Improving propagators and perturbations account in Poliastro

- **Personal information**:
  1. Nikita Astrakhantsev
  2. 2011-2017 BSc. and MSc. at Moscow institute of physics and technology and Institute for theoretical and experimental physics, department: of applied and general physics.
  3. 2015-2017 MSc. at Yandex School of Data Analysis (YSDA).
  4. **Current status**: 2017-... PhD student at Moscow institute of physics and technology and Institute for theoretical and experimental physics.
     - Area of research: numerical methods (simulations) in strongly correlated systems.
  5. **Background:** Since 2013 I work as a researcher within the ITEP Lattice Group. I do lattice QCD and graphene supercomputer simulations, it requires knowledge of computer science, high energy physics and statistical methods. We create parallel code for our cluster, use physics and statistics to interpret the data. I have 5 papers so far.
  6. **Contacts**: nikita.astronaut@gmail.com, nikita.astronaut at matrix.org, nikita-astronaut on github.

- **Programming and other**:
  - Three merged pull requests to Poliastro (# 322, # 320 and # 334).
  - **Computer skills**: experienced Linux user. Version control: Git. Languages: C, C++, Python. Parallel programming: CUDA. Knowledge of fundamental algorithms and data structures (see YSDA sertificate and my github), deep learning frameworks: theano/tensorflow (keras), computer algebra: Mathematica.
  - **Physics and math**: advanced calculus, complex variable analysis, statistics, numerical analysis, general/theoretical physics, high energy physics.
  - **Data analysis**: image processing, deep learning frameworks, statistics, bayesian methods. A golden and a silver medals on kaggle competitions. See my github and kaggle account.
  - **English**: TOEFL score is 109 out of 120, speak and write fluently.
  - **Page-proofs**: master of LaTeX.

- **About project**
  - **Short project description**: The project is aimed at improvement of currently implemented Poliastro propagators and also implementation of various perturbation effects such as J2-coefficients, Solar pressure drag, low-thrust orbits.
  - **My qualification for the project**: My current research as a PhD student is connected to use of numerical methods (solving PDEs), my major at the university was applied math and physics. I also have a long programming experience starting from my school times and participation in programming olympiads.
  - **Why it is interesting to me**: I do really love physics and writing an open-source software on space-dynamics is almost a dream to me. As a part of physics, this is not a very difficult problem theoretically, but still very important for people who study space and the outcome is easy to understand. I believe I will manage to make a big contribution to the project and after that feel that I helped people using this software.

- **My commitment:** As a participant of GSoC, I will work not less than 40 hours a week and I believe that all the primary goals of the project will be achieved (and, hopefully, all the secondary, too).

- **Detailed project timeline:**
  - **Community bonding period**: As we discussed with mentor, the next release of Poliastro and its presentation on the «Python for Astronomy» workshop in NY will happen at the end of April. So, during this period I will start working on my summer plan so that some of the goals were already completed and shown at the workshop.
  - **1st evaluation (propagators improvement)**:
    1. investigate `kepler` propagator long-time convergence problems and cure them if possible (primary)
    2. compare long-time propagation of `kepler` (if cured) and `mean_motion` propagators with the the analytical expressions for orbits (primary)
    3. rewrite `cowell` propagator with new `SciPy` ODE solvers (primary)
    4. put all code and documentation on GitHub
  - **2nd evaluation (working on perturbations)**:
    1. adding `J2` propagator accounting for obliteness of the attactor (primary)
    2. adding `J4` and `J6` propagators (secondary)
    3. adding third body perturbations: moon for anything orbiting Earth, giant planets (for instance, Jupiter) for anything orbiting Sun (primary)
    4. move solar drag and solar pressure from `https://groups.io/g/poliastro-dev/message/27` (primary)
    5. move low-thrust trajectories from `https://github.com/juanlu001/pfc-uc3m` to poliastro (secondary)
    6. put all code and documentation on GitHub
  - **3rd evaluation (propagators benchmarking)**:
    1. implement propagator from `https://link.springer.com/article/10.1007/BF01238923` and `https://link.springer.com/article/10.1007/s10569-013-9476-9` (primary)
    2. benchmark these algorithms and already implemented propagators in terms of memory usage, iterations needed e t.c. for various `ecc` (primary)
    3. write a beautiful ipython notebook demonstrating performance change during variation of `ecc` (primary)
    4. put all code, documentation and examples on GitHub