Feagin's Order 10, 12, and 14 Methods

Chris Rackauckas

September 7, 2020

DifferentialEquations.jl includes Feagin's explicit Runge-Kutta methods of orders 10/8, 12/10, and 14/12. These methods have such high order that it's pretty much required that one uses numbers with more precision than Float64. As a prerequisite reference on how to use arbitrary number systems (including higher precision) in the numerical solvers, please see the Solving Equations in With Chosen Number Types notebook.

0.1 Investigation of the Method's Error

We can use Feagin's order 16 method as follows. Let's use a two-dimensional linear ODE. Like in the Solving Equations in With Chosen Number Types notebook, we change the initial condition to BigFloats to tell the solver to use BigFloat types.

```
using DifferentialEquations
const linear_bigα = big(1.01)
f(u,p,t) = (linear_bigα*u)

# Add analytical solution so that errors are checked
f_analytic(u0,p,t) = u0*exp(linear_bigα*t)
ff = ODEFunction(f,analytic=f_analytic)
prob = ODEProblem(ff,big(0.5),(0.0,1.0))
sol = solve(prob,Feagin14(),dt=1//16,adaptive=false);
```

```
retcode: Success
Interpolation: 3rd order Hermite
t: 17-element Array{Float64,1}:
 0.0
 0.0625
 0.125
 0.1875
 0.25
 0.3125
 0.375
 0.4375
 0.5
 0.5625
 0.625
 0.6875
 0.75
 0.8125
 0.875
 0.9375
 1.0
```

```
u: 17-element Array{BigFloat,1}:
 0.50
 0.5325799879535391294151756922663100127575701279367674657299889518448727031384926
 0.5672828871371837684101760938113319028166849848855565501534311815341659662084392
 0.6042470263955404578588403482610764707515845791230606683491210486718827818716904
 0.6436197480773975548106951513331344372869387819583752748846228020793777935497762
 0.6855579953554405579820963492463707369308089500709797508025071898185633516420024
 0.7302289378157059336084349405886010308809987247048142151879999305882306169718902
 0.777810637810428680311119693945339493753691675945670302006794689942184819812709
 0.8284927602304253869042839670047645226327964053287150891084535034410969675378266
 0.8824773285262228669757698486772395431354193449529466082277853631417758596194293 6
 0.9399795299915405153402248568618947272920440084137115643604676307008419853257788
 1.001228573518936040932179654581424903118245024943528049619445960282439336451516
 1.06646860324690824654467242384670608266022713505450493171046906577618017533485
 1.135959671740132190447259472795962527154069506739201242159983984002811333382739
 1.209978776582131731617034761065124311009358614495571372335128166979326800445978
 1.288820964512299462579706253102020998553494397009716401282671347081826473196884
 1.372800507508458259187784036450083092262707343306282981254591926965493411372691
```

```
println(sol.errors)
```

```
Dict{Symbol,BigFloat}(:l∞@*( =>
2.197510403426609917814702632649560560683659367683780324635801610297349872909655e-23,:final
=>
2.197510403426609917814702632649560560683659367683780324635801610297349872909655e-23,:l2
=> 1.061501597814768635894514677590712762248364686527596359902826841740549975688161e-23)
```

Compare that to machine ϵ for Float64:

```
eps(Float64)
```

2.220446049250313e-16

The error for Feagin's method when the stepsize is 1/16 is 8 orders of magnitude below machine ϵ! However, that is dependent on the stepsize. If we instead use adaptive timestepping with the default tolerances, we get

```
sol =solve(prob,Feagin14());
println(sol.errors); print("The length was $(length(sol))")
```

```
Dict{Symbol,BigFloat}(:l∞@*( =>
1.545738883943140962546537598609759219816414790728029220638828884206395861982752e-09,:final
=>
1.545738883943140962546537598609759219816414790728029220638828884206395861982752e-09,:l2
```

```
=>
8.925066870202330409924421192162193462506388332261074725109949218067763405137993e-10)The
length was 3
```

Notice that when the stepsize is much higher, the error goes up quickly as well. These super high order methods are best when used to gain really accurate approximations (using still modest timesteps). Some examples of where such precision is necessary is astrodynamics where the many-body problem is highly chaotic and thus sensitive to small errors.

0.2 Convergence Test

The Order 14 method is awesome, but we need to make sure it's really that awesome. The following convergence test is used in the package tests in order to make sure the implementation is correct. Note that all methods have such tests in place.

```julia
using DiffEqDevTools
dts = 1.0 ./ 2.0 .^(10:-1:4)
sim = test_convergence(dts,prob,Feagin14())
```

```
DiffEqDevTools.ConvergenceSimulation{DiffEqBase.ODESolution{BigFloat,1,Arra
y{BigFloat,1},Array{BigFloat,1},Dict{Symbol,BigFloat},Array{Float64,1},Arra
y{Array{BigFloat,1},1},DiffEqBase.ODEProblem{BigFloat,Tuple{Float64,Float64
},false,DiffEqBase.NullParameters,DiffEqBase.ODEFunction{false,typeof(Main.
##WeaveSandBox#255.f),LinearAlgebra.UniformScaling{Bool},typeof(Main.##Weav
eSandBox#255.f_analytic),Nothing,Nothing,Nothing,Nothing,Nothing,Nothing,No
thing,Nothing,Nothing,Nothing,Nothing},Base.Iterators.Pairs{Union{},Union{}
,Tuple{},NamedTuple{(),Tuple{}}},DiffEqBase.StandardODEProblem},OrdinaryDif
fEq.Feagin14,OrdinaryDiffEq.InterpolationData{DiffEqBase.ODEFunction{false,
typeof(Main.##WeaveSandBox#255.f),LinearAlgebra.UniformScaling{Bool},typeof
(Main.##WeaveSandBox#255.f_analytic),Nothing,Nothing,Nothing,Nothing,Nothin
g,Nothing,Nothing,Nothing,Nothing,Nothing,Nothing},Array{BigFloat,1},Array{
Float64,1},Array{Array{BigFloat,1},1},OrdinaryDiffEq.Feagin14ConstantCache{
BigFloat,Float64}},DiffEqBase.DEStats}}(DiffEqBase.ODESolution{BigFloat,1,A
rray{BigFloat,1},Array{BigFloat,1},Dict{Symbol,BigFloat},Array{Float64,1},A
rray{Array{BigFloat,1},1},DiffEqBase.ODEProblem{BigFloat,Tuple{Float64,Floa
t64},false,DiffEqBase.NullParameters,DiffEqBase.ODEFunction{false,typeof(Ma
in.##WeaveSandBox#255.f),LinearAlgebra.UniformScaling{Bool},typeof(Main.##W
eaveSandBox#255.f_analytic),Nothing,Nothing,Nothing,Nothing,Nothing,Nothing
,Nothing,Nothing,Nothing,Nothing,Nothing},Base.Iterators.Pairs{Union{},Unio
n{},Tuple{},NamedTuple{(),Tuple{}}},DiffEqBase.StandardODEProblem},Ordinary
DiffEq.Feagin14,OrdinaryDiffEq.InterpolationData{DiffEqBase.ODEFunction{fal
se,typeof(Main.##WeaveSandBox#255.f),LinearAlgebra.UniformScaling{Bool},typ
eof(Main.##WeaveSandBox#255.f_analytic),Nothing,Nothing,Nothing,Nothing,Not
hing,Nothing,Nothing,Nothing,Nothing,Nothing,Nothing},Array{BigFloat,1},Arr
ay{Float64,1},Array{Array{BigFloat,1},1},OrdinaryDiffEq.Feagin14ConstantCac
he{BigFloat,Float64}},DiffEqBase.DEStats}[retcode: Success
Interpolation: 3rd order Hermite
t: [0.0, 0.0009765625, 0.001953125, 0.0029296875, 0.00390625, 0.0048828125,
 0.005859375, 0.0068359375, 0.0078125, 0.0087890625  ...@*( 0.9912109375, 0.9921875,
0.9931640625, 0.994140625, 0.9951171875, 0.99609375, 0.9970703125, 0.998046875,
0.9990234375, 1.0]u: BigFloat[0.50,
0.500493407353274144224016740778348649218060302161584129452202794115660211219599,
0.5009873016081808184403552812231889505849105976833766213492949239260841368026092,
0.50148168324520171427198257099835421214533336316613095293438972188206055292586,
0.5019765527452926697413863031219664328338088601822445017121888607974310367490053,
0.5024719105898841371652150179563239002143316649080207098547071166987523430816898,
0.5029677572608816515106435191675246046427648760687772460544794831604624949834855,
```

```
0.5034640932406662992139192389830628016370303392435659971381052766108783597241391,
0.5039609190120951874615456397052524112091807245795198906708350457432594781653915,
0.5044582350585019139345591885116484178028684241257599533660004519217986099443769
(*@...@*(
1.3606681506044349509999505354431852352163391999270684367421064140446792395154054,
1.3620108779461832688270011692258645341572435150730399511531778614726314096172425,
1.3633549303110193083837246378459568275219553760965929199275414926893533047625626,
1.3647003090064953391499466705889272374033129891334984765672432756507922099013163,
1.3660470153414539424145962795447727080505515244080511473449990202705531113402152,
1.3673950506260292845743750156552256873077006209146215024148192878231714392273021,
1.3687444161716483916889360774111902193244980984984290058765042822106566254749811,
1.3700951132910324252938132146019285207946298110241080897757925861837246835730522,
1.3714471432981979594723405931109132038126263705943216051179502955698393132713753,
1.3728005075084582591878060115541173583618854903329449123147950628609452659697666],
retcode: SuccessInterpolation: 3rd order Hermitet: [0.0, 0.001953125, 0.00390625,
0.005859375, 0.0078125, 0.009765625, 0.01171875, 0.013671875, 0.015625, 0.017578125
(*@...@*( 0.982421875, 0.984375, 0.986328125, 0.98828125, 0.990234375, 0.9921875,
0.994140625, 0.99609375, 0.998046875, 1.0]u: BigFloat[0.50,
0.5009873016081808184403552812231889505849105976869970947216582055241988418330043,
0.5019765527452926697413863031219664328338088601894997464536470643897825028575861,
0.5029677572608816515106435191675246046427648760796816025110697498811198930671033,
0.5039609190120951874615456397052524112091807245940877416493974506008240426203329,
0.5049560418636970370163568930910938780783930787757357867798953848057143406933377,
0.5059531296880823345471746020389471304909424091056168364541340183078255553277085,
0.5069521863652926583134503856740792313682294541010285050589346576968266665528748,
0.5079532157830311291546838581046674388309058430590545310621938691525938019872,
0.5089562218366775389970434828264868922469483072826175429449686273069735423750136
(*@...@*( 1.3486430154585924796750960230717636740164164121047504887569533744992944078691 8,
1.3513060502946406731535984145479258809626934123101230869242070340229624137540224,
1.3539743435678414104792857791059277027193819496294701761760294386460400140289953,
1.3566479056615216061007661584646523396275317937594638681026074809719857026216721,
1.3593267469795111251884521666056237846699367573855162062542349394295971516480642,
1.3620108779461832688270011692258645341572435200731887436021860304191663256308353,
1.3647003090064953391499466705889272374033129941533828271370545949805491245693654,
1.3673950506260292845743750156552256873077006259542998550290571206539339338473544,
1.3700951132910324252938132146019285207946298160836390422151163694340862252063,
1.3728005075084582591878060115541173583618854954123872165966372639390576448095851],
retcode: SuccessInterpolation: 3rd order Hermitet: [0.0, 0.00390625, 0.0078125,
0.01171875, 0.015625, 0.01953125, 0.0234375, 0.02734375, 0.03125, 0.03515625 (*@...@*(
0.96484375, 0.96875, 0.97265625, 0.9765625, 0.98046875, 0.984375, 0.98828125, 0.9921875,
0.99609375, 1.0]u: BigFloat[0.50,
0.5019765527452926697413863031219664328338089596679027919888741103974398448576829,
0.5039609190120951874615456397052524112091809243373910752677848686355143273445556,
0.5059531296880823345471746020389471304909427099049813605553848314515187463009496,
0.5079532157830311291546838581046674388309062457103293891502691351497465467261813,
0.5099612084293035092327852594150244065249580423491103311415186380692501996764351,
0.5119771388823309236419857502755201438545747729808604815102428298131085841261152,
0.5140010385211008389138964373154846258498358689286147531125652428267236364526061,
0.5160329388486451692230830905819969144868470465940686994015887118451172282452837,
0.5180728714925306371741201332924871054851977111133534904794724509949799099094199
(*@...@*(
1.3249106284541775466999950247592459404409190358693535321782749796093868686357812,
1.3301481399340546286752688034447125220434892237399552826842250963687309003865583,
1.3354063558493198164900848637876127679604780498119585885764231213717590053539773,
1.3406853580467903223350417026133709423190166359132147298499263705334289364664321,
1.3459852286968324609091831402451653325099084463360226629376573437098670177716884,
1.3513060502946406731535984145479258809627608961265042016895160145132115591278354,
1.3566479056615216061007661584646523396275998131980258628666925664483056520998713,
1.3620108779461832688270011692258645341573120783141010848761380196376688671053072,
```

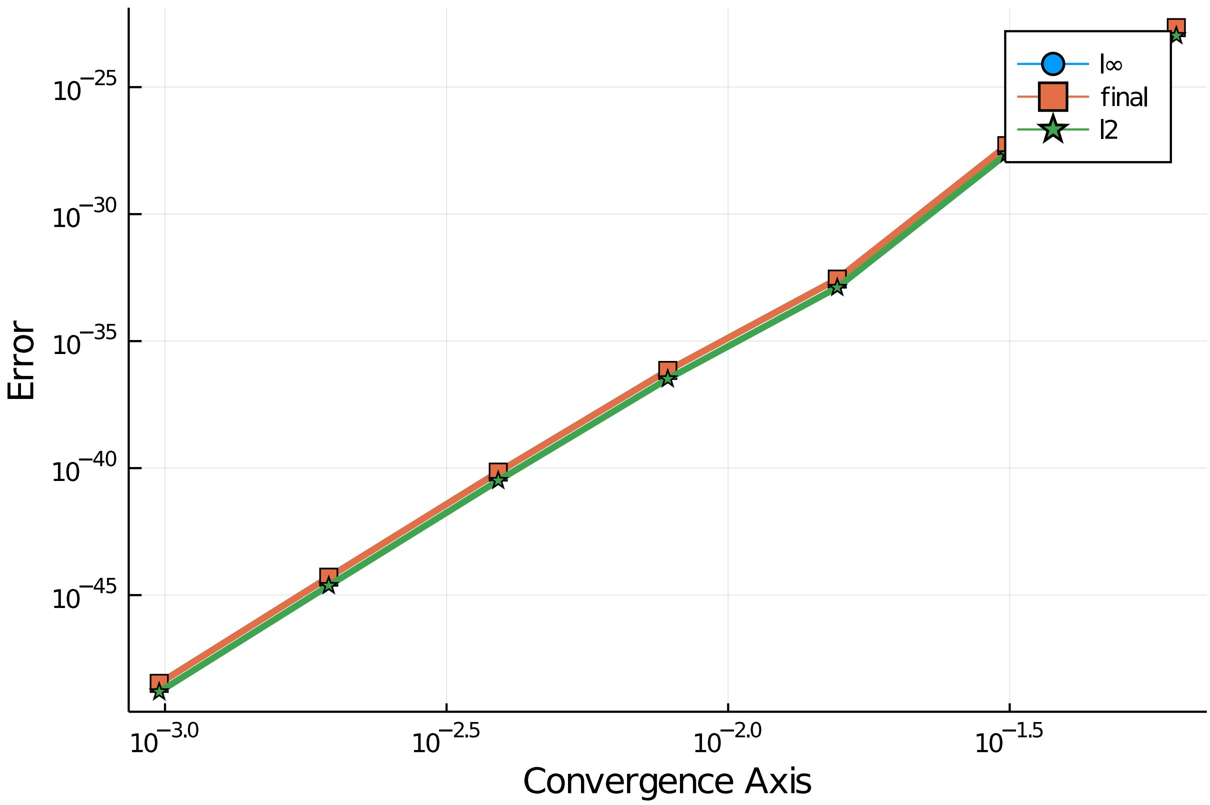
```
1.3673950506260292845743750156552256873077697261945049092191653877874628467060066,
1.3728005075084582591878060115541173583619551408656831634446641617258365621108O7],
retcode: SuccessInterpolation: 3rd order Hermitet: [0.0, 0.0078125, 0.015625, 0.0234375,
0.03125, 0.0390625, 0.046875, 0.0546875, 0.0625, 0.0703125 (*@...@*( 0.9296875, 0.9375,
0.9453125, 0.953125, 0.9609375, 0.96875, 0.9765625, 0.984375, 0.9921875, 1.0]u:
BigFloat[0.50,
0.5039609190120951874615456397052524112111879130152678185972878772909801438539725,
0.5079532157830311291546838581046674388349520211445279229156914347577732228682629,
0.5119771388823309236419857502755201438606915110986662360617789298294706379239618,
0.516032938848645169223083090581996914495067305162617236525550500055015942692349 4,
0.5201208682053510730365643024626851557639184785285451286948438374521217071188324,
0.5242411814762751337774973218812134360586606206771934231862500076957565363308646,
0.5283941352015403769607464142521774116458265525572890561728945609507560270251758,
0.5325799879535391294151762250959426613068339220070424136607469719025179880272659,
0.536799000353032327510195834525255448616954748053680777867461346364161391270671 (*@...@*(
1.2786913785286666306141956773481021922505726846718819789074951128603981468296 1,
1.288820964512299462579725594478432162502887151708268911920429452719237031672889,
1.299030795435346710270061887384885906851118472865428198598034506313240420753 54,
1.309321506985220708360876782599885617305680877664407188219768945081296125363463,
1.319693739885146473355596260460075276820407682877165061218322278576772888141989,
1.330148139934054628675268803444712522700344601505102906797926571988117634577709,
1.340685358046790322335041702613370942986414700099829148697106600907447993876335,
1.351306050294640673153598414547925881640827460343653263042386265204459670148168,
1.362010877946183268827001169225864534846174287909173052147154581054328379019894,
1.372800507508458259187806011554117359061741486128519077134901878437026032283726],
retcode: SuccessInterpolation: 3rd order Hermitet: [0.0, 0.015625, 0.03125, 0.046875,
0.0625, 0.078125, 0.09375, 0.109375,0.125, 0.140625 (*@...@*( 0.859375, 0.875, 0.890625,
0.90625, 0.921875, 0.9375, 0.953125, 0.96875, 0.984375, 1.0]u: BigFloat[0.50,
0.5079532157830311291546838581046674228648435931525498190278213939882467623443407,
0.5160329388486451692230830905819968820467955755537640078345474392168602557520595,
0.5242411814762751337774973218812133866120487057078897985952628075072800031634668,
0.5325799879535391294151762250959425943293258372491737121300708996931105182686892,
0.5410514350853763614943750093333620112324860397855474272871650047471573253335466,
0.5496576327112816769034687550989237307955190439443562958216377582793355171347836,
0.5584007242307674626882271691434832264014636154224009359108553180731368099218107,
0.5672828871371837684101772289089291091453492795766925840388246324369470541329958,
0.5763063335600296017498461036438225844495159268615137578461101403075168130161427
(*@...@*(
1.191033680844897138480138745131362631104182123430362493341676788830879226337858,
1.209978776582131731617051708708410892779229149607193600997132354929002847799 81,
1.229225221188223144617416241361990712052620148424773445644365505495011922745439,
1.248777808048331358756513829839404819440872534223897340978037770853624093056804,
1.268641406793269372330171123473022743877631147466907756310015221824454321458462,
1.288820964512299462579725594478429731261988424253379253261167688863620014837669,
1.309321506985220708360876782599883106227194078928307355592757175919985164258564,
1.330148139934054628675268803444709929859546828182075655756245810675609875660 62,
1.351306050294640673153598414547923205071964935467594455236996453561098281594227,
1.372800507508458259187806011554114596757211232433432893362855858032659142110O7],
retcode: SuccessInterpolation: 3rd order Hermitet: [0.0, 0.03125, 0.0625, 0.09375,
0.125, 0.15625, 0.1875, 0.21875, 0.25, 0.28125 (*@...@*( 0.71875, 0.75, 0.78125, 0.8125,
0.84375, 0.875, 0.90625, 0.9375,0.96875, 1.0]u: BigFloat[0.50,
0.516032938848645169223083090576164551016602523605979342081272221990055728561679,
0.5325799879535391294151762250839038946419311698176330530726315794171525645034893,
0.5496576327112816769034687550802866340567114355433590002071694871709322410405931,
0.5672828871371837684101772288832828270073935705216193322632159538121017711969774,
0.5854733108158904616210196190827662219055210008829792702890722633372001092380584,
0.6042470263955404578588421618108943195685209948140048324759492077870732303989337,
0.6236227376428912250695033937265522498285111444862489660878884721013969830771631,
0.6436197480773975548106977269607081521296225799859728739856374485966663631800 53,
```

0.664257980202808203250883074825851325341067048736253888803428690068076152641046
(*@...@*(1.0333338465044267120825803549601079350439706137564931634511294626091384818214,
1.066468603246908246544685227179493318280916696425537832438762276651139095129374,
1.100665855046623660308919551395610359208286716684431112070414408183611765253205,
1.135959671740132190447274246860075974536172903481653049159364319427066929995183,
1.172385215643205349198769729770167158307196511889785786439894918782978322870963,
1.209978776582131731617051708325497527117244553498434756316642412641437610370797,
1.248777808048331358756513829430099007097842940139504429315922973879445769691784,
1.28882096451229946257972559404143259768018368155787883476891071715795301279543,
1.330148139934054628675268802978666438630864641254421813595879534440847470751688,1.372800507508458259
retcode: SuccessInterpolation: 3rd order Hermitet: [0.0, 0.0625, 0.125, 0.1875, 0.25,
0.3125, 0.375, 0.4375, 0.5, 0.5625, 0.625, 0.6875, 0.75, 0.8125, 0.875, 0.9375, 1.0]u:
BigFloat[0.50,
0.5325799879535391294151756922663100127575701279367674657299889518448727031384926,
0.5672828871371837684101760938113319028166849848855565501534311815341659662084392,
0.6042470263955404578588403482610764707515845791230606683491210486718827818716904,
0.643619748077397554810695151333134437286938781958375274884622802079377793549762,
0.685557995355440557982096349246370736930808950070979750802507189818563351642024,
0.7302289378157059336084349405886010308809987247048142151879999305882306169718902,
0.777810637810428680311119693945339493753691675945670302006794689942184819812709,
0.8284927602304253869042839670047645226327964053287150891084535034410969675378266,
0.8824773285262286697576984867723954313541934495294660822778536314177585961942936,
0.9399795299915405153402248568618947272920440084137115643604676307008419853257788,
1.001228573518936040932179654581424903118245024943528049619445960282439336451516,
1.06646860324690824654467242384670608266022713505450493171046906577618017533485,
1.135959671740132190447259472795962527154069506739201242159983984002811333382739,
1.209978776582131731617034761065124311009358614495571372335128166979326800445978,
1.288820964512299462579706253102020998553494397009716401282671347081826473196884,
1.372800507508458259187784036450083092262707343306282981254591926965493411372691]],
Dict(*@{Any,Any
}(:l∞@*(=>
BigFloat[3.354354545962993017750167938278129130201240818733894747246416797762893693012556e-49,
5.0797773973643850037988739563364757867015516067866884666732578123388975047482e-45,
6.965053307368658446539816666631293486902250519752824131061460869466635839790668e-41,
6.998559957959096002747031148778557831077884643242480170494801380170765753974579e-37,
2.761604674257899176583497342905526511107065867988065457455490380825745507598647e-33,
4.965060749672954837420128986603252643572639175648660423260171211890675951960709e-28,
2.197510403426609917814702632649560560683659367683780324635801610297349872909655e-23],:final
=>
BigFloat[3.354354545962993017750167938278129130201240818733894747246416797762893693012556e-49,
5.0797773973643850037988739563364757867015516067866884666732578123388975047482e-45,
6.965053307368658446539816666631293486902250519752824131061460869466635839790668e-41,
6.998559957959096002747031148778557831077884643242480170494801380170765753974579e-37,
2.761604674257899176583497342905526511107065867988065457455490380825745507598647e-33,
4.965060749672954837420128986603252643572639175648660423260171211890675951960709e-28,
2.197510403426609917814702632649560560683659367683780324635801610297349872909655e-23],:l2
=>
BigFloat[1.557658061895966325846207347700821566122250234951867982385249845278493662676944e-49,
2.360411657197547333498547223212880765989953910523198376084992961121455787643313e-45,
3.240607605160746766371785542718280708237168066098329304267778983901623487961701e-41,
3.264565979149024498598621687244084221464920048688554495210368172485686228822379e-37,
1.294777666747386485263611419731111056071389864984176915402703871046417929063523e-33,
2.351485030191003061425949446982335648801181524332244933545614443786091245762492e-28,
1.061501597814768635894514677590712762248364686527596359902826841740549975688161e-23]),
7, Dict(:dts =>[0.0009765625, 0.001953125, 0.00390625, 0.0078125, 0.015625, 0.03125,
0.0625]), Dict(*@{Any,Any}(:l∞@*(=>
14.29332754610385243500089313284816040556504816254374715376150534187461411604701,:final
=> 14.29332754610385243500089313284816040556504816254374715376150534187461411604701,:l2
=> 14.30280974051840423232019057634315242594313233119811212889763182960978082577156),

```
[0.0009765625, 0.001953125, 0.00390625, 0.0078125, 0.015625, 0.03125, 0.0625])
```

For a view of what's going on, let's plot the simulation results.

```
using Plots
gr()
plot(sim)
```



This is a clear trend indicating that the convergence is truly Order 14, which is the estimated slope.

0.3 Appendix

This tutorial is part of the SciMLTutorials.jl repository, found at: https://github.com/SciML/SciMLTutor For more information on doing scientific machine learning (SciML) with open source software, check out https://sciml.ai/.

To locally run this tutorial, do the following commands:

```
using SciMLTutorials
SciMLTutorials.weave_file("ode_extras","02-feagin.jmd")
```

Computer Information:

```
Julia Version 1.4.2
Commit 44fa15b150* (2020-05-23 18:35 UTC)
Platform Info:
  OS: Linux (x86_64-pc-linux-gnu)
```

```
  CPU: Intel(R) Core(TM) i7-9700K CPU @ 3.60GHz
  WORD_SIZE: 64
  LIBM: libopenlibm
  LLVM: libLLVM-8.0.1 (ORCJIT, skylake)
Environment:
  JULIA_LOAD_PATH = /builds/JuliaGPU/DiffEqTutorials.jl:
  JULIA_DEPOT_PATH = /builds/JuliaGPU/DiffEqTutorials.jl/.julia
  JULIA_CUDA_MEMORY_LIMIT = 2147483648
  JULIA_NUM_THREADS = 8
```

Package Information:

```
Status `/builds/JuliaGPU/DiffEqTutorials.jl/tutorials/ode_extras/Project.toml`
[f3b72e0c-5b89-59e1-b016-84e28bfd966d] DiffEqDevTools 2.27.0
[0c46a032-eb83-5123-abaf-570d42b7fbaa] DifferentialEquations 6.15.0
[961ee093-0014-501f-94e3-6117800e7a78] ModelingToolkit 3.20.0
[76087f3c-5699-56af-9a33-bf431cd00edd] NLopt 0.6.0
[2774e3e8-f4cf-5e23-947b-6d7e65073b56] NLsolve 4.4.1
[429524aa-4258-5aef-a3af-852621145aeb] Optim 1.0.0
[1dea7af3-3e70-54e6-95c3-0bf5283fa5ed] OrdinaryDiffEq 5.42.7
[91a5bcdd-55d7-5caf-9e0b-520d859cae80] Plots 1.6.3
[37e2e46d-f89d-539d-b4ee-838fcccc9c8e] LinearAlgebra
[2f01184e-e22b-5df5-ae63-d93ebab69eaf] SparseArrays
```