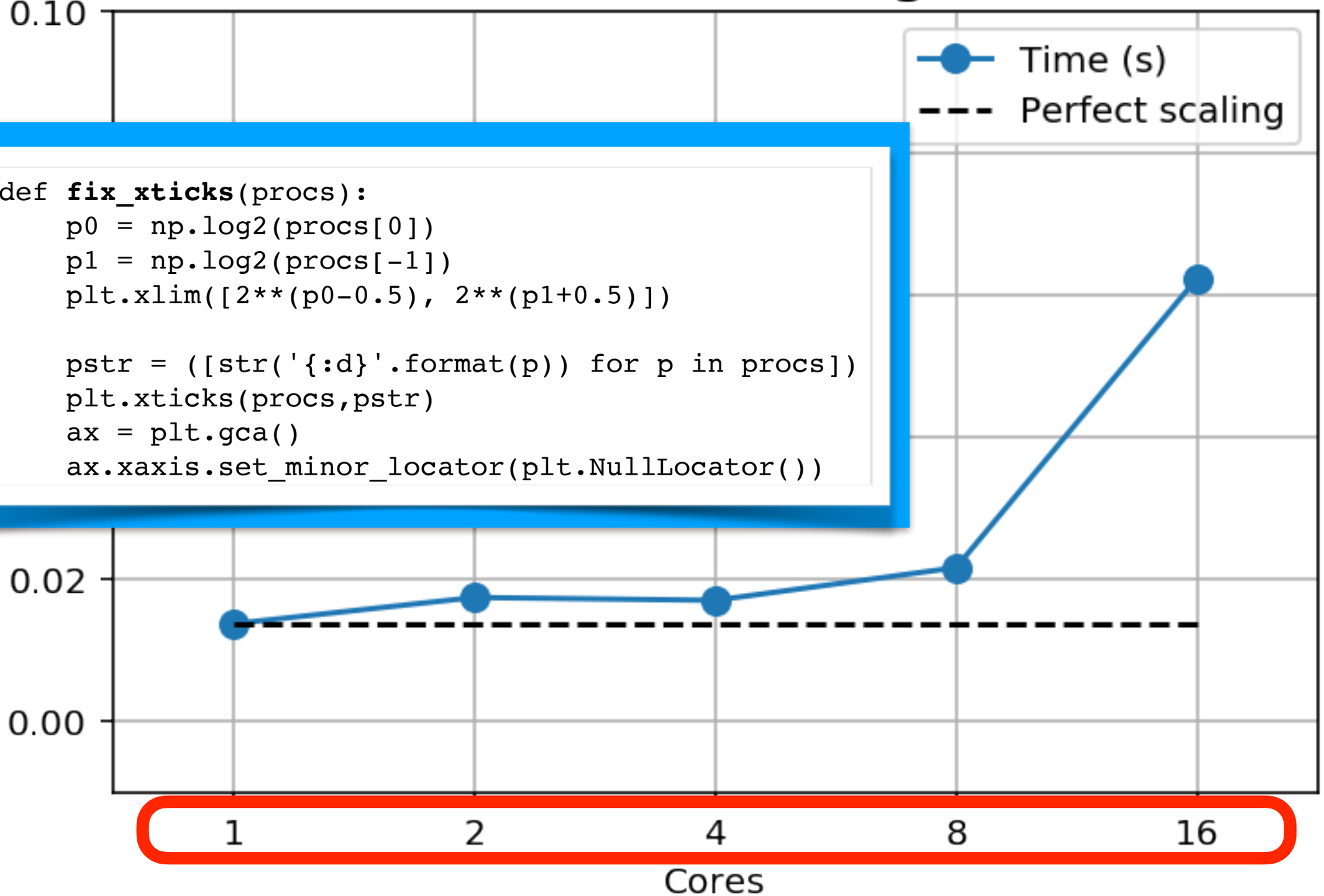
In Python

```
# Read binary output
dt = dtype([('N','int32'), ('err','d',3)])
fout = open(output_fname,"rb")
N,err = fromfile(fout,dtype=dt,count=1)[0]
fout.close()
```

See notebook **create_data.ipynb**

Weak scaling

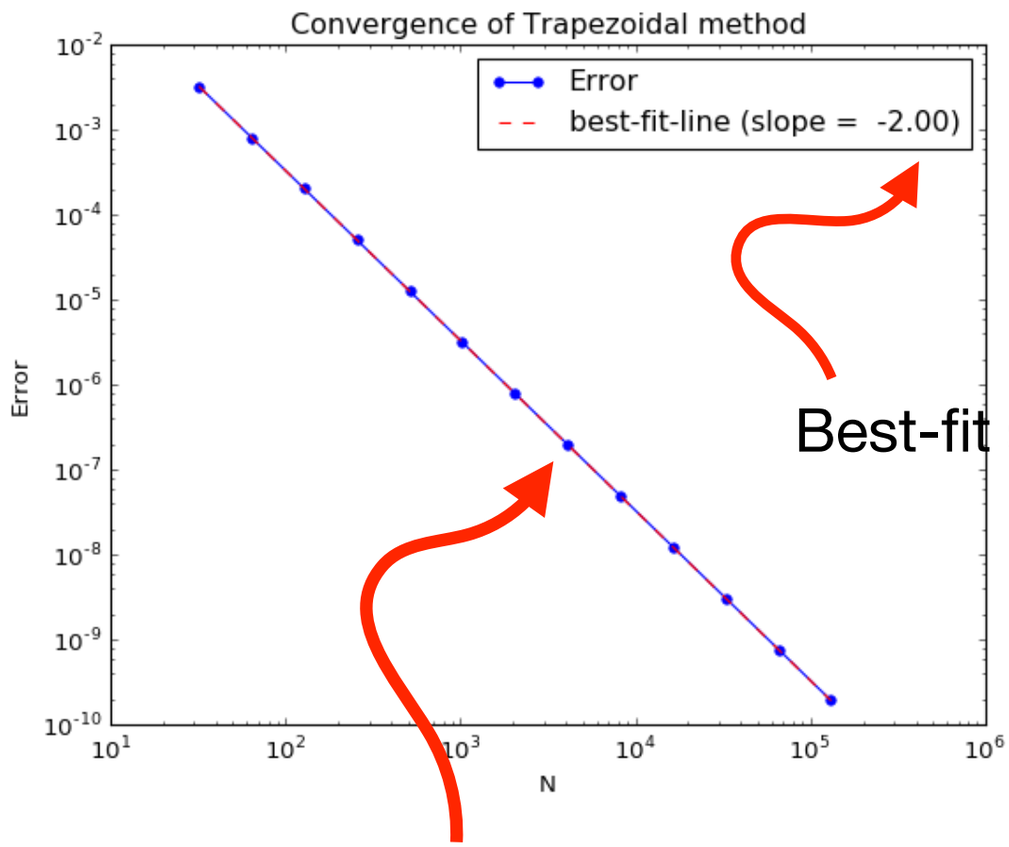
```
def fix_xticks(procs):
    p0 = np.log2(procs[0])
    p1 = np.log2(procs[-1])
    plt.xlim([2**(p0-0.5), 2**(p1+0.5)])

    pstr = ([str('{:d}'.format(p)) for p in procs])
    plt.xticks(procs,pstr)
    ax = plt.gca()
    ax.xaxis.set_minor_locator(plt.NullLocator())
```



See **scaling.ipynb**

Verification

- **Verification.** When solving PDEs, this typically involves doing a convergence study to show that expected convergence results are achieved.
- **Slope** of the line is the *numerical order of convergence.*
- *Not all data points need to be used*



Best-fit slope shown

Discrete data points shown

- Axis labels, and a title
- Both x and y axes use *log coordinates*

See **accuracy.ipynb**

Convergence

Greens (o=2, slope = -2.00)
Greens (o=4, slope = -4.01)
FD (o=2, slope = -2.00)
FD (o=4, slope = -4.04)
Machine precison

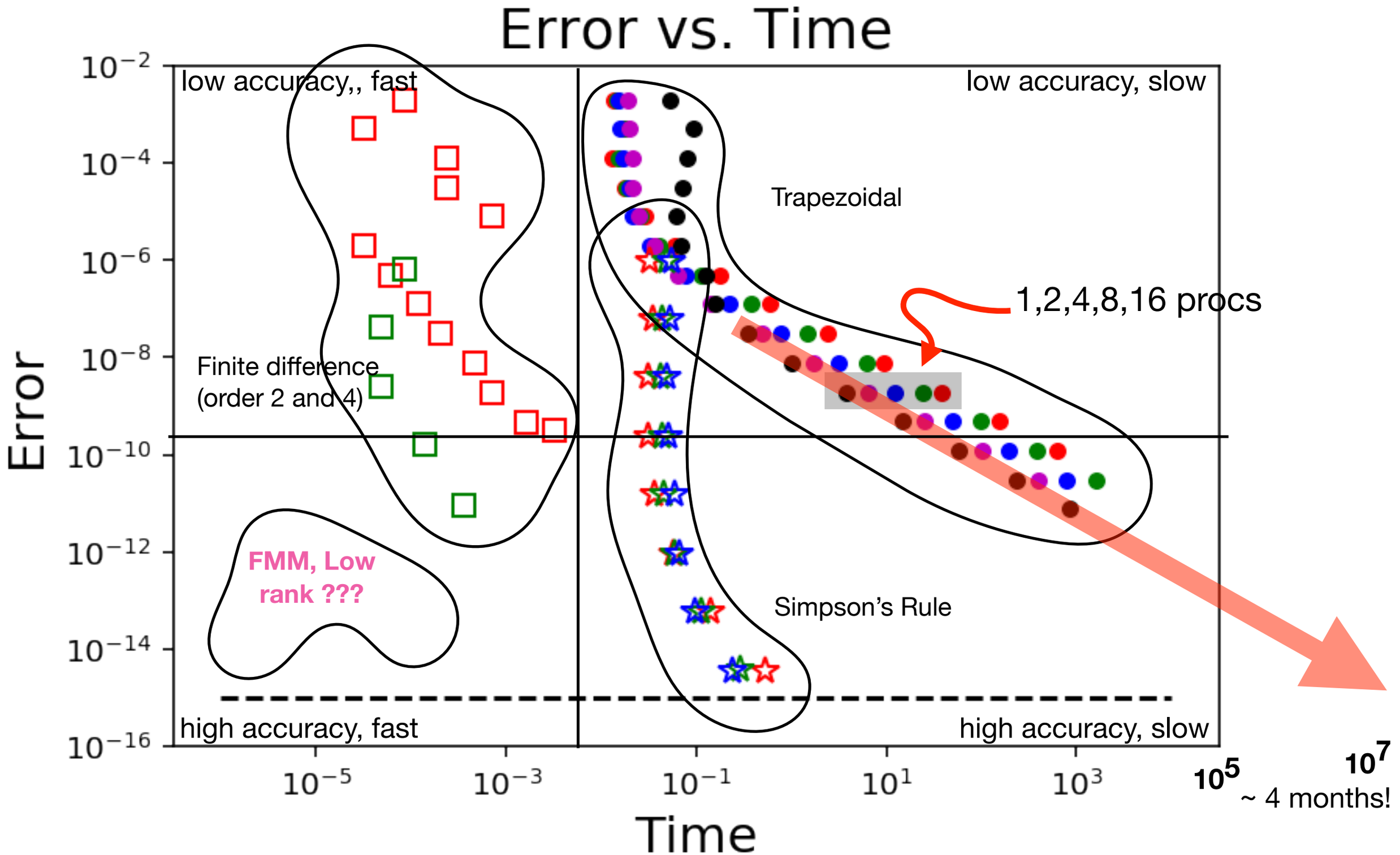
Error

10^{1}
10^{-2}
10^{-5}
10^{-8}
10^{-11}
10^{-14}
10^{-17}

10^{2}
10^{3}
10^{4}
10^{5}
10^{6}
10^{7}

N

$\mathrm{O(N)}$ $O(N^2)$



MPI Routines discussed so far

Communication

- MPI_Send, MPI_Recv
- MPI_Bcast
- MPI_Scatter
- MPI_Gather, MPI_Allgather
- MPI_Reduce, MPI_Allreduce

Data handling

- MPI_Type_create_struct
- MPI_Type_create_subarray
- MPI_Type_extent
- MPI_Aint
- MPI_Type_free
- MPI_Type_commit

Parallel I/O

- MPI_File_set_view
- MPI_File_write_all
- MPI_File_open, MPI_File_close
- ...