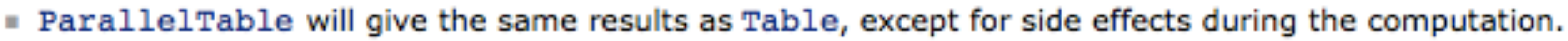
Working with Mathematica
a phd student’s experience

Journal Club, 15th October 2013

Robert Zielke

“Side effects”

- Parallel computation:
  - Table[f[x] , {x, xmin, xmax}]
  - ParallelTable[f[x] , {x, xmin, xmax}]



- ParallelTable will give the same results as Table, except for side effects during the computation.

Bug evolution

- Integrate[Exp[x Exp[I y]], {y, -Pi, Pi}, Assumptions -> {x > 0}]
- Integrate[Exp[x Exp[I y]], {y, -Pi, Pi}, Assumptions -> {x > 1}]
- Integrate[Exp[x Exp[I y]], {y, -Pi, Pi}, Assumptions -> {1 > x > 0}]
- Integrate[Exp[(0.5) Exp[I y]], {y, -Pi, Pi}]
- Integrate[Exp[(1/2) Exp[I y]], {y, -Pi, Pi}]

Bug evolution

- Integrate[Exp[x Exp[I y]], {y, -Pi, Pi}, Assumptions -> {…}]

Version	x > 0	x > 1	1 > x > 0	x = 0.5	x = ½
5.2	0	0	0	6.28 + 0. i	2 Pi
7	2 Pi	2 Pi	2 Pi	0. + 0. i	2 Pi
8 (and 6)	2 Pi ; True 0 ; x < 1	2 Pi	0	6.28 + 0. i	0
9	2 Pi	2 Pi	2 Pi	6.28 + 0. i	2 Pi

Set parameters for a calculation

- a = 1 (*nm*);
  …
  m’ = func[m0, m*, E0, …];
- parameters = {a -> 1 (*nm*), …, m’->func[m0, m*, E0, …]}
  - /.parameters
  - //.parameters

Pure functions

- f[x_]:=x^2
  - f[y]
    - y^2
- f[x1_,x2_]:=x1^2+x2
  - f[y,z]
    - y^2+z

- (#^2)&
  - (#^2)&[y]
    - y^2
- (#1^2+#2)&
  - (#1^2+#2)&[y,z]
    - y^2+z

Numerical integration

-
  ```
  NumIntFast[fun_]:=
  NIntegrate[ fun, {x, xmin, xmax}, …, {z, zmin, zmax},
        Method -> {“GlobalAdaptive”,
                          Method-> “GaussKronrodRule”,
                                “SingularityDepth”->Infinity},
        MinRecursion->4, AccuracyGoal -> Automatic]
  ```
  - SingularityDepth: number of recursive bisections before applying a singularity handler.
  - GaussKronrodRule avoides warnings if the integrand is ~0 and gives a huge speedup.

Output of huge expressions

- HugeExpression
  - Java interface/front end is not responding for some time. Box “show more” is displayed.
  - Mathematica consumes lots of memory for “Show Full Output”.
- HugeExpression;
- Short[HugeExpression , n]
  - Truncated output with ~n ‘lines’.
- Print[HugeExpression];
  - Writes HugeExpression to the notebook.

Saving expressions

- Useful directory commands:
  - Directory[]
  - SetDirectory[]
- Export[“~/dir/filename.dat”, SomeTable];
  - Different file formats
- DumpSave[“~/dir/filename.mx”, SomeTable];
  - Internal Mathematica format
  - Restriction: Not compatible with different Mathematica versions
- Clear[SomeTable];
- Get[“~/dir/filename.mx”]; OR <<“~/dir/filename.mx”

Loading files with similar name

- FileNames[]

-
  ```
  For[ x = 1, x < Length[ FileNames[] ] + 1, x++,
   If[
          StringMatchQ[ FileNames[][[x]], ”myFile*.mx"],
          Get[FileNames[][[x]]]
          ]
  ]
  ```

Timing long calculations

- Timing[calculation] / AbsoluteTiming[calculation]
  - { CPU time / real time , result }
- Input:
  - Date[]
  - myTable = LongCalculation/ParallelTable[..];
  - Date[]
  - DateDifference[%%%,%] * 24 "hours"
- Output:
  - {year,month,day,hour,minute,second}
  - {year,month,day,hour,minute,second}
  - i.e. 5.4 "hours"

Addressing previous output

- In[n]
  - Out[n]
- … more In[…] & Out[…]
- %n (%321)
  - Out[n] (Out[321])

Mathematica allocates more and more memory during the evaluation of a notebook. The memory is freed internally if there exist no more links to the expression/result.

Known as “memory hogging”.

Prevent memory hogging

- `$HistoryLength=0;`

-
  ```
  clearMemory:= Module[ {}, Unprotect[ In, Out ];
                            Clear[ In, Out ];
                            Protect[ In, Out ];
                            ClearSystemCache[]; ];
  ```
- `<<Utilities`CleanSlate``
  - `CleanSlate[]` will attempt to restore the kernel to the state when the package was loaded

Clear kernel – evaluation queue trick

Notebook containing definitions: “~/dir/definitions.nb”

- SetOptions[$FrontEnd,
  ”ClearEvaluationQueueOnKernelQuit” -> False]
- NotebookEvaluate[“~/dir/definitions.nb”];
  …calculation1
  Quit[];
- NotebookEvaluate[“~/dir/definitions.nb”];
  …calculation2
  Quit[];
- …

Conclusion

- Mathematica has bugs!
- Mathematica consumes a lot of memory.
- The Documentation explains the functions.
- For solutions to your problems consult http://mathematica.stackexchange.com