

Scientific Application Webserver

<https://bitbucket.org/SAWs/saws>

- Turns any C/C++ or Fortran program into a webserver that publishes variables, images etc to the web
- Uses standard HTTP and JSON to communicate with a browser
- GUIs are programmed with standard Javascript
- Powered by Mongoose (<http://code.google.com/p/mongoose/>) as its embedded webserver and cJSON (<http://sourceforge.net/projects/cjson>) for parsing JSON in C.

Example use of SAWs in Browser

localhost:8080

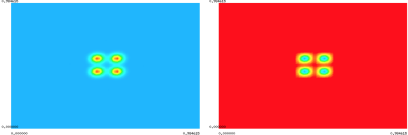
PETSc Application Web server powered by SAWs

This is the default PETSc application dashboard, from it you can access any published PETSc objects or logging data

Petsc Development GIT revision: v3.6.1-410-ga1f63b8 GIT Date: configured with --download-saws --download-sowing=0 --with-afterimage --with-fc=0 --with-mpi=0 PETSC_ARCH=arch-saws

Running `/ex5 -draw_save -draw_save_single_file -malloc_test -saws_root . -stack_view saws -ts_monitor_draw_solution -ts_view_pre saws -x_virtual`

Demonstrates Pattern Formation with Reaction-Diffusion Equations.



Update all variables from server Update server with changes below

PETSc Stack

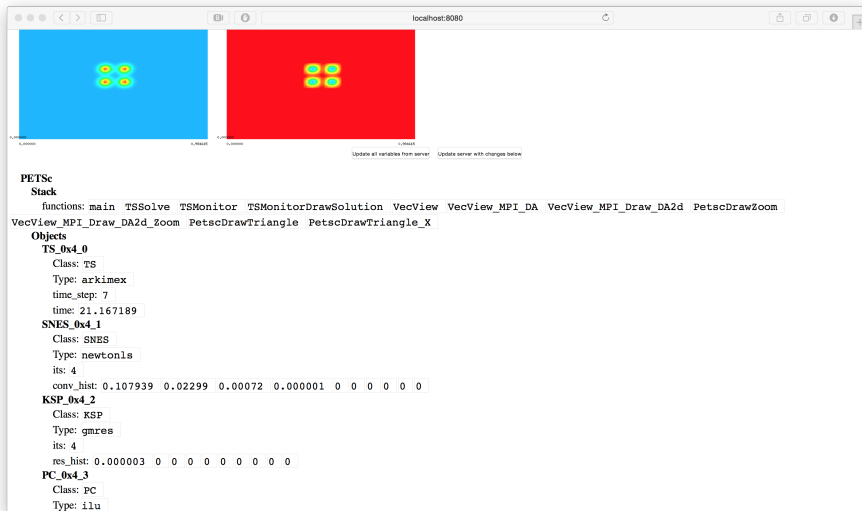
functions: main TSSolve TSMonitor TSMonitorDrawSolution VecView VecView_MPI_DA VecView_MPI_Draw_DA2d PetscDrawzoom VecView_MPI_Draw_DA2d_Zoom PetscDrawTriangle PetscDrawTriangle_X

Objects

TS_0x4_0
Class: TS
Type: arkimex
time_step: 7
time: 21.167189

SNES_0x4_1
Class: SNES

Example use of SAWs in Browser



Sample Code

Serving a variable

```
SAWs_Initialize();  
SAWs_Register("/Time",&t,1,SAWs_READ,SAWs_DOUBLE);  
sleep(60);
```

Accessing the variable

```
http://hostname:8080/SAWs/Time
```

Output as JSON:

```
{
  "directories": {
    "SAWS_ROOT_DIRECTORY": {
      "variables": {
        "Time": {
          "data": [3.140000],
          "dtype": "SAWS_DOUBLE",
          "length": "1",
          "mtype": "SAWS_READ",
          "alternatives": []
        }
      }
    }
  }
}
```

Output in GUI:

